Product Documentation

Embarcadero® DBArtisan™

User Guide

Version 9.7.5/XE6
1st Edition May, 2015

May, 2015

CONTENTS

Welcome to DBArtisan

Welcome to DBArtisan, the database administration and development solution that lets you manage IBM DB2 for Linux, Unix, and Windows, IBM DB2 for z/OS and OS/390, Interbase/Firebird, Microsoft SQL Server, MySQL, Oracle, PostgreSQL, Sybase IQ, and Sybase ASE databases.

The list below describes the major sections of this document.

- Getting Started - Provides an introduction for new users and helps you set up DBArtisan. Included is an investigation of the major user interface elements, detailed tutorial exercises, instructions on configuring DBArtisan features, and additional information on using this document and third-party documentation.
- Datasource and Server Management - Shows you how to register the datasources that DBArtisan will work against.This chapter also describes the extensive datasource management facilities.
- DB Team Server Support - Shows you how to make use of datasource definitions and ER/Studio model data in a Team Server Repository.
- Database Object Management - Describes how to create new database objects, edit existing objects, and provides details on the operations you can perform against database objects.
- Coding Environments/Editors - Describes the development/execution/testing environments offered.
- Data, Schema, and Miscellaneous Utilities (DBArtisan) - Provides detailed information about the extraction, migration and publication of your schema, the loading and unloading of data, database searching, the use of the Script Execution Facility and Query Builder, the scheduling of processes, the backing up and restoration of databases, the use of the Database Monitor, Database Manager, Resource Limits, and Logfile.
- Tools - Tools provide search, script and file execution, scheduling, visual difference, a visual query builder, data import, a data editor, and an auto-replace tool for the scripting environment.
- Project Management - DBArtisan provides project support and version control system integration.
- Add-on tools - License-dependent add-ons include:
- **Debuggers** - DBArtisan provides debugging environments for DB2 LUW, SQL Server, Oracle, and Sybase ASE.
- **Oracle PL/SQL Profiler** - captures metrics of various PL/SQL programmable objects as they are executed
- **Capacity Analyst** - collects storage and performance metrics and provides visual analysis and reporting for historical trend analysis and forecasting.

- **Performance Analyst** - presents a complete picture of a database's performance and lets you investigate performance bottlenecks.
- **Space Analyst** - helps you effectively locate and troubleshoot bottlenecks and performance inefficiencies that result from space management issues.
- **Code Analyst** - helps you identify time-consuming lines of code.

Note: For information on access to third party documentation and descriptions of conventions used in this document, see More Information on Using this Document

Additional Product Information

The Embarcadero Web site provides access to information and resources useful through the life of your product. This includes access to:

- Technical Support
- Case studies, video presentations, webinars and white papers
- Product communities through the Embarcadero Developer Network
- Free trials of related products

See www.embarcadero.com/support.

You can find general information about system requirements, licensing information, and details on DBMS product support in Prerequisites and Preliminary Tasks.

Prerequisites and Preliminary Tasks

You can find general information about system requirements, licensing information, and details on DBMS product support in:

- Introducing DBArtisan
- Technical Requirements and DBMS Support
- Installing and Licensing DBArtisan
- More Information on Using this Document

For more information, you can also see the **Read Me** and **Quick Start** at http://docs.embarcadero.com/products/dbartisan/.

Introducing DBArtisan

DBArtisan is an industry-leading database administration solution for managing Oracle, Microsoft SQL Server, Sybase, PostgreSQL, IBM DB2, and MySQL databases. Its cross-platform capability allows users to efficiently manage heterogeneous database platforms easily using a single front-end tool. Using DBArtisan, users boost their productivity by using a single tool for all their databases, regardless of vendor.

About DBArtisan

Your new application provides tools that can be used by a number of functions within an organization using DBMS from multiple vendors, in testing, development, or production environments.

Datasource and Server Management - Datasources must be registered. Datasources can be registered manually or semi-automatically. Storage can be registry-based or file-based, and a network storage option facilitates shared use of datasource catalogs.

Object management tools - Your application supports a wide range of database objects and related elements for each DBMS. You can create new objects, edit existing objects, and use a range of object operations that support common, general or DBMS-specific actions.

Coding Environments/Editors - Fully-featured environments dedicated to SQL, DDL and Active Script development are provided. Execution options are available as appropriate, and environment-specific related features, such as rollback/commit and query plan options, are provided. Related coding aids include on-the-fly semantic and syntactic validation, text substitution shortcuts, and visual query building tools. Related execution tools include script and file execution facilities.

Schema and data utilities (DBArtisan only) - Schema extraction, migration, and publication wizards let you quickly extract and move schema and data across environments and even across DBMS types. DBMS-specific import/export, load/unload, and backup/restore utilities are also available. A Data Editor lets you edit your tables in real-time, supports all editable datatypes, and provides Unicode options. Table data can also be imported from formats such as spreadsheets.

Project Management,Version Control, and Script Library - Database project management facilities help you organize, alter, and keep track of changes to database objects or SQL scripts. The project management facilities act as a repository to maintain all source code for a database project. DBArtisan also incorporates version control functions and build management facilities to help you manage and build projects.

Other Time-savers and Productivity Tools - Working within the DBArtisan environment, you have access to database and file search facilities, use visual difference to compare files or database objects, and access the Microsoft Windows task scheduler.

License-specific Add-ons - Depending on the licenses you purchased, you have access to the following tools:

- The DBArtisan Analyst Series includes advanced space, capacity, performance and backup management capabilities. They let you predict and prevent storage-related problems, trend data and object growth over time, and pinpoint performance bottlenecks.

  - Debuggers, available for IBM DB2 for Linux, Unix, and Windows, Oracle, Sybase, and SQL Server, let you test functions and procedures. A Profiler is available for Oracle datasources.

Technical Requirements and DBMS Support

For system requirements and details on DBMS product support in DBArtisan, see the **Read Me** at http://docs.embarcadero.com/products/dbartisan/.

Embarcadero DB Team Server Support

DBArtisan can make use of datasource definitions stored on an Embarcadero DB Team Server. For details, see the online help.

Generic JDBC and ODBC Support

Generic JDBC/ODBC connectivity to non-dedicated DBMS systems or non-database datasources is also provided. DBArtisan can connect to a datasource using a customer-provided, third-party JDBC version 3.0 or ODBC version 3.0 driver. Minimal DBArtisan functionality is provided, including a basic Explorer tree and SQL querying.

IBM DB2 for Z/OS Stored Procedure Requirements

When working against an IBM DB2 for z/OS data source, DBArtisan relies on the following stored procedures, provided as an optional installation step in setting up the DB2 subsystem:

- `DSNWZP`
- `DSNUTILS`
- `ADMIN_COMMAND_DSN`
- `ADMIN_COMMAND_DB2`

Prior to using DBArtisan against an IBM DB2 for z/OS data source, ensure that these components are installed on the server. See IBM DB2 for z/OS documentation for more information.

Installing and Licensing DBArtisan

Before you can register any Embarcadero Technologies application, you must meet the minimum technical requirements. If you meet all the minimum technical requirements, you can install DBArtisan. After installation, you must license the application.

The following topics walk you through this process:

- Technical Requirements and DBMS Support
- Installing DBArtisan
- Licensing Your Product

Installing DBArtisan

To install the application successfully, you must complete each panel of the installation wizard. 64-bit and 32-bit versions of DBArtisan are available with the trial version, which can be downloaded from http://www.embarcadero.com/products.

The DBArtisan installation wizard installs all corresponding files on your machine. The installer offers common options such as license agreement, file and folder selections, and shortcut options. An application-specific option lets you associate file suffixes of elements such as SQL scripts with DBArtisan.

NOTE: Before installing DBArtisan from Embarcadero ToolCloud, see http://docs.embarcadero.com/products/xe/for additional requirements or revised instructions.

The panels presented by the wizard vary depending on whether you are upgrading or performing an initial installation.

Caution: If you have not registered a license when you start a DBArtisan installation, you will be prompted to register a license. For more information, see Licensing Your Product.

After the installation is complete, we recommend that you reboot your machine before using DBArtisan.

Licensing Your Product

Each Embarcadero client application requires one or more licenses in order to run. An Embarcadero product, such as DBArtisan, has a baseline license which provides basic feature support for that product. In addition, incremental licenses may be required to support specific DBMS platforms, product add-ons, or other functions.

For more information, see the following topics:

- Licensing Overview
- Licensing Your Application
- Selecting a License Category During Startup
- Online/Offline Mode and Concurrent License Checkout

Licensing Overview

The following topics provide a high-level discussion of key licensing topics and directs you to sources of more detailed information.

- Viewing Your License Type and Modules
- Understanding Trial, Workstation, and Networked Licenses
- DBArtisan License Modules, DBMS Support, and Feature Availability
- DBArtisan XE License Modules, DBMS Support, and Feature Availability
- Directing Queries Regarding Licenses

Viewing Your License Type and Modules

The **About...** dialog, available from the **Help** menu, displays your license type and each individual license module currently registered.

Understanding Trial, Workstation, and Networked Licenses

Three kinds of licenses are available: Trial, Workstation, and Networked.

Trial licenses	A license for a 14-day, full-featured trial version of the product. The trial license must be registered before you can use the product.
Workstation licensing	A license or set of licenses is tied to a particular workstation. The product can only be used on that workstation.
Networked licensing	Networked licenses are administered and distributed by a central License Server (Embarcadero License Center or Acresso FLEXnet Publisher). There are two types of networked licenses: *Concurrent* and *Networked Named User*. With Concurrent licensing, users on different machines take turns using licenses from a shared pool. With Networked Named User licensing, licenses are pre-assigned to specific users setup on the license server's user list. Those users are guaranteed to have licenses available any time. **NOTE:** Concurrent licenses can be borrowed for use without a network connection. For details, see Online/Offline Mode and Concurrent License Checkout.

For a detailed description of licensing options, see http://www.embarcadero.com/software-licensing-solutions.

DBArtisan License Modules, DBMS Support, and Feature Availability

Individual license modules correspond to the DBMS platforms you are licensed to use. In addition, each DBMS license module corresponds to a product edition. Feature availability for each license module edition is as follows:

- **Standard** - Provides baseline support including datasource management, object management, SQL editing and execution, and standard tools and utilities.
- **Professional** (DBArtisan) - Standard edition features plus SQL Debugger and SQL Profiler.

- **Workbench** (DBArtisan) - Professional edition features plus Performance Analyst, Space Analyst, and Capacity Analyst.

DBArtisan XE License Modules, DBMS Support, and Feature Availability

An XE license provides access to all supported DBMS platforms. XE module and feature availability is as follows:

- **XE Pro** (DBArtisan) - Provides baseline support including datasource management, object management, SQL editing and execution, and standard tools and utilities, SQL Debugger and SQL Profiler.
- **Analyst Pack** (DBArtisan) - Provides access to Performance Analyst, Space Analyst, and Capacity Analyst.

Directing Queries Regarding Licenses

Questions regarding license availability, feature availability, and client or server licensing, should be directed as follows:

- If you work in an organization that uses networked licensing, direct any questions to your site's DBArtisan administrator.
- If you are using workstation licensing, direct licensing questions to your Embarcadero Technologies representative.

Licensing Your Application

See the following topics for details on registering your product:

- Registering a Trial or Workstation License during Installation
- Registering a Workstation License After Application Startup
- Registering By Phone
- Registering a Networked License

Registering a Trial or Workstation License during Installation

Shortly after initiating download of a trial version of an Embarcadero product, you should receive an email with a serial number you must register during installation. Similarly, if you purchase an Embarcadero product while no trial version is active, you will receive a serial number that must be registered during installation.

1 Start the installation. An **Embarcadero License Registration** dialog appears.

2 Copy the serial number from the email and paste it in the **Serial Number** field.

3 Enter your Embarcadero Developer Network account credentials in the **Login or Email** and **Password** fields. If you have not previously created an EDN account, or have forgotten your password, click **I need to create ...** or I've lost my password.

4 Click **Register**.

Your activation file should be downloaded and installed automatically. If this does not happen, click the **Trouble Connecting? Try Web Registration** link and follow the prompts. If you still have problems, see Registering By Phone.

Registering a Workstation License After Application Startup

The following instructions assume that you have received a workstation license by email and that you currently have a valid trial license. If you did not install a trial version or the trial period has expired, follow the instructions in Registering a Trial or Workstation License during Installation instead.

1 On the **Help** menu, select **About**, and then on the dialog that opens, click **Manage** to open a license manager dialog.

2 On the **Serial** menu, select **Add**.

3 Copy the serial number from the email and paste it in the **Add Serial Number** dialog, then click **OK**.

4 Right-click on the serial number you added, and then select **Register** from the context menu. A registration dialog opens.

Note: The **Registration Code** box shows a machine-specific identifier required with other registration methods.

5 Ensure that the **Register using Online Registration** radio box is selected.

6 Provide Developer Network credentials in the **DN Login name or Email** and **DN Password** boxes. If you have not previously created an EDN account, or have forgotten your password, click **I need to create ...** or I've lost my password.

7 Click **Register**.

8 If prompted to restart the application, click **Yes**.

Your activation file should be downloaded and installed automatically. If this does not happen, click the **Trouble Connecting? Try Web Registration** link and follow the prompts. If you still have problems, Registering By Phone.

Registering By Phone

If you have problems with either of the above procedures, you can register licenses by phone. You will have to provide:

- Developer Network credentials
- The registration code displayed in the Embarcadero License Registration dialog that appears when you start an unlicensed application
- The product base license serial number
- The license serial numbers for any additional features you have purchased.

For North America, Latin America and Asia Pacific, call (415) 834-3131 option 2 and then follow the prompts. The hours are Monday to Friday, 6:00 A.M. to 6:00 P.M. Pacific time.

For Europe, Africa and the Middle East, call +44 (0)1628-684 494. The hours are Monday to Friday, 9 A.M. to 5:30 P.M. U.K. time.

Shortly after phoning in, you will receive an email containing an activation file. Then do the following:

1 Save the file to the desktop or a scratch directory such as `c:\temp`.

2 On the **Help** menu select **About** and then on the dialog that opens, click **Register**. A registration dialog opens.

3 Select the **I have received an activation file (*.slip or reg*.txt)** radio box.

4 Click the **Browse** button and use the **Select License Activation File** dialog to locate and select the activation file you installed.

5 Click the **Import** button to import the activation file and when complete, click the **Finish** button.

6 If prompted to restart the application, click **Yes**.

Registering a Networked License

If you work in an organization using Networked licensing, an administrator, department head, or someone performing a similar function will provide you with an activation file.

Once you receive the file, save it to the license subfolder of your product's main installation folder (typically `C:\Program Files\Embarcadero\`*<product><version>*`\license\`), then restart the application.

No additional steps are necessary.

Selecting a License Category During Startup

During startup, if multiple concurrent license categories are available, you are prompted to select a category to use for this DBArtisan session. Multiple license categories can be set up to provide differing feature access or access to different DBMS versions. Feature and DBMS version access are typically distributed across multiple license categories to optimize use of a site's purchased licenses.

Note: This dialog also includes the option to remember your selection on subsequent startups. If you select that option, you can subsequently use the **Select Licenses** button on the **About**... dialog (**Help > About**) to select a different license.

Contact your License Administrator for details on individual license categories or requests for additional feature or DBMS support.

Online/Offline Mode and Concurrent License Checkout

Concurrent licenses can be used in both online and offline modes. In online mode, you must have a continuous network connection to your License Center. Licenses are checked out on startup and checked back in on shutdown.

You can also use a license in offline mode. When you explicitly check out a license for offline use, you can use the license without a connection to your License Center for a specified duration. This lets you work while travelling or commuting, work away from your primary work area, or use the license when a network connection is unavailable or not required.

Note: Contact your site administrator for information on offline license availability, the maximum duration, offline license policy at your site, or any other issues arising from online license usage.

To check out a license for offline use:

1 On the **Help** menu, select **Checkout License**. The **Check Out Licenses For Offline Use** dialog opens.

2 Select the check box associated with each individual license you want to check out.

3 In the **Checkout Duration** box, type the number of hours that you can use the offline license without a network connection to the License Center.

4 Click **OK**.

You can work offline for the specified duration. The duration period begins immediately.

If you subsequently establish a network connection to the License Center before the license duration expires, you can indicate to the License Center that the offline license is no longer required.

To indicate that an offline license is no longer required:

1 On the **Help** menu, select **Checkin License**.

There is no interruption in DBArtisan usage. The license is not actually checked in until you shut down DBArtisan.

More Information on Using this Document

The following topics provide additional information that is useful when working with product documentation:

- Accessing Third Party Documentation
- Shorthand for Third Party Product References

Accessing Third Party Documentation

Many DBArtisan features provide support for functionality available in the supported third party DBMS platforms. In object management for example, properties available when creating or editing objects and actions available against object types, are direct equivalents of clauses, options, or keywords available for use with the third party equivalent.

In such cases, no attempt is made to duplicate detailed documentation by third parties. Casual descriptions are provided, noting the third party equivalent. You can consult the third party documentation for details or clarifications. The following links provide access to online documentation for supported DBMS platforms:

- IBM DB2 for Linux, Unix, and Windows documentation
- IBM DB2 for z/OS documentation
- InterBase/Firebird documentation
- Microsoft SQL Server documentation
- MySQL documentation
- Oracle documentation
- PostgreSQL documentation
- Sybase ASE and Sybase IQ documentation

Shorthand for Third Party Product References

To save space in headings and table columns, shorthand is used to represent versions and variations of the DBMS platforms supported.

- DB2 LUW
- DB2 z/OS
- ITB/FBD *
- MySQL
- ORCL
- PSTGRS *
- SQL SVR
- SYB ASE
- SYB IQ

For specific DBMS version support for DBArtisan, see the **Read Me** at http://docs.embarcadero.com/products/dbartisan/.

DB2 LUW

DB2 LUWis used as shorthand for IBM DB2 for Linux, Unix, and Windows.

DB2 z/OS

DB2 z/OS is used as shorthand for IBM DB2 for z/OS.

MySQL

MySQL is used as shorthand for the MySQL RDBMS.

ORCL

ORCL is used as shorthand for Oracle.

PSTGRS

PSTGRS is used as shorthand for PostgreSQL.

Specifically-supported PostgreSQL-based database products include Greenplum, Pivotal HAWQ, and BigSQL.

SQL SVR

SQL SVR is used as shorthand for Microsoft SQL Server.

SYB ASE

SYB ASE is used as shorthand for Sybase Adaptive Server.

SYB IQ

SYB IQ is used as shorthand for Sybase IQ.

Getting Started

The following sets of topics are intended for new users. They provide learning curve material and tell you how to configure your new application:

- Prerequisites and Preliminary Tasks - Provides information on installation and licensing as well as general information on using this document.
- Starting DBArtisan - Describes the various ways to start the application.
- DBArtisan Application Basics - Introduces the major user interface elements.
- DBArtisan Tutorial exercises - Tutorial exercises that show the major functional areas of DBArtisan.
- Specifying Application Preferences and Feature Options - Tells you how to configure DBArtisan.

Starting DBArtisan

There are three ways to start the application:

- Starting an Installed version of DBArtisan from the Start Menu
- Starting an Installed version of DBArtisan from the Command Line

Staring an Installed version of DBArtisan from the Start Menu

The **Start** menu sequences for DBArtisan startup are always in the form

- **Programs > Embarcadero DBArtisan***version identifier***> Embarcadero DBArtisan***version identifier*

where *version identifier* reflects the version you are running.

Staring an Installed version of DBArtisan from the Command Line

Command line startup is primarily aimed at identity management systems working in conjunction with DBArtisan. Such systems can be configured to start DBArtisan and connect to a specific datasource using credentials that are invisible to the user. Other applications that hide credentials, force connection to a specific datasource, administer usage audits, or otherwise automate startup of DBArtisan can make use of this feature.

Two forms of command line startup are available for each product. Syntax and parameters are as follows:

DBArtisan

- **dbart.exe -D***datasource***-U***username* [**-P***password*]
- **dbart.exe -R***connectionstring***-D***datasource***-U***username* [**-P***password*]

For example

- **rsql.exe -D torlabsy01-Umyusername-P mypassword**
- **dbart.exe -R "oracle://(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=TORLABORCL10g_2)(PORT=1521))(CONNECT_DATA=(SID=ORASID)))" -D torlabora10g_2 -U sys -P extreme**

Note: For a listing of valid connection string values, see -R Switch Values by Platform.

Keep the following points in mind when using command line startup:

- Credentials remain valid until the user disconnects from the datasource.
- When you try disconnect, a confirmation message box includes a warning message that the connection has external credentials that will be lost if you choose to proceed with disconnect.
- Once disconnected, a user must connect to the previously disconnected datasource using specified, valid credentials or rely on Auti-connect. For more information, see Manually Registering or Editing Datasources.
- Command line credentials override the default credentials provided if a datasource is registered using the Auto-connect setting.

Regarding the switch specifications:

- The password switch/value pair is optional and the **P** switch can be provided without a value.
- The switches are case-sensitive while the datasource is case-insensitive;
- The order of these switches doesn't matter;

- Switches can be specified with the minus sign or with a slash and any combinations are allowed. For example, **DBArt -D ROMLABSQL08 /U sa /P**.
- Switches can be specified with the minus sign or with a slash and any combinations are allowed. For example, **RSQL.-D ROMLABSQL08 /U sa /P**.
- If you specify a switch multiple times, it's last valid occurrence from left to right is used. We auto-connect a single datasource, no matter the number of times a -D switch is specified;
- The -p switch (with lower case) prints a file;
- If -R is specified then one of the following will occur:
- If -D is also specified, it will check to see if that datasource already exists. If it does exist, it will check to see if its connection information matches the provided -R parameter. If it matches, startup behavior will be the same as if -R was not specified. If it doesn't match, a modified form of the datasource name is generated using the -D-provided name to ensure uniqueness and give it the connection information specified by -R. If the datasource doesn't exist, a new datasource is generated using the name provided by -D and the remaining connection information provided by -R.
- If -D is not specified, the -R-provided datasource is checked for a match against existing datasources. If a match is found, ithat datasource is connected. If no match is found, a new datasource is generated using the -R-provided information to create a unique name.

Note: If connecting to an existing datasource with auto-connect enabled, if -U and -P values were provided, then auto-connect will be disabled and the provided -U and -P values are used.

-R Switch Values by Platform

The following table lists the valid connection strings that can be used with the **-R** switch for a command line startup.

Platform	Connect String	Notes
DB2 LUW and DB2 z/OS	db2://<HOST>/<DB> db2://<HOST>:<PORT>/<DB> db2:@<ALIAS> where ALIAS is the name from the DB2 client registration	If the DB2 ODBC/CLI driver is installed on the system, that driver will be used for DB2 connections. Otherwise the IBM Data Server Driver for JDBC will be used. Using "db2p" will make the datasource created permanent
MySQL	mysql://<HOST> mysql://<HOST>/<DB> mysql://<HOST>:<PORT> mysql://<HOST>:<PORT>/<DB>	If the MySQL ODBC driver is installed on the system, that driver will be used to make the connection. Otherwise, the MySQL Connector/J JDBC driver will be used. Using "mysqlp" will make the datasource created permanent
ODBC	odbc://<CONNECT_STRING> where connect string can be the a System DSN name on it's own or the full ODBC connection string containing at least one of the following tags: DSN=, FILEDSN= or DRIVER=	Using "odbcp" will make the datasource created permanent.
ORCL	**oracle://<CONNECT_DESCRIPTOR>** **oracle:@<ALIAS>** **where CONNECT_DESCRIPTOR is in the same form as in the TNSNames.ora file. i.e. (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=<HOST>)(PORT=<PORT>))(CONNECT_DATA=(SID=<SID>)))** **where ALIAS is the name from the TNSNAMES.ora file**	**If OCI is installed on the system, that driver will be used. Otherwise, the Oracle Instant Client will be used. Using "oraclep" will make the datasource created permanent.**
PSTGRS *	postgresql://<HOST> postgresql://<HOST>/<DB> postgresql://<HOST>:<PORT> postgresql://<HOST>:<PORT>/<DB>	If <HOST> is a IPv6 address, it must be enclosed in square brackets. If the PostgresSQL ODBC driver is installed on the system, that driver will be used., Otherwise, the PostgreSQL JDBC driver. will be used Using "postgresqlp" will make the datasource created permanent
SQL SVR	sqlserver://<HOST> sqlserver://<HOST>/<DB> sqlserver://<HOST>:<PORT> sqlserver://<HOST>:<PORT>/<DB> sqlserver://<HOST>\<INSTANCE> sqlserver://<HOST>\<INSTANCE>/<DB> sqlserver://<HOST>\<INSTANCE>:<PORT> sqlserver://<HOST>\<INSTANCE>:<PORT>/<DB> sqlserver:@<ALIAS> sqlserver:@<ALIAS>/<DB> where ALIAS is the name from the Network Library Configuration	The latest Microsoft ODBC driver for SQL Server installed on the system is used to make the connection. Using "sqlserverp" will make the datasource created permanent.

SQL SVR**Azure**	sqlserverazure://<HOST> sqlserverazure://<HOST>/<DB>	The latest Microsoft ODBC driver for SQL Server installed on the system is used to make the connection. Using "sqlserverazurep" will make the datasource created permanent
SYB ASE	sybase://<HOST> sybase://<HOST>/<DB> sybase://<HOST>:<PORT> sybase://<HOST>:<PORT>/<DB> sybase:@<ALIAS> sybase:@<ALIAS>/<DB> where ALIAS is the name from the SQL.ini file	If CTLIb is installed on the system that driver will be used. Otherwise the Sybase jConnect JDBC driver will be used. Using "sybasep" will make the datasource created permanent
SYB IQ	sybaseiq://<HOST> sybaseiq://<HOST>/<DB> sybaseiq://<HOST>:<PORT> sybaseiq://<HOST>:<PORT>/<DB>	If the Sybase IQ ODBC driver is installed on the system it will use that, otherwise it will use the Sybase jConnect JDBC driver. Using "sybaseiqp" will make the datasource created permanent

The following table lists the valid connection strings that can be used with the **-R** switch for a command line startup.

Platform	Connect String	Notes
DB2 LUW and DB2 z/OS	db2://<HOST>/<DB> db2://<HOST>:<PORT>/<DB> db2:@<ALIAS> where ALIAS is the name from the DB2 client registration	If the DB2 ODBC/CLI driver is installed on the system, that driver will be used for DB2 connections. Otherwise the IBM Data Server Driver for JDBC will be used. Using "db2p" will make the datasource created permanent
ITB/FBD *	interbase://<HOST>/<DB> interbase://<HOST>:<PORT>/<DB> firebirdsql://<HOST>/<DB> firebirdsql://<HOST>:<PORT>/<DB>	If the InterBase ODBC driver is installed on the system, that driver will be used for an InterBase connection. Otherwise, the InterBase JDBC driver will be used. Using "interbasep" will make the datasource created permanent. If the Firebird ODBC driver is installed on the system, that driver will be used for Firebird connections. Otherwise, the Jaybird JDBC driver will be used. Using "firebirdsqlp" will make the datasource created permanent
MySQL	mysql://<HOST> mysql://<HOST>/<DB> mysql://<HOST>:<PORT> mysql://<HOST>:<PORT>/<DB>	If the MySQL ODBC driver is installed on the system, that driver will be used to make the connection. Otherwise, the MySQL Connector/J JDBC driver will be used. Using "mysqlp" will make the datasource created permanent
ODBC	odbc://<CONNECT_STRING> where connect string can be the a System DSN name on it's own or the full ODBC connection string containing at least one of the following tags: DSN=, FILEDSN= or DRIVER=	Using "odbcp" will make the datasource created permanent.
ORCL	**oracle://<CONNECT_DESCRIPTOR>** **oracle:@<ALIAS>** **where CONNECT_DESCRIPTOR is in the same form as in the TNSNames.ora file. i.e. (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=<HOST>)(PORT=<PORT>))(CONNECT_DATA=(SID=<SID>)))** **where ALIAS is the name from the TNSNAMES.ora file**	**If OCI is installed on the system, that driver will be used. Otherwise, the Oracle Instant Client will be used. Using "oraclep" will make the datasource created permanent**
PSTGRS *	postgresql://<HOST> postgresql://<HOST>/<DB> postgresql://<HOST>:<PORT> postgresql://<HOST>:<PORT>/<DB>	If <HOST> is a IPv6 address, it must be enclosed in square brackets. If the PostgresSQL ODBC driver is installed on the system, that driver will be used., Otherwise, the PostgreSQL JDBC driver. will be used Using "postgresqlp" will make the datasource created permanent

SQL SVR	sqlserver://<HOST> sqlserver://<HOST>/<DB> sqlserver://<HOST>:<PORT> sqlserver://<HOST>:<PORT>/<DB> sqlserver://<HOST>\<INSTANCE> sqlserver://<HOST>\<INSTANCE>/<DB> sqlserver://<HOST>\<INSTANCE>:<PORT> sqlserver://<HOST>\<INSTANCE>:<PORT>/<DB> sqlserver:@<ALIAS> sqlserver:@<ALIAS>/<DB> where ALIAS is the name from the Network Library Configuration	The latest Microsoft ODBC driver for SQL Server installed on the system is used to make the connection. Using "sqlserverp" will make the datasource created permanent
SQL SVR**Azure**	sqlserverazure://<HOST> sqlserverazure://<HOST>/<DB>	The latest Microsoft ODBC driver for SQL Server installed on the system is used to make the connection. Using "sqlserverazurep" will make the datasource created permanent
SYB ASE	sybase://<HOST> sybase://<HOST>/<DB> sybase://<HOST>:<PORT> sybase://<HOST>:<PORT>/<DB> sybase:@<ALIAS> sybase:@<ALIAS>/<DB> where ALIAS is the name from the SQL.ini file	If CTLIb is installed on the system that driver will be used. Otherwise the Sybase jConnect JDBC driver will be used. Using "sybasep" will make the datasource created permanent
SYB IQ	sybaseiq://<HOST> sybaseiq://<HOST>/<DB> sybaseiq://<HOST>:<PORT> sybaseiq://<HOST>:<PORT>/<DB>	If the Sybase IQ ODBC driver is installed on the system it will use that, otherwise it will use the Sybase jConnect JDBC driver. Using "sybaseiqp" will make the datasource created permanent

Application Basics

Application Basics is designed to situate you within the application and to provide information about what DBArtisan offers in the way of design, navigation, and application features. The information presented here is high-level and conceptual.

Application Basics is divided into two sections. The table below describes each section:

Section	Description
DBArtisan Product Design	This section describes key interface elements and concepts, including the Datasource Navigator and Datasource Explorer, parallel query execution, and how to navigate and make the most use of DBArtisan's Multiple Document Interface (MDI) environment.
Specifying Application Preferences and Feature Options	This section describes how you can configure the application to meet your specific needs.

DBArtisan Product Design

This section describes key interface elements and concepts, such as the Datasource Navigator and Datasource Explorer, parallel query execution, and how to navigate and make the most use of DBArtisan's Multiple Document Interface (MDI) environment.

For more information, see:

- Datasource Navigator and Datasource Explorer
- DBArtisan Toolbars and Menus
- DBArtisan Dialog Boxes
- Customizing the DBArtisan Display
- Parallel Query Execution/multithreading
- Describe Window
- DBArtisan Script Library
- DBArtisan Wizards
- DBArtisan Editors

Datasource Navigator and Datasource Explorer

By default, the first time you start DBArtisan, a single pane window is displayed on the left of the main application window. The Datasource Navigator provides a hierarchy that descends from the datasource level to the level of individual database objects. It provides a visual method for navigating or drilling down through a datasource. From the Navigator, you can connect to and disconnect, from datasources as well as browse, filter, and initiate other actions against objects in all supported datasources on your network.

The Datasource Explorer provides an alternative to the Datasource Navigator. It's a two-paned window. Like the Navigator, it provides a tree-view of your datasources. When you select an object on the Datasource Explorer, you can view corresponding data in the right pane. The Explorer is designed to give you an overall picture of the contents and configuration of your datasources.

You hide and display the Navigator and Explorer using **View** menu commands. As you progress along your DBArtisan learning curve, you will find the each has it's own benefits or that you have a preference. For details on the Navigator and Explorer and datasource management, in general, see the following topics:

- Datasource and Server Management
- Using the Datasource Explorer/Navigator

Parallel Query Execution/multithreading

DBArtisan supports parallel query execution or multithreading.

DBArtisan accelerates the development process by executing multiple queries simultaneously using separate operating system threads and improving response time because most operations can start immediately and return control to the primary application thread.

DBArtisan lets you run multiple, concurrent queries against the same datasource.

DBArtisan employs a database connection pooling technique that dynamically spawns connections as needed and disconnects them when the associated query has completed.

DBArtisan increases throughput by automatically assigns query threads to separate processors if your system has multiple CPUs. When you execute parallel queries against different database servers using a multi-processor computer, you experience near linear gains in throughput.

Describe Window

The Describe Window is a floating window that lets you view procedures, tables, views, and functions (Oracle and IBM DB2 for Linux, Unix, and Windows only). In the Describe Window you can view columnar information (for tables and views) or input parameter information (for procedures and functions).

Opening the Describe Window

DBArtisan offers three ways to open the Describe window:

1 In an editor, right-click an object and then click **Describe**.

2 On the **View** menu, select an object and then click **Describe**.

3 On the right pane of the Explorer or in an editor, select an object and then press CTRL+D.

Using the Describe Window

In the Describe window:

1 Click the **Name** list and then click a name to view a list of types of objects in the database.

2 Click the **Owner** list and then click an owner to view a list of all owners of objects in the database.

Click the **Type** list and then click a type to view columnar information (for tables and views) or input parameter information (for functions and procedures).

DBArtisan Script Library

The Script Library provides a drag-and-drop library interface of all supported DBMS syntax, SQL syntax, built-in functions, optimizer hints, and SQL-conditional syntax. Additionally, it provides the ability to create custom folders to store commonly-used code for quick and efficient access, as needed.

To open the Script Library:

1 Select **View > Script Library**. The **Script Library** window opens.

2 Expand the **Microsoft** node and then expand the **Schema** sub-node.

3 Right-click the **Procedures** node and select **Open**. The selected code opens in the SQL Editor window and is ready for execution.

4 Right-click in the editor window and select **Close** from the context menu.

To add a custom folder to the Script Library

1 Right-click the **Script Library** folder.

2 Select **New Folder** from the context menu. A new folder is added to the **Script Library** folder.

To close the Script Library window:

1 Select **View > Script Library**.

DBArtisan Toolbars and Menus

DBArtisan offers multiple ways to navigate through the application:

- Main Menu
- Toolbars
- Shortcut Menus

For more information on customizing these items, see Customizing General User Interface Appearance.

Main Menu

You can access DBArtisan's features from the Main Menu by clicking the menu name, and then selecting the target command on the sub-menu. The menus are context-sensitive, changing as you perform tasks. DBArtisan menu items include:

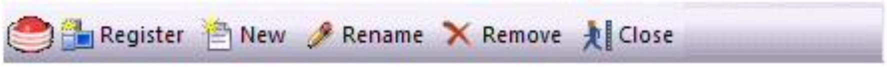
Menu Item	Description
File	Create, open, close, print and save script files and result sets. Set application options and defaults. Note: When you open New, you have easy access to a new ISQL window, the DDL Editor, a Java file, or the Active Script.
Datasource	Create, modify, select, manage, connect to, and disconnect from datasources. Access database object editors. Access session recording for Tivoli integration.
Utilities	Access utilities to extract schema, migrate schema, export and import data, execute scripts, search databases, monitor process activity and locks, schedule tasks, and restore, manage, and backup databases.
Logfile	Activate/deactivate, open, flush, and annotate the DBArtisan application log.
View	Configure DBArtisan to hide or show toolbars, status bars, and windows. Access full screen mode.
Tools	Access customizable tools menu. Access installed DBMS connectivity utilities and server documentation.
Analyst	Opens the Space Analyst, Capacity Analyst, and Performance Analyst.
Bookmarks	Access and manage bookmarks.
Window	Cascade and tile open windows. Arrange icons. Toggle between open windows.
Help	Lets you access context-sensitive Help, link to Embarcadero on the Web, and see license information when you click **About..**.
Edit	Edit and manipulate the text in your SQL scripts. Available only when an ISQL Window is selected.
Query	Execute and set query options for a SQL script. Available only when an ISQL Window is selected.
Format	Format the contents of result sets. Available only when a Result Window is selected.

For more information, see **DBArtisan Toolbars and Menus**.

Toolbars

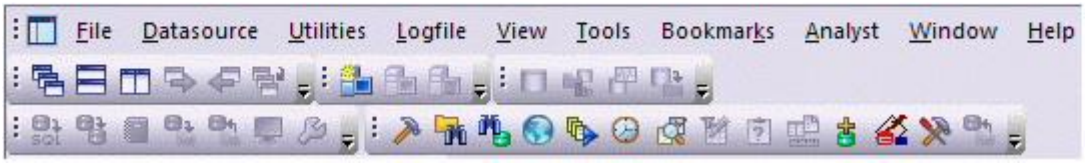
DBArtisan supports two classes of toolbars:

- Feature-specific toolbars appearing at the top of the window or panel of tools such as the Datasource Explorer and Datasource Navigator, Object editors, the ISQL editor, and the Analyst tools:



For information on the functions provided by feature-specific toolbars, see the documentation for the associated feature.

- Optional-display toolbars, appearing in one or more rows below the menu bar.



::

You can hide or display the individual optional-display toolbars and they change to reflect the element of the application you are using. These toolbars contain buttons that are the fastest way to access commonly used features of DBArtisan.

The following topics provide information you need to work with optional-display toolbars:

- Using Toolbar Viewing Options
- Understanding the Shortcuts Available on Specific Toolbars

Note: You can set preferences as to the default toolbars displayed and their appearance. For details, see Customizing General User Interface Appearance.

Using Toolbar Viewing Options

DBArtisan offers standard Windows toolbar options such as docking, floating, and positioning toolbars.The only application-specific viewing option is the hiding or display of the individual toolbars.

To hide or display a toolbar

1 On the **View** menu, select **Toolbars** and then select the specific toolbar you want to display or hide.

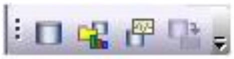
For information on the toolbars available and the commands available on each, see Understanding the Shortcuts Available on Specific Toolbars.

Understanding the Shortcuts Available on Specific Toolbars

The following optional-display toolbars are available using the **Toolbars** command on the **View** menu.

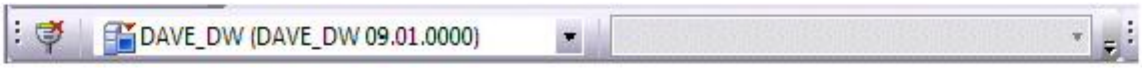
Analyst Series Toolbar

This toolbar offer shortcuts that open the Space Analyst, Capacity Analyst, and Performance Analyst tools.



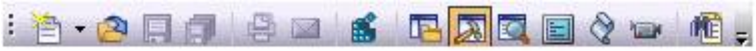
Datasource Toolbar

This toolbar offers shortcuts that let you connect to or disconnect from a datasource, select a datasource and for Sybase ASE and SQL Server, select a specific database.



Main Toolbar

This toolbar has common application functions.

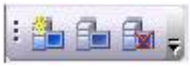


It offers the following shortcuts:

- New
- Open
- Save
- Save All
- Print
- Send Mail
- Options
- New Datasource Explorer
- Output
- Describe Window
- Full Screen
- Log SQL
- Session Recording
- Find Object

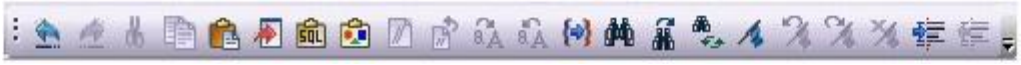
Registration Toolbar

This toolbar offers shortcuts to registering a datasource, editing a datasource registration, and unregistering a datasource.



SQL Edit Toolbar

This toolbar offers functions used in editing SQL:

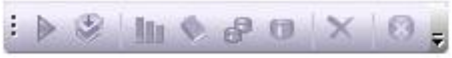


It offers the following shortcuts:

- Undo
- Redo
- Cut
- Copy
- Paste
- Insert File
- Paste SQL Syntax
- Paste SQL Statement
- Comment
- Undo Comment
- Upper Case
- Lower Case
- Match Token
- Find
- Repeat
- Replace
- Toggle Bookmark
- Previous Bookmark
- Next Bookmark
- Clear Bookmark
- Increase Indent
- Decrease Indent

SQL Profiler Toolbar

This toolbar offer functions used when working with the Oracle Profiler.



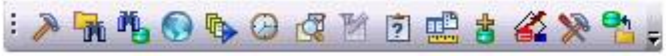
It offers the following shortcuts:

Start Profiler Flush Profile Run Summary Run Detail

Unit Summary Unit Detail Clear Profile Tables Stop Profiling

Tools Toolbar

This toolbar offers functions otherwise available on the **Tools** menu.



It offers the following shortcuts:

Customize Tools Menu Find in Files Database Search Script Execution Facility

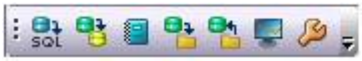
File Execution Facility Scheduler Visual Difference Edit Data

Build Query Table/Index Size Estimator SQL* PLus Code Workbench

Import Data From File

Utilities Toolbar

This toolbar offers functions otherwise available on the **Utilities** menu.



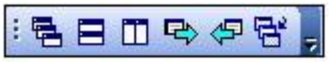
It offers the following shortcuts:

Schema Extraction Schema Migration Shema Publication Data Unload

Data Load Database Monitor Server Configuration

Windows Toolbar

This toolbar offers functionality for working with windows.



It offers the following shortcuts:

Cascade Tile Windows Horizontally Tile Windows Vertically Next

Previous Windows

Customizing General User Interface Appearance

DBArtisan lets you choose from a set of general visual application styles, dictate hiding or display of particular items, and select preferences for specific user interface elements.

To customize the general look and feel of DBArtisan

1 On the **View** menu, select **Toolbars > Customize**. The **Customize** dialog opens.

2 Use the following table as a guide to understanding and setting options on tabs of the Customize dialog:

Tab	Settings and tasks	
Toolbars	Select the toolbars you want to display in the application. For information on the toolbars available, see Understanding the Shortcuts Available on Specific Toolbars.	
Application Visual Style	Select a visual style such as Microsoft Windows XP or one of the .NET options from the dropdown. Depending on your selection, the following options may or may not be enabled: **Use default WIndows XP colors**, **OneNote style tabs**, **Docking Tab Colors**, **Allow MDI Tab Swapping**, **Enable Smart Docking**, **Enable Tab Menu**, and **3D Rounded Docking Tabs**.	
	Menu animations	Lets you specify a menu animation style of UNFOLD, SLIDE, or FADE.
	Menu Shadows	Displays shadowed 3D effects.
Tools	Lets you define external applications to run from the **Tools** menu of DBArtisan: the text displayed on the Tools menu command (**Menu contents**), the path and file name of the executable (**Command**), optional **Arguments**, and an optional **Initial DIrectory**.	
Keyboard	**Category**	Select a general category for a hot key for the command.
	Commands	Select a hot key command, based on the general category.
	Description	Displays the command description.
	Set Accelerator for	Select application area where you want new hot key to be active.
	Current Keys	Displays current hot key.
	Press New Shortcut Key	Press keyboard key or an unassigned F key.
Options	**Show ScreenTips on toolbars**	Select to display a ScreenTip when you hover your mouse over a button. For example, when you hover your mouse over the New button, DBArtisan displays the ScreenTip "New."
	Show shortcut keys in ScreenTips	Select to display a shortcut key in the ScreenTip when you hover your mouse over a button. For example, when you hover your mouse over the New button, DBArtisan displays the ScreenTip "New (CTRL+N)."

3 Click **Apply** at any time to implement any changes you have made and when finished, click **Close**.

Shortcut Menus

DBArtisan incorporates shortcut menus to give you another way to access object functionality. These menus mirror the functionality that you can access from application toolbars or the main menu.

Accessing Shortcut Menus

1 On Desktop, right-click DBArtisan.

DBArtisan opens the shortcut menu.

2 Click the target shortcut menu command.

For more information, see **DBArtisan Toolbars and Menus**.

Keyboard Commands

DBArtisan provides a number of Keyboard commands that expedite ISQL Editor development. The table below describes the available keyboard commands:

Category	Action	Keyboard shortcut
General editing	Delete one character to the left	BACKSPACE
	Delete one character to the right	DELETE
	Cut selected text to the Clipboard	CTRL+X
	Undo the last action	CTRL+Z
	Redo the last undo operation	CTRL+Y
	Copy text	CTRL+C
	Paste the Clipboard contents	CTRL+V
Extending selections	One character to the right	SHIFT+RIGHT ARROW
	One character to the left	SHIFT+LEFT ARROW
	To the end of a word	CTRL+SHIFT+RIGHT ARROW
	To the beginning of a word	CTRL+SHIFT+LEFT ARROW
	To the end of a line	SHIFT+END
	To the beginning of a line	SHIFT+HOME
	One line down	SHIFT+DOWN ARROW
	One screen up	SHIFT+PAGE UP
	To the beginning of a document	CTRL+SHIFT+HOME

	To the end of a document	CTRL+SHIFT+END
	To include the entire document	CTRL+A
Moving the insertion point	One character to the left	LEFT ARROW
	One character to the right	RIGHT ARROW
	One word to the left	CTRL+LEFT ARROW
	One word to the right	CTRL+RIGHT ARROW
	Up one line	UP ARROW
	Down one line	DOWN ARROW
	To the end of a line	END
	To the beginning of a line	HOME
	Up one screen (scrolling)	PAGE UP
	Down one screen (scrolling)	PAGE DOWN
	To the end of a document	CTRL+END
	To the beginning of a document	CTRL+HOME
Bookmarks	Toggle bookmark on/off	CTRL+F2
	Go to next bookmark	F2
	Go to previous bookmark	SHIFT+F2
Splitter windows	Go to next pane	F6
	Go to previous pane	SHIFT+F6
Debugger operations	Start Debugging	CTRL+F5
	Stop Debugging	SHIFT+F5
	Step Over	F10
	Step Into	F11
	Run to Cursor	CTRL+F10
	Step Out	SHIFT+F11
	Describe from Cursor	CTRL+D
	Insert or Remove Breakpoint	F9
	Toggle (Enable or Disable) Breakpoint	CTRL+F9

	Edit Breakpoint	ALT+F9
	Go	F5
	Restart	CTRL+SHIFT+F5
Debugger windows	Open or Close Watch Window	ALT+3
	Open or Close Variables Window	ALT+4
	Open or Close Call Stack Window	ALT+5
	Open or Close Dependency Tree Window	ALT+6
SQL Window	Go to the Result tab	CTRL+ALT+R
	Go to the Query tab	CTRL+ALT+Q

For more information, see **DBArtisan Toolbars and Menus**

Customizing the Tools Menu

The **Customize Tools Menu** dialog lets you add up to sixteen commands to the Tools menu. You can associate any program that running on your local machine. You can use this feature to associate frequently used programs and commands, thereby adding functionality directly to the DBArtisan application. In addition, you can specify arguments for any command that you add to the **Tools** menu. You can use these arguments to decipher how you want to execute commands. DBArtisan includes many pre-defined arguments that let you customize the application.

During the initial installation, DBArtisan automatically adds the following items (if installed locally) to the **Tools** menu.

- ODBC Administrator
- Oracle Service Name Manager
- Sybase Open Client/Server Config
- Sybase Directory Service Edit (DSEDIT)
- SQL Server Network Library Config
- SQL Server Books Online
- DB2 Client Configuration Assistant

For more information, see **DBArtisan Toolbars and Menus**.

Customize User Tools Menu Dialog: DBArtisan lets you customize the **Tools** menu.

The table below describes the options and functionality on the **Customize User Tools Menu** dialog:

Option	Description
Menu Contents	In the box, click the target command.
Menu Text	In the box, type the name of the tool as you want to appear on the Tools menu.
Command	In the box, type the path and name of the program for the new tool or click **Browse** to locate the program. For example, C:\WINDOWS\NOTEPAD.EXE
Arguments	To pass an argument to the command, in the box, type any argument or click the Arrow, and then click the target argument. NOTE: To display a menu of pre-defined arguments, click the drop-down arrow next to the Argument box, and then select an argument from the list to insert the syntax into the box.
Initial Directory	To specify an initial directory, in the box, type the path and name of the file directory or click **Browse** to locate the file directory.
Add Button	Click to add a new command to Tools menu.
Remove Button	In the Menu Contents box, click the command you want to remove, and then click the button.
Move Up Button	Click to move the command up one position in the menu.
Move Down Button	Click to move the command down one position in the menu.

Completing the Customize User Tools Menu dialog

To complete the **Customize User Tools Menu** dialog, do the following:

1 On the **Tools** Menu, select **Customize**.

DBArtisan opens the **Customize User Tools** dialog.

2 Select options.

3 Click **OK**.

For more information, see **DBArtisan Toolbars and Menus**.

Specifying Arguments

DBArtisan provides a number of pre-defined arguments that you can pass to programs that you have added to the **Tools m**enu. You can specify an argument to be passed to a program for newly added commands by choosing one of DBArtisan's pre-defined arguments or typing in your own arguments. The table below describes the pre-defined arguments:

Argument	Description
$$FilePath$$	The complete filename of the current source (defined as drive+path+filename); blank if a non-source window is active.
$$FileDir$$	The directory of the current source (defined as drive+path); blank if a nonsource window is active.
$$FileName$$	The filename of the current source (defined as filename); blank if the non-source window is active.
$$FileExt$$	The filename extension of the current source; blank if a nonsource window is active.
$$CurLine$$	The current pointer line position within the active window.
$$CurCol$$	The current pointer column position within the active window.
$$CurText$$	The current text (the word under the current pointer position, or the currently selected text, if there is one.)
$$CurDir$$	The current working directory (defined as drive+path.)
$$CurDatasource$$	The name of the current datasource as defined in DBArtisan.
$$CurUserID$$	The name of the current datasource user.
$$CurPwd$$	The current datasource password.
$$CurConString$$	The current connection string or server name.

Note: Arguments are case-sensitive.

The table below describes examples for using the pre-defined arguments and command-line arguments:

Command	Argument	Description
NOTEPAD.EXE	$$FilePath$$	Launches Microsoft Notepad displaying the contents of the $$FilePath$$ argument.
ISQL.EXE	-U$$CurUserID$$ -P$$CurPwd$$ -S$$CurConString$$ -i$$FilePath$$	Launches ISQL, automatically connects to the current datasource using the current user name and password, and executes the contents of $$FilePath$$.
SQLPLUS.EXE	$$CurUserID$$/$$CurPwd$$@$$CurConString$$ @$$FilePath$$	Launches SQL*Plus, connects to the current datasource using the current user name and password, and executes the contents of $$FilePath$$.

Tip: DBArtisan lets you specify an argument in the DBArtisan opens the Customizing General User Interface Appearance.

For more information, see **DBArtisan Toolbars and Menus**.

DBArtisan Dialog Boxes

DBArtisan offers the following common dialogs.

DBArtisan Wizards

DBArtisan offers several wizards that let you create objects without knowing the underlying SQL commands. DBArtisan also offers wizards to perform common tasks.

DBArtisan Editors

DBArtisan stores information about server and object types in object editors. An editor is a tabbed dialog box that groups related information together. Each editor contains a context-sensitive **Commands** menu with pertinent functionality for the object. Many of the object editors contain a **DDL** tab. This tab displays the underlying target object's SQL.

Tip: The Object Editor tool bar has a refresh button, that lets you refresh Object Editor contents, clear the Object Editors, and log SQL if SQL Logging is turned on.

If an objects has dependent objects, such as tables, triggers, procedures and views, you can view and access their dependent objects in the editor.

For more information, see Modifying objects using editors.

Customizing the DBArtisan Display

DBArtisan lets you customize your display to suit your work habits.

For more information, see:

Full Screen Mode

Active Windows Dialog Box

Navigating DBArtisan Windows

Output Window

Set Sort Columns Dialog Box

Full Screen Mode

DBArtisan is equipped with Full Screen mode capabilities that let you conceal the application framework and take advantage of the entire monitor area. Full screen mode hides any other applications running on the computer and uses every available pixel for DBArtisan. Main menu functionality is accessible through keyboard commands when you use Full Screen Mode.

Activating Full Screen Mode

1 On the **View** menu, select **Full Screen**.

DBArtisan expands the application to fit the entire monitor area, and displays the Full Screen Mode icon as a stand-alone floating toolbar.

Dismissing Full Screen Mode

1 Click **Full Screen**.

DBArtisan reduces the size of the application.

Tip: If you closed the **Full Screen Mode** toolbar, right-click the top of the DBArtisan desktop to display the full screen.

For more information, see **Customizing the DBArtisan Display**.

Active Windows Dialog Box

DBArtisan’s multi-tasking environment lets you open multiple active windows. DBArtisan's **Active Windows** dialog lets you see all the open windows in the application workspace. DBArtisan displays active windows as tabs in the Datasource Explorer. Each tab represents a separate window that lets you execute, create or view the objects or schema in your database.

Tip: DBArtisan lets you use a mouse with a wheel to navigate in active windows.

Opening the Active Windows dialog

1 On the **Window** menu, select **Windows**.

DBArtisan opens the **Active Windows** dialog and any open windows in the application workspace.

The table below describes the options and functionality on the **Active Windows** dialog:

Option	Description
Activate	Sets the focus on the selected window and closes the **Active Windows** dialog.
OK	Closes the **Active Windows** dialog and makes any changes you have made to the windows in the workspace.
Save	Saves the contents of the selected window. If you are saving a file, DBArtisan prompts you to provide a name and location for the file.
Close Window	Closes the selected window. If you have not saved the contents of the window, DBArtisan prompts you to do so.
Help	Initiates and displays this Help topic in the DBArtisan Help.

For more information, see **Customizing the DBArtisan Display**.

Navigating DBArtisan Windows

DBArtisan offers functionality through the Windows toolbar that lets you:

- Tile windows vertically or horizontally.
- Cascade open windows.
- Minimize all windows and move to the next active window.

Tip: DBArtisan lets you use a mouse with a wheel to navigate in active windows.

The table below describes the commands available on the **Windows** menu:

Command	Description
Window	Activates the **Active Windows** dialog.
Cascade	Layers all active windows on the desktop so that you can view all active windows simultaneously.
Tile Horizontal	Tiles all active windows on the desktop horizontally.
Tile Vertical	Tiles all active windows on the desktop vertically.
Next	Moves the focus of the application to the next active window.
Previous	Moves the focus of the application to the previous active window.

For more information, see **Customizing the DBArtisan Display**.

Output Window

DBArtisan has a separate window that captures all messages returned by the server and to log entries about the progress of operations started by the application.

The Output Window includes the following tabs:

- General
- Find In Files 1
- Find In Files 2

The Output Window lets you do the following to server messages:

- View
- Save
- Print
- Copy
- Flush

For more information, see **Customizing the DBArtisan Display**.

CONFIGURING THE OUTPUT WINDOW

Basic Output window options you can configure include:

- Floating the Output Window (with Smart Docking)
- Docking the Output Window
- Undocking the Output Window
- Hiding the Output Window
- Displaying the Output Window

Floating the Output Window (with Smart Docking)

This is the default view when you first open DBArtisan.

1 Place your pointer on the Output Window and drag it to another location.

Arrows appear and you can move the window to the top, bottom, or sides of the view within DBArtisan or to another location on your desktop.

2 To change the location of the Output Window, repeat the first step.

3 To disable the feature, right-click and choose **Docking** or **Hide**.

Docking the Output Window

1 Place your pointer on the Output Window.

2 Right-click, and then click **Docking View**.

DBArtisan docks the Output Window to the bottom of the application frame.

Undocking the Output Window

1 Place your pointer on the Output Window.

2 Right-click, and then click **Docking View**.

DBArtisan floats the Output Window.

Hiding the Output Window

1 On the **View** menu, select **Output**.

DBArtisan hides the Output Window.

Displaying the Output Window

1 On the **View** menu, select **Output**.

DBArtisan displays the Output Window.

For more information, see **Customizing the DBArtisan Display**.

SAVING OUTPUT WINDOW MESSAGES

DBArtisan lets you save messages in the Output window to a file.

1 Right-click a message in the Output Window, and then click **Save**.

DBArtisan opens the **Save As** dialog.

2 In the **File Name** box, type the location and name of the file.

DBArtisan defaults the file extension to.msg.

3 Click **Save**.

DBArtisan saves the file.

For more information, see **Customizing the DBArtisan Display**.

PRINTING OUTPUT WINDOW MESSAGES

DBArtisan lets you print the messages you see in the Output window.

1 Right-click a message in the Output Window, and then click **Print**.

DBArtisan opens the **Print** dialog.

2 Configure your print job.

3 Click **Print**.

DBArtisan prints the file.

For more information, see**Customizing the DBArtisan Display**.

COPYING OUTPUT WINDOW MESSAGES

DBArtisan lets you copy all or part of the contents of the Output window to the clipboard.

1 Select the target Server Messages.

2 Right-click, and then click **Copy**.

DBArtisan copies the selected messages to the Windows clipboard.

3 Paste the text into other Windows applications.

For more information, see**Customizing the DBArtisan Display**.

CLEARING OUTPUT WINDOW MESSAGES

DBArtisan lets you clear server messages from the Output Window.

1 Right-click a message in the Output Window, and then click **Clear**.

DBArtisan clears the server messages.

For more information, see **Customizing the DBArtisan Display**.

Set Sort Columns Dialog Box

The **Set Sort Columns** dialog lets you sort multiple columns, and sort column identification.

Completing the Set Columns dialog

To complete the **Set Columns** dialog, do the following:

1 In the right pane of the application, right-click a column heading and select **Sort Multiple Columns**.

DBArtisan opens the **Set Sort Columns** dialog.

2 In Column Name select the column(s) to use to sort the information in the right pane of the application.

3 Click the right arrow to move the column(s) to the **Column Name** box.

4 Click the up and down arrows to change the order of the sort columns.

DBArtisan Tutorial exercises

The exercises that follow walk you through DBArtisan's major functional areas. After completing these exercises, you will have the foundation you need to explore the many features and benefits of DBArtisan. You'll have learned how to competently manage the major database administration disciplines using DBArtisan's standard cross-platform console. In addition, you will have a solid understanding of DBArtisan's more advanced Space, Performance, and Capacity management capabilities.

This guide is divided into 10 sessions:

- Session 1: Getting Started with DBArtisan
- Session 2: Object and Schema Management
- Session 3: Database Security Management
- Session 4: Space Management
- Session 5: SQL Management
- Session 6: Job Management
- Session 7: Data Management
- Session 8: Performance Management
- Session 9: Capacity Management
- Session 10: General Utilities and Tools

You can use this basic tutorial as a roadmap of product highlights, but also to help you find your own path to explore DBArtisan.

Once you've started, you can select **Help Topics** from the **Help** menu to find many additional resources that complement and build on many of the activities shown in this tutorial.

Session 1: Getting Started with DBArtisan

Your first tasks in getting familiar with DBArtisan are as follows:

- Starting the DBArtisan Application
- Registering Cross-Platform Datasources to DBArtisan
- Setting up the Datasource Explorer
- Exploring General Usability Features

Starting the DBArtisan Application

How you start DBArtisan depends on the type of application you are evaluating:

- InstantOn version – start the application by double-clicking the file you downloaded.
- Fully-installed version – The **Start** menu sequence for DBArtisan is always in the form **Programs > Embarcadero DBArtisan***version identifier***> DBArtisan***version identifier*, where *version identifier* reflects the version you are running.

To get started

1 Run DBArtisan.

The first time DBArtisan starts, a dialog opens, prompting you to set up datasources. In addition to letting you manually set up individual datasources, a number of more automated methods are available. If you have installed and used other Embarcadero tools, DBArtisan can find any active datasources being used by those tools. In addition, DBArtisan provides a Discover Datasources feature that automatically searches the DBMS configuration files on your system for datasources that are not currently registered. Since other Embarcadero tools let you export datasource definitions to file, you also have the option of importing these definitions.

2 For the purpose of this tutorial, click **Cancel**. You will be registering a datasource manually.

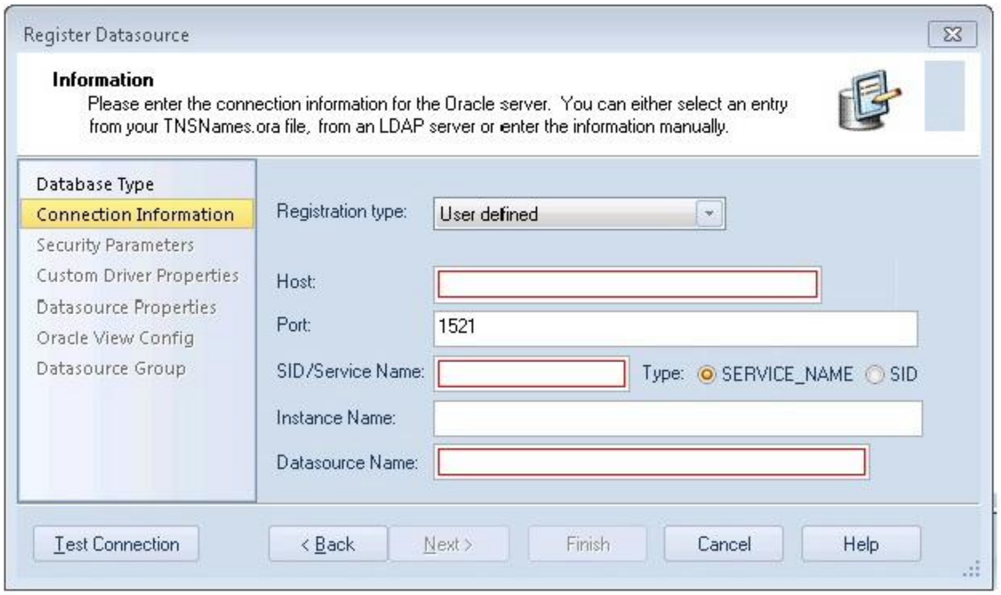
Proceed to Registering Cross-Platform Datasources to DBArtisan.

Registering Cross-Platform Datasources to DBArtisan

The Datasource Registration Wizard walks you through the process of registering a datasource for use with DBArtisan.

1 On the **Datasource** menu, select **Register Datasource**. DBArtisan opens the wizard, initially prompting you for the DBMS type.

2 Select **Oracle** as the database type and then click **Next**. DBArtisan opens the next panel of the Datasource Registration Wizard.



Register Datasource

Information

Please enter the connection information for the Oracle server. You can either select an entry from your TNSNames.ora file, from an LDAP server or enter the information manually.

Registration type: User defined

Host:

Port: 1521

SID/Service Name:

Type: ◉ SERVICE_NAME ○ SID

Instance Name:

Datasource Name:

3 Ensure that **User defined** is selected and provide the **Host** machine name associated with an Oracle datasource.

4 Type a **Port** number. The default is 1521, but you can change it to wherever the Oracle listener is set up.

5 Specify a **Type** of **SERVICE_NAME** or **SID** and enter the corresponding value in the **SID/Service** box.

6 In the **Datasource Name** text box, type **SAMPLE_DATASOURCE** for the purpose of this example.

7 Click **Next**. DBArtisan saves your selections and opens the next panel of the wizard. This is where you enter login credentials.

8 Provide a **User ID** and **Password**.

9 To save and encrypt your password so that you do not have to provide credentials each time you connect, select **Auto-Connect?**.

10 Click **Datasource Group** to display the last panel.

11 In the **Managed Datasources** tree, select the folder corresponding to the type of DBMS you are registering.

12 Click **Finish**. DBArtisan prompts you as to whether you want to connect to the datasource.

13 Click **Yes**.

The same easy-to-use Datasource Registration Wizard is available for directly-supported DBMS platforms as well as JDBC and ODBC datasources. Datasource information is stored in the registry, is editable, and datasources can be deleted. If you run multiple Embarcadero products on the same machine, they can share the same datasource catalog.

A number of datasource catalog storage options are available:

- A local, file-based alternative lets you easily move a datasource catalog between users.
- A remote, file-based alternative lets multiple users share a single catalog. Settings can be changed locally but are not propagated to other users or retained over shutdown/startup

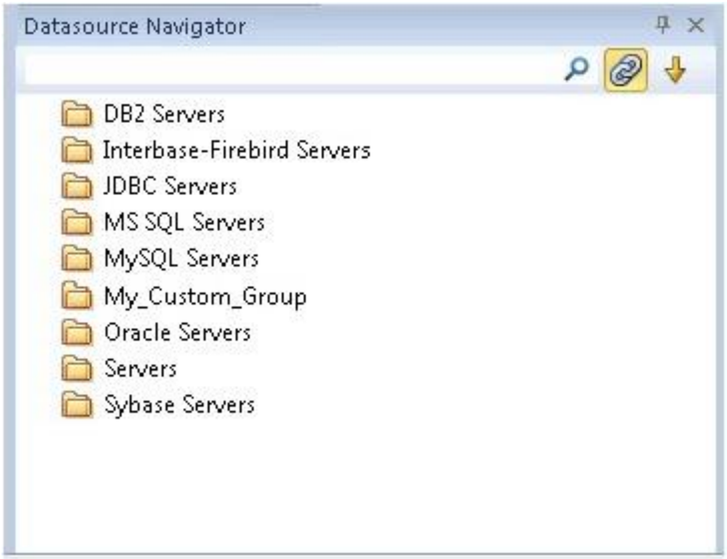
Another way of centralizing datasource definition maintenance is available if you use Embarcadero DB Team Server. Datasource definitions can be stored on DB Team Server and subsequently registered as local datasources. Local connection details for a datasource are synchronized with the Team Server version on each attempt to connect to a datasource.

For details on datasource options, refer to the DBArtisan online Help.

Proceed to Setting up the Datasource Explorer.

Setting up the Datasource Explorer

The first time that you start DBArtisan, the Datasource Navigator occupies the left side of the main window. It is a single pane window that lets you navigate the datasources that you register and initiate actions against databases and database objects.



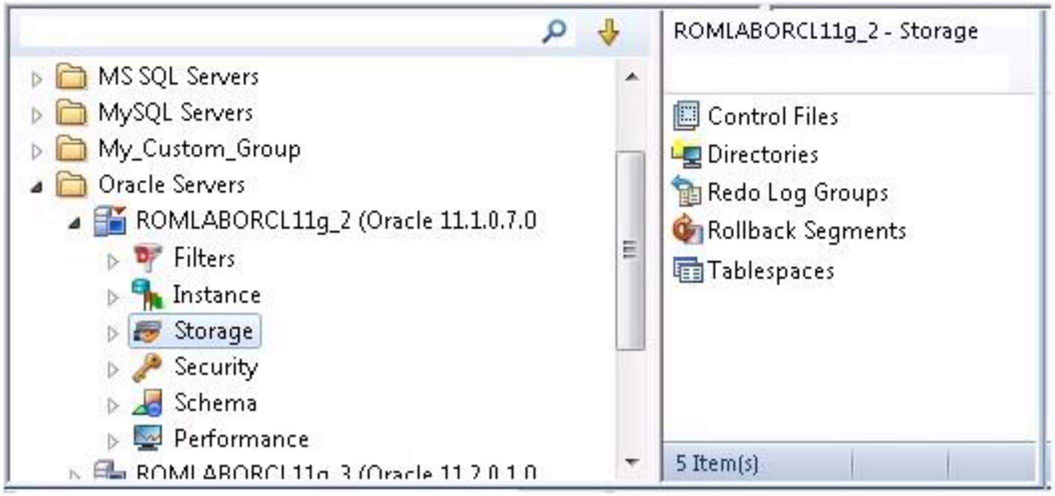
For the following set of exercises, you will be using a similar aid.

To prepare for subsequent exercises

1 Dismiss the Navigator by clicking the close button (**X**) at the top of the Navigator window.

2 Select **View > Activate Explorer**.

The Datasource Explorer is a two pane window that provides most of the same functions as the Navigator.



One key difference is the right pane of the window. It displays details for the node currently selected in the left pane. This might be individual database objects such as tables or primary keys, configuration parameters, and so on.

Before proceeding, familiarize yourself with the **View** menu commands for opening the Navigator and Explorer, and the ways to close those windows. As you proceed with DBArtisan, you will develop a preference for either the Navigator or Explorer, or find that each has benefits that are preferable in different situations.

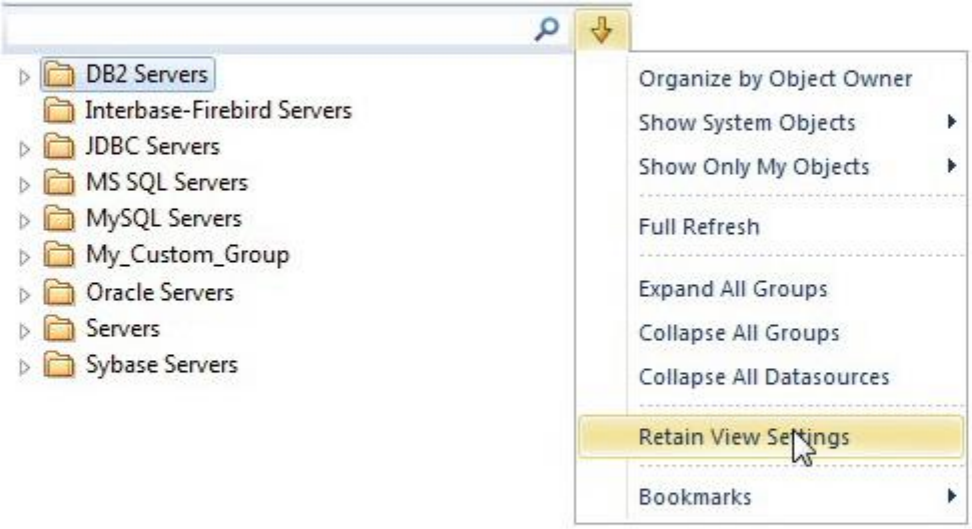
Proceed to Exploring General Usability Features.

Exploring General Usability Features

DBArtisan provides many "user in mind" features that make the product configurable to meet individual needs and preferences. These features are designed to shave time off the tasks that you perform many times on any given working day.

Retaining Datasource Explorer View Settings

1 At the top of the Explorer tree, click to expand the drop-down menu.

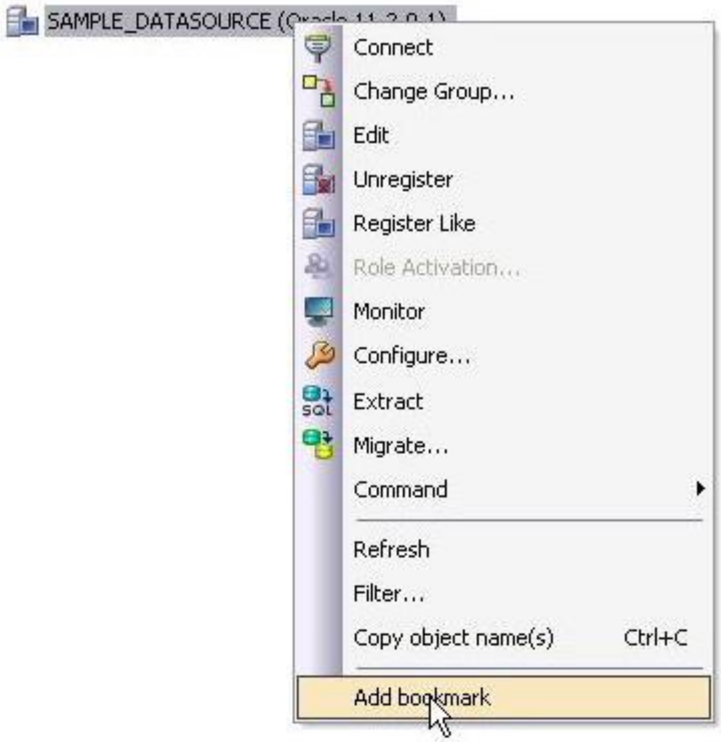


2 Select **Retain View Settings**.

The next time you open DBArtisan, the Explorer appears just as you left it.

Working with Datasource Explorer Bookmarks

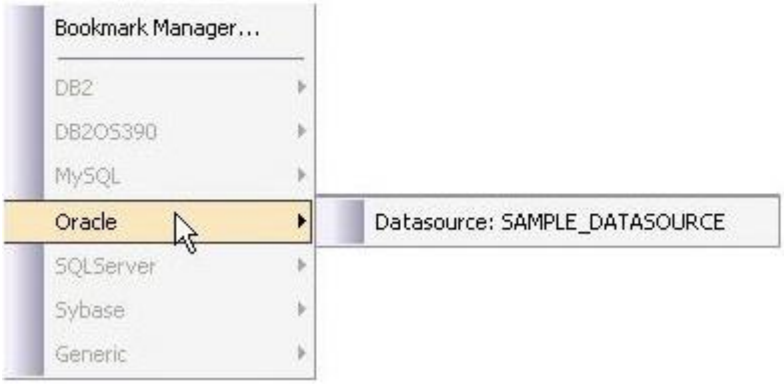
1 In the Explorer tree, right-click the datasource you registered and select **Add bookmark** from the context menu.



DBArtisan opens the **Add Friendly Bookmark Name** dialog.

2 Click **OK**.

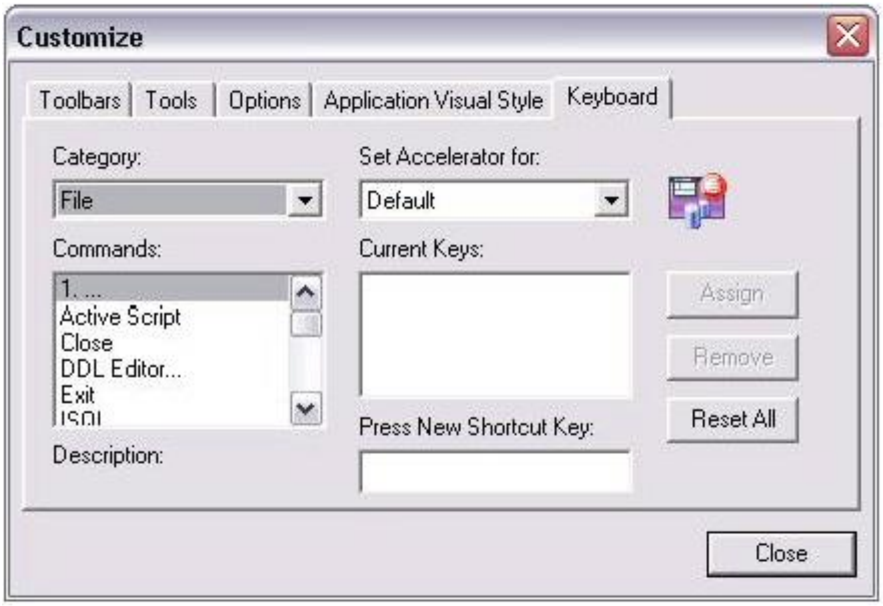
After Bookmarks are defined you can use them to easily navigate to commonly used datasource resources using the dropdown at the top of the Explorer and selecting the **Bookmarks** item.



Setting Keyboard Shortcuts and Hotkeys

1 From the **Tools** menu, select **Customize**. The **Customize** dialog opens.

2 In the **Customize** dialog, open the **Keyboard** tab.



Category: File

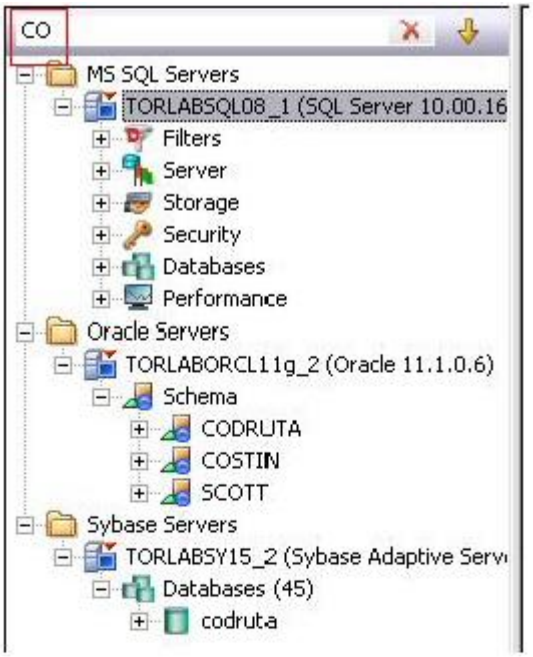
Set Accelerator for: Default

The **Keyboard** tab can be used to set keyboard shortcut hot keys for all areas of DBArtisan functionality.

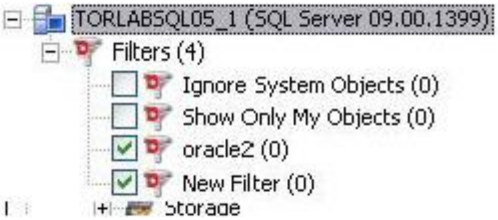
3 Click **Close**.

Datasource Explorer tree filtering

DBArtisan provides a number of ways to filter the Datasource Explorer tree, easing navigation. During a DBArtisan session, you can use the **Filter** box at the top of the Datasource Explorer to display only those objects whose name contains characters typed in the **Filter** box.

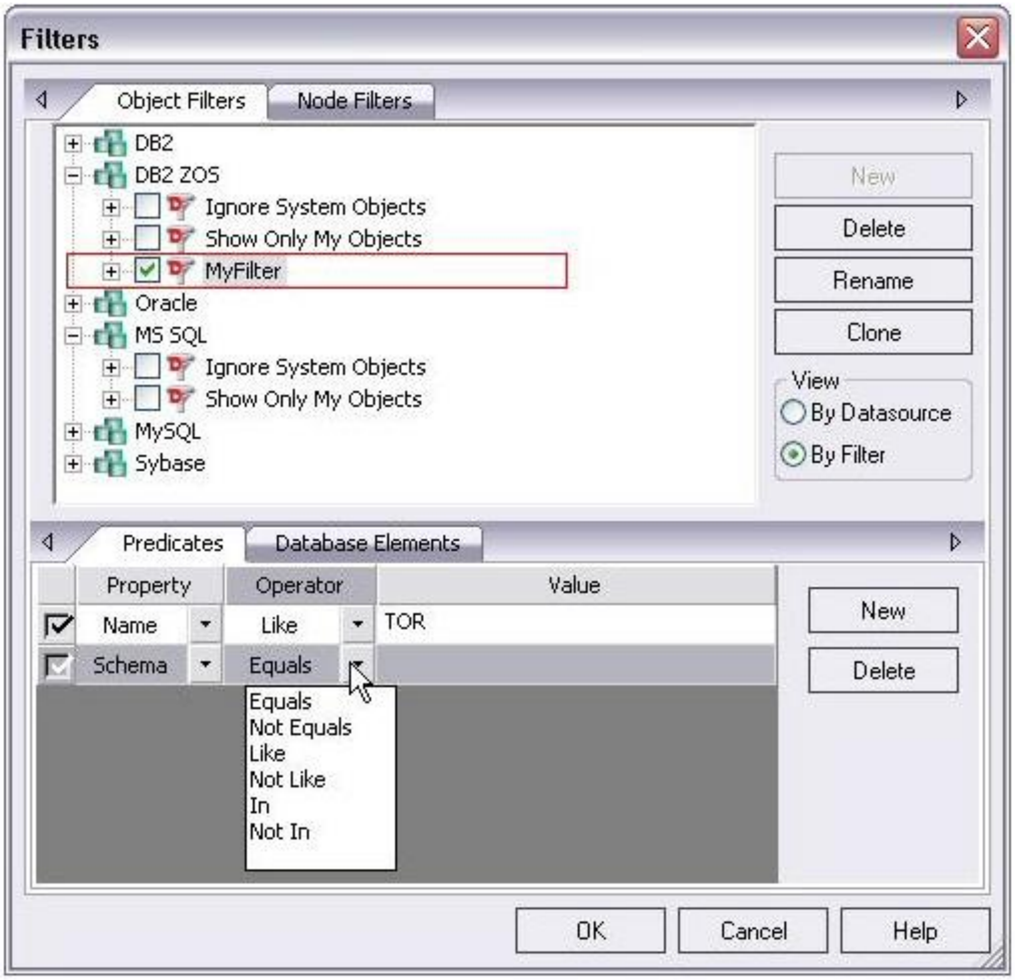


If you are interested in a more permanent filtering method, you can create filters at the DBMS or datasource level. Filters are available under the **Filters** node for each datasource.


TORLABSQL05_1 (SQL Server 09.00.1399)
Filters (4)
☐ Ignore System Objects (0)
☐ Show Only My Objects (0)
☑ oracle2 (0)
☑ New Filter (0)
Storage

Right-clicking in the tree and selecting **Filter** opens the **Filters** dialog. It lets you filter out object types at the DBMS platform level. It also lets you create object filters.

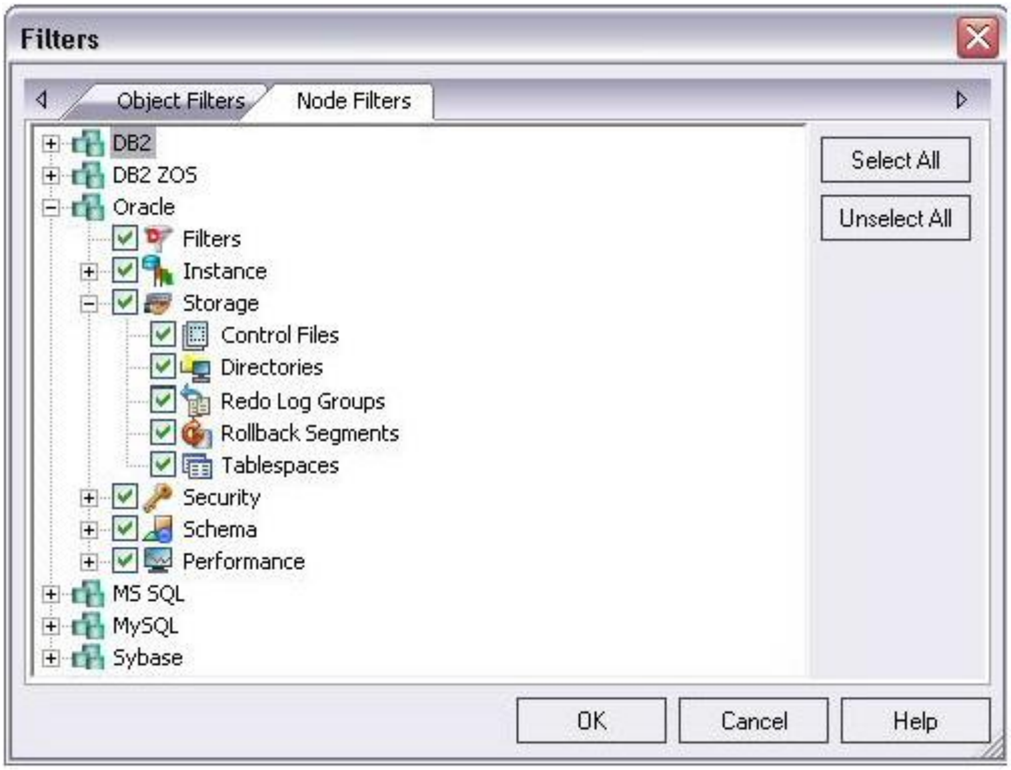
Object filters can be object name-based, schema-based, or some combination thereof. A name-based or schema-based filter consists of one or more ANDed conditions. You also specify the object types to which the filter is to apply.



	Property	Operator	Value
☑	Name	Like	TOR
☑	Schema	Equals	

Object filters are created at the DBMS or datasource level and can be enabled and disabled for individual datasources.

The **Filters** dialog also lets you hide or display nodes based on object type. Node filtering lets you filter out object types for all datasources of the same DBMS type.



In addition to any custom object filters you create, the **Filters** node also provides access to two default filters: **Ignore System Objects** and **Show Only My Objects**. They can be applied at the datasource level and cannot be deleted or modified.

Proceed to Session 2: Object and Schema Management.

Session 2: Object and Schema Management

DBArtisan provides unparalleled database object management capabilities. Its database platform-specific and version-specific graphical object editors and wizards enable you to easily create, drop or alter any of your accessible database objects.

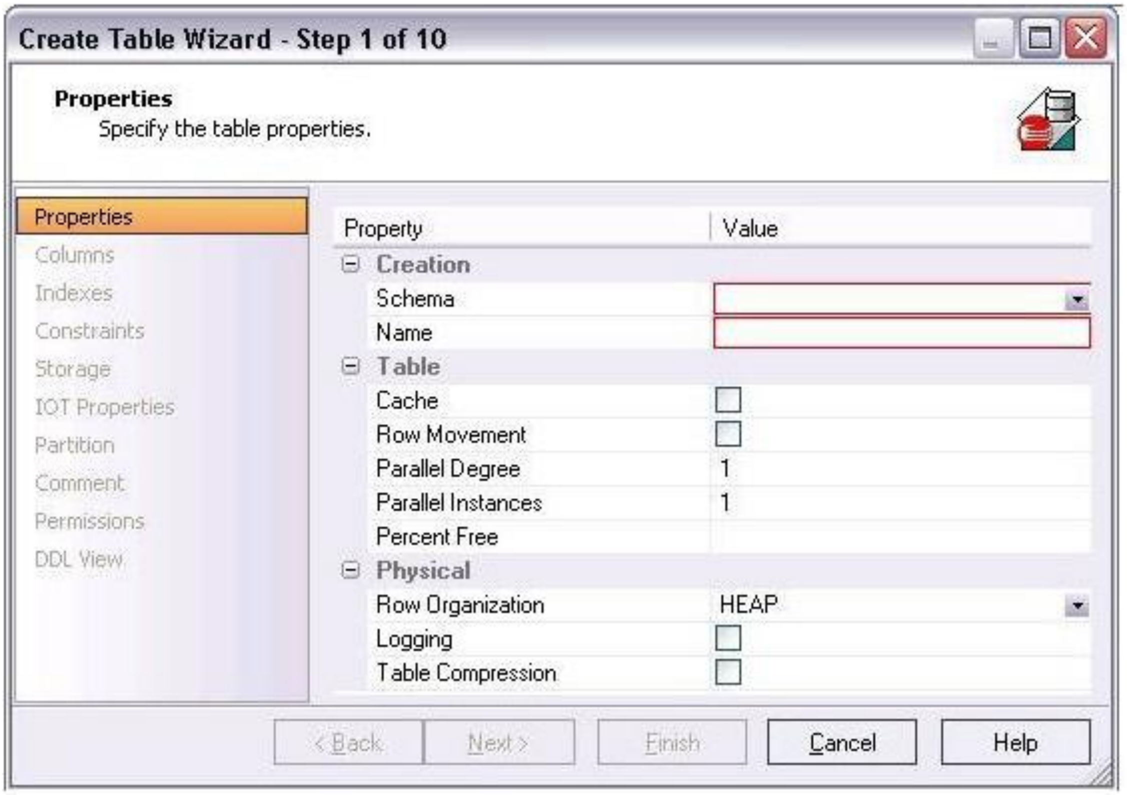
Creating a Table Object

The following example walks you through creating a standard Oracle table object. This concept carries across all of the supported object types, across all of the supported platforms.

1 On the Datasource Explorer, expand the **Schema** node of an Oracle datasource.

2 Right-click the **Tables** node, and then select **Create**.

3 DBArtisan opens the Table wizard and leads you through the process of creating a table object.



Property	Value
Creation	
Schema	
Name	
Table	
Cache	☐
Row Movement	☐
Parallel Degree	1
Parallel Instances	1
Percent Free	
Physical	
Row Organization	HEAP
Logging	☐
Table Compression	☐

4 Complete the wizard panels, and ensure that you create two or more columns in the table.

5 Click **Finish**.

DBArtisan lets you preview any and all generated scripts before you submit them to the database. This is standard for all object-related scripts.

Making changes to an existing table object

Changes to database tables, such as modifying column lengths, inserting new columns, or deleting unneeded ones, can require dropping of a table. This requires knowledge of the underlying object dependencies so that these dependent objects are rebuilt after the table has been recreated. DBArtisan provides the ability to perform "extended" table alterations by constructing a SQL script with the steps necessary to save off the original data, create the new table, and populate it with the original data. Once these steps are complete, all dependent objects are then rebuilt and permissions re-applied.

1 From the Explorer, **Tables** node, select the table you created in the previous example.

2 Double-click the table or from the **Command** menu, click **Open**.

DBArtisan opens the Table Editor. The Table Editor provides access to basic table properties, the list of table columns as well as any constraints, storage parameters, space allocation, partitioning, table dependencies, object privileges, table DDL and other attributes of the table.

3 Ensure that the **Columns** tab is selected and select one of the columns you created in this table you want to modify.

Details for the column are shown in the **Column Attributes** area on the right side of the pane.

4 In the **Width** or **Scale** text box, type a new value.

5 On the Table Editor toolbar, select the **Alter** button.

DBArtisan lets you preview the SQL script before you submit it to the database.

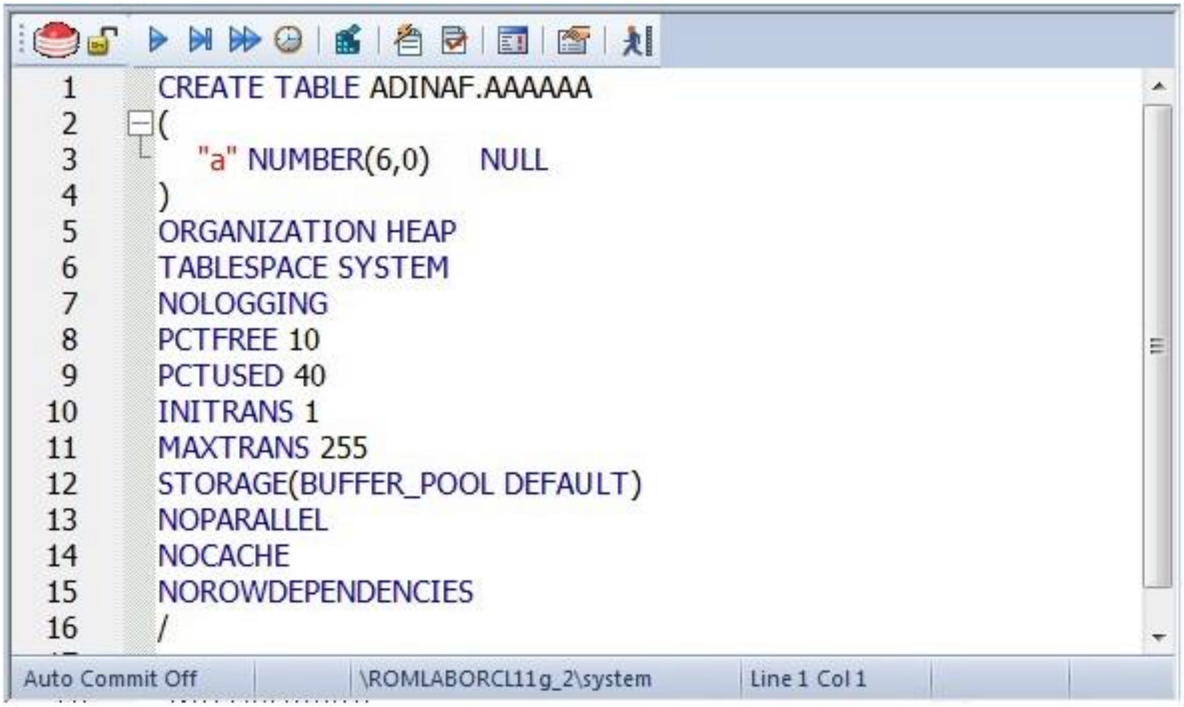
6 Close the Table Editor pane.

Working with Object DDL

DBArtisan allows you to easily extract DDL for single or multiple objects using several methods. The most straight-forward is described here:

1 On the Explorer, right-click the table you created earlier, and then select **Extract**.

The DDL for all highlighted objects is extracted directly in a DDL Editor where it can be altered, executed and saved to the database, with no intermediary steps required. For example:



```
CREATE TABLE ADINAF.AAAAAA
(
    "a" NUMBER(6,0)     NULL
)
ORGANIZATION HEAP
TABLESPACE SYSTEM
NOLOGGING
PCTFREE 10
PCTUSED 40
INITRANS 1
MAXTRANS 255
STORAGE(BUFFER_POOL DEFAULT)
NOPARALLEL
NOCACHE
NOROWDEPENDENCIES
/
```

2 Close the PL/SQL Editor pane.

Proceed to Session 3: Database Security Management.

Session 3: Database Security Management

DBArtisan can help you efficiently establish and maintain database security. Whether you are managing an existing production database or setting up a new environment, you'll find consistent support across all of the supported platforms.

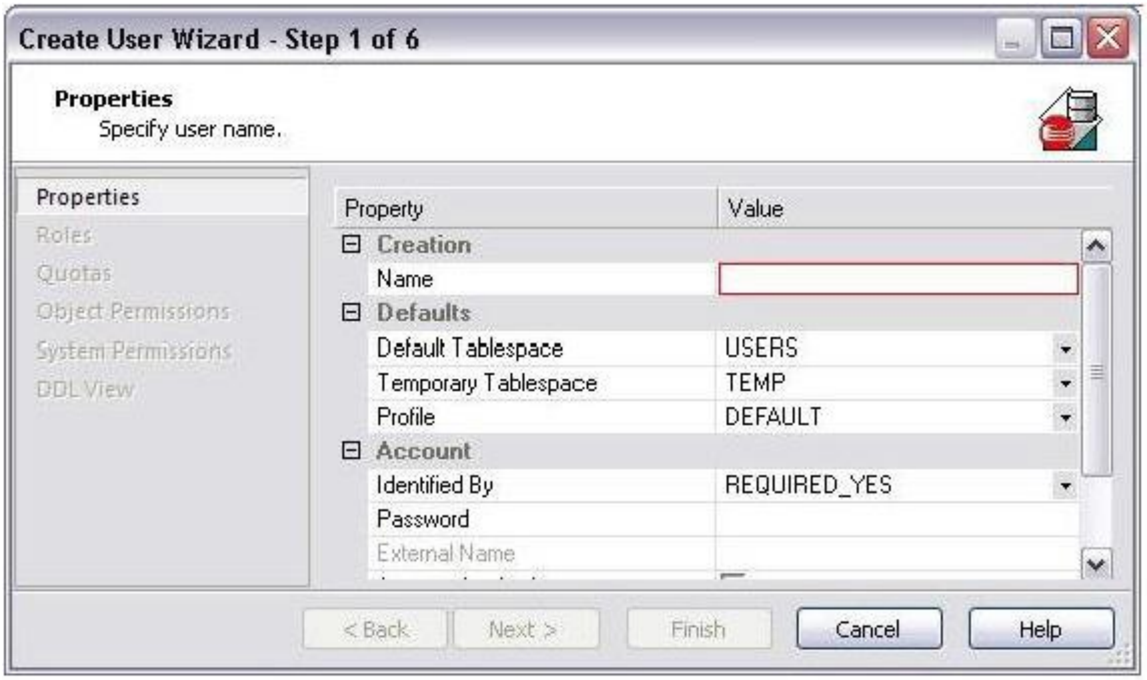
Adding a new database user

While this example focuses on creating a new Oracle user, the same wizard-driven principle applies to all security objects (groups, roles, etc).

1 On the Datasource Explorer, expand an Oracle datasource, and then the **Security** node.

2 On the **Security** node, right-click **Users**, and then click **Create**.

DBArtisan opens the **Create User Wizard** and leads you through the process of adding a user.



Property	Value
⊟ Creation	
Name	
⊟ Defaults	
Default Tablespace	USERS
Temporary Tablespace	TEMP
Profile	DEFAULT
⊟ Account	
Identified By	REQUIRED_YES
Password	
External Name	

3 Provide the information on each panel of the **Create User Wizard** until you reach the **DDL View** panel.

DBArtisan allows you to preview any and all generated scripts before they are submitted to the database. This is standard for all object-related scripts.

4 Click **Execute** to create the new user.

DBArtisan automatically opens the User Editor for the new user. The standard User Editor can be used to manage existing database users as shown below.

Granting and Editing User Privileges

Privileges can be easily granted, revoked, or viewed from within either of two editors within DBArtisan; the User Editor, or the individual object editor (Table, Procedure, etc.). The User Editor provides a tabbed interface, which can be used to view and modify individual attributes of the user.

1 In the User Editor, open the **Object Permissions** tab.

2 Use the **Object Type** dropdown to select a set of objects such as tables or views.

3 Right-click a cell (corresponding to a specific object and a specific permission, such as DELETE), and then click **Grant**.

A distinctive icon is shown in the cell, indicating that this permission is granted. You use a similar process to revoke privileges and perform other permission-based activities.

4 On the Object Editor toolbar, click **Alter** to implement the changes.

Proceed to Session 4: Space Management.

Session 4: Space Management

Managing space is vital to ensuring the availability and performance of your databases. DBArtisan incorporates many built-in space features that let you smartly manage and exploit all aspects of your database's storage. The following examples walk you through a review of DBArtisan's built-in support for reporting Oracle tablespace storage and space data.

- Built-in Space Management
- Advanced Space Management (Oracle and SQL Server only)

Built-in Space Management

While this example is specific to Oracle tablespaces, the same concept applies to all of the supported platforms.

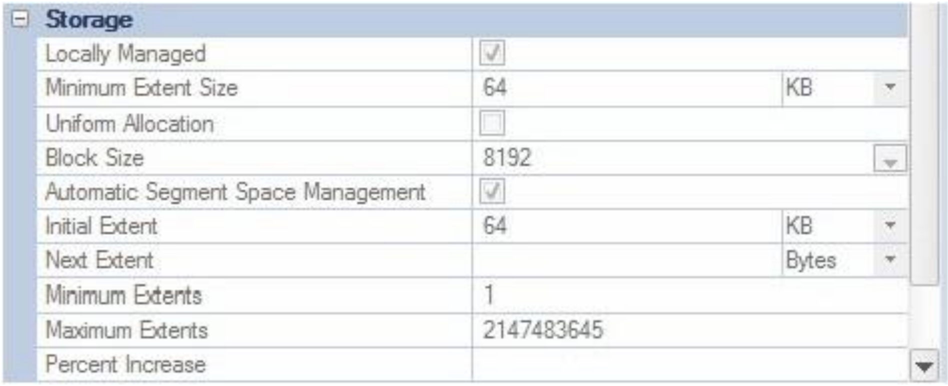
1 On the Datasource Explorer, expand any Oracle datasource.

2 On the Oracle datasource, expand the **Storage** node, and then select **Tablespaces**.

3 Right-click any tablespace listed in the right pane of the Explorer window, and then click **Open**.

Embarcadero DBArtisan opens the Tablespaces Editor.

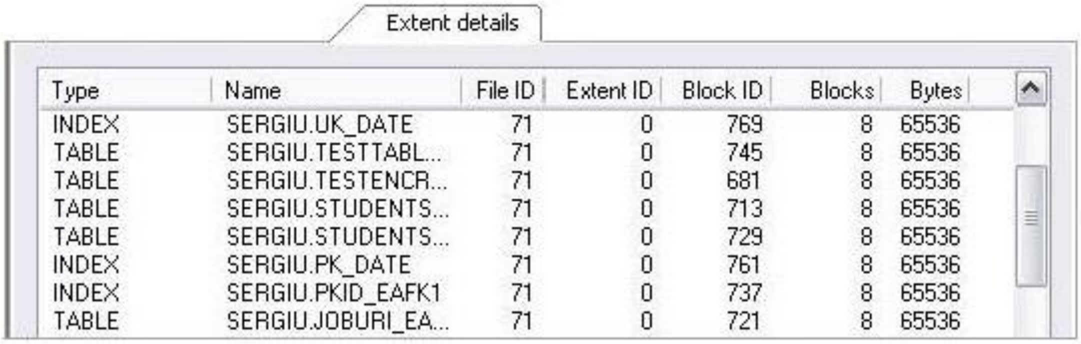
4 Select the **Properties** tab. The **Storage** group on that tab provides a summary of the space settings for the tablespace.



Storage

Locally Managed	☑	
Minimum Extent Size	64	KB
Uniform Allocation	☐	
Block Size	8192	
Automatic Segment Space Management	☑	
Initial Extent	64	KB
Next Extent		Bytes
Minimum Extents	1	
Maximum Extents	2147483645	
Percent Increase		

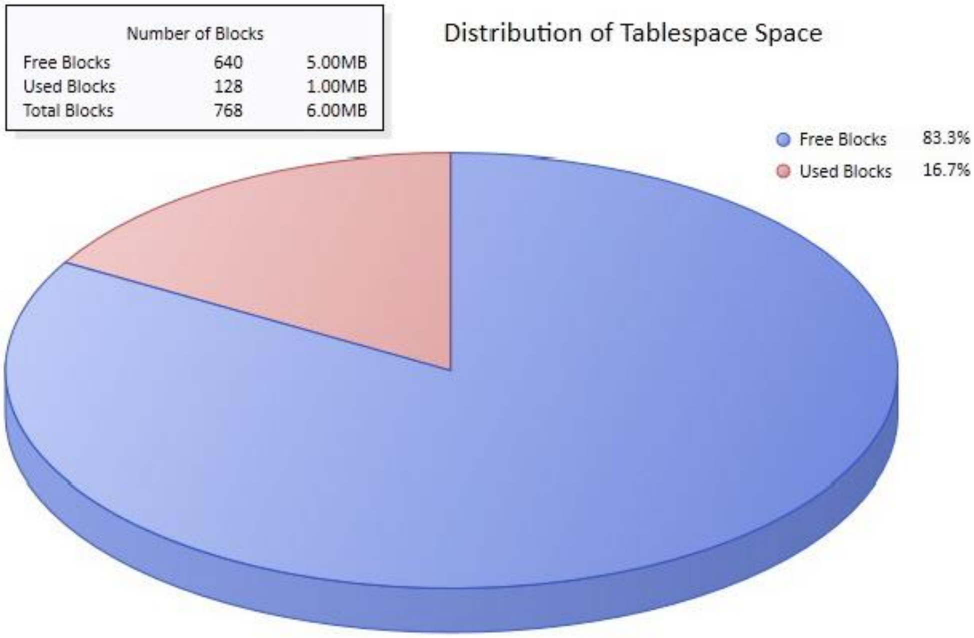
5 Select the **Extent Details** tab. It provides more detail on an extent-by-extent basis.



Type	Name	File ID	Extent ID	Block ID	Blocks	Bytes
INDEX	SERGIU.UK_DATE	71	0	769	8	65536
TABLE	SERGIU.TESTTABL...	71	0	745	8	65536
TABLE	SERGIU.TESTENCR...	71	0	681	8	65536
TABLE	SERGIU.STUDENTS...	71	0	713	8	65536
TABLE	SERGIU.STUDENTS...	71	0	729	8	65536
INDEX	SERGIU.PK_DATE	71	0	761	8	65536
INDEX	SERGIU.PKID_EAFK1	71	0	737	8	65536
TABLE	SERGIU.JOBURI_EA...	71	0	721	8	65536

6 On the Tablespace Editor, click the **Space** tab.

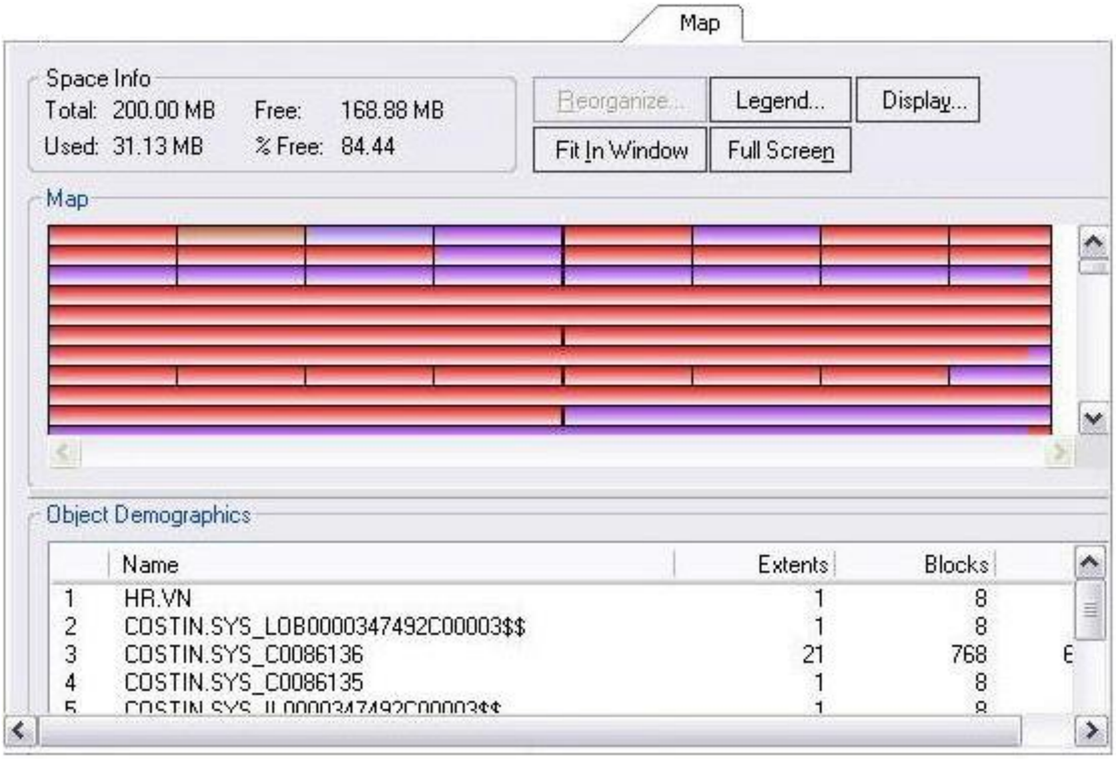
The **Space** tab displays a graphical view of the free and used blocks for the target tablespace.



Number of Blocks		
Free Blocks	640	5.00MB
Used Blocks	128	1.00MB
Total Blocks	768	6.00MB

7 Finally, on the Tablespace Editor, click the **Map** tab.

The **Map** tab displays a color-coded map of the objects contained on the tablespace. The map segments are proportional to the actual size of the objects on the tablespace.



Space Info
Total: 200.00 MB Free: 168.88 MB
Used: 31.13 MB % Free: 84.44

	Name	Extents	Blocks
1	HR.VN	1	8
2	COSTIN.SYS_LOB0000347492C00003$$	1	8
3	COSTIN.SYS_C0086136	21	768
4	COSTIN.SYS_C0086135	1	8
5	COSTIN.SYS_IL0000347492C00003$$	1	8

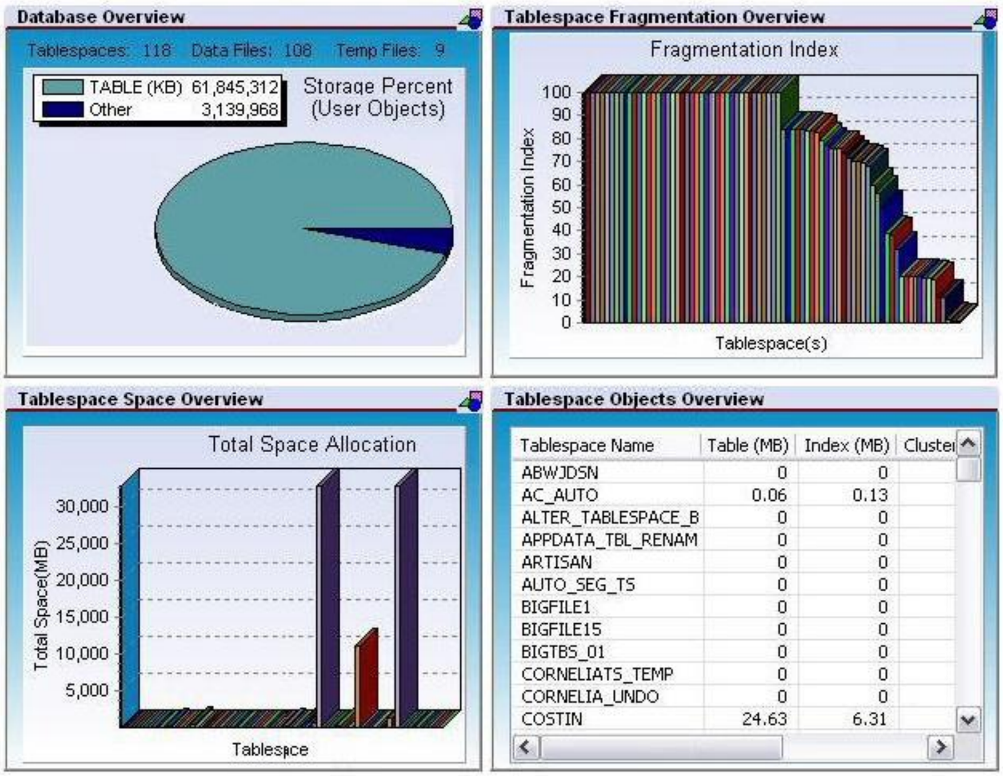
Note the additional **Reorganize** and display options.

8 Close the Tablespaces Editor pane.

Proceed to Advanced Space Management (Oracle and SQL Server only).

Advanced Space Management (Oracle and SQL Server only)

For advanced space analysis and management, DBArtisan's optional Space Analyst component contains sophisticated diagnostics to help you pinpoint all space-related problems in your database, as well as an intelligent reorganization wizard that can reorganize all or selected parts of your database.



Tablespace Name	Table (MB)	Index (MB)	Cluster
ABWJDSN	0	0	
AC_AUTO	0.06	0.13	
ALTER_TABLESPACE_B	0	0	
APPDATA_TBL_RENAM	0	0	
ARTISAN	0	0	
AUTO_SEG_TS	0	0	
BIGFILE1	0	0	
BIGFILE15	0	0	
BIGTBS_01	0	0	
CORNELIATS_TEMP	0	0	
CORNELIA_UNDO	0	0	
COSTIN	24.63	6.31	

Note: Space Analyst is a separately-licensed component. If you are not working with an evaluation copy of DBArtisan or do not have a specific license, proceed to Session 5: SQL Management. Contact your license administrator or your Embarcadero representative about obtaining a license.

To see the sort of information provided by the Space Analyst

1 On the **Analyst** menu, select **Space Analyst.**

Note: If prompted to install components on the server, allow DBArtisan to install the components.

The Space Analyst launches in the DBArtisan workspace.

Embarcadero's Space Analyst provides sophisticated diagnostic capabilities to troubleshoot bottlenecks and performance inefficiencies that result in poor space management. See the DBArtisan online help for a detailed walkthrough of all available features and functionality.

2 Select the **Data Files**, **Tablespace Map**, **Tables**, and **Indexes** tabs available at the bottom of the Space Analyst pane.

3 Close the Space Analyst pane.

Proceed to Session 5: SQL Management.

Session 5: SQL Management

DBArtisan provides powerful visual tools for creating and analyzing complex SQL statements and server-side code objects. The following examples walk you through DBArtisan's Visual Query Builder, feature-rich ISQL facility and some of the advanced analysis and debugging capabilities provided by the Embarcadero SQL Debugger and SQL Profiler:

- Working with Query Builder
- ISQL Window
- SQL Debugging, Analysis and Tuning.

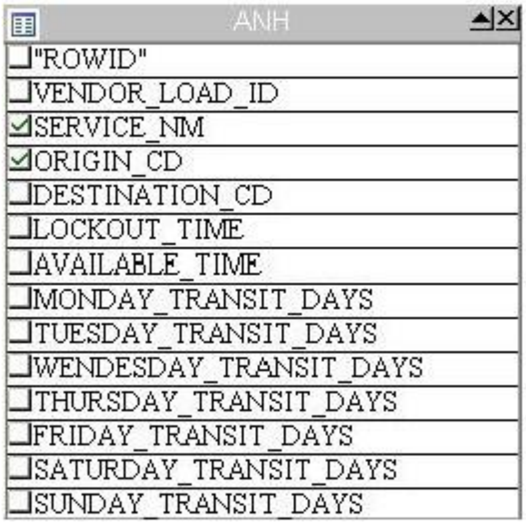
Working with Query Builder

1 From the **Tools** menu, select **Query Builder**.

DBArtisan opens the Query Builder.

2 In the **Tables/Views** tab, right-click a table or view and select **Add**.

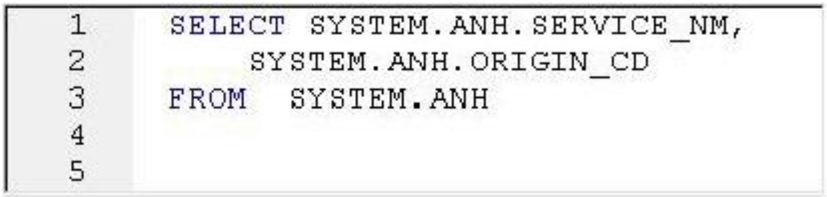
3 In the window that opens, select the columns to return in the result. For example:



ANH

- ☐ "ROWID"
- ☐ VENDOR_LOAD_ID
- ☑ SERVICE_NM
- ☑ ORIGIN_CD
- ☐ DESTINATION_CD
- ☐ LOCKOUT_TIME
- ☐ AVAILABLE_TIME
- ☐ MONDAY_TRANSIT_DAYS
- ☐ TUESDAY_TRANSIT_DAYS
- ☐ WENDESDAY_TRANSIT_DAYS
- ☐ THURSDAY_TRANSIT_DAYS
- ☐ FRIDAY_TRANSIT_DAYS
- ☐ SATURDAY_TRANSIT_DAYS
- ☐ SUNDAY_TRANSIT_DAYS

Query Builder generates the query text in the lower SQL window.



```
SELECT  SYSTEM.ANH.SERVICE_NM,
        SYSTEM.ANH.ORIGIN_CD
FROM    SYSTEM.ANH
```

You can build advanced queries using the options supplied on the **DML** tab. You choose the type of query (SELECT, INSERT, and so on) using the dropdown on the Query Builder toolbar.

4 After the query is built, click the **Execute** button (green arrow) on the Query Builder toolbar.

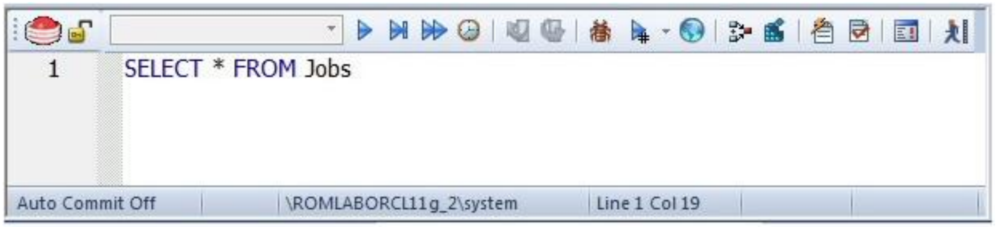
Query Builder displays results in the lower SQL window.

5 Close the Query Builder pane.

Proceed to ISQL Window.

ISQL Window

DBArtisan offers a fully-featured SQL editor. In addition to basic text processing features, the editor environment offers testing/debugging and execution facilities. In combination with supporting features, the ISQL editor eases and speeds up the process of developing working scripts.



```
SELECT * FROM Jobs
```

== Automated error detection and coding assistance ==

DBArtisan provides a range of features that detect or help you avoid errors and save keystrokes in developing your scripts.

To enable these features:

1 On the **File** menu select **Options**. The **Options** dialog opens.

2 In the left-hand pane, expand the **ISQL** node and then select **Code Assist**.

3 On the **Code Assist** panel:

- Ensure that **Enable Code Complete** is selected.
- Ensure that **Enable Real-time syntax checking** is selected.
- Ensure that **Severity levels for semantic validation problems** has **Warning** selected.

4 Click **OK**.

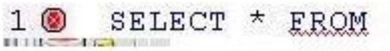
To see these features in action:

1 On the **File** menu, select **New > ISQL**.

DBArtisan opens the ISQL Editor window. You can add SQL code via your method of choice (free-form typing, retrieve from a file, paste copied code, etc.).

2 Experiment with the automated error detection and coding assistance as follows:

- Type SELECT * FROM and stop typing. Note the error condition.



```
SELECT * FROM
```

By default, DBArtisan runs a syntax check any time there is an interval of 1.5 seconds between keystrokes. You can also disable automatic syntax checking and only run a check when you manually initiate it. Syntax error annotation persists until you correct the problem.

- This time type a fragment that includes the name of a nonexistent object, SELECT * FROM NON.OBJECT, for example. For now, ignore any popups. The warning condition is a result of on-the-fly semantic validation. DBArtisan notifies you when a script contains a reference to an object that DBArtisan cannot resolve.

- Type SELECT * FROM followed by a space and then stop typing. If no popup appears, press CTRL+SPACE. The Code Complete suggestion box lets you select from objects or object name components such as databases or schema. This feature saves keystrokes and minimizes typing errors. See the online help for full descriptions of these features.

3 If you want to disable the automated error detection and coding assistance features, select **File > Options** and on the **Code Assist** page, click **Restore defaults**.

Feature Summary

The ISQL Editor window includes the following features and options:

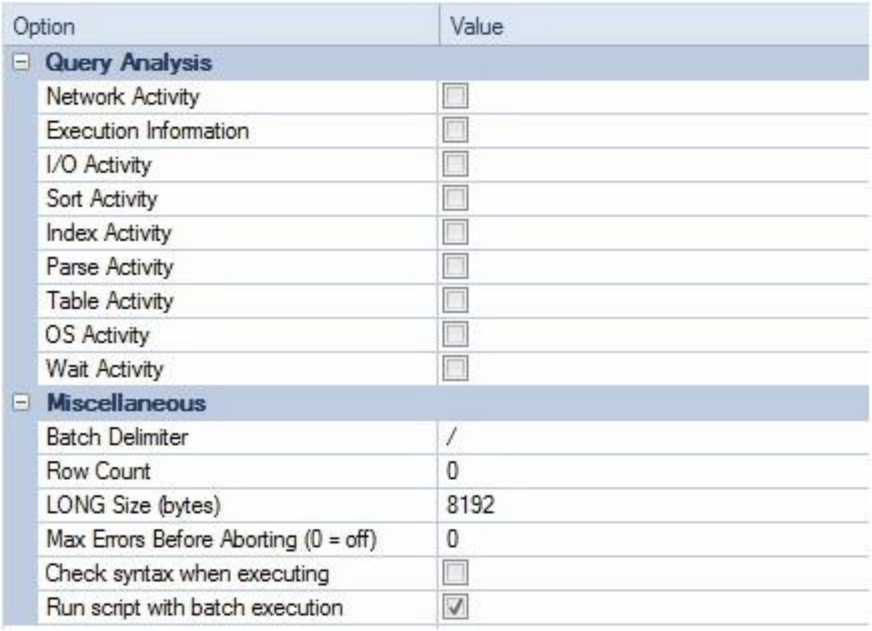
- The ISQL window highlights all platform-specific and general keywords, and provides options for SQL code formatting and analysis.
- Once code is executed you have control over whether your transaction is committed or rolled back from the database.
- For all open ISQL windows, there are also options for connection locking, scheduling, executing your code across multiple datasources, explain plan generation, and SQL Tuning.

Setting environment options

DBArtisan lets you set platform-specific query options to immediately determine if your code is optimized.

1 Press F8.

DBArtisan opens the **Query Option** dialog.



Option	Value
⊟ **Query Analysis**	
Network Activity	☐
Execution Information	☐
I/O Activity	☐
Sort Activity	☐
Index Activity	☐
Parse Activity	☐
Table Activity	☐
OS Activity	☐
Wait Activity	☐
⊟ **Miscellaneous**	
Batch Delimiter	/
Row Count	0
LONG Size (bytes)	8192
Max Errors Before Aborting (0 = off)	0
Check syntax when executing	☐
Run script with batch execution	☑

2 Either close the **Query Options** dialog and then the SQL Editor window or enter a valid query in the ISQL editor and then execute your query by clicking the **Execute** button.

▶

3 When complete, ensure that only the Datasource Explorer window is open.

Proceed to SQL Debugging, Analysis and Tuning.

SQL Debugging, Analysis and Tuning

While this example is specific to Oracle PL/SQL Debugging, the same interface and functionality applies to all of the supported platforms.

To see debugging in action

1 On the Datasource Explorer, expand any Oracle datasource node.

2 On the Oracle datasource, select the **Procedures** node.

3 In the right pane of the Explorer, right-click any stored procedure, and then select **Debug**.

4 If prompted, enter any input parameters and then click **Continue**.

After the SQL Debugger interface is displayed you can step through code, step into dependencies, set and watch variables, and even perform basic code profiling for each line of executed code. See the DBArtisan online help for a detailed listing of all available SQL features.

5 Experiment with debugging features and then close the debugger window.

Proceed to Session 6: Job Management.

Session 6: Job Management

DBArtisan freely integrates with the Microsoft Windows Task Scheduler, which allows you to virtually schedule any task to run on your own computer whenever and how often you'd like. While this example is specific to an Oracle table redefinition, the same concept applies to any job or script that can be scheduled.

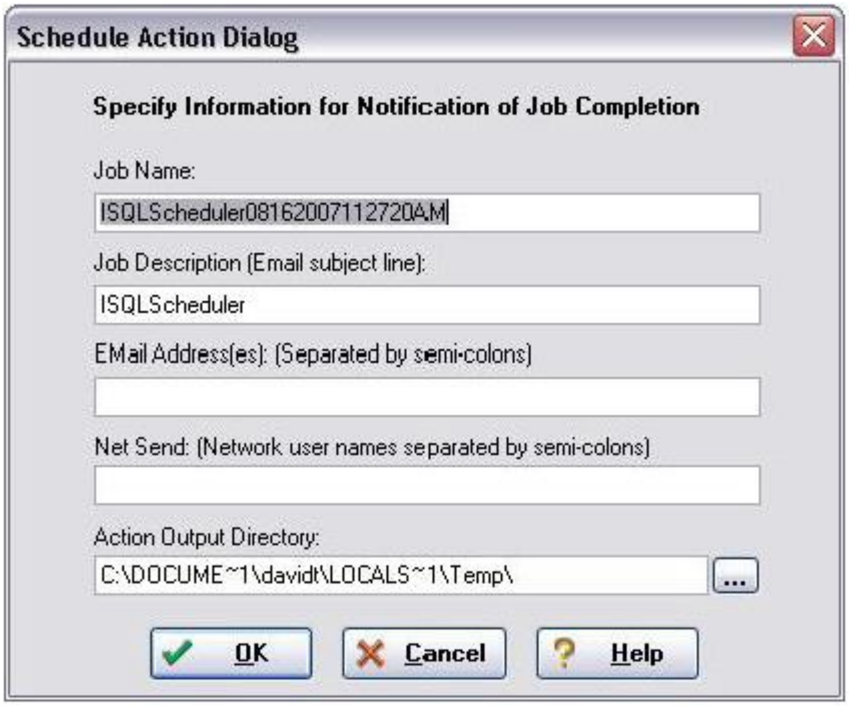
To schedule a job, do the following:

1 On the Explorer, expand any Oracle datasource.

2 On the Oracle datasource, expand the **Tables** node, right-click any table, and select **Extract**. DBArtisan extracts the DDL that defines the selected table and opens an ISQL Editor window on the DDL.

3 From the ISQL window toolbar, click the **Schedule** button.

The **Schedule Action** dialog opens. You can provide a name, set notifications, and specify an output directory for the new job.


Schedule Action Dialog

Specify Information for Notification of Job Completion

Job Name: ISQLScheduler08162007112720AM

Job Description (Email subject line): ISQLScheduler

EMail Address(es): (Separated by semi-colons)

Net Send: (Network user names separated by semi-colons)

Action Output Directory: C:\DOCUME~1\davidt\LOCALS~1\Temp\

4 After you have completed the dialog, click **OK**. The **Job Scheduler** dialog opens.

5 Proceed as prompted to schedule your job. When finished, close the ISQL window.

You can use the **Oracle Job Queue** explorer node to monitor your job.

Proceed to [Session 7: Data Management](#).

Session 7: Data Management

DBArtisan provides comprehensive facilities to help you manage the data in all of your databases. A visual data editor helps you add, change, and delete data from your tables with all referential integrity enforced. You can create INSERT statements for tables using current data and also extract data as XML documents for certain databases. Rounding out its rich Schema Management capabilities, DBArtisan also allows you to migrate schema objects and associated table data from one database server to another, across the same or different platforms.

The following topics walk you through some of the data management features:

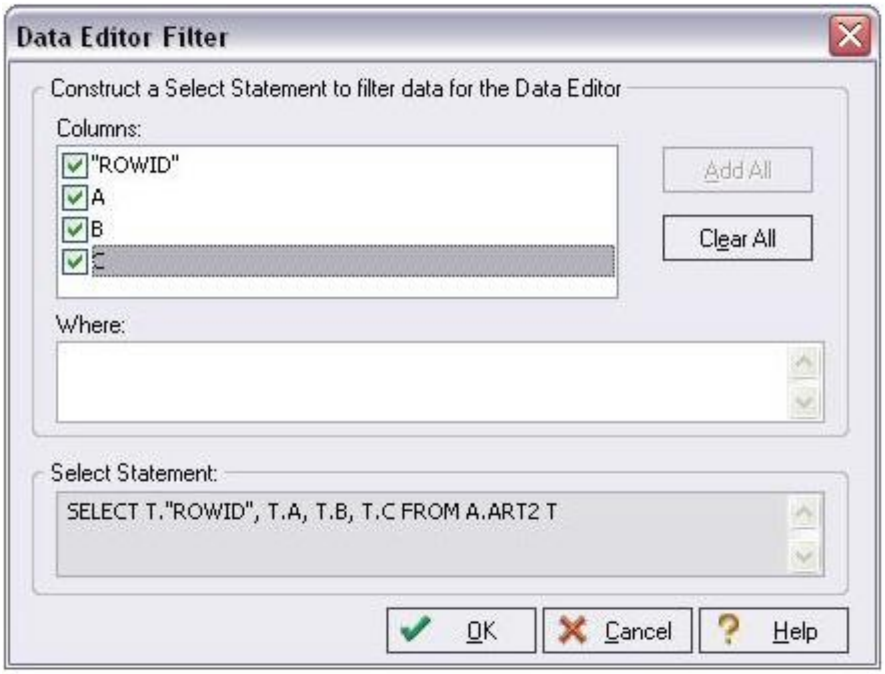
- Visual Data Editing
- Working with Table Data - Create Insert Statements
- Working with Table Data - Extract Data as XML
- Advanced Data Management - Schema and Data Migration

Visual Data Editing

To start the Visual Data Editor, do the following:

1 In the Datasource Explorer, right-click any table and select **Edit Data**. DBArtisan opens the **Data Editor Filter** dialog.

2 In **Columns**, select the columns that are to be included in the editable data.



☑ "ROWID"
☑ A
☑ B
☑ C

SELECT T."ROWID", T.A, T.B, T.C FROM A.ART2 T

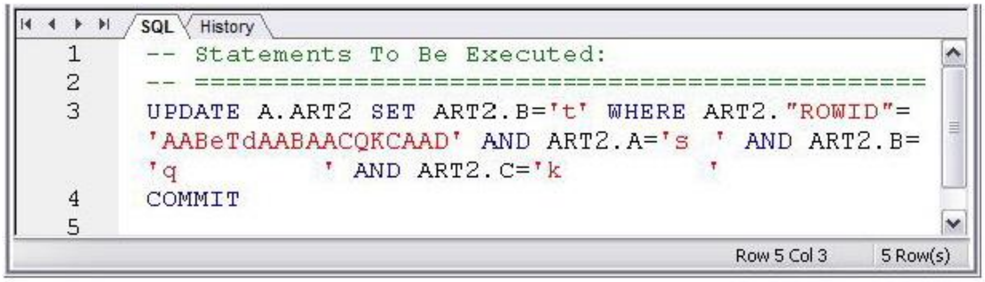
You can also filter the editable rows by including your own SELECT statement.

3 Click **OK**.

In Live mode all changes are applied to the database when you move off of an updated or inserted row. Deleted rows are immediately removed from the database. Batch mode allows you to make changes and then save all when all are complete. Mode is controlled by a dropdown in the Data Editor toolbar.

4 Experiment with editing your data, and when complete, on the Data Editor toolbar, click the **Execute** (blue arrow) button.

DBArtisan commits your changes. Regardless of mode, all of the generated DML statements are viewable in the lower SQL window.



```
-- Statements To Be Executed:
-- =============================================
UPDATE A.ART2 SET ART2.B='t' WHERE ART2."ROWID"=
'AABeTdAABAACQKCAAD' AND ART2.A='s  ' AND ART2.B=
'q          ' AND ART2.C='k          '
COMMIT
```

5 Close the Data Editor pane.

Proceed to Working with Table Data - Create Insert Statements.

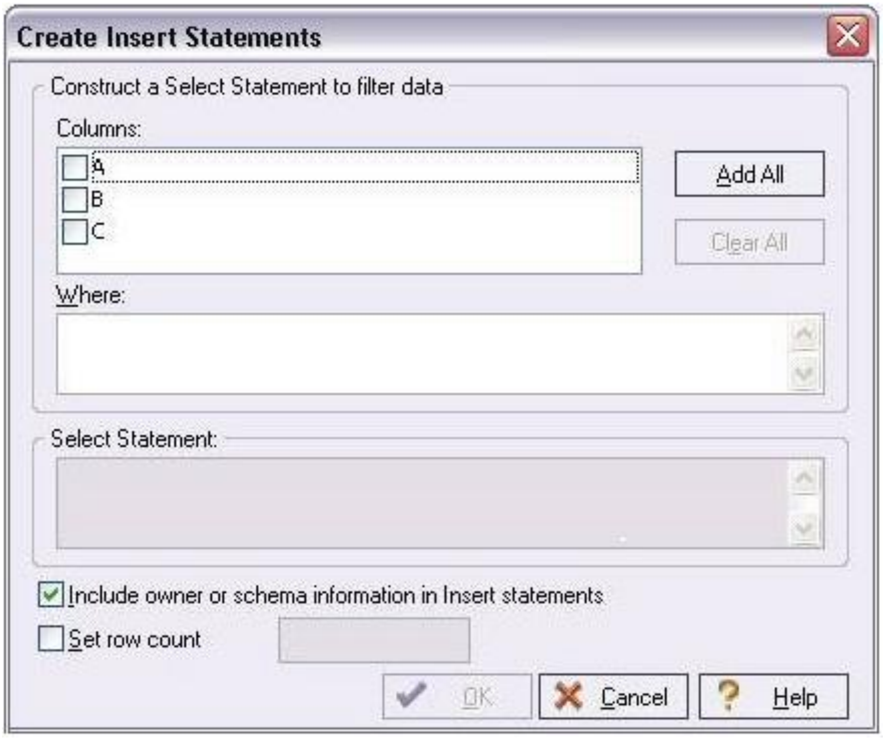
Working with Table Data - Create Insert Statements

DBArtisan lets you generate a set of INSERT statements based on rows of an existing table.

1 On the Datasource Explorer, select any Oracle datasource.

2 On the Oracle datasource, expand the **Tables** node.

3 In the right pane of the Explorer window, right-click any table, and then select **Create Insert Statements**. DBArtisan opens the **Create Insert Statements** dialog. For example:



☐ A
☐ B
☐ C

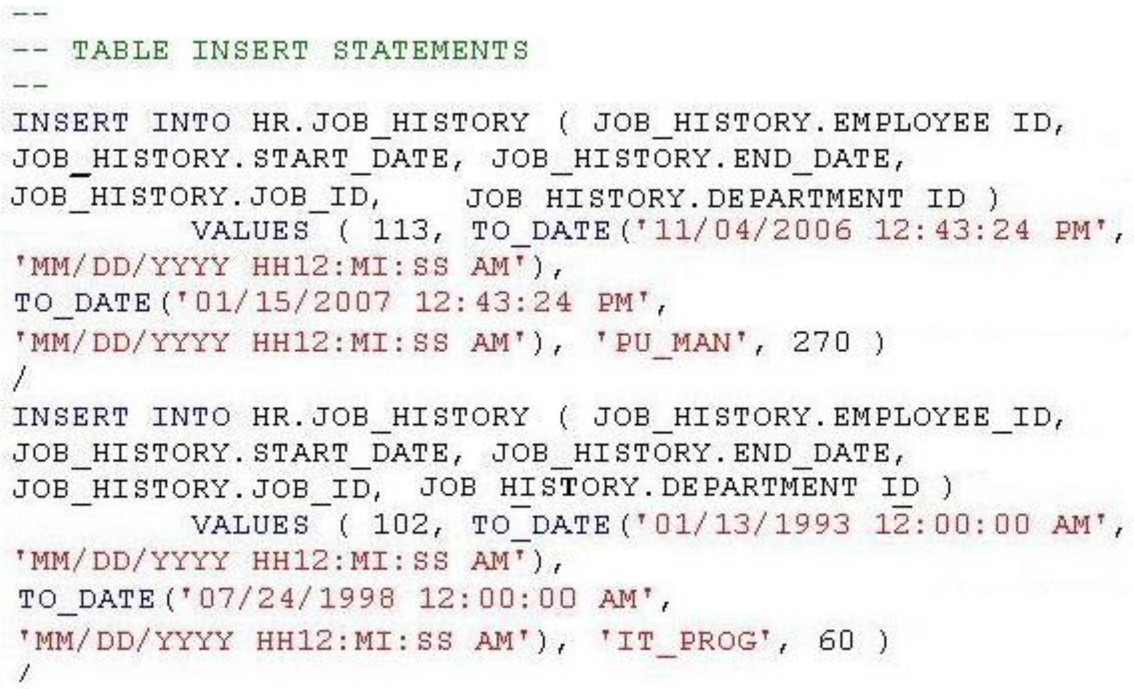
☑ Include owner or schema information in Insert statements
☐ Set row count

4 In **Columns**, select the columns you want to include in the INSERT statement.

5 You can also filter what rows are included by adding your own WHERE clause.

6 Click **OK**.

The resulting INSERT statements are created and presented in an active ISQL window. At this point they can be executed immediately, scheduled to execute later, or saved. Note that all extracted INSERT statements can be run against the same or different databases containing a similar schema.



```
--
-- TABLE INSERT STATEMENTS
--
INSERT INTO HR.JOB_HISTORY ( JOB_HISTORY.EMPLOYEE ID,
JOB_HISTORY.START_DATE, JOB_HISTORY.END_DATE,
JOB_HISTORY.JOB_ID,    JOB HISTORY.DEPARTMENT ID )
         VALUES ( 113, TO_DATE('11/04/2006 12:43:24 PM',
'MM/DD/YYYY HH12:MI:SS AM'),
TO_DATE('01/15/2007 12:43:24 PM',
'MM/DD/YYYY HH12:MI:SS AM'), 'PU_MAN', 270 )
/
INSERT INTO HR.JOB_HISTORY ( JOB_HISTORY.EMPLOYEE_ID,
JOB_HISTORY.START_DATE, JOB_HISTORY.END_DATE,
JOB_HISTORY.JOB_ID,  JOB HISTORY.DEPARTMENT ID )
         VALUES ( 102, TO_DATE('01/13/1993 12:00:00 AM',
'MM/DD/YYYY HH12:MI:SS AM'),
TO_DATE('07/24/1998 12:00:00 AM',
'MM/DD/YYYY HH12:MI:SS AM'), 'IT_PROG', 60 )
/
```

7 Close the editor pane.

Proceed to Working with Table Data - Extract Data as XML.

Working with Table Data - Extract Data as XML

DBArtisan lets you generate an XML representation of table data.

To extract data as XML

1 On the Datasource Explorer, expand any Oracle datasource.

2 On the Oracle datasource, expand the **Tables** node.

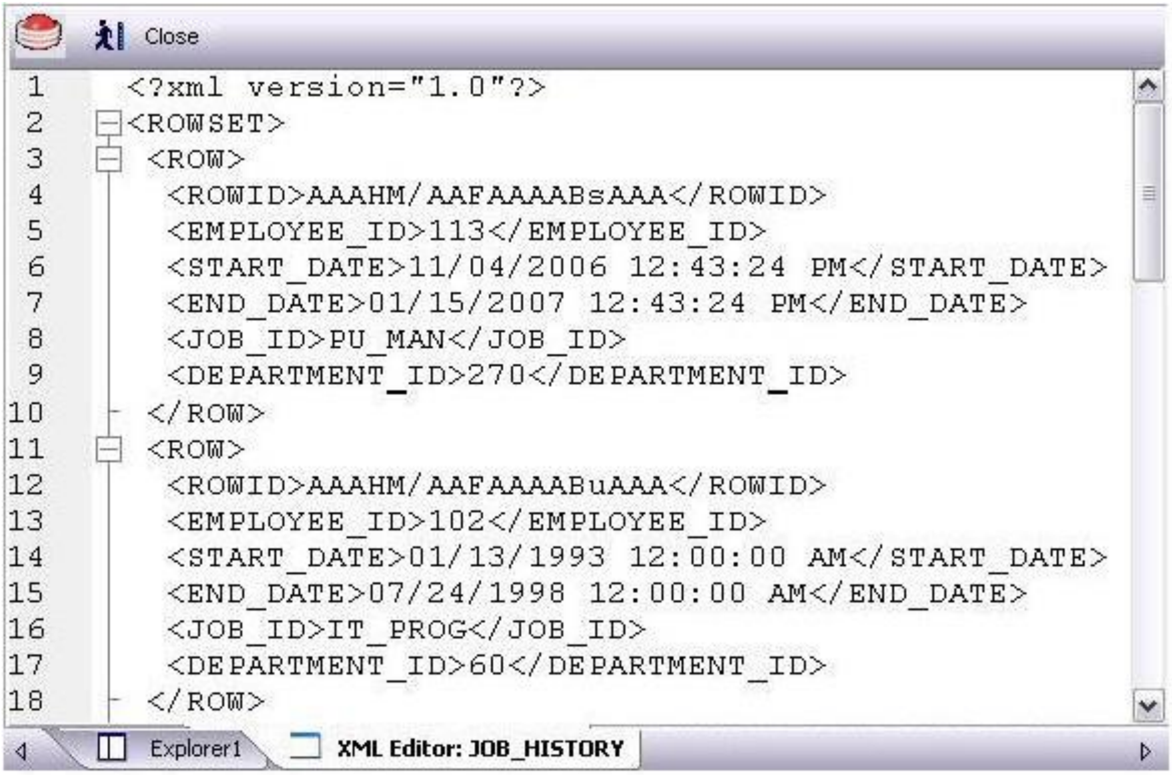
3 In the right pane of the Explorer window, right-click any table listed, and then select **Extract Data as XML**.

4 Select the columns to include in the SELECT statement.

5 You can also filter what rows are included by adding a WHERE clause in the **Select Statement** box.

6 Click **OK**.

The resulting XML document is created and presented in an active XML Editor. At this point the document can be saved in XML format.



```
<?xml version="1.0"?>
<ROWSET>
 <ROW>
  <ROWID>AAAHM/AAFAAAABsAAA</ROWID>
  <EMPLOYEE_ID>113</EMPLOYEE_ID>
  <START_DATE>11/04/2006 12:43:24 PM</START_DATE>
  <END_DATE>01/15/2007 12:43:24 PM</END_DATE>
  <JOB_ID>PU_MAN</JOB_ID>
  <DEPARTMENT_ID>270</DEPARTMENT_ID>
 </ROW>
 <ROW>
  <ROWID>AAAHM/AAFAAAABuAAA</ROWID>
  <EMPLOYEE_ID>102</EMPLOYEE_ID>
  <START_DATE>01/13/1993 12:00:00 AM</START_DATE>
  <END_DATE>07/24/1998 12:00:00 AM</END_DATE>
  <JOB_ID>IT_PROG</JOB_ID>
  <DEPARTMENT_ID>60</DEPARTMENT_ID>
 </ROW>
```

7 Close the editor pane.

Proceed to [Advanced Data Management - Schema and Data Migration]().

Advanced Data Management - Schema and Data Migration

DBArtisan provides advanced data management tools that help you to move schema and corresponding table data across the same or different platforms. You can copy a single database object, all objects owned by a specific user, or an entire database, all guided by a wizard-driven process.

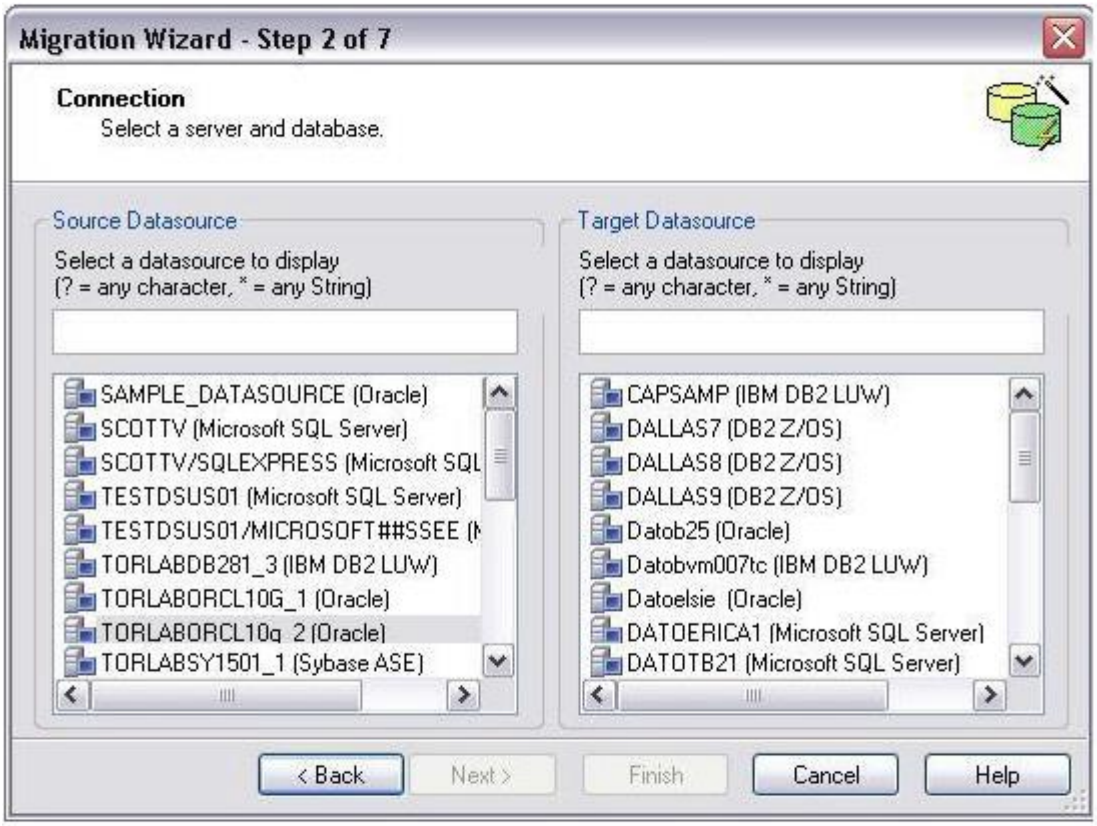
While this example is specific to an Oracle to SQL Server schema and data migration, the same concept applies to any migration involving any combination of the supported platforms.

To open the Schema Migration Wizard:

1 On the **Utilities** menu, select **Schema Migration**. DBArtisan opens the Migration Wizard.

2 Select the **Perform new migration** option.

3 Click **Next**. DBArtisan opens the next panel of the Migration Wizard. For example:

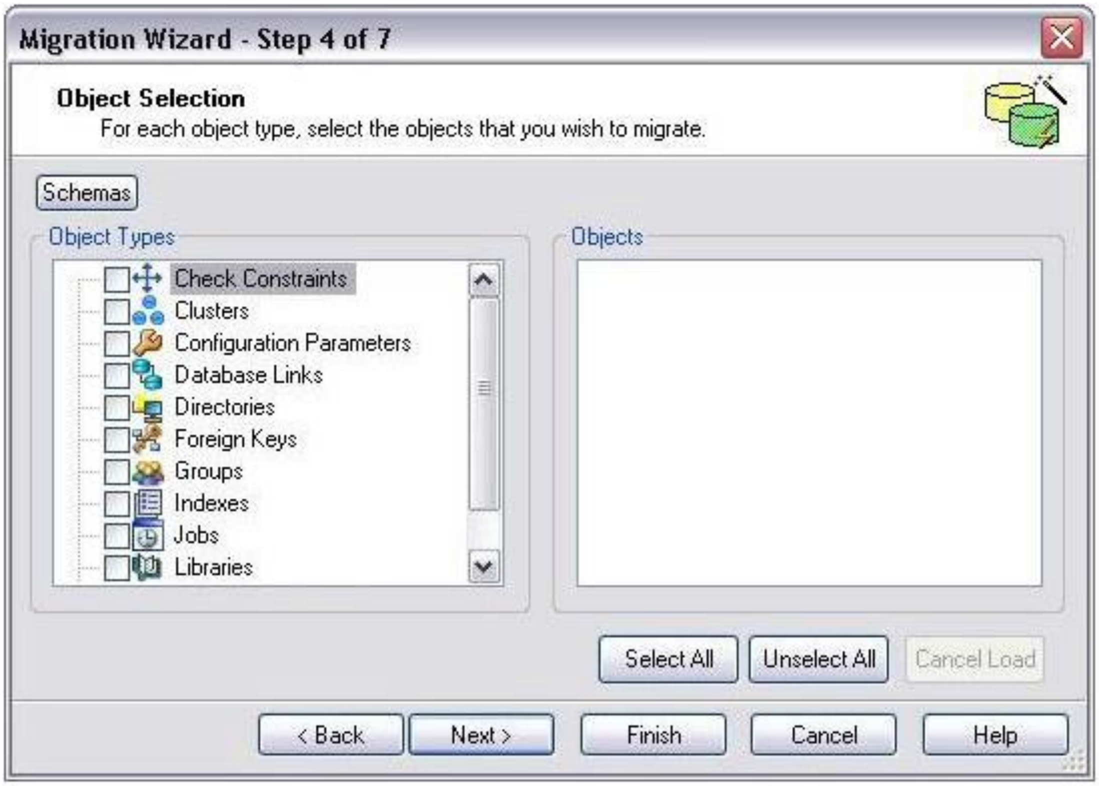

Migration Wizard - Step 2 of 7
Connection
Select a server and database.
Source Datasource
Select a datasource to display
(? = any character, * = any String)
SAMPLE_DATASOURCE (Oracle)
SCOTTV (Microsoft SQL Server)
SCOTTV/SQLEXPRESS (Microsoft SQL
TESTDSUS01 (Microsoft SQL Server)
TESTDSUS01/MICROSOFT##SSEE (M
TORLABDB281_3 (IBM DB2 LUW)
TORLABORCL10G_1 (Oracle)
TORLABORCL10g_2 (Oracle)
TORLABSY1501_1 (Sybase ASE)
Target Datasource
Select a datasource to display
(? = any character, * = any String)
CAPSAMP (IBM DB2 LUW)
DALLAS7 (DB2 Z/OS)
DALLAS8 (DB2 Z/OS)
DALLAS9 (DB2 Z/OS)
Datob25 (Oracle)
Datobvm007tc (IBM DB2 LUW)
Datoelsie (Oracle)
DATOERICA1 (Microsoft SQL Server)
DATOTB21 (Microsoft SQL Server)
< Back
Next >
Finish
Cancel
Help

4 Select an Oracle datasource from the **Source Datasource** list and a Microsoft SQL Server datasource from the **Target Datasource** list.

5 Click **Next** and provide login credentials if prompted. DBArtisan opens the next panel of the Migration Wizard.

Since Microsoft SQL Server supports multiple databases per datasource, you must choose a database.

6 Select an entry from the **Target Database** list and then click **Next**. DBArtisan opens the next panel of the Migration Wizard.

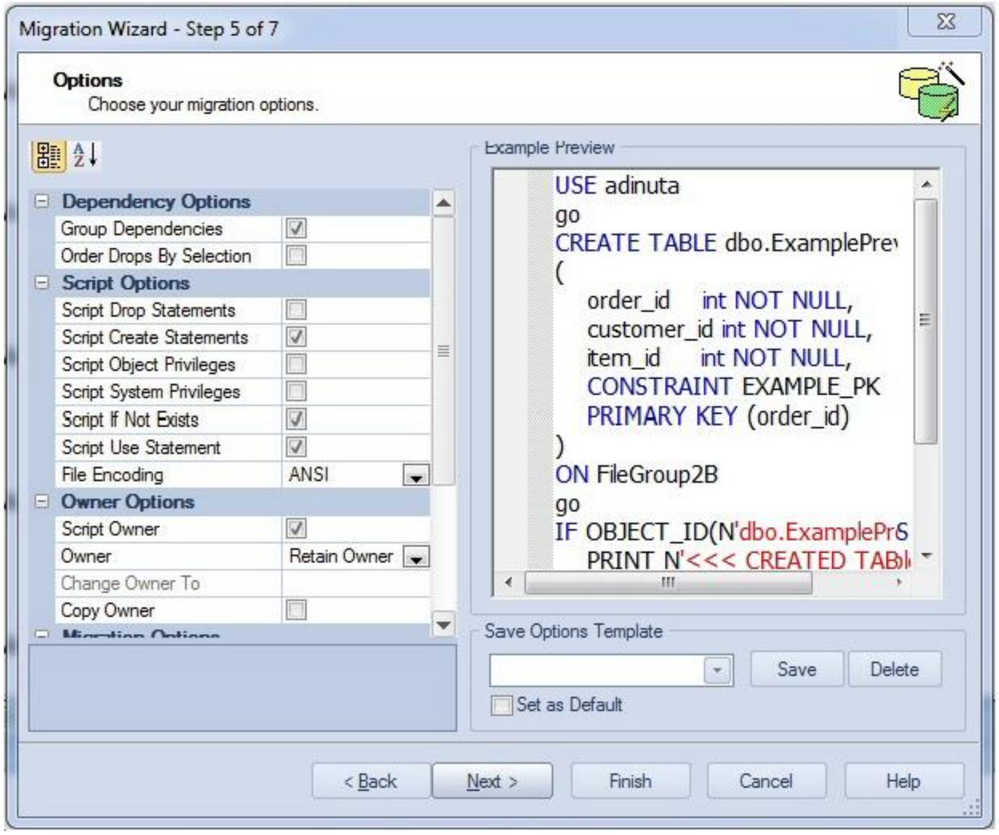


7 In the **Object Types** list, select the object types to be migrated to the target datasource.

As you select object types, the **Objects** list is updated to let you select specific objects to migrate. The **Expand/Collapse** icons let you drill down through dependencies/containment.

8 Use the **Objects** list to select the specific objects to migrate.

9 Click **Next**. DBArtisan opens the next panel of the Migration Wizard.



Options
Choose your migration options.

Dependency Options
- Group Dependencies ☑
- Order Drops By Selection ☐

Script Options
- Script Drop Statements ☐
- Script Create Statements ☑
- Script Object Privileges ☐
- Script System Privileges ☐
- Script If Not Exists ☑
- Script Use Statement ☑
- File Encoding: ANSI

Owner Options
- Script Owner ☑
- Owner: Retain Owner
- Change Owner To
- Copy Owner ☐

Example Preview

```
USE adinuta
go
CREATE TABLE dbo.ExamplePre
(
    order_id    int NOT NULL,
    customer_id int NOT NULL,
    item_id     int NOT NULL,
    CONSTRAINT EXAMPLE_PK
    PRIMARY KEY (order_id)
)
ON FileGroup2B
go
IF OBJECT_ID(N'dbo.ExamplePr
    PRINT N'<<< CREATED TAB
```

Save Options Template

☐ Set as Default

10 Examine the migration options offered but leave the default settings for now.

11 Click **Next**. DBArtisan opens the next panel of the Migration Wizard.

Clicking **Finish** executes the migration and lets you view the progress of the job.

Proceed to Session 8: Performance Management.

Session 8: Performance Management

DBArtisan offers a number of different options to help you manage the performance of your databases. First, DBArtisan ships with a built-in process monitor that helps you understand who is connected to your database along with each user's current activity and session-related data. For more robust performance details DBArtisan's Performance Analyst add-on is a powerful client-side database monitor that runs fully contained in the DBArtisan console.

The following topics walk you through some of the performance management features:

- Monitoring Sessions
- Advanced Client-Side Performance Monitoring

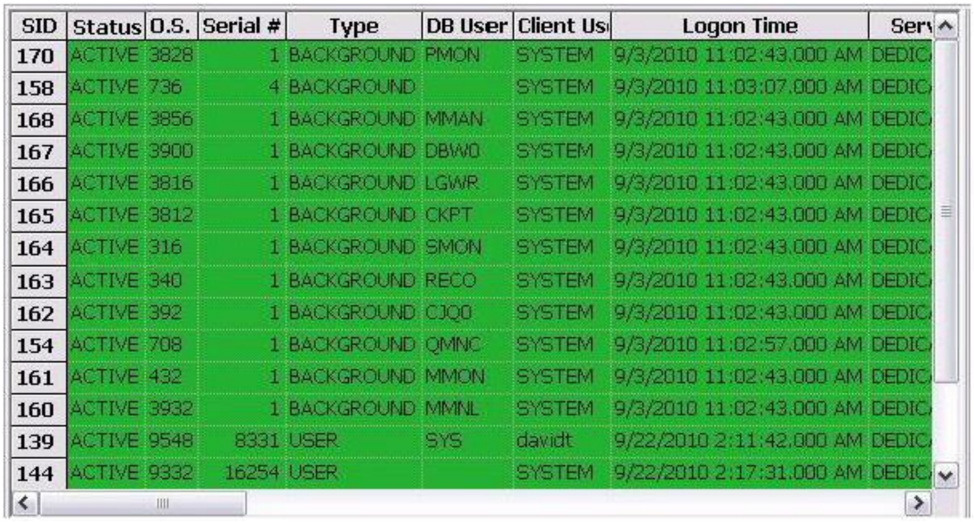
Monitoring Sessions

While this example is specific to Oracle the Process Monitor is available for all of the supported platforms.

To start the DBArtisan Process Monitor:

1 On the Datasource Explorer, select any Oracle datasource.

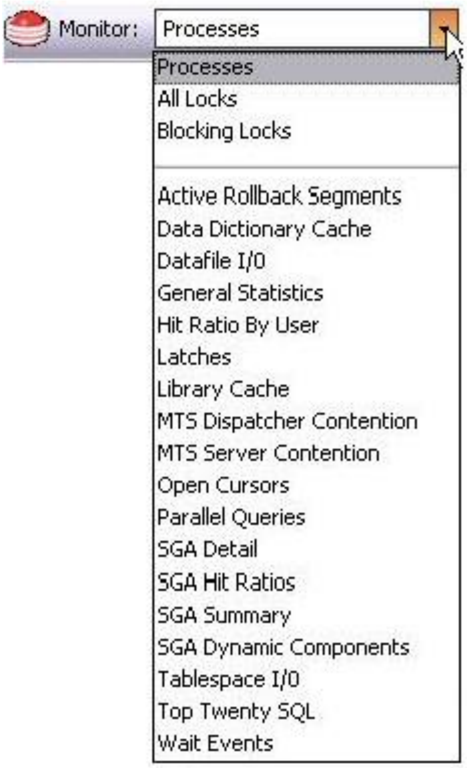
2 From the **Utilities** menu, select **Database Monitor**.



SID	Status	O.S.	Serial #	Type	DB User	Client Us	Logon Time	Serv
170	ACTIVE	3828	1	BACKGROUND	PMON	SYSTEM	9/3/2010 11:02:43.000 AM	DEDIC
158	ACTIVE	736	4	BACKGROUND		SYSTEM	9/3/2010 11:03:07.000 AM	DEDIC
168	ACTIVE	3856	1	BACKGROUND	MMAN	SYSTEM	9/3/2010 11:02:43.000 AM	DEDIC
167	ACTIVE	3900	1	BACKGROUND	DBW0	SYSTEM	9/3/2010 11:02:43.000 AM	DEDIC
166	ACTIVE	3816	1	BACKGROUND	LGWR	SYSTEM	9/3/2010 11:02:43.000 AM	DEDIC
165	ACTIVE	3812	1	BACKGROUND	CKPT	SYSTEM	9/3/2010 11:02:43.000 AM	DEDIC
164	ACTIVE	316	1	BACKGROUND	SMON	SYSTEM	9/3/2010 11:02:43.000 AM	DEDIC
163	ACTIVE	340	1	BACKGROUND	RECO	SYSTEM	9/3/2010 11:02:43.000 AM	DEDIC
162	ACTIVE	392	1	BACKGROUND	CJQ0	SYSTEM	9/3/2010 11:02:43.000 AM	DEDIC
154	ACTIVE	708	1	BACKGROUND	QMNC	SYSTEM	9/3/2010 11:02:57.000 AM	DEDIC
161	ACTIVE	432	1	BACKGROUND	MMON	SYSTEM	9/3/2010 11:02:43.000 AM	DEDIC
160	ACTIVE	3932	1	BACKGROUND	MMNL	SYSTEM	9/3/2010 11:02:43.000 AM	DEDIC
139	ACTIVE	9548	8331	USER	SYS	davidt	9/22/2010 2:11:42.000 AM	DEDIC
144	ACTIVE	9332	16254	USER		SYSTEM	9/22/2010 2:17:31.000 AM	DEDIC

The Database Monitor includes the following options and features:

- Highlight any session and any currently running SQL is displayed in the lower pane.
- You can drill-down into a specific session to display session-level statistical details, historical and current wait events along with a working copy of the currently running SQL that can be copied to an ISQL for Explain Plan generation.
- By using the **Monitor** dropdown options you can display more advanced database-level monitoring data such as locks, blocking locks, hit ratio by user, Top 20 SQL etc.



3 Close the Database Monitor pane.

Proceed to Advanced Client-Side Performance Monitoring.

Advanced Client-Side Performance Monitoring

For advanced performance monitoring and management, DBArtisan's optional Performance Analyst provides intelligent diagnostic information and strong drilldown details to help you get to the heart of any episode of performance degradation. Performance Analyst integrates completely with DBArtisan, so you can fix any performance problems with a few clicks of the mouse.

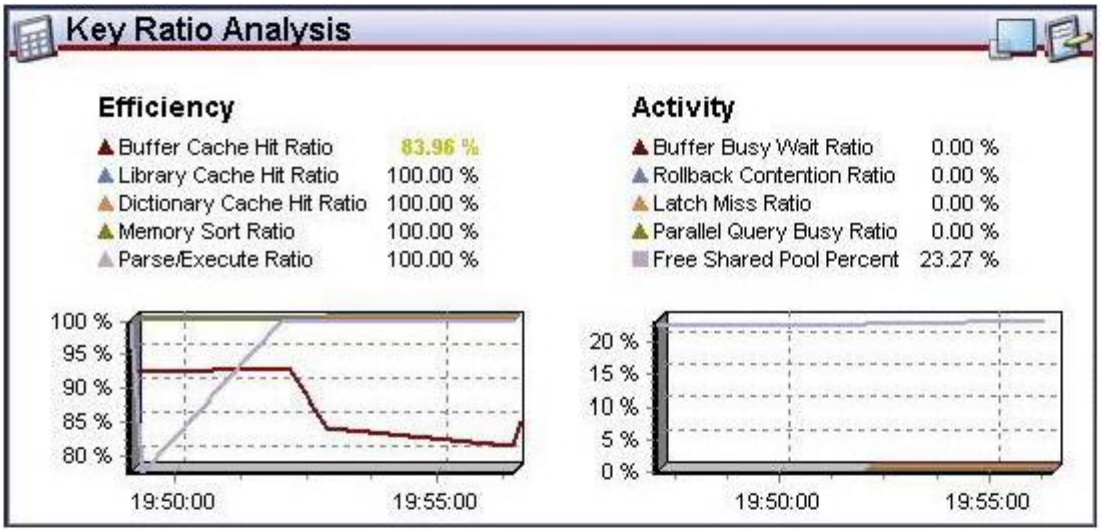
Note: Performance Analyst is available for Oracle, SQL Server, Sybase ASE and DB2 for Unix, Windows, and Linux.

Note: Performance Analyst is a separately-licensed component. If you are not working with an evaluation copy of DBArtisan or do not have a specific license, proceed to Session 10: General Utilities and Tools. Contact your license administrator or your Embarcadero representative about obtaining a license.

1 On the Datasource Explorer, select any Oracle datasource.

2 On the **Analyst** menu, select **Performance Analyst**. The Performance Analyst opens in the DBArtisan workspace for the target datasource.

In addition to the **Home** page displayed initially, Performance Analyst presents several pages of statistics. Page categories include **Memory**, **I/O**, **Space**, **Objects**, **Users**, and **OS**. Each page contains several panes of related statistics, the **Home** page's **Key Ratio Analysis** pane, for example:



Key Ratio Analysis

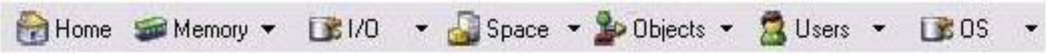
Efficiency

Buffer Cache Hit Ratio	83.96 %
Library Cache Hit Ratio	100.00 %
Dictionary Cache Hit Ratio	100.00 %
Memory Sort Ratio	100.00 %
Parse/Execute Ratio	100.00 %

Activity

Buffer Busy Wait Ratio	0.00 %
Rollback Contention Ratio	0.00 %
Latch Miss Ratio	0.00 %
Parallel Query Busy Ratio	0.00 %
Free Shared Pool Percent	23.27 %

Note the toolbar at the top of the Performance Analyst. It lets you navigate among the Performance Analyst statistics pages.



3 Experiment with displaying the various pages available from the toolbar dropdowns.

See the DBArtisan online help Performance Analyst topics for a detailed walkthrough of all available features and functionality.

Proceed to Session 9: Capacity Management.

Session 9: Capacity Management

Planning for the future of your critical databases used to be a difficult task. However, DBArtisan's optional Capacity Analyst tool makes it easy to understand where your databases are today and where they are headed in the future. Capacity Analyst lets you track key database metadata and performance metrics over time so you can perform trend analysis on key areas like growth, object fragmentation, I/O and session load. Like all of the Analyst Series products, Capacity Analyst runs fully contained within DBArtisan so you have access to smart, built-in forecasting mechanisms that allow you to predict when your databases run out of space and the ability to proactively manage your storage assets, all from the same console.

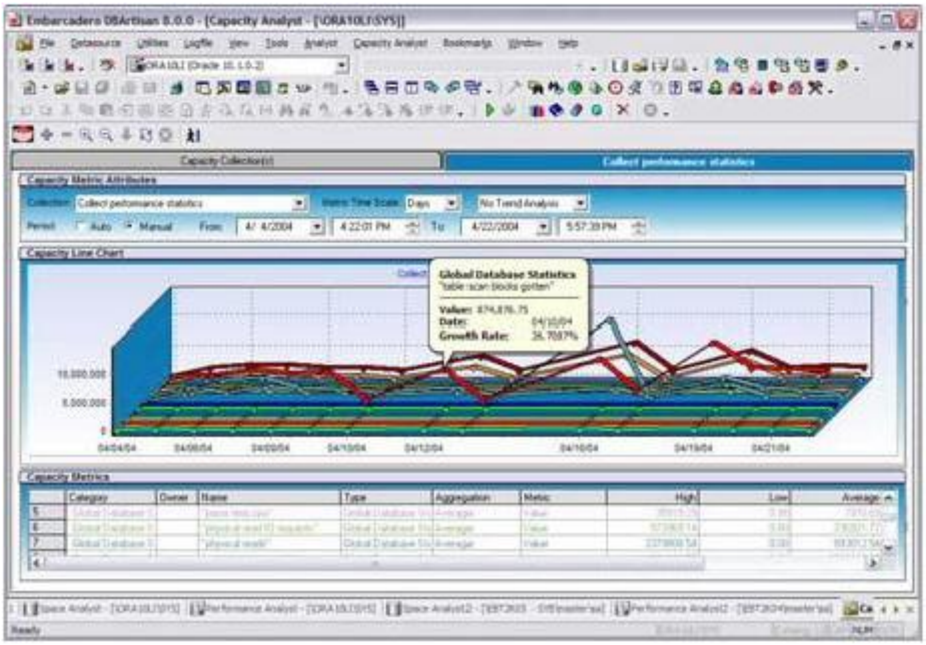
Performance Analyst is available for DB2 for Unix, Windows, and Linux, Sybase ASE, Oracle, and SQL Server.

1 On the Datasource Explorer, select any Oracle datasource.

2 From the **Analyst** menu select **Capacity Analyst**.

Note: Capacity Analyst is a separately-licensed component. If you are not working with an evaluation copy of DBArtisan or do not have a specific license, proceed to Session 10: General Utilities and Tools. Contact your license administrator or your Embarcadero representative about obtaining a license.

The Capacity Analyst opens in the DBArtisan workspace for the target Oracle datasource.

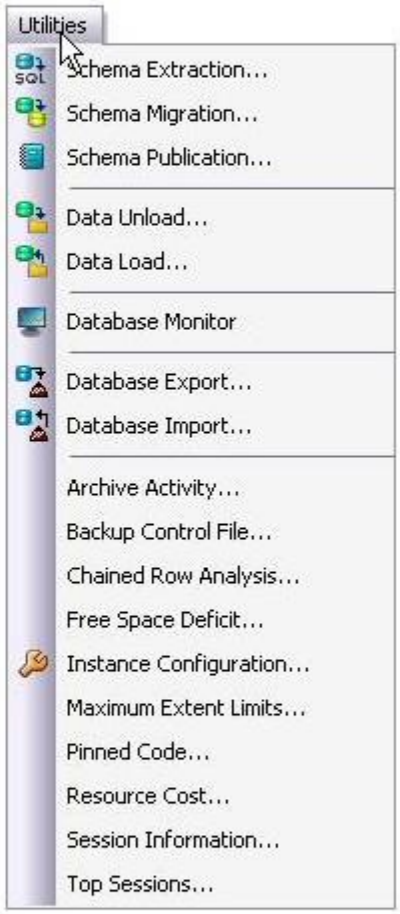

Embarcadero DBArtisan 8.0.0 - [Capacity Analyst - [ORA10LT/SYS]]
Capacity Collector(s)
Collect performance statistics
Capacity Metric Attributes
Capacity Line Chart
Global Database Statistics
Capacity Metrics

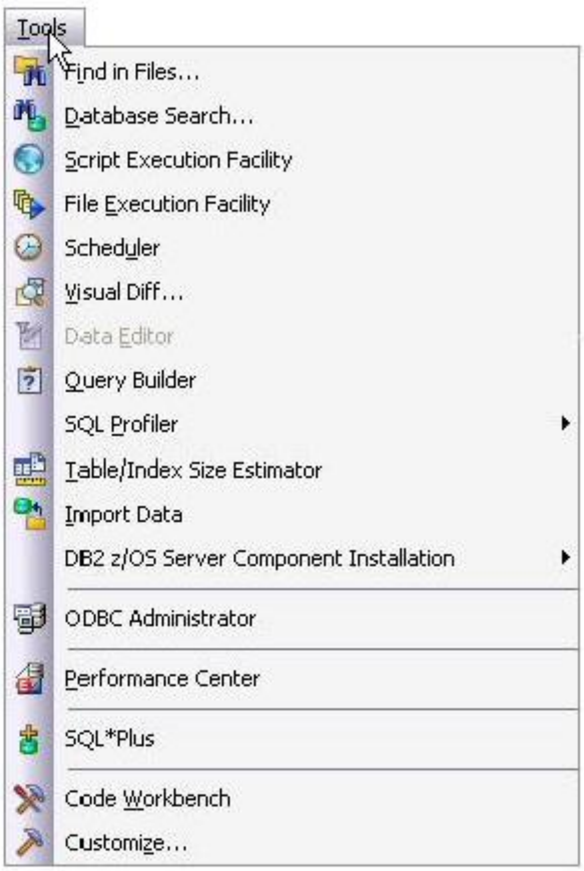
See the relevant DBArtisan online help topics for a detailed walkthrough of all available features and functionality.

Proceed to Session 10: General Utilities and Tools.

Session 10: General Utilities and Tools

No evaluation of DBArtisan would be complete without a mention of the general utilities and tools that are available across all of the supported platforms. The **Utilities** menu offers primarily DBMS-specific operations while those offered on the **Tools** menu are more generic.





The **Utilities** menu contains the more advanced DBArtisan features. The available menu items are context-sensitive and version-specific for the selected datasource DBMS platform. This example above shows the **Utilities** menu features that are available for Oracle.

The **Tools** menu contains those features that are common across all DBMS platforms. The example above shows the **Tools** menu features that are available for all supported DBMS platforms. Note that if any other Embarcadero products are installed on your client they will be available on the **Tools** menu.

All DBArtisan utilities and tools provide a common interface that walks you through all input and execution requirements. All results are consistently presented so you can easily move between features without effort or confusion.

Specifying Application Preferences and Feature Options

The following topics show you how to customize the application configuration for your specific needs:

- Configuring Feature Options
- Locking Down Features Using the Registry

Configuring Feature Options

All application settings are available in the Options Editor, which is organized in a tabular format based on feature sets.

To specify options for the DBArtisan application or a particular feature

1 On the **File** menu, select **Options**.

The Options Editor opens.

2 Click the tab corresponding to the feature you want to customize:

Connection Options	Specifies the timeout parameters, packet size for a connection, and ANSI to OEM settings.
Data Editor Options	Specifies settings for Data Editor.
Data Transfer Options	Sets default directories when performing data unload and load and data export and import operations.
Datasource Options	Specifies general datasource management options and Datasource Explorer/Navigator preferences.
DB Monitor Options (DBArtisan)	Lets you customize the Database Monitor display on a DBMS platform basis.
DBMS Java Options	Specifies load Java files and drop Java files.
DB2 Utilities Options	Sets default directories for executables used by DB2 LUW.
DDL Extract Options	Specifies whether or not DROP statements should be included when extracting the schema of different database object types.
Debug Options	Sets the duration of your debug initialization and debug session, enable or disable DBMS_OUTPUT, and enable the refresh option.
Directories Options	Sets the default directories for placing the output of different operations such as HTML reports or schema extractions.
Explorer Options	Sets defaults for the organization of objects in the Datasource Explorer/Navigator.
General Options	Sets defaults for automatic login, restoring the last session, and other general application options.
Grid Properties (Results window) Options	Dictates the physical appearance of the results window grid.
ISQL Options	Sets defaults for the maximum allowable errors before aborting the execution of a SQL script, executing selected text, and the position of Query and Results tabs.
Java Options	Lets you specify Java Virtual Machine options.
LDAP Configuration Options	Sets preferences for LDAP Server usage.
Logging Options	Sets defaults for SQL Logging.
Migration Options (DBArtisan)	Sets the default options for schema migration operations.
MySQL Utilities Options	Specifies paths to MySQL utilities.
Oracle Utilities Options	Specifies the location of the Oracle Utilities.
Query Builder Options	Specifies global settings for Query Builder.

Results (ISQL) Options	Specifies auto format result sets, sets the display and format of Result Windows, and the mail file type and default fonts.
SMTP Mail Options	Lets you specify defaults for your outgoing mail notifications.
Space Analyst Options (DBArtisan)	Sets Space Analyst options.
Team Server Options	Lets you integrate with the Embarcadero DB Team Server data governance matadata repository.
Version Control Options	Lets you select which version control system you want DBArtisan to use as the underlying version control system
Warnings Options	Activates specific warnings when undesirable actions are attempted against a database.

3 Set feature options on the tab and then click **OK**.

Note: If there is an open document, the **Update Document Statement Properties** dialog opens. The **Update Document Statement Properties** dialog lets you override changes you made to a current document or documents with new setting you made in the Options Editor.

Connection Options

The **Connection** tab of the Options Editor lets you specify common connectivity options and provides access to DBMS-specific options.

Note: For DBArtisan system requirements and details on DBMS product support, see the **Read Me** at http://docs.embarcadero.com/products/dbartisan/.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

The table below describes the options and functionality on the **Connection** tab:

Option	Description	Default
Login Timeout	Specifies the number of seconds that the application should wait for a response to a connection request from server. If server does not respond within the specified period, the application aborts the connection and returns an error message.	30
Query Timeout	Specifies the number of seconds that the application should wait for a response to a query from the server. If the server does not respond within the specified period, the application terminates its query process and returns an error.	0
Host Name	Name of the client computer.	Local name

See the following topics for DBMS platform-specific options:

- Connection Options - DB2
- Connection Options - MySQL
- Connection Options - PostgreSQL
- Connection Options - Sybase
- Connection Option - Sybase IQ
- Connection Option - SQL Server
- Connection Options - Oracle

Connection Options - DB2

The **Connection > DB2** tab of the Options Editor lets you specify DBMS-specific connection options.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For more general connectivity options, see **Connection Options**.

Group	Setting	Description
Default Driver Selection	By default, when you register a datasource, the native driver is automatically selected as the default connection option. **NOTE:** For details on native driver and JDBC driver support, see the **ReadMe** file selecting **View > Release Notes**. The controls in this group let you specify one of the installed or packaged DBMS platform-specific JDBC drivers as the default connection option.	
	Connect using JDBC instead of the DB2 CLI driver	For platforms for which a native driver is supported, this check box enables selection of a JDBC driver as the default connection option. For platforms for which no native driver is available, this check box is disabled.
	JDBC driver to use	If multiple JDBC drivers are available for this platform, this control lets you select the specific JDBC driver to use as the default.
	Assign to all disconnected Datasources	This control lets you assign either the native driver or the currently selected JDBC driver as the connection option for all datasources defined on the relevant DBMS platform that are currently not connected. If the **Connect using JDBC instead of the DB2 CLI driver** setting is selected, all disconnected datasources are assigned the driver selected from the **JDBC driver to use** list as the connection option. Otherwise, the native driver is assigned to all disconnected datasources on this DBMS platform.

Connection Options - MySQL

The **Connection > MyQSL** tab of the Options Editor lets you specify DBMS-specific connection options.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For more general connectivity options, see **Connection Options**.

Group	Setting	Description
Default Driver Selection	By default, when you register a datasource, the native driver is automatically selected as the default connection option. **NOTE:** For details on native driver and JDBC driver support, see the **ReadMe** file selecting **View > Release Notes**. The controls in this group let you specify one of the installed or packaged DBMS platform-specific JDBC drivers as the default connection option.	
	Connect using JDBC instead of the MySQL Connector/ODBC driver	For platforms for which a native driver is supported, this check box enables selection of a JDBC driver as the default connection option. For platforms for which no native driver is available, this check box is disabled.

JDBC driver to use	If multiple JDBC drivers are available for this platform, this control lets you select the specific JDBC driver to use as the default.
Assign to all disconnected Datasources	This control lets you assign either the native driver or the currently selected JDBC driver as the connection option for all datasources defined on the relevant DBMS platform that are currently not connected. If the **Connect using JDBC instead of the MySQL Connector/ODBC driver** setting is selected, all disconnected datasources are assigned the driver selected from the **JDBC driver to use** list as the connection option. Otherwise, the native driver is assigned to all disconnected datasources on this DBMS platform.

Connection Options - PostgreSQL

The **Connection > PostgreSQL** tab of the Options Editor lets you specify DBMS-specific connection options.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For more general connectivity options, see **Connection Options**.

Group	Setting	Description
Default Driver Selection	By default, when you register a datasource, the native driver is automatically selected as the default connection option. **NOTE:** For details on native driver and JDBC driver support, see the **ReadMe** file selecting **View > Release Notes**. The controls in this group let you specify one of the installed or packaged DBMS platform-specific JDBC drivers as the default connection option.	
	Connect using JDBC instead of the PostgreSQL ODBC driver	For platforms for which a native driver is supported, this check box enables selection of a JDBC driver as the default connection option. For platforms for which no native driver is available, this check box is disabled.
	JDBC driver to use	If multiple JDBC drivers are available for this platform, this control lets you select the specific JDBC driver to use as the default.
	Assign to all disconnected Datasources	This control lets you assign either the native driver or the currently selected JDBC driver as the connection option for all datasources defined on the relevant DBMS platform that are currently not connected. If the **Connect using JDBC instead of the PostgreSQL ODBC driver** setting is selected, all disconnected datasources are assigned the driver selected from the **JDBC driver to use** list as the connection option. Otherwise, the native driver is assigned to all disconnected datasources on this DBMS platform.

Connection Options - Sybase

The **Connection > Sybase** tab of the Options Editor lets you select an installed client other than the active/default client version for Embarcadero driver connections. It also lets you override default client/server settings.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For more general connectivity options, see **Connection Options**.

Group	Setting	Description
	Use Quoted Identifiers	If you plan to use delimited identifiers, select this option.
Sybase Open Client Selection	This group of controls apply only to use of the Embarcadero drivers with native Sybase clients.	
	Enable Client Selection	Selecting this check box lets you override the default/active Sybase client specified by the SYBASE, SYBASE_OCS, and/or SYBROOT environment variables. This control enables and disables the **Client Home Directory** and **Client OCS Directory** controls.
	Client Home Directory	Browse to the root directory of the client to be used.
	Client OCS Directory	Select the **OCS-** directory corresponding to the client version that you want to use.
Connection Settings	**SQL INI File**	Browse to locate and select the specific, SQL.INI-formatted file containing server-configuration information. The file does not have to be named **SQL.INI**.
	Packet Size, **Max Connections**, and **Client Character Set**	These settings let you create a set of default parameters used in setting up connections to Sybase ASE datasources. These settings can be overridden when registering a Sybase ASE datasource. For details, see [RapArt_Datasource_Management.backup.htm#Ram61836 Providing Sybase ASE Connection Information When Registering Datasources].

<table>
<tr>
<td>Default Driver Selection</td>
<td>
By default, when you register a datasource, the native driver is automatically selected as the default connection option.
<b>NOTE:</b> For details on native driver and JDBC driver support, see the <b>ReadMe</b> file selecting <b>View > Release Notes</b>.
The controls in this group let you specify one of the installed or packaged DBMS platform-specific JDBC drivers as the default connection option.
<table>
<tr><td><b>Connect using JDBC instead of the Sybase Open Client</b></td><td>For platforms for which a native driver is supported, this check box enables selection of a JDBC driver as the default connection option.
For platforms for which no native driver is available, this check box is disabled.</td></tr>
<tr><td><b>JDBC driver to use</b></td><td>If multiple JDBC drivers are available for this platform, this control lets you select the specific JDBC driver to use as the default.</td></tr>
<tr><td><b>Assign to all disconnected Datasources</b></td><td>This control lets you assign either the native driver or the currently selected JDBC driver as the connection option for all datasources defined on the relevant DBMS platform that are currently not connected. If the <b>Connect using JDBC instead of the Sybase Open Client</b> setting is selected, all disconnected datasources are assigned the driver selected from the <b>JDBC driver to use</b> list as the connection option. Otherwise, the native driver is assigned to all disconnected datasources on this DBMS platform.</td></tr>
</table>
</td>
</tr>
</table>

Connection Options - Sybase IQ

The **Connection > Sybase IQ** tab of the Options Editor lets you specify DBMS-specific connection options.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For more general connectivity options, see **Connection Options**.

<table>
<tr><th>Setting</th><th>Description</th></tr>
<tr>
<td></td>
<td>
<table>
<tr><td><b>Use Quoted Identifiers</b></td><td>If you plan to use delimited identifiers, select this option.</td></tr>
</table>
</td>
</tr>
<tr>
<td>Default Driver Selection</td>
<td>
By default, when you register a datasource, the native driver is automatically selected as the default connection option.
<b>NOTE:</b> For details on native driver and JDBC driver support, see the <b>ReadMe</b> file selecting <b>View > Release Notes</b>.
The controls in this group let you specify one of the installed or packaged DBMS platform-specific JDBC drivers as the default connection option.
<table>
<tr><td><b>Connect using JDBC instead of the Sybase IQ ODBC Driver</b></td><td>For platforms for which a native driver is supported, this check box enables selection of a JDBC driver as the default connection option.
For platforms for which no native driver is available, this check box is disabled.</td></tr>
</table>
</td>
</tr>
</table>

JDBC driver to use	If multiple JDBC drivers are available for this platform, this control lets you select the specific JDBC driver to use as the default.
Assign to all disconnected Datasources	This control lets you assign either the native driver or the currently selected JDBC driver as the connection option for all datasources defined on the relevant DBMS platform that are currently not connected. If the **Connect using JDBC instead of the Sybase IQ ODBC Driver** setting is selected, all disconnected datasources are assigned the driver selected from the **JDBC driver to use** list as the connection option. Otherwise, the native driver is assigned to all disconnected datasources on this DBMS platform.

Connection Options - SQL Server

The **Connection > SQL Server** tab of the Options Editor lets you specify DBMS-specific connection options.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For more general connectivity options, see **Connection Options**.

Setting	Description
SQL Server Native Client Selection	These controls let you specify a default native driver. **NOTE:** For details on native driver and JDBC driver support, see the **ReadMe** file selecting **View > Release Notes**.

Enable Client Selection	Where multiple native drivers are supported, this check box enables selection of a native client to use as the default option.
Clients	Select the native client to be used as default.
Use Quoted Identifiers	If you plan to use delimited identifiers, select this option.

Connection Settings	**Packet Size**	This control lets you specify a default packet size parameter used with SQL Server connections. This setting can be overridden when registering a SQL Server datasource. For details, see [RapArt_Datasource_Management.backup.htm#Ram94263 Providing SQL Server Connection Information When Registering Datasources].
Default Driver Selection	By default, when you register a datasource, the native driver is automatically selected as the default connection option. **NOTE:** For details on native driver and JDBC driver support, see the **ReadMe** file selecting **View > Release Notes**. The controls in this group let you specify one of the installed or packaged DBMS platform-specific JDBC drivers as the default connection option.	
	Connect using JDBC instead of the SQL Server Native Client	For platforms for which a native driver is supported, this check box enables selection of a JDBC driver as the default connection option. For platforms for which no native driver is available, this check box is disabled.
	JDBC driver to use	If multiple JDBC drivers are available for this platform, this control lets you select the specific JDBC driver to use as the default.
	Assign to all disconnected Datasources	This control lets you assign either the native driver or the currently selected JDBC driver as the connection option for all datasources defined on the relevant DBMS platform that are currently not connected. If the **Connect using JDBC instead of the DB2 CLI driver** setting is selected, all disconnected datasources are assigned the driver selected from the **JDBC driver to use** list as the connection option. Otherwise, the native driver is assigned to all disconnected datasources on this DBMS platform.

Connection Options - Oracle

The **Connection > Oracle** tab of the Options Editor lets you select an installed client other than the active/default client version for Embarcadero driver connections. It also lets you override default client/server settings.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For more general connectivity options, see **Connection Options**.

Group	Setting	Description
Hide time portion of DATE datatype if zero		If selected, the time portion (HH:MM:SS) of DATE datatype results are suppressed in the ISQL Editor's Results window, if the time portion is all zeroes.
OCI Client Selection	This group of controls apply only to use of the Embarcadero drivers with native Oracle clients. If no client is specified using these controls, the following sequence is used in determining the client to use for Oracle connections. - The driver specified in the ORACLE_HOME environment variable - The first driver found in the PATH environment variable - The bundled Oracle Instant Client driver (v. 11)	
	Enable Client Selection	Selecting this check box lets you override the default/active client specified by Oracle environment variables. This control enables and disables the **Oracle Home** control.
	Oracle Home	Select the directory of the Oracle client you wish to use.
TNSNAMES.ORA	**Directory Path of Oracle configuration files**	Browse to locate and select the directory containing the **TNSNAMES.ORA** file to use. Ensure that no file name is included with the path. If no file is specified using this control, the following sequence is used in determining the **TNSNAMES.ORA** file to use for Oracle connections. -The directory specified in the TNS_ADMIN environment variable - The directory specified in the TNS_ADMIN key at the top level in HKLM\SOFTWARE\ORACLE - The directory specified in the TNS_ADMIN key in the client subkey (HOME or KEY_ prefixed) corresponding to the value of the ORACLE_HOME (if there is a particular client specified by the user, its TNS_ADMIN subkey will be used if present) - The directory ORACLE_HOME\NETWORK\ADMIN (if there is a particular client specified by the user, ORACLE_HOME is the client's full path) - The directory ORACLE_HOME\net80\ADMIN (if there is a particular client specified by the user, ORACLE_HOME is the client's full path) - All directories in PATH environment variable
Default Driver Selection	By default, when you register a datasource, the native driver is automatically selected as the default connection option. **NOTE:** For details on native driver and JDBC driver support, see the **ReadMe** file selecting **View > Release Notes**. The controls in this group let you specify one of the installed or packaged DBMS platform-specific JDBC drivers as the default connection option.	
	Connect using JDBC instead of the OCI client	For platforms for which a native driver is supported, this check box enables selection of a JDBC driver as the default connection option. For platforms for which no native driver is available, this check box is disabled.
	JDBC driver to use	If multiple JDBC drivers are available for this platform, this control lets you select the specific JDBC driver to use as the default.

Assign to all disconnected Datasources	This control lets you assign either the native driver or the currently selected JDBC driver as the connection option for all datasources defined on the relevant DBMS platform that are currently not connected. If the **Connect using JDBC instead of the OCI client** setting is selected, all disconnected datasources are assigned the driver selected from the **JDBC driver to use** list as the connection option. Otherwise, the native driver is assigned to all disconnected datasources on this DBMS platform.

Data Editor Options

The **Data Editor** tab of the Options Editor lets you provide preferences that dictate behavior of the table data editor.

Note: Option parameters set in the Options Editor override options set within Query Builder and Data Editor.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using the Data Editor, see Data Editor.

The table below describes the options and functionality on the **Data Editor tab**:

Interface Element	Option	Description	Default
Default Execution Mode	**Live Mode** lets you execute changes one row at a time. **Batch Mode** lets you make unlimited changes prior to execution. Batch Mode offers three sub-options: **Ignore errors-continue processing**, **Prompt on Error**, and **Stop after error(s)** (number of errors allowed before stopping execution)		Live Mode
Data Editor File	**Mail File Type**	Sets the default mail output style as Results, Tab Delimited, Comma Separated, or HTML.	Results
	Include column titles when saving	Includes column titles when saving.	Not selected
Grid Font	Customizes font style, and size for the Data Editor and the Results Grid.		Available
Printer Font	Sets font style, and size for printing output.		Available
Auto Format (Best Fit)	Fits formatting to match your desktop.		Selected
Begin and End Transaction Statements	Adds a beginning and ending transaction on each statement.		Selected
Default Date/Time Format	Displays the current date/time format and lets you customize the date/time display.		Results
	Use Calendar Control as default	If selected, the Calendar Control window is used.	Not selected
	2 digit year system setting warning	If selected, a warning is issued when you use a two digit year system setting.	Selected
Confirmation Dialog Options	Enabling **Show Delete Confirmation Dialog**, **Show Update Confirmation Dialog**, or **Show Update LOB Confirmation Dialog** lets you display a confirmation dialog when you use a delete command, update a row, or update a LOB, respectively.		Selected

Data Transfer Options

After opening the Options editor (see Specifying Application Preferences and Feature Options), you can make changes to the **Data Transfer** tab.

You can configure DBArtisan to use default directories when performing data unload and load and data export and import operations. Setting a default directory saves time because it describes a single reference point for loading or unloading, exporting and importing table or view data files.

The table below describes the options and functionality on the **Data Transfer tab**:

Option	Description
Data Unload	Specifies the name and location of the default directory in the Data Unload box.
Data Load	Specifies the name and location of the default directory in the Data Load box.
Oracle Export (for Oracle)	Specifies the name and location of the default directory in the Oracle Export box.
Oracle Import (for Oracle)	Specifies the name and location of the default directory in the Oracle Import box.

Datasource Options

The **Datasource** tab of the Options Editor lets you provide datasource categorization and storage preferences.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Option	Description	Default
Datasource Storage Management	Registry-based and file-based datasource catalog management is available. Prior to changing methods, see [RapArt_Datasource_Management.backup.htm#Ram86862 Specifying a Datasource Catalog Storage Option]/ The following options are offered: **Windows Registry datasource catalog -** The datasource catalog is stored and managed locally, in the Windows registry. This user can add and delete datasources and edit the registration details of existing resources. Any changes are retained across startups. **File based datasource catalog -** The datasource catalog is stored locally and is file-based (*%APPDATA%***\Embarcadero\Data Sources** in a default installation). This user can add and delete datasources and edit the registration details of existing resources. Any changes are retained across startups. **Network shared datasource catalog -** The datasource catalog is file-based, and obtained from the location specified in the **Location of shared datasources file** control. Changes such as new datasources, deleted datasources or edited datasource registrations are lost on shutdown. Changing methods does not delete datasource definitions stored using the current method. When you change methods, existing catalog definitions are not automatically converted to the new method	Windows Registry
Recent Datasource List Contains	Lets you specify number of datasources to display when you select **Recent Datasources** from the **File** menu.	8
Default to Alias Usage When Defining New Datasources	For DBMS that support alias usage, defaults the **Connection Information** tab to alias usage selection.	Unselected
Datasource Category Settings	Lets you select the optional user interface elements that are to be color-coded or labelled for datasources that have been assigned a category: **Color** the status bar using the datasource’s category colorColor the tabs using the datasource’s category colorInclude category short name in the **datasource toolbar combo** For an introduction to datasource categorization, see Categorizing Datasources using Color-coding and Labeling.	N/A

Note: For information on using Embarcadero Team Server repository-based datasource definitions, see DB Team Server Support.

DB Monitor Options (DBArtisan)

The Options Editor's **DB Monitor** tab lets you specify color and refresh rate Database Monitor settings for a DBMS platform.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using Database Monitor, see Database Monitor (DBArtisan).

The table below describes the options and functionality on the **DB Monitor**tab:

Option	Description
DBMS type dropdown	Lets you select the DBMS type that this set of custom settings applies to.
Refresh Rate	The initial rate at which the display is refreshed when you open a Database Monitor window on the specified platform.
Process types, **Item foreground**, and **Item background**	Lets you specify the colors of text and background for specific process types.

DBMS Java Options

The **DBMS_JAVA** tab of the Options Editor lets you choose between use of the DBMS_JAVA package and the loadJava.bat batch file when loading or dropping Java classes.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

The table below describes the options and functionality on the **DBMS_Java** tab:

Interface Element	Option	Description	Default
Load Java files	**Use the DBMS_JAVA package**	Lets you schedule your SQL.	Selected
	Use batch file (oracle_home/bin/loadJava.bat)	Uses batch file (oracle_home/bin/loadJava.bat)	Not selected
	Default Encoding option	Leave blank to use the default.	Default
Drop Java files	**Use the DBMS_JAVA package**	Lets you schedule your SQL.	Selected
	Use batch file (oracle_home/bin/dropJava.bat)	Uses batch file (oracle_home/bin/dropJava.bat)	Not selected

For related information, see the following object actions:

- Drop Java
- Load Java

DB2 Utilities Options

Some functionality against DB2 LUW datasources makes use of native DB2 utilities.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

The table below describes the options and functionality of the **DB2 Utilities** tab:

Option	Description
db2cmd.exe	Use this control to specify the location of the directory containing the **db2cmd.exe** utility.
db2.exe	Use this control to specify the location of the directory containing the **db2.exe** utility.

DDL Extract Options

The **DDL Extract** tab of the Options Editor lets you specify results-handling preferences for use of the **Extract** object action.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using the **Extract** object action, see Extract.

Common options

The table below describes the options and functionality on the **DDL Extract tab**, main view, of the Options Editor:

Option	Description
Extract to multiple windows	Select to extract the schema for each object into separate DDL windows. NOTE: This option only works when you extract DDL for multiple objects.
Extract in dependency order	This is the default. When you select a number of objects as part of an extraction job, this option ensures objects will be extracted in the proper dependency order. If the option is not selected, errors may result when you execute the script. It's also true, however, that loading the dependencies can add significant overhead when you are extracting numerous objects.
Script Use Statement	This option is for Sybase. Optimizes extraction through generating Use statements in the generated script.
Grouping Extracted DDL	There are two option as to how the DDL is generated for an extract operation: **Group the Drop and Create DDL for each object together** - Objects are extracted one at a time. In the DDL generated by the extract, the Drop statement for a selected or referenced object is followed by the Create statement for that object. **Group DDL statements by type - all Drop statements, then all Create statements** - In the DDL generated by the extract, Drop statements for all selected or referenced objects are grouped to precede Create statements for those objects. This method ensures that all dependencies are respected. For more information on DROP options and dependencies, see DBMS-specific DDL Extract Options.

DBMS-specific DDL Extract Options

The DBMS-specific views of the DDL Extract tab let you specify:

- The object types to include DROP statements for
- The default dependent object types for each object type included in extraction/migration operations

You can choose to include DROP statements before you perform ad hoc DDL extractions. You can use this feature to modify and to re-compile database objects. To recompile a database object, drop it before recreating it. This option drops any existing objects of the same name before recreating the object. The data in the existing table is not saved when you specify a DROP statement for extracted DDL.

Caution: Because dropping an object is a destructive action, you should carefully consider including drop statements before activating this option.

You can also specify dependent object types included in extraction/migration operations. Each DBMS-specific view of the DDL Extract tab lets you specify the object dependencies for each object type. Dependent object types selected on that view are by default selected when you run an extraction/migration operation and can be overridden using those wizards. For more information see Extract.

Debug Options

The **Debug tab** of the Options Editor lets you specify operational preferences for the Debugger.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using the Debugger, see Embarcadero SQL Debugger.

The table below describes the options and functionality on the **Debug** tab:

Tab	Option	Description	Default
General	**Dependency Tree Option**	Lets you specify pre-fetch levels.	Pre-Fetch All Dependencies
Profiler	**Profiler Time Unit**	Lets you select a unit of milliseconds, seconds or minutes.	Milliseconds
	Save Profiler Reports	Lets you save profiler reports and type or browse for the report path.	Not Selected
Oracle	**Initialization Timeout (seconds)**	Specifies the point at which the application stops trying to initialize the debugger. If it cannot initialize the debugger in the specified time, it displays message in the Debug Output window.	60
	Debug Session Timeout (seconds)	Specifies the point at which the application terminates your debug session due to idle time.	7200
	Enable DBMS Output	Enables the Oracle built-in package, DBMS_OUTPUT, letting you send messages from stored procedures, packages, and triggers.	Selected
	Refresh Dependencies for each run	Refreshes the dependencies each time you run the debugger.	Not selected
	Compile with Debug Option	When debugging a procedure against an Oracle datasource, a compilation is performed before debugging. During a compilation, Oracle invalidates referencing objects. For example, if procedures A2 and A3 both reference procedure A1, debugging procedure A1 will result in procedures A2 and A3 being marked as invalidated. The **Compile with Debug Options** settings let you control compilation of dependent objects while debugging.	Compile dependent options
DB2	**Debug Session Timeout (seconds)**	Specifies the point at which the application terminates your debug session due to idle time.	300
	Compile with Debug Option before Debug Session	Lets you specify options.	Prompt Always

Directories Options

The **Debug tab** of the Options Editor lets you configure default directories when performing certain operations. You can set the default directories for:

- Wizard operations
- Report generation
- Schema extraction
- HTML templates for customizing reports

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

The table below describes the options and functionality on the **Directories** tab.
The %APPDATA%, %MYDOCS%, and %PROGFILES% placeholders represent the location of the Application Data, My Documents, and Program Files folders relevant to your operating system :

Option	Description	Default
Wizard Definitions	Specifies the name and location of the default directory for saving wizard operations. This option applies to wizard operations in which you have the option to save the definition file, such as data unload/load or schema migration operations.	*%APPDATA%***Embarcadero\DBArtisan\DefFiles**
HTML Reports	Specifies the name and location of the default directory for the output from generated HTML reports. For information on generating reports, see Report.	*%APPDATA%***Embarcadero\DBArtisan\Report**
Schema Extraction	Specifies the name and location of the default directory for placing the output from schema extraction operations.	*%MYDOCS%***Embarcadero\DBArtisan\Extract**
HTML Template	Specifies the name and location of the default directory where the HTML template on which to base HTML reports can be found. This feature lets you customize your HTML reports.	*%PROGFILES%***Embarcadero\DBA**<*version identifier*>**HtmlTpl**
User SQL Scripts	Specifies the name and location of the default directory for SQL Scripts.	*%MYDOCS%***Embarcadero\DBArtisan\UserSQLScripts**
Job Config Files	Specifies the location of any ETSQLX job configuration files you have set on your local machine. For more information, see ETSQLX Command Line Utility.	*%MYDOCS%***Embarcadero\DBArtisan\Directories\ETSQLXJobCfg**
#include Files	Specifies the name and location of the directory searched for files specified by a #include directive in the ISQL editor, Procedure Object Editor, or Package Body Object Editor if there are no paths specified on the Datasource Properties tab of the Datasource Registration Editor. For more information, see [RapArt_Datasource_Management.backup.htm#Ram41384 Registering Datasources]. For more information on compiler directives, see SQL Preprocessing: #define and #include.	*%MYDOCS%***Embarcadero\DBArtisan\Directories\IncludeFiles**

Explorer Options

The Explorer options are available on the following Options Editor tabs:

- Explorer Options - Main Tab
- Explorer Options - Organization Tab
- Explorer Options - Limits Tab

Explorer Options - Main Tab

The Options Editor's **Explorer** tab controls the appearance and behavior of the Datasource Navigator/Explorer.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For related information, see [RapArt_Datasource_Management.backup.htm#Ram19710 Using the Datasource Explorer/Navigator].

Use the following table as a guide to modifying settings on this tab

Setting	Description
Refresh after Object Commands	Refreshes the Explorer automatically after an object has been modified or created.
Retain View Setting on Close	Select to retain the current state of the Datasource Explorer so that it opens the same way on the next startup.
Column filters use regular expressions by default	Lets you use regular expressions when defining criteria for display of database objects in the right-hand pane of the Explorer.
Display Explorer on startup	Activates the Datasource Explorer on startup. For more information, see [RapArt_Datasource_Management.backup.htm#Ram19710 Using the Datasource Explorer/Navigator].
Landing Page View	Enables landing page views on a DBMS-by-DBMS basis. Landing pages display informative metadata that gives you a quick overview of your server/database.

In DBArtisan, additional settings are available:

Setting	Description
Display Explorer on startup	Activates the Datasource Explorer on startup. For more information, see [RapArt_Datasource_Management.backup.htm#Ram19710 Using the Datasource Explorer/Navigator].
Landing Page View	Enables landing page views on a DBMS-by-DBMS basis. Landing pages display informative metadata that gives you a quick overview of your server/database.

Explorer Options - Organization Tab

The Options Editor's **Explorer > Organization** tab controls the appearance and behavior of the Datasource Navigator/Explorer.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For related information, see [RapArt_Datasource_Management.backup.htm#Ram19710 Using the Datasource Explorer/Navigator].

Use the following table as a guide to modifying settings on this tab

Setting	Description
Explorer organization	Lets you select a default grouping option of **Organize by Object Owner** (groups objects, by object type, for each user) or **Organize by Object Type** (groups objects by object type for all users in the same list). Grouping by object owner is most efficient if you are working with databases containing a high number of objects. You can change the organization manually during a session. For details, see Basic Viewing Options in the Navigator/Explorer.
Show System Objects and **Show Only My Objects** (available if you are organizing the Datasource Explorer by object type)	Lets you select a default Datasource Explorer tree display options for system objects and objects that you own. Available options in the two groups are: **Never** - the relevant objects are not displayed by default. **Always** - the relevant objects are always displayed by default. **Use datasource filter** - The relevant default datasource filter is enabled. For more information, see Filtering in the Navigator/Explorer. You can change the **Show System Objects** and **Show Only My Objects** settings manually during a session. For details, see Basic Viewing Options in the Navigator/Explorer.

In DBArtisan, the following setting is also available

Setting	Description
Sort Type	Lets you select between alphanumeric and lexicographic options.

Explorer Options - Limits Tab

The Options Editor's **Explorer > Limits** tab controls the appearance and behavior of the Datasource Navigator/Explorer..

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For related information, see [RapArt_Datasource_Management.backup.htm#Ram19710 Using the Datasource Explorer/Navigator].

While the DBMS is queried as necessary when you navigate the node hierarchy, it also caches nodes for the purpose of quickly repopulating if you move away from and then back to a node. For example if you open a **Tables** node and then an **Indexes** node and then again select the **Tables** node, the cached **Tables** sub-nodes are displayed.

The following setings are available

Setting	Description
Cache Expiration Time	The duration a node is cached. Selecting a node after this expiration interval results in a fresh population.
Maximum number of children to display per tree node	Lets you place an upper limit of the number of nodes to display per node.

In DBArtisan, the following setting is also available

Setting	Description
Number of tree nodes to cache	Lets you place an upper limit of the number of nodes to cache. For example, if set to **3**, and you navigate to **Tables**, then **Indexes**, then **Primary Keys**, then **Triggers**, going back to **Tables** would force a fresh population since the **Triggers** node pushed out **Tables** from the cache;

General Options

The **General** tab of the Options Editor lets you specify general application options.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

The table below describes the options and functionality on the **General tab**:

Tab	Option	Description	Default
Main tab	**Confirm on Exit**	Ensures that a prompt is issued confirming the operation before exiting the application.	Selected
	Max Editors in a Single Open Operation	Specifies the maximum number of editors allowable from a single Open operation.	5
	Max Entries in Output Window	Specifies the maximum number of messages that can appear in the Output Window before the contents are flushed. This option conserves memory resources. You can clear the Output window or raise the maximum number allowed at any time.	1500

The **General** tab also offers the following DBMS-specific aub-tabs:

Oracle sub-tab

Option	Description	Default
Data Dictionary View Usage	Lets you synchronize use of the Oracle Data Dictionary DBA and ALL/USER views with the privileges associated with the role used in creating and using the datasource. **Role-based Views -** Views based on the user's role will be used. That is, users with DBA privileges will use DBA views. **DBA Views** - DBA views are always used. **All Views** - ALL views are always used. **Use Datasource Configuration for Views -** lets you set up Data Dictionary usage on a view-by-view basis when editing or registering a datasource. For more information, see Registering Datasources.	
Preserve Case in Object Identifiers	Preserves case of the database object.	Not selected

Sybase and SQL Server sub-tabs

Option	Description	Default
Rename action style	Lets you specify whether user-defined objects or datatypes are renamed using an extended ALTER or using the `sp_rename` procedure.	extended alter

Grid Properties (Results window) Options

The **Grid Properties** tab of the Option editor lets you set preferences for the layout and appearance of the grid in an Results window.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

The table below describes the options and functionality on the **Grid Properties** tab:

Interface Element	Options and descriptions		Default
Titles and Gridlines	**3D-Buttons**	Enables or disables a 3-D appearance on row and column headings in the grid.	Set
	Horizontal Gridlines	Enables or disables ruling between rows of the grid.	Set
	Vertical Gridlines	Enables or disables ruling between columns of the grid.	Set
	Mark Headings	Enables or disables highlighted row and column headings.	Set
Preview	Displays a preview of the settings currently selected in the TItles and Gridlines group.		

ISQL Options

The ISQL options are available on the following Options Editor tabs:

- ISQL Options - Main Tab
- ISQL Options - Oracle Tab
- ISQL Options - DB2 Tab
- ISQL Options - Sybase Tab
- ISQL Options - SQL Server Tab
- ISQL Options - Sybase IQ Tab
- ISQL Options - MySQL Tab
- ISQL Options - DB2 OS390 Tab
- ISQL Options - Editor Tab
- ISQL Options - Auto format Tab
- ISQL Options - Code Assist Tab

ISQL Options - Main Tab

The Options Editor's **ISQL** tab and its sub-tabs control the appearance and behavior of the SQL Editor.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using the SQL Editor, see Coding Environments/Editors.

Use the following table as a guide to modifying settings on this tab:

Max Errors Before Aborting Execution	Sets the maximum number of errors allowed before aborting the execution of a script. A zero indicates that the option is inactive and that you recognize no limit to the number of errors allowed. This value is used when step executing SQL scripts.
Execute Selected Text	Executes a portion of a selected SQL script.
Check Syntax When Executing	For DB2, required to execute DB2 call statements in the ISQL Window.
Automatically lock connection	When disabled, a prompt to commit or rollback the transaction is displayed when you close the ISQL editor window. Enabling this options disables the prompts and locks the connection automatically.
Prompt to lock database connection	Locks the database connection on execution.
Tabs	Sets the appearance of the ISQL Window tabs to either the top or bottom of the ISQL Window.
File Association	Specifies whether the application should open an unknown file type automatically into an ISQL Window or prompt you with a message that the file type is not recognized.
Query Plan Layout	Sets the default orientation of a graphical query plan. For more information, see Setting up the Execution Environment with Query options.

ISQL Options - Oracle Tab

The Options Editor's **ISQL > Oracle** tab controls the appearance and behavior of the SQL Editor when executing against an Oracle environment.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using the SQL Editor, see Coding Environments/Editors.

Use the following table as a guide to modifying settings on this tab:

Option	Description
Enable DBMS Output	Lets you specify **Buffer** size. 0 is the default.
Auto-Commit changes	Applies auto commit status changes to all open windows.
View xmltype as clob	When enabled, xmltype columns are displayed as CLOBs in the Results grid and the Data editor. Without this option selected, SELECT statements that qualify xmltype columns produce an **OciTypeBinder** conversion error. With the option selected, the SELECT submitted is modified to include a getclobval() method call.
Retain connection in pool after use	This option determines what happens to a connection after it is unlocked by the ISQL window. If selected, the connection will be returned to the application connection pool. If unselected, the unlocked connection will be disconnected.
Default Query Plan	Lets you select the default display, tree-based or graphical, when you generate a query plan. For more information, see Viewing a Tree-based or Graphical Query Plan (Oracle).
Load Query Options	If this option is not enabled, a set of default query options is loaded that customize your execution environment, and periodically sends those settings to the server. Enabling this option and specifying an XML file, forces load of a previously saved query options file each time an ISQL windows opens against this DBMS platform. This lets you override default query options. You can also enable this option when saving a query options file. For information on other load and save options, specific query options offered, manually updating query options, and conditions for which query options are sent to the server, see Setting up the Execution Environment with Query options.

ISQL Options - DB2 Tab

The Options Editor's **ISQL > DB2** tab controls the appearance and behavior of the SQL Editor when executing against aDB2 LUW environment.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using the SQL Editor, see Coding Environments/Editors.

Use the following table as a guide to modifying settings on this tab:

Option	Description
Set Isolation Level	Sets the default for the value of the **Isolation Level** option on the Query Options dialog. For details, see Setting up the Execution Environment with Query options.
Auto-Commit changes	Applies auto commit status changes to all open windows.
Create Explain plan tables if required	If set to TRUE, Explain Plan tables are created, as necessary. If set to FALSE and you don't manually create tables, Explain Plan operations will fail.
Create explain plan tables on the SYSTOOLS schema	If set to TRUE, Explain Plan tables are created on the SYSTOOLS schema. If the tables already exist in the user's default schema, those tables continue to be used. Refer to DB2 documentation for a listing of Explain Plan tables that must be deleted in order to use the SYSTOOLS option. If set to FALSE, Explain Plan tables are created under the user's default schema.
Retain connection in pool after use	This option determines what happens to a connection after it is unlocked by the ISQL window. If selected, the connection will be returned to the application connection pool. If unselected, the unlocked connection will be disconnected.
Load Query Options	If this option is not enabled, a set of default query options is loaded that customize your execution environment, and periodically sends those settings to the server. Enabling this option and specifying an XML file, forces load of a previously saved query options file each time an ISQL windows opens against this DBMS platform. This lets you override the default query options. You can also enable this option when saving a query options file. For information on other load and save options, specific query options offered, manually updating query options, and conditions for which query options are sent to the server, see Setting up the Execution Environment with Query options.

ISQL Options - Sybase Tab

The Options Editor's **ISQL > Sybase** tab controls the appearance and behavior of the SQL Editor when executing against a Sybase ASE environment.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using the SQL Editor, see Coding Environments/Editors.

Use the following table as a guide to modifying settings on this tab:

Option	Description
Auto-Commit changes	Applies auto commit status changes to all open windows.
Retain connection in pool after use	This option determines what happens to a connection after it is unlocked by the ISQL window. If selected, the connection will be returned to the application connection pool. If unselected, the unlocked connection will be disconnected.
Load Query Options	If this option is not enabled, a set of default query options is loaded. That set of query options customize your execution environment, and periodically sends those settings to the server. Enabling this option and specifying an XML file, forces load of a previously saved query options file each time an ISQL windows opens against this DBMS platform. This lets you override the default query options. You can also enable this option when saving a query options file. For information on other load and save options, specific query options offered, manually updating query options, and conditions for which query options are sent to the server, see Setting up the Execution Environment with Query options.

ISQL Options - Sybase IQ Tab

The Options Editor's **ISQL > Sybase IQ** tab controls the appearance and behavior of the SQL Editor when executing against a Sybase IQ environment.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using the SQL Editor, see Coding Environments/Editors.

Use the following table as a guide to modifying settings on this tab:

Option	Description
Enable Set Query Options	Sets the default value of the **Send Set Options** setting on the Query Options dialog. For details, see Setting up the Execution Environment with Query options. Changing this value does not affect ISQL sessions currently open.
Auto-Commit changes	Applies auto commit status changes to all open windows.
Retain connection in pool after use	This option determines what happens to a connection after it is unlocked by the ISQL window. If selected, the connection will be returned to the application connection pool. If unselected, the unlocked connection will be disconnected.
Load Query Options	If this option is not enabled, a set of default query options is loaded. That set of options customize your execution environment, and periodically sends those settings to the server. Enabling this option and specifying an XML file, forces load of a previously saved query options file each time an ISQL windows opens against this DBMS platform. This lets you override the default query options. You can also enable this option when saving a query options file. For information on other load and save options, specific query options offered, manually updating query options, and conditions for which query options are sent to the server, see Setting up the Execution Environment with Query options.

ISQL Options - SQL Server Tab

The Options Editor's **ISQL > SQL Server** tab controls the appearance and behavior of the SQL Editor when executing against a SQL Server environment.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using the SQL Editor, see Coding Environments/Editors.

Use the following table as a guide to modifying settings on this tab:

Option	Description
Enable Set Query Options	Sets the default value of the **Send Set Options** setting on the Query Options dialog. For details, see Setting up the Execution Environment with Query options. Changing this value does not affect ISQL sessions currently open.
Auto-Commit changes	Applies auto commit status changes to all open windows.
Retain connection in pool after use	This option determines what happens to a connection after it is unlocked by the ISQL window. If selected, the connection will be returned to the application connection pool. If unselected, the unlocked connection will be disconnected.
Load Query Options	If this option is not enabled, a set of default query options is loaded. That set of options customize your execution environment, and periodically sends those settings to the server. Enabling this option and specifying an XML file, forces load of a previously saved query options file each time an ISQL windows opens against this DBMS platform. This lets you override the default query options. You can also enable this option when saving a query options file. For information on other load and save options, specific query options offered, manually updating query options, and conditions for which query options are sent to the server, see Setting up the Execution Environment with Query options.

ISQL Options - MySQL Tab

The Options Editor's **ISQL > MySQL** tab controls the appearance and behavior of the SQL Editor when executing against a MySQL environment.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using the SQL Editor, see Coding Environments/Editors.

Use the following table as a guide to modifying settings on this tab:

Option	Description
Retain connection in pool after use	This option determines what happens to a connection after it is unlocked by the ISQL window. If selected, the connection will be returned to the application connection pool. If unselected, the unlocked connection will be disconnected.
Load Query Options	If this option is not enabled, a set of default query options is loaded. That set of options customizes your execution environment, and periodically sends those settings to the server. Enabling this option and specifying an XML file, forces load of a previously saved query options file each time an ISQL windows opens against this DBMS platform. This lets you override the default query options. You can also enable this option when saving a query options file. For information on other load and save options, specific query options offered, manually updating query options, and conditions for which query options are sent to the server, see Setting up the Execution Environment with Query options.

ISQL Options - DB2 OS390 Tab

The Options Editor's **ISQL > DB2 OS390** tab controls the appearance and behavior of the SQL Editor when executing against a DB2 z/OS environment.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using the SQL Editor, see Coding Environments/Editors.

Use the following table as a guide to modifying settings on this tab:

Option	Description
Retain connection in pool after use	This option determines what happens to a connection after it is unlocked by the ISQL window. If selected, the connection will be returned to the application connection pool. If unselected, the unlocked connection will be disconnected.
Load Query Options	If this option is not enabled, a set of default query options is loaded. That set of options customizes your execution environment, and periodically sends those settings to the server. Enabling this option and specifying an XML file, forces load of a previously saved query options file each time an ISQL windows opens against this DBMS platform. This lets you override the default query options. You can also enable this option when saving a query options file. For information on other load and save options, specific query options offered, manually updating query options, and conditions for which query options are sent to the server, see Setting up the Execution Environment with Query options.

ISQL Options - Teradata Tab

The Options Editor's **ISQL > Teradata** tab controls the appearance and behavior of the SQL Editor when executing against a Teradata environment.

Note: Teradata datasources support is intended as a technical preview. Minimal functionality is provided. For information on technical previews, see the **Read Me** at http://docs.embarcadero.com.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using the SQL Editor, see Coding Environments/Editors.

Use the following table as a guide to modifying settings on this tab:

Option	Description
Retain connection in pool after use	This option determines what happens to a connection after it is unlocked by the ISQL window. If selected, the connection will be returned to the application connection pool. If unselected, the unlocked connection will be disconnected.
Load Query Options	If this option is not enabled, a set of default query options is loaded. That set of options customizes your execution environment, and periodically sends those settings to the server. Enabling this option and specifying an XML file, forces load of a previously saved query options file each time an ISQL windows opens against this DBMS platform. This lets you override the default query options. You can also enable this option when saving a query options file. For information on other load and save options, specific query options offered, manually updating query options, and conditions for which query options are sent to the server, see Setting up the Execution Environment with Query options.

ISQL Options - Editor Tab

The Options Editor's **ISQL > Editor** tab controls the appearance and behavior of the SQL Editor window.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using the SQL Editor, see Coding Environments/Editors.

The table below describes the options and functionality on the **Editor** tab:

Group	Option	Description	Default
Window	**Show Toolbar** and **Show Status Bar**	Enables/disables these ISQL Window user interface elements.	Selected
	Maximize on new or open	Indicates that the ISQL Editorwindow should be maximized on opening. If you already have an active MDI Window that is maximized, the default behavior is to maximize a new child window. To deactivate this option, ensure you do not have any active MDI Windows, such as the Explorer.	Selected
	File Tracking	Indicates that the ISQL Editor should use the File Tracking Facility to monitor the status of a file. If a file has been modified and saved outside the application, the application loads the most current version of the file into the IISQL Editor based on the options set for Auto-Reload File (see above.)	Selected

	Auto-Reload File	If File Tracking is enabled, indicates that the application should automatically reload a file that has been externally modified without prompting you. If you turn this option off, you are prompted before reloading your file if external changes have been saved.	Not selected
	Auto-Save File	Indicates that files in the ISQL Editor should automatically be saved at the indicated time interval. Backup files are saved to: **In DBArtisan, %APPDATA%\DBArtisan\Backup\**. Backup files use a naming convention of the form **~ET***xxxx***.tmp**.. For a listing of backup file names and the path of the file to which they correspond, see the registry key: In DBArtisan, **HKEY_CURRENT_USER\Software\Embarcadero\DBArtisan\***version***\Backup**	Selected (5 minutes)
Command History	**Save File Before Overwriting**	Specifies the action you want the application to take when selecting a command from the Command History box. You have the option to be reminded to save a file before overwriting (Ask First), to automatically save a file before overwriting (Always), or to automatically overwrite the file with the command (Never).	Ask First
	Save Most Recent	Specifies the number of commands you want to save in the Command History list in the top of the ISQL Window toolbar. The maximum value is 99.	15
Printing	Lets you select common printer options.		
Appearance	**Enable Syntax Highlighting**	Sets syntax highlighting on so that all keywords and comments are colored for easier reading and debugging.	Selected
	Show Line Numbers	Places line numbers in the left column of an ISQL Window.	Selected
	Enable Outlining	Enables and disables outlining.	Selected
	Enable Text Wrapping	Enables and disables a typical text wrap feature.	Selected
	Editor Font	Sets the font face, style, and size displayed in the editor.	Available
	Syntax Colors	Sets syntax coloring for keywords, comments, quotes, and default text for various file types and scripts from the **Syntax Coloring** dialog.	Available
Formatting	**Auto Indent**	Sets automatic indentation for each carriage return, new line in your SQL script.	Selected
	Expand Tabs and **Tab Size**	Sets tabs as the specified number of spaces in result sets.	Selected (4)

Clipboard Line Endings	**None**	When copying text from the editor, no substitution is made for end-of-line characters. This provides the fastest cut and paste operation.	Selected
	Line Endings	When copying text from the editor, end-of-line characters are replaced with platform-specific equivalents (Windows - CR LF, UNIX - LF, Macintosh - CR). This lets you copy text into environments such as Notepad, that require platform-specific end-of-line characters. This option avoids unnecessary characters and incorrect display, but cut-and-paste operations take more time to complete.	

ISQL Options - Auto format Tab

After opening the Options editor, you can make changes to the **Auto Format** tab. of the **ISQL** pane.

The Options Editor's **ISQL > Auto Format** tab lets you specify the style and spacing of SQL statements in an ISQL window when you choose to auto format SQL in the ISQL editor.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using the SQL Editor, see Coding Environments/Editors.

On opening, the **Preview** area shows an SQL statement formatted according to the current **Auto Format** settings. Click **Edit** to open the **Auto Format Options** dialog. Use the following table as a guide to understanding and modifying the settings in this dialog:

Setting	Description
Keywords	Lets you select a character case (UPPERCASE, LOWERCASE, INITIALCAPS, or NOCHANGE) formatting treatment for SQL keywords.
Right Margin	Specifies the maximum number of characters per line.
New line before keyword	If enabled, a new line is forced before every SQL keyword. If disabled, lines are only forced for statement type and clause type keywords.
BEGIN..END block	If enabled, empty BEGIN..END blocks occupy a single line.
Stack lists	If enabled, a new line and indenting is forced for each item in comma-separated list such as argument lists.
List indent size	If **Stack lists** is enabled, this control lets you specify the number of spaces each list item will be indented from the current offset.
Parenthesis indent size	Lets you specify the number of indenting spaces for forced lines following an open parenthesis.
Conditions format style	Specifies how conditions in clauses are formatted: CONDITIONS_WRAPPED - lines are not forced before conditions. CONDITIONS_STACKED_WITH_LEADING_OPERATORS - a new line is forced for each condition in a clause, with logical operators, if present for a line, displayed at the start of the line. CONDITIONS_STACKED_WITH_TRAILING_OPERATORS - a new line is forced for each condition in a clause, with logical operators, if present for a line, displayed at the end of the line.
Conditions stack threshold	For **Conditions format style** selections that specify stacking, this value specifies the minimum number of conditions that must appear in a clause before conditions are stacked.
THEN statements	If enabled, simple THEN clauses are kept to a single line.

ISQL Options - Code Assist Tab

Code assist options are available on the following tabs:

- ISQL Options - Code Assist - Main Tab
- ISQL Options - Code Assist - Advanced Tab

ISQL Options - Code Assist - Main Tab

The Options Editor's **ISQL > Code Assist** tab lets you activate and select options for coding assistance and error-detection features available from the ISQL editor:

- **Code Complete** - lets you insert or replace object names, selected from suggestion lists, as you edit a script. For more information, see Code Complete.

- **Semantic validation** - detects object name references to objects not present in the datasource index. For more information, see Semantic Validation.
- **Real-time syntax checking** - on-the-fly syntax checking is performed as you type. For more information, see Syntax Checking.
- **Hyperlinks** - lets you invoke object actions against objects referenced in a script. For more information, see Hyperlink Object Actions.
- **Object metadata display** - lets you view Connect server metadata for referenced objects. For details, see Viewing Embarcadero Team Server Model Metadata in the SQL Editor.

The **ISQL - Code Assist** tab has the following settings:

Group/Option		Description
Code Complete	**Enable Code Complete**	Lets you enable or disable the Code Complete feature. When disabled, Code Complete cannot be invoked automatically or manually. Because of the overhead in maintaining internal resources used in implementing this feature, a slight performance increase can result from disabling the feature.
	Enable auto-activation and **Auto-activation delay**	When auto-activation is disabled, the Code Complete feature must be invoked manually. When auto-activation is enabled, the Code Complete feature is invoked automatically each time the interval between keystrokes exceeds the specified **Auto-activation delay**.
	Insert single proposals automatically	Specifies that if a Code Complete suggestion list would contain only a single suggestion, that suggestion is inserted automatically.
	Fully qualify completions automatically	Specifies that Code Complete results are returned fully qualified, rather than the minimum required to identify the object.
Code Validation	**Enable Real-time syntax checking**	If enabled, syntax-checking is performed as you type. If disabled, syntax checks must be initiated manually.
	Severity levels for semantic validation problems	Lets you select the severity level (ERROR, WARNING, or IGNORE) associated with detected semantic errors. The WARNING option provides a contrast to syntax errors, which are always flagged with a severity level of ERROR.The IGNORE setting disables the feature. For more information, see Semantic Validation.
	Perform syntax and semantic validation on files smaller than	Places an upper limit on the size of files for which automatic syntax checking and semantic validation are active.

DB Action Menu	**Enable DBAction menu in the editor**	Enables the Hyperlinks feature.
	Show DB Actions in submenu	If unselected, Hyperlink menu actions are available as first level options in the context menu. If selected, Hyperlink menu actions are available as a **DB Actions** submenu.
	Key used in conjunction with right-mouse	Lets you select the key (CTRL, SHIFT, or CTRL+SHIFT) used in conjunction with right-clicking, that activates the object action menu against a supported object type reference in a script. The default keystroke combination is CTRL+right-click.
Restore defaults	Restores all settings on the tab to the original defaults.	
Metadata	Enables display/use of metadata for referenced objects, obtained from a Connect server repository, in the following user interface elements: tooltips, window gutter, code and complete proposals. You can also have metadata added as comments when formatting code. For detailed information on setting up and using this feature, see Viewing Embarcadero Team Server Model Metadata in the SQL Editor.	

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using the SQL Editor, see Coding Environments/Editors.

ISQL Options - Code Assist - Advanced Tab

Coding assistance and error detection features such as Hyperlink object actions, Code Complete, and Semantic Validation rely on:

- Parsing of SQL entered in the SQL Editor into keywords, object name references, data, and so on
- A database object index generated for a connected datasource. On the first connection to a datasource, an index of specified object types is generated, with an imposed, configurable, upper limit on the number of objects contained in the index. On each subsequent connection, the index is updated to reflect any additions or changes on the datasource.

Note: The object index is stored in a persistent database with a database path of **%TEMP%**. With that in mind, the index may be deleted by the system administrator or automated administration-related processes.

When coding assistance and error detection features are enabled, the index is queried for the names of objects referenced in your scripts queries. This information is then used for tasks such as recognizing object types, detecting misspelled object names, and populating suggestion lists.

The Options Editor's **ISQL > Code Assist > Advanced** tab lets you force parsing, if required, and customize how the index is generated. Since there is performance overhead associated with generation and usage of the index, you can control the properties and usage of the index.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

The table below describes the options and functionality on the **ISQL > Code Assist > Advanced** Tab of the Options Editor:

Option	Description
Scan entire file for context	This control is enabled only if the **ISQL > Code Assist** tab's **Enable Real-time syntax checking** check box is deselected and the **Severity levels for semantic validation problem**s is set to IGNORE. When this control is enabled, SQL Editor content continues to be parsed. That is, if you have disabled real-time syntax checking and effectively disabled semantic validation, you can still use the Hyperlinks feature.
Limit scanning to files smaller than	Lets you place an upper limit on the size of files parsed for Code Assist features.
Object types to be indexed	Lets you select specific object types (aliases, synonyms, functions, packages, procedures, tables, or views) that will be included in the object index. While this can be used to reduce the index size and therefore improve performance, Hyperlinks, Semantic Validation, and Code Complete functionality associated with object types not included in the index will not be available.
Recycle cache space when index size exceeds	This imposes a soft ceiling on the number of objects stored in the index. After updating the index on connection to a datasource, the total number of objects in the index is compared to this value. If the value is exceeded, the oldest definitions are deleted.
The maximum number of results to return when querying the object index	Specifies the number of results when querying against the cache databases, to improve performance when working with large databases. This limit is applied per object type.
Refresh	Each time you open an ISQL editor window, schema information for the object types supported is obtained for the Code Complete and Semantic Validation features. That information is updated with any modifications that you make and persists until the window is closed. The **Refresh** button ensures that the information is current in case other users or applications are modifying the schema while the current SQL editor session window is open.

Java Options

DBArtisan requires JDBC connections. The Options Editor's **Java** tab lets you set or change options that apply to the Java virtual machine (JVM) that is running on the client.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

The table below describes the options and functionality of the **Java** tab:

Option	Description	Default
JVM Options:		
Initial Heap Size	Set the size, in MB, for the repository where live and dead objects comingle with free memory. If memory runs out, JVM executions stop so the garbage can be collected and expunged. Each platform responds differently, so trial and error can help you maximize performance.	64 MB
Maximum Heap Size	Set the upper limit for your heap size.	64 MB
Maximum Perm Gen	Set the upper limit for the permanent generation size.	Version-specific
Additional Options	Add options here ONLY in consultation with Embarcadero Technical Support.	N/A

LDAP Configuration Options

When using the Discover Datasources feature, you can have an LDAP Server searched for datasources. Similarly, when manually registering a datasource, you can select from server names obtained from an LDAP Server. The **LDAP Configuration** tab of the Options Editor lets you provide the information required to establish a session with the LDAP Server and obtain the required datasource information.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

The table below describes the options and functionality on the **LDAP Configuration** tab:

Group	Option	Description
	Enable LDAP	Enables LDAP lookup. This check box must be selected in order for the Discover Datasources feature to search an LDAP Servers or for server names obtained from an LDAP server to be available when manually registering datasources.
Server	**Host Port**	The FQDN or IP address of the LDAP server.
	Search Base DN and Suggest	Lets you provide the distinguished name to be used as the search base. The **Suggest** button is only enabled after you select an authentication type and provide details, as required. It queries the server for possible DNs and populates the combo box with the results.
Authentication	**Anonymous Login**	Selecting this check box directs DBArtisan to send LDAP requests without performing a bind. This also disables the other **Authentication** controls.
	User, **Password** , and Remember	Let you provide the credentials used when sending LDAP requests.
Options	**Size Limit** and Time Limit	Lets you specify the maximum number of entries returnable and the maximum time, in seconds, to wait for the search to complete.
	Protocol Version	The LDAP protocol version.

For more information, see:

- Manually Registering or Editing Datasources
- Automatically Discovering Datasources

Logging Options

After opening the Options editor (see Specifying Application Preferences and Feature Options), you can make changes to the **Logging** tab. The Logging tab lets you set defaults that specify the behavior and placement of SQL, output and trace file logging.

SQL logging provides an audit trail. You can examine this log to determine the SQL executed to complete a task. The table below describes the SQL Logging options and functionality on the **Logging** tab:

Option	Description
Log all SQL Statements to a File	Indicates that the application should log all of the SQL that it executes to a file. SQL logging provides an audit trail. You can examine this log to determine the SQL executed to complete a task.
Logfile Path	If you choose to log the SQL generated by the application, specify the drive, directory, and file name.
Max File Size	Specifies the maximum size for the logfile. When the logfile reaches this threshold, it automatically starts deleting lines in the logfile (starting with the oldest statements) to remain within the specified size limit.
Truncate	Empties the entire contents of the logfile.

Output logging lets you monitor only messages issued by the server versus all SQL logged by the application. You can examine this log to determine all messages the server issued. The table below describes the output logging options and functionality on the **Logging** tab:

Option	Description
Log all Output Messages to a File	Indicates that the application should log all server messages sent to the Output window. This type of logging lets you monitor only messages issued by the server versus all SQL logged by the application. You can examine this log to determine all messages the server issued.
Logfile Path	If you choose to log the server messages generated in the Output window, specify the drive, directory, and file name.
Max File Size	Specifies the maximum size for the output logfile. When the output logfile reaches this threshold, it automatically starts deleting lines in the file (starting with the oldest statements) to remain within the specified size limit.
Truncate	Empties the entire contents of the output logfile.

Tracing is typically disabled (**Trace Level** set to **Off**) unless you are trying to diagnose a problem. When enabled, the sequence of trace messages generated by the application, severity specified, is written to a **.log** file. THe file name includes a date stamp.

A trace file is created on each run, one per day. If multiple instances of the same application are running, each instance will also have its own trace file for that day. The trace file will be written to until the OS denies writing to it due to the file being too big or not enough disk space. Unless the

Trace Level is set to the higher levels such as **Trace** or **Max**, there won't be much in this file normally, so space should not be a problem. Up to 10 days of trace files are kept.

If tracing is turned on in the application (any **Trace Level** other than **Off** is set) the trace file will be locked until the application exits or the **Trace Level** is set to **Off** . The table below describes the Trace File options:

Option	Description	
Trace Path	Fully specify the directory where trace files are to be stored.	
Trace Level	In descending order of severity (and ascending order of total messages logged), **Trace Level** options are **Fatal Errors**, **Errors**, **Warning**, **Info**, **Debug**, **Trace**, and **Max**.	
Truncate	Empties the entire contents of the current trace file.	Not available

Migration Options (DBArtisan)

After opening the Options editor (see Specifying Application Preferences and Feature Options), you can make changes to the **Migration** tab. The **Migration** tab sets the default values effective when you open the Migration Wizard, a utility that lets you migrate schema and data. For information on this utility, see Using the Schema Migration Wizard (DBArtisan).

MySQL Utilities Options

DBArtisan integrates with MySQL utilities. For access to these utilities, you need to specify their location in the **MySQL Utilities** tab of the Options Editor. You can use the MySQL Dump and Import Utilities.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

The table below describes the options and functionality on the **MySQL Utilities** tab

Option	Description
mysqldump	Specifies a path for the MySQL dump utility, mysqldump.exe. By default, MySQL installs this utility in the C:\mysql\bin directory.
mysqlimport	Specifies a path for the MySQL import utility, mysqlimport.exe. By default, MySQL installs this utility in the C:\mysql\bin directory.

Oracle Utilities Options

To make use of features that integrate with Oracle utilities, you need to specify their location on the **Oracle Utilities** tab of the Options Editor. You can use the Oracle Export, Import Utilities, and SQL * Loader.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

The table below describes the options and functionality on the **Oracle Utilities** tab:

Option	Description
Data Pump Export	Specifies a path for the Oracle **expdp.exe** utility.
Data Pump Import	Specifies a path for the Oracle **impdp.exe** utility.
Export	Specifies a path for the Oracle Export utility. By default, Oracle installs to C:\Orant\Bin directory.
Import	Specifies a path for the Oracle Import utility. By default, Oracle installs to C:\Orant\Bin directory.
SQL*Loader	Specifies a path for the SQL * Loader utility. By default, Oracle installs to C:\Orant\Bin directory.
Default Bind Size	Specifies the bind size. Default is set to 70KB.

Query Builder Options

The Options Editor's **Query Builder Options** tab lets you specify operational and execution preferences for the Query Builder. Option parameters set on the Options Editor elicit a prompt if there are different options set on an open individual session. Global options override properties set within individual Query Builder sessions.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using the Query Builder, see Query Builder.

The table below describes the options and functionality on the **Query Builder tab**:

Interface Element	Option	Description	Default
Code Generation	**Generate Use Database statement**	Adds a line of SQL code indicating which database or instance is used in the statement.	Selected
	Generate owner names	Adds a line of SQL code showing the table owner name as part of the query.	Selected
	Include Row Count limits	Includes the output row limit set in the Execution settings.	Not selected
	Generate SQL/92 if supported by DBMS	SQL/92 is a standard for relational database management systems.	Not selected
Execution	To set the maximum number of rows in your result set, type the number in the dialog. This lessens congestion of server processes when queries execute by setting row count limits.		1000 rows
General	**Show Column Data types in Query Diagram**	Reveals the data type in each column for tables in the SQL Diagram pane.	Not selected
	Confirm on Item delete	Opens a **Confirm Delete** dialog when an item is deleted.	Selected
	Auto Populate Views	Checks syntax every time an execute statement, refresh or copy statement begins.	Not Selected
	Auto Format	Automatically sets style and spacing of display.	Selected
Auto Join	**Run Automatically**	Automatically detects names and data types, and create joins for multiple tables.	Selected
	Require Indexes	Joins only indexed columns. Requires an indexed column for joins.	Selected
	Require same data type	Automatically joins columns with the same data type.	Selected

Syntax Checker	**Automatic Syntax Check**	Automatically checks SELECT and CREATE VIEW statements for errors.	Selected
	Warn on non index join	Returns a warning when it detects a join against a non-indexed column, or a column not participating in a primary key	Not selected
Display	Lets you sets the style, size, and color of **Column Font** and **Title Font**. Also lets you set the background **Table Color** for the SQL Diagram Pane.		Available

Reports Options

The Options Editor's **Reports** tab lets you:

- Append a date and timestamp to report file names. This prevents existing reports from being overwritten.
- Save reports in date-specific folders.
- Save reports in PDF format as opposed to HTML.
- Let you include a provided image and title to reports.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on generating reports, see Report.

Results (ISQL) Options

The Options Editor's **Results** tab lets you specify operational and appearance details for the SQL Editor.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using the SQL Editor, see Coding Environments/Editors.

The table below describes the options and functionality on the **Results tab**:

Interface Element		Options and descriptions
Results Option pane	**Result Window**	**Single Window** - Displays all results in one tabbed result window. Multiple result sets together in the window. Single Window and Multiple Windows options are mutually exclusive. **Multiple Windows** - Displays multiple result sets one result set per window. **Attached to Editor** - Sets results as tabbed windows attached to the ISQL window. Used in conjunction with Single Window option or Multiple Window option. Attached and Unattached options are mutually exclusive. **Unattached** - Sets results appear separate from the ISQL Window. Used in conjunction with Single Window option or Multiple Windows option. **Reuse Window** - Sets new result sets to overwrite any current result sets in an open Result Window. Only valid for Single and Attached to Editor combination.
	Results File	**Mail File Type** - Selects the file type to use when sending result sets via a MAPI-compliant mail package. Valid formats include the proprietary Results type, Tab delimited, Comma separated, and HTML. **Schedule File Type** - Selects the schedule file type. Valid formats include Tab delimited, Comma separated, and HTML. **Include column titles when saving** - Includes column titles when saving a result set. If this option is turned off, result sets are not saved.
	Result Set Options	**Default Rowcount** - Lets you limit the number of rows returned to the result window of the ISQL window (default 0). **Sybase ASE and SQL Server: Text Size (bytes)** - Lets you specify the text size (default 8192). **Oracle:LONG Size (bytes)** - Lets you specify the LONG size (default 8192). **LOB Preview: Text Size (bytes)** - Specifies the length of the preview of LOB column data (default 4096).
	Sort Type	Lets you select a result sorting: **Alphameric** or **Lexicographic**.
Format pane	**Column Formatting**	**Auto Format (Best Fit)** - Sets column widths automatically to accommodate the longest piece of data in a column. Large queries depend on the longest row formatting, so activating this option can affect performance. **Use pre-defined column** - Lets you select column type and character length
	Enable Date/Time Format	Lets you select the date/time format.

Format	**Standard Grid** - Displays all result sets in a standard grid format. Result sets are displayed in grid format in SQL Editors that are opened after you have selected this option. This has no effect on ISQL Editors that are already open. **HTML** - Displays all result sets as HTML tables. Result sets are displaedin HTML format in ISQL Editors that are opened after you have selected this option. This has no effect on SQL Editors that are already open. **ASCII Text** - Displays all result sets as ASCII Text. Result sets are displayed in ASCII Text format in ISQL Editors that are opened after you have selected this option. There is no effect on SQL Editors that are already open. **Grid Font** and **Printer Font** buttons - Open a **Font** dialog, letting you select the font, style, and size for the result sets grid or printed result sets. **Enable Locale** - If selected, numbers are formatted (thousands separator, decimal separator) as locale-specific strings. If unselected, numbers are formatted according to the **C** locale.

SMTP Mail Options

The Options Editor's **SMTP Mail Options** tab lets you specify outgoing notification e-mail message options.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

The table below describes the options and functionality on the **SMTP Mail** tab:

Option	Description	Default
Send messages through SMTP	Enables SMTP messaging and makes the other controls on this tab available.	Unselected
Name	Name that appear as the e-mail sender.	Name you specified during installation.
E-mail Address	Address to send e-mails.	E-mail address you specified during installation.
Authentication	Lets you specify authentication options.	None
User Name	User name for authentication.	Not available
Password	Password for authentication.	Not available
Host Name	SMTP server for outgoing messages. For Microsoft Outlook, select **Tools**, and then **Accounts**. On the **Mail** tab, select target account, and then click **Properties**. On **Servers** tab, copy the **Outgoing Mail(SMPT)** and paste.	Host Name you specified during installation.
Port Number	Port number you connect to on your outgoing SMTP server.	25
Test	Opens an SMTP Configuration Test e-mail addressed to your e-mail address. Click **Send Mail** to send the e-mail.	Available
Bind to	Your IP address the message is bound to.	ANY_IP_ADDRESS
Encoding	E-mail encoding.	Western Europe (ISO)
Send messages Mime encoded	Messages encoded using Multipurpose Internet Mail Extensions (MIME) support enriched content and attachments.	Selected
Send all messages as HTML	Messages include text formatting.	Selected
Auto Connect to the Internet	An internet connection is established at launch.	Selected

Space Analyst Options (DBArtisan)

The Options Editor's **Space Analyst** tab lets you specify DBMS-specific parameters.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

Note: For information on using Space Analyst, see Space Analyst (DBArtisan).

The table below describes the options and functionality on the **Space Analyst tab**:

Tab	Option	Description	Default
Oracle	**Exclude objects owned by SYS in Reorganization Wizard.**	The Space Analyst Reorganization Wizard excludes SYS objects.	Selected
	Auto populate Table Reorganization Details Grid	Space Analyst auto-populates table reorganization details.	Not Selected
	Auto populate Index Reorganization Details Grid	Space Analyst auto-populates index reorganization details.	Not Selected
	Populate Index Reorganization Details Grid with Deleted Leaf Statistics	Space Analyst gathers deleted leaf row statistics for an index and displays details.	Not Selected
SQL Server, Sybase ASE, and IBM DB2 for Windows, Unix, and Linux	Offers the option to update the Table Details Grid and Index Details Grid.		Not Selected

Team Server Options

The **Team Server** tab of the Options Editor integrates DBArtisan with Embarcadero Team Server, a data governance metadata repository.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

The table below describes the options and functionality on the **Team Server** tab:

Option	Description
Use Team Server	Enables use of the Embarcadero Team Server repository.
Connect URL	The login URL of the Team Server repository.

Note: These controls can be set or disabled using registry entries. For details, see Locking Down Features Using the Registry.

For more information on repository-based resources, see DB Team Server Support.

Version Control Options

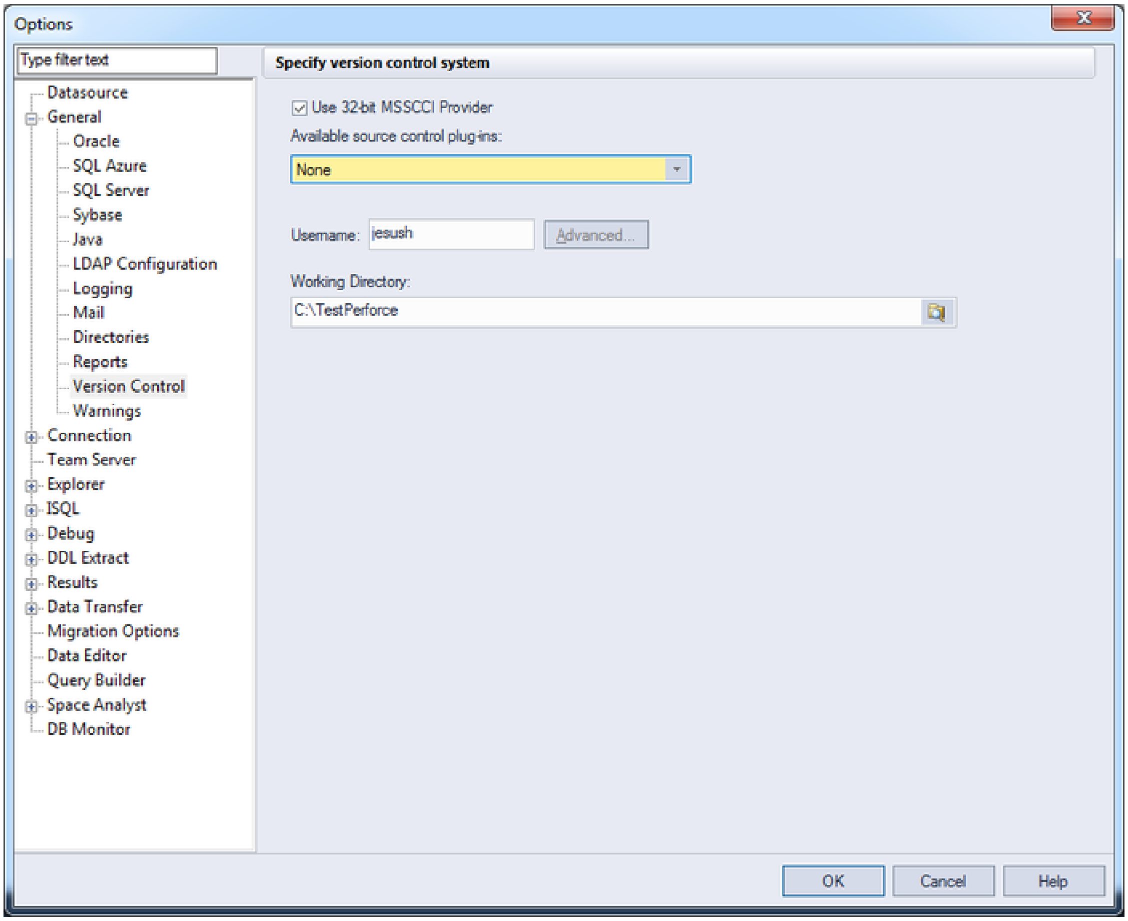
The Options Editor's **Version Control tab** lets you select which version control system you want to use as the underlying version control system.

For configuring the Version Control functionality:

1. Open the **Options Editor** Clicking **File** menu and selecting **Options**.
2. Select **General** and **Version Control** to open the **Version Control Tab**.
3. Select the Version Control System in the **Available source control plug-ins** drop-down list.

Note: If you are using a 64 bit version of DBArtisan with a 32 bit MSSCCI version control provider, the **Use 32-bit MSSCCI Provider" checkbox must be checked.**

1. Select the **Username** for the Version Control System.
2. Specify the local **Working Directory**
3. Click **Ok**



Specify version control system

☑ Use 32-bit MSSCCI Provider

Available source control plug-ins: None

Username: jesush

Working Directory: C:\TestPerforce

Note: After Adding a project or file to the version control, the Version Control System launches its Configuration Wizard to start to work with it and add that project or file to the version control.

Topics

- Version Control

Warnings Options

You can have warning messages issued to users whenever improper operations are attempted in a database. Warning messages differ by platform. The Options Editor's **Warnings** tab lets you select the DBMS-specific events that generate warnings.

Note: For information on opening the Options Editor, see Specifying Application Preferences and Feature Options.

The tables below describes the options of the **Warnings** tab:

Option	Description	Platform
Create an index on the same tablespace as the associated table	Issues a warning message whenever a user does not create an index on a different tablespace than the table. This makes it possible for the server to read the index and table data in parallel.	DB2
Create an index on the same tablespace as the associated table	Issues a warning message whenever a user does not create an index on a different tablespace than the table. This makes it possible for the server to read the index and table data in parallel.	Oracle
Create an object in the SYSTEM tablespace	Issues a warning message whenever a user tries to create or place an object on the SYSTEM tablespace.	Oracle
Create a user with default or temp tablespace as the SYSTEM tablespace	Issues a warning message when a user is created with a default or temp tablespace on the SYSTEM tablespace.	Oracle
Create an object in the master database	Issues a warning message when an object is created in the master database.	Microsoft SQL Server and Sybase ASE
Create a table or index on the default segment	Issues a warning message when a table or index is created on the default segment.	Microsoft SQL Server and Sybase ASE
Create a non-clustered index on same segment as the associated table	Issues a warning message when a non-clustered index is created on the same segment as the associated table.	Microsoft SQL Server and Sybase ASE

Locking Down Features Using the Registry

Administrators can restrict certain features on individual machines through registry settings. Those settings are stored in the following location:

- In DBArtisan, HKLM\Software\Embarcadero\DBArtisan\Admin

The following general datasource and connection settings are available:

Name (and Type)	Description and Supported Values
RestrictedPlatforms (REG_SZ)	A comma-separated list (**db2=0,mssqlserver=1**, for example) dictating DBMS platform support on the machine. Valid names for the *name=value* pairs are **db2**, **os390**, **mssqlserver**, **mysql**, **oracle**, **sybase**, **mssqlazure**, **odbc**, **jdbc**, and **sybaseiq**. Any non-zero value for a **name=value** pair results in that platform not being unavailable on that machine. A zero value (the default) results in the platform being available on that machine.
DisableAutoConnect (REG_DWORD)	When set to any non-zero value, users cannot use the auto-connect feature when registering or logging in to a datasource.
SharedDatasourceCatalogPath (REG_SZ)	Providing a path to the network shared datasource catalog forces DBArtisan to use that datasource catalog storage option. Registry-based and file-based catalog storage will be unavailable on this machine.

The following settings control configuration options on the **Team Server** page of the Options Editor and the **Embarcadero Team Server Log In** dialog. For details, see Team Server Options and Setting up and Connecting to DB Team Server.

Name (and Type)	Description and Supported Values
ConnectURL (REG_SZ)	The full Embarcadero Team Server URL that the user must use if Team Server is enabled. If a value is specified, the **Team Server URL** control on the **Team Server** page of the Options Editor is set to read-only.
UseConnect (REG_DWORD)	If a value of 1 (TRUE) is specified, use of Embarcadero Team Server is enabled. If a value of 0 (FALSE) is specified, use of Team Server is disabled. If either value is specified, the **Use Team Server** control on the **Team Server** page of the Options Editor is disabled.
AllowKeepMeSignedIn (REG_DWORD)	If a value of 1 (TRUE) is specified, the **Keep me signed in** control on the **Embarcadero Team Server - Log In** dialog is is set to an unchecked state and the control is disabled.

UseConnect and **AllowKeepMeSignedIn** is type REG_DWORD (legal values for both are 0 for FALSE and 1 for TRUE).

Administrators not wanting to use these settings from the HKLM root key, can use the same options in HKCU root key:

- In DBArtisan, **HKCU\Software\Embarcadero\DBArtisan\Admin**

If the option is specified in both locations, HKLM takes precedence over HKCU.

Datasource and Server Management

After installation, you must set up datasource definitions to establish reusable connections to your database servers. The following topics describe the function of datasources, the process of establishing datasource connections, and the related options provided:

Registering Datasources	Provides instructions on how to set up datasource definitions, manually and automatically.
Connecting to Datasources	Provides instructions on how to how to connect to datasources, manually and automatically.
Trouble-shooting Client Setup	Provides advice on solving common connectivity problems.
Using the Datasource Explorer/Navigator	Describes the user interface mechanism that lets you browse among datasources, drill down to the database object level, and initiate datasource or object actions.
Using the Manage Datasources Utility	Describes a utility offering shortcuts to datasource options.

In addition, DBArtisan offers a set of server operation and configuration options. For details, see Server Management Options (DBArtisan).

Registering Datasources

A datasource is the connection profile that lets you have access to a database, and, therefore, to your data. So, when you register a datasource, you're telling the application how to get at the database(s) you want to operate on. In fact, all database activities are performed through datasources, so you need a datasource profile for each database instance (or database server) in your enterprise. Each connection profile consists of basics such as unique name, DBMS-specific connection parameters, and user credentials.

Collectively, datasource definitions are stored in a datasource catalog. The catalog can be registry-based, local file-based, or network shared file-based, with each method addressing particular advantages. For details, see Specifying a Datasource Catalog Storage Option.

You register datasources using the following techniques:

Manual registration	You can register datasources individually. For details, see Manually Registering or Editing Datasources.
Automatic discovery	You can have search computers or your network for datasources. For details, see Automatically Discovering Datasources.
Import/Export	You can export and import sets of datasource registrations. For details, see Importing and Exporting Datasource Definitions.
Embarcadero DB Team Server	Integrating with DB Team Server lets you work with datasource definitions on a data governance metadata repository. For details, see DB Team Server Support.

As an aid in maintaining your datasource registrations, you can quickly view basic registration details of a datasource. For details, see Viewing Datasource Properties.

Lastly, when a datasource registration becomes obsolete, you can unregister the datasource. For details, see Unregistering Datasources.

Specifying a Datasource Catalog Storage Option

A Datasource Catalog is a collection of defined datasources. On startup, a Datasource Catalog is loaded using one of three methods, specified using the Options editor:

- **Windows Registry datasource catalog** - The datasource catalog is stored and managed locally, in the Windows registry. This user can add and delete datasources and edit the registration details of existing resources. Any changes are retained across startups.

This method is compatible with other Embarcadero products using registry-based datasource definition storage. If more than one of those products is installed on the same machine, they can share the same Datasource Catalog.

- **File based datasource catalog** - The datasource catalog is stored locally and is file-based. Each datasource definition included in the catalog is stored in a single file (`.dsd` file suffix) in the *%APPDATA%***Embarcadero\Data Sources** folder, in a default installation. This user can add and delete datasources and edit the registration details of existing resources. Any changes are retained across startups.

This method is compatible with other Embarcadero products using file-based datasource definition storage. If more than one of those products is installed on the same machine with DBArtisan, all can share the same Datasource Catalog.

- **Network shared datasource catalog** - The datasource catalog is file-based, and obtained from a user-specified location such as a network share. At each startup, the definitions are imported from the network shared file (.etsds file suffix).

Changes such as new datasources, deleted datasources or edited datasource registrations are lost as soon as this user shuts down DBArtisan. This method allows centralized maintenance of the Datasource Catalog and sharing of a single catalog among multiple users across a network.

Note: The definition files used in both the **File based datasource catalog** and **Network shared datasource catalog** methods do not store user ID and password credentials. Credentials, links to definition files in the current user's catalog, and folder structure are stored in the *%APPDATA%***Embarcadero\Data Sources\metadata\dsuri.xml** file.

Note: On individual computers, administrators can force use of the network shared option. For details, see Locking Down Features Using the Registry.

For information on choosing a method and specifying the network location of a Shared Datasource Management definition file, see Datasource Options.

Regardless of the catalog storage method, the Datasource Catalog can be built manually by explicitly registering datasources or using automated methods such as the Discover Datasources feature. For details, see Registering Datasources.

Keep in mind that changes made with **Network shared datasource catalog** in effect, are lost on shutdown.

In order to set up a **Network shared datasource catalog** scheme, you can build a catalog first using the **File based datasource catalog** or **Windows Registry datasource catalog** method. You can subsequently use the **Manage Datasources** facility to export your **Network shared datasource catalog** file to the location that others will use to load the file. For details, see Importing and Exporting Datasource Definitions.

More generally, the **Manage Datasources** facility's import/export functions can be used to exchange datasource definitions between users employing the different storage methods. Even more simply, these functions save time in allowing datasource definitions to be defined once and then shared with multiple users.

Manually Registering or Editing Datasources

Each database instance must be registered. Whether you are registering a new datasource, registering a new datasource based on an existing datasource definition, or editing preexisting connection information, you use the Datasource Registration dialog box.

To register a new datasource or edit an existing datasource

1 Use one of the following actions:

- To register a new datasource, from the **Datasource** menu, select **Register Datasource**.
- To register a new datasource based on an existing definition, select the existing datasource in the Navigator/Explorer and then from the **Datasource** menu, select **Register Like**.
- To edit registration information, on the Navigator/Explorer, right-click a datasource and select **Edit**.

2 Use the following topics as a guide to providing the information required to register or edit the datasource:

- Selecting Database Type/Connectivity Options When Registering Datasources
- Providing Connection Information When Registering Datasources
- Providing Security Parameters When Registering Datasources
- Providing Custom Driver Properties When Registering Datasources
- Specifying Datasource Properties When Registering Datasources
- Specifying Oracle Views Configuration When Registering Datasources
- Specifying Group and Category Details When Registering Datasources

3 When ready, click **Finish** and respond to the prompt to connect immediately.

Note: The **Test Connection** button can be used when creating or editing a datasource to verify that a connection can be established. Keep in mind that the information currently provided in the wizard is used to test the connection and therefore connection information, credentials, and custom JDBC driver properties if appropriate, must be valid.

Selecting Database Type/Connectivity Options When Registering Datasources

The **Datasource Registration** Wizard's **Database Type** panel lets you specify the basic connectivity method for the datasource you are creating.

Prior to registering datasources, you should be familiar with the connectivity options available for each DBMS platform.

- For DBArtisan system requirements and details on DBMS product support, see the **Read Me** at http://docs.embarcadero.com/products/dbartisan/.

Note: The Options Editor lets you indicate preferences, select options, and provide details not available with the Datasource Registration Wizard. For details, see Connection Options.

Note: On individual computers, administrators can restrict the available DBMS platforms available using registry settings. For details, see Locking Down Features Using the Registry.

To use Embarcadero's default drivers requiring dedicated client software installed on the **client machine**

1 From the **Available database types** list, select one of the following:

- **IBM DB2 Universal Database**
- **Microsoft SQL Server**
- **MySQL**
- **Oracle**
- **PostgreSQL Servers**
- **Sybase Adaptive Server**
- **Sybase IQ**
- **Teradata**

If there is currently a DBMS node selected in the Navigator/Explorer, that database type is automatically selected.

Note: If you are connected to an Embarcadero Team Server installation, you can select an entry of the selected database type from the **Related Team Server Data Source** list. This prepopulates the wizard with definitions/settings from the Connect server for that datasource and sets up the new datasource definition to be attached to the Team Server datasource. When an item is selected from the **Related Team Server Data Source** list, clicking **Unrelate** removes the Team Server settings/definitions from the current definition and cancels attachment to the Team Server datasource. When no item is selected from the **Related Team Server Data Source** list, clicking **Relate** uses the (alphabetically) first matching Team Server datasource of the selected database type to populate/attach, and if there are no matching definitions, opens a dialog to let you select a definition from Team Server. For more information on Team Server datasources, see DB Team Server Support.

2 Click **Next**.

To specify a custom JDBC driver

1 From the **Available database types** list, select **Generic JDBC**.

2 Click **Next**.

To specify a generic ODBC driver or to use an ODBC connection

1 From the **Available database types** list, select **Generic JDBC**.

2 Click **Next**.

Providing Connection Information When Registering Datasources

Connection information is provided on a DBMS platform basis. The following topics describe platform-by-platform descriptions of the required and optional connection information:

- Providing DB2 Connection Information When Registering Datasources
- Providing MySQL Connection Information When Registering Datasources
- Providing Oracle Connection Information When Registering Datasources
- Providing PostgreSQL Connection Information When Registering Datasources
- Providing SQL Server Connection Information When Registering Datasources
- Providing Sybase ASE Connection Information When Registering Datasources
- Providing Sybase IQ Connection Information When Registering Datasources
- Providing Teradata Connection Information When Registering Datasources
- Providing Connection Information When Registering a Generic JDBC Datasource
- Providing Connection Information When Registering a Generic ODBC Datasource

Providing DB2 Connection Information When Registering Datasources

The **Datasource Registration** Wizard's **Connection Information** panel lets you provide details required to establish a connection to a DB2 datasource. Available options depend on the **Registration Type** value selected.

The following table describes these settings:

Registration Type	Option/Description
Simple	Host/Instance - The network name of the machine hosting the datasource. Port - This defaults to 50000 but can be overwritten. Database - The name of the database this user should connect to. **Datasource Name** - This is the name of the datasource as it will appear in the Navigator or Explorer.
Use alias from IBM client	**Server** - Select from the names of servers defined in the IBM client. **Datasource Name** - This is the name of the datasource as it will appear in the Navigator or Explorer.
Use Data from the LDAP Server	This option is only enabled if LDAP is enabled and configured on the Options Editor's **LDAP Configuration** tab. For details, see LDAP Configuration Options. **Server** - Select from the names of servers obtained from the LDAP server Host/Instance - The network name of the machine hosting the datasource. Port - This defaults to 50000 but can be overwritten. Database - The name of the database this user should connect to. **Datasource Name** - The datasource as it will appear in the Navigator or Explorer.

Keep the following in mind when registering a DB2 datasource:

- The Options Editor lets you indicate preferences, select options, and provide details not available with the Datasource Registration Wizard. For details, see Connection Options.
- Consult DB2 documentation for detailed information on general LDAP support, supported LDAP servers, and client configuration before trying to register LDAP-specified datasources.. For assistance, see Accessing Third Party Documentation.
- For more general information on registering datasources, see Manually Registering or Editing Datasources.

Providing MySQL Connection Information When Registering Datasources

The **Datasource Registration** Wizard's **Connection Information** lets you provide details required to establish a connection to a MySQL datasource. The following table describes these setting:

Option	Description
Server	Enter the name of the host, for example doctest01.
Port (optional)	The default port is 3306. You do not need to enter this information.
Default Database	You must enter the name of the default database.
Datasource Name	This field is automatically populated with the server name, but you can rename it to whatever you want.

Note: The Options Editor lets you indicate preferences, select options, and provide details not available with the Datasource Registration Wizard. For details, see Connection Options.

For more general information on registering datasources, see Manually Registering or Editing Datasources.

Providing Oracle Connection Information When Registering Datasources

The **Datasource Registration** Wizard's **Connection Information** lets you provide details required to establish a connection to an Oracle datasource. Available options depend on the **Registration Type** value selected.

The following table describes these setting:

Registration Type	Options/Description
Simple	**Host** - Manually enter the name of the host machine. **Port** -The default is 1521, but you can change it to wherever the Oracle listener is set up. **SID/Service Name** -Enter the SID or Service Name to correspond with the option you select. **Instance Name**- The specific name used to identify the Oracle instance (the SGA and the Oracle processes). **Datasource Name** - The field is automatically populated with the host name, but you can change it.
Use Connection Descriptor	**Connect Descriptor** - provide a TNS Connect String, a connect identifier that defines the parameters that need the Oracle Net Service to connect to a database service. **Datasource Name** - The field is automatically populated with the host name, but you can change it.
Use TNS Names alias	You can specify a non-default TNSNAMES.ORA file using the Options editor. For details, see Connections Options - Oracle. Oracle Alias - The dropdown is populated with server names from the TNSNAMES.ORA file. Select an Oracle alias. **Connect Descriptor** - A read-only field that displays the TNS Connect String obtained from the alias definition from the TNSNAMES.ORA file. **Datasource Name** - The field is automatically populated with the host name, but you can change it.
Use Data from the LDAP Server ()	This option is only enabled if LDAP is enabled and properly configured on the Options Editor's **LDAP Configuration** tab. For details, see LDAP Configuration Options. Oracle Alias - The dropdown is populated with the names of servers obtained from the LDAP server specified on the Options Editor's **LDAP Configuration** tab. **Connect Descriptor** - A read-only field that displays the TNS Connect String obtained from the alias definition from the LDAP server. **Datasource Name** - The field is automatically populated with the host name, but you can change it.

Keep the following in mind when registering an Oracle datasource:

- You can register an Oracle ASM instance directly using the Datasource Registration Wizard. No Datasource Navigator/Explorer tree is available when connecting to the ASM instance. The ISQL Editor is available with this type of connection.
- Consult Oracle documentation for detailed information on general LDAP support, supported LDAP servers, and client configuration before trying to register LDAP-specified datasources.. For assistance, see Accessing Third Party Documentation.

- For more general information on registering datasources, see Manually Registering or Editing Datasources.

Providing PostgreSQL Connection Information When Registering Datasources

The **Datasource Registration** Wizard's **Connection Information** lets you provide details required to establish a connection to a PostgreSQL datasource. The following table describes these settings:

Option	Description
Host	The host name for the PostgreSQL Server.
Port	The port that the PostgreSQL server is listening on.
Database	The name for the database to connect to.
Datasource Name	This field automatically populates with the host name. You can change it to whatever you want.

Note: The Options Editor lets you indicate preferences, select options, and provide details not available with the Datasource Registration Wizard. For details, see Connection Options.

For more general information on registering datasources, see Manually Registering or Editing Datasources.

Providing SQL Server Connection Information When Registering Datasources

The **Datasource Registration** Wizard's **Connection Information** lets you provide details required to establish a connection to a SQL Server datasource. The following table describes these setting:

Note: If you are registering a named instance on a server, you are restricted to use of the **Use Network Library Configuration/Alias** method below. Prior to registering the datasource, use the SQL Server Configuration Manager to create an alias to the instance, making it available as an **Alias** selection on the **Connection Information** panel.

Option	Description
Simple	**Protocol** (**TCP/IP** or **Named Pipes**) - You need to select one or the other to register a SQL Server database. **Host** - the host name. **Port** - Depending on your section, you can optionally indicate the port for TCP/IP or the pipe name depending on your means of connection. **Default DB** - Optionally give the default name for the database. **Datasource Name** - The field is automatically populated with the host name, but you can rename to whatever you want.
Use Network Library Configuration	**Use Encryption** - enables multiprotocol encryption. **Alias** - Select an alias **Protocol** (**TCP/IP** or **Named Pipes**) - You need to select one or the other to register a SQL Server database. **Host** - Displays the host name. **Port** - Depending on your section, you can optionally indicate the port for TCP/IP or the pipe name depending on your means of connection. **Default DB** - Optionally give the default name for the database. **Datasource Name** - The field is automatically populated with the host name, but you can rename to whatever you want.

Note: The Options Editor lets you indicate preferences, select options, and provide details not available with the Datasource Registration Wizard. For details, see Connection Options.

For more general information on registering datasources, see Manually Registering or Editing Datasources.

Providing Sybase ASE Connection Information When Registering Datasources

The **Datasource Registration** Wizard's **Connection Information** lets you provide details required to establish a connection to a Sybase ASE datasource. The following table describes these settings:

Registration Type	Option/Description
Simple	**Use SSL Encryption** - If this option is selected, the JDBC connection will be established using SSL encryption **Host** - Manually enter the name of the host. **Port** - There is no default for this optional field. **Default DB** - Optionally give the default name for the database. **Datasource Name** - This field automatically populates with the server name. You can change it to whatever you want.
Use alias information from the SQL.INI FILE	**Sybase Server** - Select a server from the dropdown **Host** - Automatically populated when you select an alias. **Port** - There is no default for this optional field. **Default DB** - Optionally give the default name for the database. **Datasource Name** - This field automatically populates with the server name. You can change it to whatever you want.
Use data from the LDAP server	**Use SSL Encryption** - The connection will be established using SSL encryption. **Sybase Server** - Select a server from the dropdown **Host** - Automatically populated when you select an alias. **Port** - There is no default for this optional field. **Default DB** - Optionally give the default name for the database. **Datasource Name** - This field automatically populates with the server name. You can change it to whatever you want.

Note: The Options Editor lets you indicate preferences, select options, and provide details not available with the Datasource Registration Wizard. For details, see Connection Options.

For more general information on registering datasources, see Manually Registering or Editing Datasources.

Providing Sybase IQ Connection Information When Registering Datasources

The **Datasource Registration** Wizard's **Connection Information** lets you provide details required to establish a connection to a Sybase IQ datasource. The following table describes these settings:

Option	Description
Host	The host name for the Sybase IQ Server.
Port	The port that Sybase IQ is listening on.
Database	The name for the database to connect to.
Datasource Name	This field automatically populates with the host name. You can change it to whatever you want.

Note: The Options Editor lets you indicate preferences, select options, and provide details not available with the Datasource Registration Wizard. For details, see Connection Options.

For more general information on registering datasources, see Manually Registering or Editing Datasources.

Providing Teradata Connection Information When Registering Datasources

The **Datasource Registration** Wizard's **Connection Information** lets you provide details required to establish a connection to a Teradata datasource. The following table describes these settings:

Option	Description
Host	The host name for the Teradata Server.
Port	The port that the Teradata server is listening on.
Database	The name for the database to connect to.
Datasource Name	This field automatically populates with the host name. You can change it to whatever you want.

Note: The Options Editor lets you indicate preferences, select options, and provide details not available with the Datasource Registration Wizard. For details, see Connection Options.

For more general information on registering datasources, see Manually Registering or Editing Datasources.

Providing Connection Information When Registering a Generic JDBC Datasource

The **Datasource Registration** Wizard's **Connection Information** lets you provide basic information details required to establish a connection to a generic JDBC datasource.

Note: For requirements and functionality restrictions for generic JDBC connections, see Generic JDBC and ODBC Support.

The following table describes the settings on this panel:

Option	Description
JDBC driver to use	
Connection URL	If you are defining a datasource using a custom JDBC driver, this field is automatically populated with the value you entered in the **Template Connect URL for new data source** field. For more information. see Providing details for a new custom JDBC driver. Type a valid connection URL or edit the URL template to provide a valid connection URL.
Datasource Name	This is the name that will appear in the Datasource Navigator/Explorer.

Note: In addition to basic connection properties, you can also specify provider-specific JDBC properties. For details, see Providing Custom Driver Properties When Registering Datasources.

3 Take one of the following actions:

a To specify a JDBC driver that you have not used before, click **Manage** to register a new driver. For more information, see Providing details for a new custom JDBC driver.

b To use a custom JDBC driver you have already registered, select that driver from the **JDBC driver to use** dropdown.
== Providing details for a new custom JDBC driver ==

When registering a datasource, if you select a generic JDBC connection, you must specify the installed JDBC driver to be used. If you have not used that driver, you can register that driver while defining a datasource.

Note: Before setting up a custom driver, consult the documentation for that driver to obtain details such as the driver class and connection URL for that driver type.

To specify a new custom JDBC driver

1 Open a datasource registration dialog. For details, see Manually Registering or Editing Datasources.

1 From the **Available database types** list, select **Generic JDBC**.

2 On the **Connection Information** page, click **Manage**. The **JDBC Driver Editor** opens.

3 In the **Name** field, type a name or description for the new driver.

4 Click the **Add** button associated with the **Driver Archives** list and use the **Open** dialog to locate and select the library or archive for the driver.

5 In the **Class** field, type the driver class.

6 In the **Template Connect URL for new data source** field, provide a connection URL template. There are no constraints on the value that you enter. When providing details on the **Datasource Registration** Wizard's **Connection Information** panel, the template will automatically be entered in the **Connection URL** field, letting you provide actual values for any placeholders you provide in the URL template entered here.

Providing Connection Information When Registering a Generic ODBC Datasource

The **Datasource Registration** Wizard's **Connection Information** lets you provide details required to establish a connection to a generic ODBC datasource.

Note: For requirements and functionality restrictions for generic ODBC connections, see Generic JDBC and ODBC Support.

The following table describes the settings on this panel:

Use Connect String	Option/Description
Unselected	**Datasource Name** - Provide the name that will appear in the Datasource Navigator/Explorer for this datasource. **DSN** - Provide the DSN for the datasource.
Selected	**Datasource Name** - Provide the name that will appear in the Datasource Navigator/Explorer for this datasource. **Connect String** - Provide the full connection string for the datasource.

Note: Consult the documentation for the ODBC driver and the datasource you are connecting to for details on DSN and connect string requirements.

Providing Security Parameters When Registering Datasources

The **Datasource Registration** Wizard's **Security Parameters** panel lets you provide user credentials and specify other DBMS-specific security settings. The following table describes these settings and their DBMS availability:

Option	Description
User ID	The User ID that DBArtisan will use to connect to the datasource.
Password	The password associated with the User ID.
Connect As (Oracle)	When relevant, choose the appropriate user/administrator level: NORMAL, SYSASM, SYSDBA, or SYSOPER.
Domain (SQL Server)	For MS SQL only. Identify the domain if the user has restricted access.
Auto connect?	Spares the user from reentering the password every time you connect. On individual computers, auto connect can be disabled by administrators. For details, see Locking Down Features Using the Registry.
Omit User ID and Password (Generic JDBC and Generic ODBC only)	Selected, no user ID or password are passed when connecting to the datasource.
Connect using Windows Authentication or **Connect using OS Authentication** (IBM DB2 for Windows, Unix, and Linux, MySQL, Oracle, SQL Server)	Login to the server is verified using Windows/OS authentication
Connect using Kerberos Authentication (Sybase ASE)	If this option is selected, login to the server is verified using Kerberos authentication. **NOTE**: This is only available of the **Use alias information in the SQL.INI file/Sybase Server** option on the **Connection Information** tab is selected. For details, see Data Editor Options.

For more general information on registering datasources, see Manually Registering or Editing Datasources.

Providing Custom Driver Properties When Registering Datasources

The **Datasource Registration** Wizard's **Custom Driver Properties** panel lets you specify optional, provider-specific connection string keywords or properties beyond those offered on other panels of the wizard.

Note: Consult driver/client documentation for details on supported driver properties. For a listing of supported DBMS drivers and clients, see the **ReadMe.htm** file for this release (**View > Release Notes**).

To add a driver property for this datasource connection

1 Click **New**. An **Add Driver Property dialog** opens.

2 Select a property name from the **Name** dropdown or type the name of the property. The dropdown is automatically populated with the most commonly-used properties for a platform driver. However, all properties (name/value pairs) supported for that driver can be specified.

3 Type the property value in the **Value** field.

4 Optionally, provide a **Description**.

5 Click **OK**.

The dialog closes and the new property's name and value are displayed in the **Property Name**/**Property Value** list.

Use the **Edit** button to initiate editing of a selected property name/value pair. Similarly, use the **Delete** button to remove a selected property name/value pair from the list.

Specifying Datasource Properties When Registering Datasources

The **Datasource Registration** Wizard's **Datasource Properties** panel lets you specify properties used in conjunction with application features. The following table describes these settings and their DBMS availability:

Setting	Description
#Include Search Directories	Enter one or more paths on this datasource, which will be searched for files in conjunction with use of the **#include** directive in the ISQL editor, Procedure Object Editor, or Package Body Object Editor. Separate multiple paths using semicolons. For example: c:\myscripts;c:\Program Files\Scripts For more information on use of the #include directive, see SQL Preprocessing: #define and #include. Note that if there are no entries specified here, the directory specified on the **Directories** tab of the Options editor will be searched. For more information, see Directories Options.
Database device default path (Sybase ASE)	The database device default path is used when creating a new database device.
View in Team Server	If there is an attached datasource on a connected Embarcadero DB Team Server installation, this control opens a browser on the Team Server with the definition displayed.
View Related ER Models	Opens a dialog displaying models related to the current datasource and letting you create and delete relationships.
Edit Related Data Source	Opens a Registration Editor that will let you modify the associated DB Team Server datasource definition.

For more general information on registering datasources, see Manually Registering or Editing Datasources.

Specifying Oracle Views Configuration When Registering Datasources

The **Datasource Registration** Wizard's **Oracle View Config** panel lets you manually configure access to Oracle Data Dictionary DBA or ALL/USER views to conform to the view privileges associated with a specific role.

Note: In order to enable manual configuration of Oracle Data Dictionary views for Oracle datasources and make this panel available, you must select **Use Datasource Configuration for Views** in the Options Editor (**File > Options > General > Oracle**). For more information, see General Options.

Only DBA views with an ALL/USER equivalent are offered on this panel. You can use the following methods to select DBA or ALL/USER type views on a view-by-view basis:

Option	Description
Manual selection using the **Use DBA View** check boxes	Selecting the **Use DBA View** check box associated with an individual view makes the DBA group version of that view available. Deselecting the **Use DBA View** check box associated with an individual view makes the ALL/USER group version of that view available.
Reset to All	Deselects all **Use DBA View** check boxes, making only ALL views accessible.
Reset to DBA	Selects all **Use DBA View** check boxes, making DBA views accessible.
Resolve Automatically (only available when editing a connected datasource)	Queries the datasource and selects the **Use DBA View** check box for each DBA view to which the current user has access.

Changes take effect as soon as you click the **Finish** button.

For more general information on registering datasources, see Manually Registering or Editing Datasources.

Specifying Group and Category Details When Registering Datasources

The **Datasource Registration** Wizard's **Datasource Group** panel lets you specify properties that help you organize your datasources. The following table describes these settings and their DBMS availability:

Select the datasource group folder	Select the folder that is to contain this datasource.
Select a category	If you want to use category-based user interface item color-coding and labeling for this datasource, this dropdown lets you select an existing category to assign this datasource to that category. You can also select **Customize** to manage your categories. For details, see Customizing Datasource Categories. For an introduction to datasource categorization and related tasks, see Categorizing Datasources using Color-coding and Labeling.

For more general information on registering datasources, see Manually Registering or Editing Datasources.

Automatically Discovering Datasources

You can automate registration of datasources by importing datasource definitions from the following locations:

- DBMS configuration files/registry entries
- LDAP Servers
- TOAD product datasource definitions (specifically from TOAD for Oracle, TOAD for Microsoft SQL Server, TOAD for MySQL, TOAD for Sybase (both ASE and IQ))
- An Embarcadero DB Team Server repository. For detailed information, see DB Team Server Support

For example, if you have a multiple DBMS platform license, the Discover Datasources feature finds all unregistered datasources that you are licensed for. If you have an Oracle only license, Discover Datasources finds all unregistered Oracle datasources.

Use the information in Setting up to Automatically Discover Datasources to ensure that you are ready to use the Doscover Datasources feature.

There are two relevant scenarios for this feature.

For details, see the following topics:

- Discovering Datasources on First Application Startup
- Manually Discovering Datasources

In addition to Auto-Discovering your database servers, datasources are automatically placed within a DBMS-specific database group folder in the Navigator/Explorer. For more information, see Categorizing Datasources using Color-coding and Labeling.

For related information, see Connecting to Datasources.

Setting up to Automatically Discover Datasources

Some search targets used by the Discover Datasources feature rely on customer-provided information. The Options Editor lets you provide the following:

- The names and locations of configuration files such as the Oracle TNSNAMES.ORA file and the Sybase ASE SQL.INI file. For details, see Connection Options.
- Identification and authentication details of an LDAP Server that can publish datasources. For details, see LDAP Configuration Options.
- Enabling and configuration of an Embarcadero DB Team Server connection. For details, see DB Team Server Support. In addition, you must be logged in to the Embarcadero DB Team Server. For details, see Setting up and Connecting to DB Team Server.

In addition, there are environment details that can impact this feature. For details, see Troubleshooting Client Setup.

Discovering Datasources on First Application Startup

On first startup, a dialog box displays, giving you the option to Auto-Discover all configured datasources. If you click Yes, the Auto-Discover feature searches the DBMS configuration files on your computer and automatically discovers all the datasources that you are licensed for.

Note: Microsoft SQL Server datasources are registered through a Windows system call to your network. Provide login information (user name and password) the first time you connect to a datasource. For more information, see Disconnecting from a Datasource.

Note: IBM DB2 for Linux, Unix, and Windows databases use ODBC/CLI or DB2 (attach) to connect. Therefore, you need the proper ODBC/CLI Connection established in order for the auto-discover feature to find your IBM DB2 for Linux, Unix, and Windows databases, including registering the DB2 datasource to ODBC as a system datasource. Although your datasources are auto-discovered, provide login information (user name and password) the first time you connect to a datasource.

Manually Discovering Datasources

You can discover datasources residing on your system that are not currently registered datasources. The Discover Datasources operation can retrieve a list of discovered datasources, which includes the name of the server or instance, the type of DBMS, and source (DBMS configuration files/registry entries, LDAP Servers, TOAD datasource definitions, Embarcadero DB Team Server) of all unregistered datasources found on your network or local machine. Once discovered, you have the option to register datasources.

Completing the Discover Datasources Dialog Box

1 On the **Datasource** menu, click **Discover Datasources**.

The **Discover Datasources** dialog box opens and when the search is complete, displays all discovered datasources. The source (**Local alias**, **Network**, **LDAP**, or **Toad**) and status of each is displayed.

2 Select the check box next to any datasources you want to register.

3 Click **Register**.

Importing and Exporting Datasource Definitions

A datasource catalog is loaded on startup. It is a collection of datasource definitions that the user can work against. How the catalog is stored and whether changes to the catalog are temporary or permanent, depends on the datasource storage method used. For an introduction, see Specifying a Datasource Catalog Storage Option.

Note: On individual computers, administrators can restrict the available DBMS platforms using registry settings. For details, see Locking Down Features Using the Registry.

Regardless of the storage method used, you can add datasources to the currently loaded catalog by importing datasource definitions from the following sources:

- **Windows Registr**y - if you previously used the **Windows Registry datasource catalog** storage method but are now using one of the file-based methods
- **File-based datasource catalog** storage method files (**.dsd** file suffix)
- **Network-shared datasource catalog** storage method files (**.etsds** file suffix)

Similarly, you can export the currently loaded datasource catalog definitions to a **file-based datasource catalog** or **network-shared datasource catalog** storage method files.

The ability to export and import datasource definitions provides a number of practical advantages. For example:

- Simple sharing among users - if you have a large number of datasources to register, you can have one user walk through the process of registering each datasource and then export those definitions. Other users can then import the resulting datasource definition files.
- Sharing definitions among users employing different datasource storage methods - datasource definitions or entire catalogs created using the **Windows Registry datasource catalog** or **File based datasource catalog** methods can be exported for use by users employing any of the three methods.

To export one or more datasource definitions

1 On the **Datasource** menu, select **Manage Datasources**.

The **Manage Datasources** dialog box opens.

2 Select a single datasource, multiple datasources, or a default or custom datasource group folder.

Note: Datasource group folder information is not exported. Datasources are exported as individual files with no higher level grouping.

3 Click **Export**. An **Export Data Sources Folder** dialog opens.

4 Take one of the following actions:

- To export datasource definitions to individual files that can be imported on a datasource-by-datasource basis, select **Export Data Sources to Individual Data Source Files** and use the associated control to locate and select a target folder.
- To export datasource definitions to a single file for use by installations employing the Shared Datasource Managed catalog storage method, select **Export Data Sources To a Shared Data Source Management File** and use the associated control to locate and select a target folder and provide a file name. For information on how to designate a user as a shared datasource managed catalog storage method, see Datasource Options.

5 Click **OK**.

Note: User ID and password credentials are not exported to the datasource definition file.

After export, other users can then import the datasource definitions.

To add one or more datasource definitions to a catalog

1 On the **Datasource** menu, click **Manage Datasources**.

The **Manage Datasources** dialog box opens.

2 Select the datasource group folder where the files or files are to be imported.

3 Click **Import**. An **Import Data Sources** dialog opens.

4 Take one of the following actions:

- If you previously used the Windows Registry catalog storage method but are currently using the Locally Managed or Shared Datasource Managed method, select **Import Previously Registered Embarcadero Data Sources From Registry** to load any datasources registered when you were using the Windows Registry catalog method.
- To import datasource definitions from individual datasource definition files (`.dsd` file suffix), select **Import Previously Registered Embarcadero Data Sources From File(s)**, click **Add**, and use the **Select datasource definitions to import** dialog to locate and select the files to be added.
- To import datasource definitions from a single Shared Datasource Managed storage method file (`.etsds` file suffix) file, select **Import Data Sources From a Shared Datasource Management File** and use the associated control to locate and select the target file.

5 Click **OK**.

After importing a datasource definition file, you can either edit the datasource registration to provide security parameters or provide credentials when connecting to the datasource.

Viewing Datasource Properties

You can view the name, type, version, status and mode of a connected datasource. The **Datasource Properties** dialog also lets you view the middleware or connectivity software that is being used to establish a particular datasource connection. You can use this information to troubleshoot connectivity problems, determining vital information such as the server version, connectivity library used, and library version and date.

Note: For information on establishing a datasource connection, see Connecting to Datasources.

To view key properties of a connected datasource

1 On the Datasource Navigator or Datasource Explorer, select a datasource with an established connection. For more information, see Using the Datasource Explorer/Navigator.

2 On the **Datasource** menu, select **Properties.**

The **Datasource Properties** dialog box opens.

Unregistering Datasources

You can unregister datasources when you no longer need them.

Note: Unregistering a datasource does not delete the physical database. It simply removes the datasource definition, and connection information, from the datasource catalog.

1 In the Datasource Explorer or Datasource Navigator, right-click the target datasource and select **Unregister**. For more information, see Using the Datasource Explorer/Navigator.

2 When prompted to confirm, click **Yes**.

Note: The datasource manager is shared across Embarcadero's database management products. When you remove a datasource in any of Embarcadero's database management tools the datasource is removed across all relevant products.

For related information, see Registering Datasources.

Connecting to Datasources

There are two general categories of methods for connecting to a datasource.

Automatic	You can set datasources to automatically connect each time you open the application. You can do this when registering a datasource or as an option when explicitly connecting to a datasource. For details, see Registering Datasources and Manually Registering or Editing Datasources
Explicit	If you do not use the automatic connection option for a datasource, you must explicitly connect to a datasource before you can use that datasource.

The following topics provide instruction on explicit connection and disconnection options:

- Connecting to a Datasource From the Navigator/Explorer
- Connecting to Datasources from an Alphabetical Listing
- Disconnecting from a Datasource

For related information, see Registering Datasources

Connecting to a Datasource From the Navigator/Explorer

The first time you start the application, you are prompted to register your datasources. During this process, you can select the Auto Connect check box, which automatically connects all registered datasource each subsequent time you open the application. For details, see Moving Datasources Between Groups.

If you did not check the Auto Connect box, or if you clicked No when prompted to connect to a database after registering, you must connect manually, each time you want to access that datasource.

If you later want to automatically connect your datasources, you can edit the datasource to make that change.

Tip: To configure your datasource to login automatically, refer to Registering Datasources.

To manually connect to a datasource

1 In the Datasource Explorer or Datasource Navigator, right-click a datasource and select **Connect** from the context menu. For details on the Explorer or Navigator, see Using the Datasource Explorer/Navigator.

A **Login to...** dialog box opens.

2 Type a **Login ID** and **Password**.

3 Optionally, and only if available, provide **Login As**, **Default Schema ID**, and **Default Function Path** values.

4 Select **Auto Connect** to automatically connect to the datasource in the future.

5 Click **OK**.

When a successful connection is made, datasource display expands to show the top level of nodes available on that datasource.

Connecting to Datasources from an Alphabetical Listing

The **Select Datasource** dialog lets you sequentially connect to a set of datasources chosen from a list.

To connect to one or more datasources

1 On the **Datasource** menu, select **Select**. The **Select Datasource** dialog opens.

2 For each datasource that you want to connect to, select a datasource from the **Datasource** list, and then click **Connect** to connect to the selected datasource.

3 If the last datasource you connect to supports multiple databases, you can optionally select a database from the **Database** list, and then click **OK** to have that database selected.

The datasource display expends to show the top level of nodes available on the datasources you connected to. For more information, see Using the Datasource Explorer/Navigator.

Disconnecting from a Datasource

When you disconnect from a server, the application immediately breaks the connection between any open ISQL Windows, the servers, and databases. Although your ISQL Windows are still visible, the connections are no longer valid. If you attempt to execute a script, an attempt to reconnect to a registered datasource, if available, is made.

To disconnect from a datasource

1 In the Datasource Explorer or Datasource Navigator, right-click a datasource and select **Disconnect**. For information on the Explorer and Navigator, see Using the Datasource Explorer/Navigator.

Note: If necessary, a dialog box opens, asking if you want to commit all pending transactions for that connection or to rollback all before disconnecting. You cannot disconnect if there is an uncommitted transaction.

Trouble-shooting Client Setup

If you are using native Embarcadero drivers to connect to a datasource, improperly configured client software can lead to the following symptoms:

- Failure to Auto-Discover accessible datasources.
- Inability to properly register a datasource
- Failure to manually connect to a registered datasource
- Error messages indicating missing client software files

Note: For details on the various ways to register and connect to datasources, see Datasource and Server Management.

Regardless of the DBMS platform, your first step in trouble-shooting client connectivity is to ensure that your client is specified in the **PATH** environment variable.

For DBMS-specific help, see the following topics:

- Trouble-shooting Sybase Connectivity
- Trouble-shooting Oracle Connectivity
- Trouble-shooting DB2 LUW and DB2 z/OS Connectivity
- Trouble-shooting SQL Server Connectivity

Trouble-shooting Sybase Connectivity

A common cause of client connectivity issues against a Sybase DBMS is the client on the workstation not being compatible with DBArtisan or the Sybase environment on your workstation not being configured properly.

Verify that your Sybase environment variables have been set properly on your workstation. If you make any changes, restart DBArtisan

Ensure you at least have the following variables (Environment Variables accessible through the Control Panel):

`SCROOT`

`SYBASE`

`SYBASE_OCS`

`SYBROOT`

`LIB` - ensure that the Sybase LIB directory is listed here

`INCLUDE` - ensure that the Sybase **INCLUDE** directory is listed here

Verify that the **PATH** value has only ONE SET of Sybase directories listed. You can modify this from the same Environment Variable window above in the **PATH** value:

%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\system32\WBEM;C:\PROGRA~1\Serena\vm\win32\bin;C:\PROGRA~1\Serena\vm\common\bin\win32;C:\oracle\ora92\bin;C:\oracle\ora92\jre\1.4.2\bin\client;c:\sybase\OCS-15_0\lib3p;c:\sybase\OCS-15_0\dll;c:\sybase\OCS-15_0\bin;c:\sybase\DataAccess\ADONET\dll;c:\sybase\DataAccess\OLEDB\dll;c:\sybase\DataAccess\ODBC\dll;c:\sybase\Shared\Sybase Central 4.3;c:\sybase\ua\bin;C:\Program Files\Internet Explorer;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\Program

In order to use the Auto-Discover feature, a valid **SQL.ini** file must be located in the following location:

C:\sybase\*version_identifier_directory***\ini**

Where the *version_identifier_directory* is the name of the release-specific directory associated with this purpose, such as **OCS-15_0** or **OCS-12_5**.

Trouble-shooting Oracle Connectivity

An Oracle client must be installed on the workstation. It must be defined/specified in the following locations:

- The **ORACLE_HOME** in the Windows Registry
- The Environment **PATH** value

In addition, in order for the Auto-Discover feature to work, there must be a valid Oracle **tnsnames.ora** and **sqlnet.ora** file in the Oracle client installed. They must be installed in the `%Oracle_home\Network\Admin` directory. As well, the **sqlnet.ora** file must have the **NAMES.DIRECTORY_PATH** value of **TNSNAMES**.

A typical sqlnet.ora file:

```
# sqlnet.ora Network Configuration File:
C:\oracle\product\10.2.0\client_2\network\admin\sqlnet.ora
# Generated by Oracle configuration tools.
# This file is actually generated by netca. But if customers choose to
# install "Software Only", this file wont exist and without the native
# authentication, they will not be able to connect to the database on NT.
SQLNET.AUTHENTICATION_SERVICES= (NTS)
SQLNET.INBOUND_CONNECT_TIMEOUT = 30
SQLNET.SEND_TIMEOUT = 30
SQLNET.RECV_TIMEOUT = 30
TRACE_FILENO_SERVER=3
LOG_FILE_SERVER=svr.log
TRACE_FILE_CLIENT=clientsqlnet.trc
NAMES.DIRECTORY_PATH= (TNSNAMES)
```

Trouble-shooting DB2 LUW and DB2 z/OS Connectivity

The client must be defined in the workstation **PATH** value.

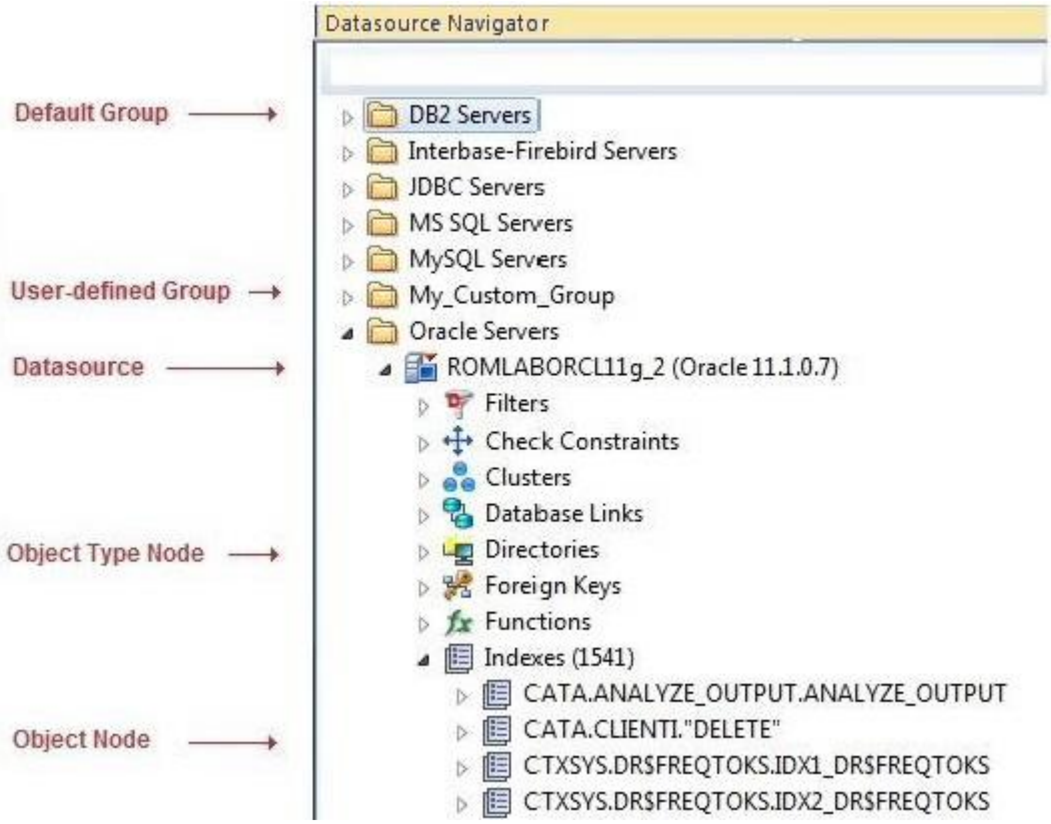
In order to Auto-Discover or manually register DB2 datasources, they must be configured in the DB2 Configuration Assistant. If DB2 datasources are not being detected, have your DBA provide a Configuration Assistant profile.

Trouble-shooting SQL Server Connectivity

In general, SQL Server datasources rarely go undetected. Each database server sends out a ping (via a service) to the network and DBArtisan is usually able to find this. The SQL Server client must be defined in the workstation **PATH** value.

Using the Datasource Explorer/Navigator

The Datasource Navigator provides a visual method for browsing, accessing, and manipulating objects. The Datasource Navigator contains a hierarchy consisting of groups, datasources, object type, and object nodes. Higher level nodes in the hierarchy have an associated Expand/Collapse icon, letting you drill down through the hierarchy to expose lower level nodes and objects.



- Groups organize the datasources available. A default set of groups correspond to the DBMS platforms supported. You can also create custom groups to organize your datasources. For more information, see Categorizing Datasources using Color-coding and Labeling.
- Within groups are the datasources on your network that you have registered. You must register datasources in order to work against them. For more information, see Registering Datasources and Connecting to Datasources.
- With the exceptions of the **Filters** node, nodes below the datasource level correspond to the object types supported on the DBMS platform. For more information on the **Filters** node, see Filtering in the Navigator/Explorer.
- At the lowest level, are the individual database objects on the datasource. For a listing of object types by DBMS platform and detailed support summaries for each type, see Supported Objects.

DBArtisan also provides the Database Explorer providing functions similar to the Navigator. Key differences are as follows:

- It is a two-pane window that lets you use the left pane to drill down to the object level while using the right pane to view details on the currently selected node. For example, if the selected node is a supported object type, the right-pane will show individual objects of that type. For details, see Navigating the Datasource Explorer and Displaying Nodes or Objects (DBArtisan).
- Intermediate level nodes correspond to a default hierarchy of object type categories. For example, the **Security** node for a given platform may contain object types such as users, logins, or roles. For an introduction to the hierarchy of datasource nodes, see Understanding the Explorer Tree Hierarchy (DBArtisan).

For information on opening and closing the Navigator/Explorer, Hiding and Displaying the Datasource Navigator/Explorer

Bookmarks provide another means to quickly navigate to specific Explorer object nodes. For details, see Working with Bookmarks.

You can also initiate object type-specific actions and open object editors from the Navigator/Explorer. For details, see Initiating Object Actions in the Navigator/Explorer.

A number of viewing and organization options are available.They let you optimize the Explorer for your own purposes. These include:

- Basic viewing options that let you toggle between type-based or owner-based display, one-step expansion and collapse of hierarchy elements, hiding or display of system objects, saving of view option settings.
- Creating custom datasource groups to provide a deeper level of organization.
- Categorizing datasources using color-coding and labeling to quickly distinguish between datasources used for different purposes in your organization.
- Filtering the Explorer tree to display only selected nodes and objects. You can filter by name, schema or object type,

For related or more detailed information, see the following topics:

- Customizing the Navigator/Explorer Display.
- Connected/Selected Datasource options
- Filtering columns in Database Object Displays
- Working with Bookmarks
- Working with Supplementary Organizer Tabs

Hiding and Displaying the Datasource Navigator/Explorer

Display of the Datasource Navigator can be toggled on and off.

To Toggle Display of the Navigator On and Off

- Select **View > Datasource Navigator**.

Whether a Datasource Explorer opens on DBArtisan startup is dictated by an Options Editor setting. For details, see Explorer Options.

Otherwise, the Explorer can be opened or closed, as needed.

To open a Datasource Explorer

- Select **View > Activate Explorer**.

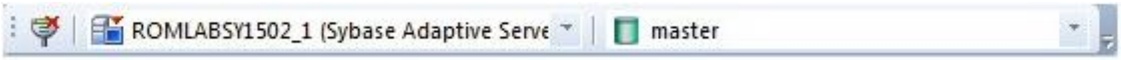
To close the two-pane Datasource Explorer window

- Click the close window button (X) at the top, left.

For context information, see Using the Datasource Explorer/Navigator.

Connected/Selected Datasource options

While you can connect to multiple datasources, at any one time only one connected datasource is the currently selected datasource. The currently selected datasource is displayed in the **Datasource** toolbar. For multiple-database datasources such as SQL Server and Sybase, a currently selected database is also displayed.



Certain features, to varying degrees, depend on the currently selected datasource or selected datasource/database combination. On a fairly trivial level, availability of some features or the initial state of a feature on being invoked is dependent on the selected datasource:

- In DBArtisan, **Utilities** menu feature availability (backup, restore, import, export, for example) depends on the DBMS platform of the selected datasource.
- Some features, Query Builder for example, have user interface controls for selecting a datasource. On opening, these user interface items default to the currently selected datasource.

More strongly dependent on the selected datasource is the SQL Editor. A SQL Editor window not locked to a datasource, will direct all transactions to the currently selected datasource.

Note: For information on how to lock and unlock a SQL Editor window, see Locking a SQL Editor Window to a Datasource

When the Datasource Explorer is active, a SQL Editor opened against a particular datasource will continue to direct transactions to that datasource, until you explicitly change the selected datasource. Whether a SQL Editor or any other window or other user interface element is active, the method of explicitly changing the selected datasource is the same.

To Explicitly Make a Datasource the Selected Datasource

1 Select the datasource from the **Select Datasource** dropdown on the Datasource toolbar. Optionally, on a multi-database DBMS platform, select a database using the **Select From Available Databases** dropdown.

With a Datasource Navigator active, while an unlocked SQL Editor window is open, the selected datasource can be changed explicitly or automatically in response to Explorer navigation or related user interface actions.

This can result in the following issues:

- Active session-specific features such as execution directives and query options being sent to the wrong datasource
- Features such as Semantic Validation returning false errors

How the selected datasource changes depends on the state of the **Link** button appearing at the top of the Explorer pane:

Link Button State	Description
🔗 The Explorer is linked to the **Select Datasource** combo	You can explicitly change the active datasource by selecting a datasource from the **Select Datasource** dropdown. The active datasource changes automatically as datasource-associated windows such as object editors become active in the Workspace or as you select nodes in the Explorer. As a side effect, the associated datasource node in the Explorer tree is selected.
🔗 The Explorer is NOT linked to the **Select Datasource** combo	When linking is inactive, you can only explicitly change the active datasource when an SQL Editor window is open. The active datasource still changes automatically as datasource-associated windows such as object editors become active in the Workspace. And when linking is inactive, the change in the selected datasource has no effect on the selected node in the Explorer tree.

To toggle between the two link states

1 Click the **Link** button.

Navigating the Datasource Explorer and Displaying Nodes or Objects (DBArtisan)

The Datasource Explorer features a hierarchical tree. All higher level nodes have an Expand/Collapse control that lets you drill down the hierarchy or hide hierarchy levels accordingly.

For a description of higher level nodes and the functionality provided for those nodes, see Understanding the Explorer Tree Hierarchy (DBArtisan).

When you select an object on the left-hand pane of the Datasource Explorer, you can view corresponding data in the right pane. This varies by node type:

- When a group is selected in the left-hand pane, the right-hand pane displays datasources and sub-groups contained in that group
- When a datasource is selected in the left-hand pane, the right-hand pane displays a set of configuration or operational parameters for the datasource, if available.
- When a node is selected in the left-hand pane, the right-hand pane display depends on the hierarchy level of the selected node. At the lowest level, when a node corresponding to an object type is selected, the right-hand pane displays all objects of that type as well as key properties of the object. Right-hand pane display for a higher level node, corresponding to a grouping of object types, may be a simple list of the available object types in that category.

Filtering columns in Database Object Displays

In DBArtisan, Datasource Explorer name filtering is available when an object type is displayed in the left pane with objects of the selected type displayed in the right panel. For more information, see Hiding and Displaying the Datasource Navigator/Explorer.

In the Database Navigator, name filtering is available with the Browser feature.

Two methods of filtering are available in the object displays. Both make use of the text boxes appearing below column names in the display.

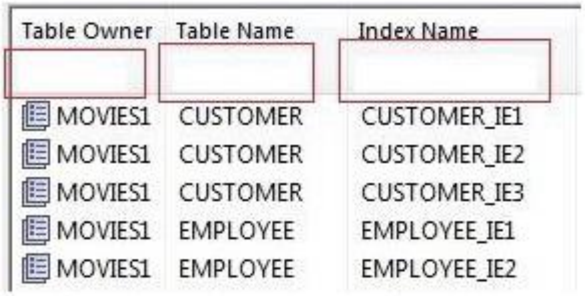


Table Owner	Table Name	Index Name
MOVIES1	CUSTOMER	CUSTOMER_IE1
MOVIES1	CUSTOMER	CUSTOMER_IE2
MOVIES1	CUSTOMER	CUSTOMER_IE3
MOVIES1	EMPLOYEE	EMPLOYEE_IE1
MOVIES1	EMPLOYEE	EMPLOYEE_IE2

To show only rows for which a given column value contains one or more contiguous characters

1 Type those characters into the text box below that column name.

To make use of regular expressions in column filtering

1 Ensure that regular expression usage is enabled. For details, see Explorer Options.

2 Type a valid regular expression search criteria into the text box below that column name. For more information, see Regular Expressions Support.

Note: The **RegEx ON/RegEx OFF** button in the lower right lets you enable or disable regular expression usage on-the-fly.

Working with Bookmarks

Bookmarks provides easy datasource navigation. You can right-click any node of the Explorer tree and add a bookmark. After a bookmark is created, you can simply select it from the **Bookmarks** menu, and be instantly positioned and connected to the resource in the Datasource Explorer tree.

Up to 256 bookmarks can be created.

To create a bookmark

1 On the Navigator/Explorer, right-click the target node, and then select **Add Bookmark**. The **Add Friendly Bookmark Name** dialog opens.

2 Type the Explorer bookmark name.

3 Click **OK**.

Note: You cannot create multiple bookmarks referencing the same node.

To navigate to a bookmarked node

1 Select the Options menu (designated with a down arrow, above the Navigator/Explorer tree), select **Bookmarks**, select a DBMS platform, and then select the bookmarked node.

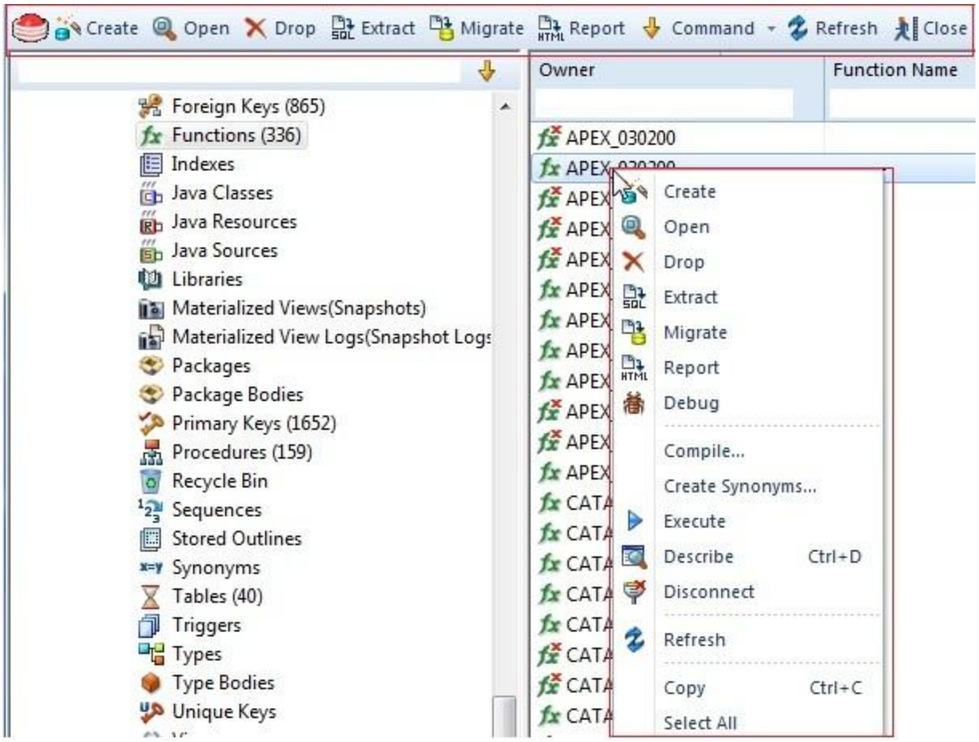
Editing Explorer Bookmarks

1 Select the Options menu (designated with a down arrow, above the Navigator/Explorer tree), and then select **Bookmarks > Bookmark Manager**. the Bookmark Manager dialog opens.

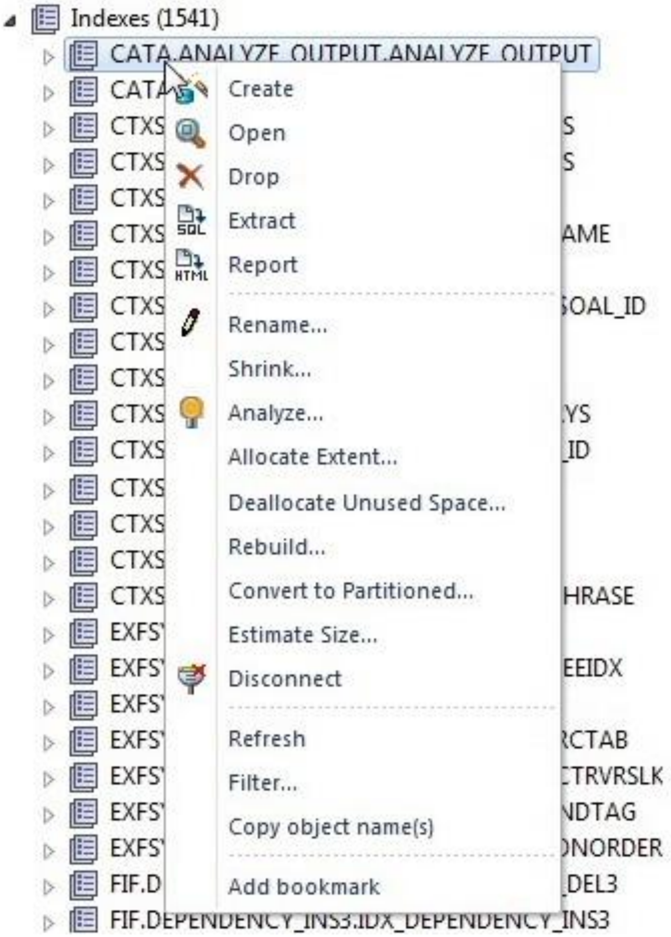
- Rename a bookmark by selecting it and then clicking **Rename**. Then use the **Edit Bookmark Name** dialog to provide a new name.
- Delete a bookmark by selecting it and then clicking **Delete**.

Initiating Object Actions in the Navigator/Explorer

In the Datasource Explorer, when a node corresponding to an object type is selected, objects of that type are displayed in the right-hand pane. Toolbar and right-click options available from the Explorer window apply to information or objects displayed in the Explorer window or to the window itself.



In the Datasource Navigator, right-click options let you initiate object actions against object types or against specific objects.



Right-click Object Actions in the Datasource Explorer

The right-click context menu for each object type lets you open creation wizards or editors or initiate object type specific actions.

For detailed instructions, see the following topics:

- Creating objects
- Modifying objects using editors
- Object actions

For context information, see Hiding and Displaying the Datasource Navigator/Explorer.

Understanding the Explorer Tree Hierarchy (DBArtisan)

The Explorer tree organizes and nests subjects as branches, or nodes. By expanding or collapsing the tree, you can efficiently browse multiple servers and databases and penetrate to the appropriate level of detail.

The Explorer tree nodes give you quick access to instance and server configuration, as well as security and performance functionality. The node names depend on the platform. For example, the **Instance** node in IBM DB2 is very similar to the **Server** node in Sybase ASE.

Tip: For IBM DB2 for Linux, Unix, and Windows, DBArtisan offers enhanced navigation of the **System**, **Instance**, and **Database** nodes of the Explorer. DBArtisan lets you view, manage, and navigate DB2 for Windows/Unix datasources at the instance node level rather than at the database level.

The following table shows the nodes immediately below the datasource level on a platform-by-platform basis:

	DB2 LUW	DB2 z/OS	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
Application Processes Explorer Node (DBArtisan)		✓						
Databases Explorer Node (DBArtisan)		✓	✓			✓	✓	
Filters Explorer Node (DBArtisan)	✓	✓	✓	✓	✓	✓	✓	✓
Instance Explorer Node (DBArtisan)	✓			✓				
Performance Explorer Node (DBArtisan)	✓	✓	✓	✓	✓	✓	✓	✓
Schema Explorer Node (DBArtisan)	✓	✓		✓	✓			✓
Security Explorer Node (DBArtisan)	✓	✓	✓	✓	✓	✓	✓	✓
Server Explorer Node (DBArtisan)			✓		✓	✓	✓	✓
Storage Explorer Node (DBArtisan)	✓	✓		✓	✓	✓	✓	✓
Subsystem Explorer Node (DBArtisan)		✓						

Application Processes Explorer Node (DBArtisan)

The **Application Processes** node of the Explorer includes portions of the mainframe DB2 subsystem that relate to application programming. The **Application Processes** node also includes two components that are industry innovations: a search tool for statements, and a means to expose previously established access paths.

DBArtisan places **Application Processes** as the first level of information under datasources.

Platform availability and hierarchy placement

DB2 LUW	DB2 z/OS	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
	✓						

Explorer node location

DBArtisan places **Application Processes** as the first level of information under the datasource node in the Explorer.

Sub-nodes

Expanding this node provides access to the following nodes:

- Plans
- Packages
- DBRM (DBArtisan)
- Procedures
- Triggers

For a support summary for those object types, see Supported Objects.

Databases Explorer Node (DBArtisan)

The **Databases** node provides access to the set of databases on a datasource.

Platform availability and hierarchy placement

DB2 LUW	DB2 z/OS	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
	✓	✓			✓	✓	

This node is placed as the first level below the datasource level.

Explorer node location

The **Databases** node is placed as the first level of information under the datasource node in the Explorer.

Sub-nodes

When expanded, this node provides access to individual databases, which in turn, can be expanded to display object types supported for that DBMS platform. For a support summary for each object type, see Supported Objects.

Filters Explorer Node (DBArtisan)

The filters node provides access to the resources you create to filter the Explorer by object name, type, or schema. For details, see Filtering in the Navigator/Explorer.

Platform availability and hierarchy placement

DB2 LUW	DB2 z/OS	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
✓	✓	✓	✓	✓	✓	✓	✓

This node is placed as the first level below the datasource level.

Explorer node location

DBArtisan places the **Filters** node as the first level of information under the datasource node in the Explorer.

Sub-nodes

The Filters node contains default Explorer tree filters and any custom filters you create for that datasource or create across that DBMS type. Controls let you enable or disable the default and custom filters. Options for creating and maintaining custom filters are provided.

Instance Explorer Node (DBArtisan)

Platform availability

DB2 LUW	DB2 z/OS	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
✓			✓				

The following topics provide node descriptions for those DBMS for which DBArtisan supports an **Instance** node:

- Instances (Oracle)
- Instances (DB2 LUW)

Instances (Oracle)

Explorer node location

DBArtisan places the **Instance** node as the first level of information under the datasource node in the Explorer.

Sub-nodes

- **ASM** and **ASM > Disk Groups** nodes (only visible if the Datasource has an ASM instance) - for a support summary, see Disk Groups (DBArtisan).
- **Version** node - provides specific version information for the datasource.
- **Configuration** node - lets you view, extract, and modify configuration/initialization parameters. For detailed information, see Working with Configuration/initialization Parameters (DBArtisan).
- **Oracle Job Queues** node - for a support summary, see Job Queues.
- **Scheduler** node - Lets you access object types in support of DBMS_Scheduler. For details, see the following topics.
- Chains
- Jobs (Oracle)
- Programs
- Schedules

Instances (DB2 LUW)

Explorer node location

DBArtisan places the **Instance** node as the first level of information under the datasource node in the Explorer.

Sub-nodes

- **Database Configuration** node - - lets you view and modify configuration/initialization parameters. For detailed information, see Working with Configuration/initialization Parameters (DBArtisan).
- **Datasources** node - lets you connect to specific datasources.

Toolbar or Right-click Actions

Right-clicking the **Instance** node also lets you take the following actions:

- **New UDB Database** - see Databases Wizard (DB2 LUW)
- Quiesce (Instance) and Unquiesce - for more information on objects actions, see Object actions.

Performance Explorer Node (DBArtisan)

This node provides access to a set of DBMS-specific performance-related objects.

Explorer node location

DBArtisan places the **Performance** node as the first level of information under the datasource node in the Explorer.

Sub-nodes and Platform Availability

Expanding this node provides access to the following nodes:

	DB2 LUW	DB2 z/OS	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
Active Threads Node		✓						
Blocking Locks Node				✓		✓	✓	
Bufferpools (DBArtisan)	✓	✓						
Connected Users Node								✓
Database Locks		✓						
Event Monitors (DBArtisan)	✓							
Image Copies (DBArtisan)		✓						
Inactive Threads		✓						
Indoubt Threads		✓						
IQ Status Node								✓
Locks	✓	✓	✓	✓	✓	✓	✓	
Performance Databases Node		✓						
PL/SQL Code Profiling Node				✓				
Postponed Threads Node		✓						
Processes	✓		✓	✓	✓	✓	✓	
Prepared Transactions					✓			
Restricted Tablespaces		✓						
Table Locks Node								✓
Traces		✓						
Utilities		✓						

For a support summary for the **Event Monitors**, **Image Copies**, and **Buffer Pools** object types, see Supported Objects.

The following topics provide details on all sub-nodes except the **Event Monitors**, **Image Copies**, and **Buffer Pools** nodes.

Note: For DB2 z/OS, performance is a result of the DB2 API calls. The DB2 API calls are subject to change depending on the version and DB2 SDK call interface of the DB2 server. Performances do not appear in the system catalog tables.

Active Threads Node

Active threads are the processes running on your database. When you select the Active Thread node on the Explorer, DBArtisan displays the active threads that are currently running on the IBM DB2 for z/OS and OS/390 database. To utilize all Database Monitor functionality, on the **Utilities** menu, select **Database Monitor**.

Platform availability and Explorer node location

DB2 z/OS	Performance > Active Threads

Toolbar or Right-click Actions

The following object-specific action is available for this object type:

- Cancel Thread (DBArtisan)

For a complete listing of object actions and detailed instructions, see Object actions.

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

Blocking Locks Node

The Blocking Locks node on the Explorer displays only the locks that are blocked. To utilize all Database Monitor functionality, on the **Utilities** menu, select **Database Monitor**.

Platform availability and Explorer node location

ORCL	Performance > Blocking locks
SQL SVR	Performance > Blocking Locks
SYB ASE	Performance > Blocking Locks

Toolbar or Right-click actions

The following object-specific action is available for this object type:

- Kill (DBArtisan)

For a complete listing of object actions and detailed instructions, see Object actions.

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

Connected Users Node

This node displays `sp_iqconnection` procedure output including

- User details (ConnHandle, Name, Userid)

- Connection details (ReqType, LastReqTime, ConnCreateTime, CommLink, NodeAddr, DBremote)
- Command details (IQCmdType, LastIQCmdTime)
- Cursor details (IQCursors, LowestIQCursorState)
- Thread details (IQthreads)
- Transaction details (TxnID, LastIdle)
- Storage space details (TempTableSpaceKB, TempWorkingSpaceKB)
- Message identifier details (IQConnID)
- Internal counter details (satoiq_count, iqtosa_count)

Platform availability and Explorer node location

SYB IQ Database Monitor > Connected Users

Toolbar or Right-click actions

The following object-specific action is available for this object type:

- Kill (DBArtisan)

For a complete listing of object actions and detailed instructions, see Object actions.

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

Database Locks

The **Database Locks** node lists `-DISPLAY DATABASE(*) LIMIT(*) LOCKS` output, showing status codes, connection ID, correlation ID, and lock information (**LockInfo** field) for all programs holding locks on databases or spaces.

Note:LockInfo consists of lock status (**H**, **W**, or **R**), a lock state (**IS**, **IX**, **S**, **U**, **SIX**, **X**, or **NSU**), lock type (**S**, **T**, **C**, **R**, **W**, **P**, **PP**) and lock duration (**A**, **C**, **H**, **M**, **P**, or **I**). Similarly, status codes display standard `-display` command output (**ARBDP**, **AREO***, **ACHKP**, **AREST**, **AUXW**, **CHKP**, **COPY**, **DBETE**, **GRECP**, **ICOPY**, **LPL**, **LSTOP**, **PSRBD**, **RBDP**, R**BDP***, **RECP**, **REFP**, **RELDP**, **REORP**, **RESTP**, **RO**, **RW**, **STOP**, **STOPE**, **STOPP**, **UT**, **UTRO**, **UTRW**, **UTUT**, and **WEPR**). For detailed descriptions of these items, connection IDs and correlation IDs, consult DB2 for z/OS documentation. For more information, see Accessing Third Party Documentation.

Platform availability and Explorer node location

DB2 z/OS**V8^** Performance > Database Locks

Toolbar or Right-click actions

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

Related Information

- Performance Databases Node
- Restricted Tablespaces

Inactive Threads

Inactive threads are processes that are idle. When you select the Inactive Thread node on the Explorer, DBArtisan displays the inactive threads that are currently running on the IBM DB2 for z/OS and OS/390 database. To utilize all Database Monitor functionality, on the **Utilities** menu, click **Database Monitor**.

Platform availability and Explorer node location

DB2 z/OS Performance > Inactive Threads

Toolbar or Right-click actions

The following object-specific action is available for this object type:

- Cancel Thread (DBArtisan)

For a complete listing of object actions and detailed instructions, see Object actions.

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

Indoubt Threads

Indoubt threads are processes waiting for resources to become active. When you select the Indoubt Thread node on the Explorer, DBArtisan displays the indoubt threads that are currently running on the IBM DB2 for z/OS and OS/390 database. To utilize all Database Monitor functionality, on the **Utilities** menu, select **Database Monitor**.

Platform availability and Explorer node location

DB2 z/OS Performance > Indoubt Threads

Toolbar or Right-click actions

The following object-specific actions are available for this object type:

- Recover Abort (DBArtisan)
- Recover Commit (DBArtisan)

For a complete listing of object actions and detailed instructions, see Object actions.

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

IQ Status Node

This node displays `sp_iqstatus` procedure output, including:

Version	Time Now	Build Time	File Format
Server mode	Catalog Format	Stored Procedure Revision	Page Size
Number of Main DB Files	Main Store Out Of Space	Number of Temp DB Files	Temp Store Out Of Space
DB Blocks...	Temp Blocks...	Create Time	Update Time
Main IQ Buffers	Temporary IQ Buffers	Main IQ Blocks Used	Temporary IQ Blocks Used
Main Reserved Blocks Available	Temporary Reserved Blocks Available	IQ Dynamic Memory	Main IQ Buffers
Temporary IQ Buffers	Main IQ I/O	Temporary IQ I/O	Other Versions
Active Txn Version	Last Full Backup ID	Last Full Backup Time:	Last Backup ID
Last Backup Type	Last Backup Time	DB Updated	Blocks in next ISF Backup
Blocks in next ISI Backup	DB File Encryption Status		

Platform availability and Explorer node location

SYB IQ Database Monitor > IQ Status

Locks

The **Locks** node on the Explorer displays the lock objects held on the table and row IDs for your database. DBArtisan lets you view all locks associated with a database.

Platform availability and Explorer node location

DB2 LUW	Performance > Locks
DB2 z/OS	Performance > Locks
MySQL	Performance > Locks
ORCL	Performance > Locks
SQL SVR	Performance > Locks
SYB ASE	Performance > Locks

Toolbar or Right-click actions

The following object-specific actions are available for this object type:

	DB2 LUW	DB2 z/OS	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
Kill (DBArtisan)			✓	✓		✓	✓	

For a complete listing of object actions and detailed instructions, see Object actions.

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

Performance Databases Node

The **Performance Databases** node lists `-DISPLAY DATABASE(*) LIMIT(*)` output, showing status codes for all tablespaces and indexes on the datasource.

Platform availability and Explorer node location

DB2 z/OS**v8^** Performance > Performance Databases

Toolbar or Right-click actions

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

Related Information

- Restricted Tablespaces
- Database Locks

PL/SQL Code Profiling Node

This node provides access to resources created using the Oracle SQP Profiler. For detailed information on this DBArtisan feature, see PL/SQL Profiler.

Platform availability and Explorer node location

ORCL Performance > PL/SQL Profiling

Toolbar or Right-click actions

The following object-specific actions are available for this object type:

- Start
- Stop (PL/SQL Profiler)
- Flush
- Clear Profile Table
- Run Summary

- Run Detail
- Unit Summary
- Run Detail

For a complete listing of object actions and detailed instructions, see Object actions.

Postponed Threads Node

Postponed threads are units of work for which back-out processing is currently postponed.

Platform availability and Explorer node location

DB2 z/OS Performance > Postponed Threads

Toolbar or Right-click actions

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

Prepared Transactions Node

These are transactions that have been prepared for two-phase commit. The following pg_prepared_xacts table columns are displayed for each transaction: transaction ID, global ID, date and time that the PREPARE TRANSACTION statement was executed, and the owner and database associated with the transaction.

Platform availability and Explorer node location

PSTGRS Performance > Prepared Transactions

Processes

The **Processes** node on the Explorer displays processes that are currently running on your DBMS.

Platform availability and Explorer node location

DB2 LUW	Performance > Processes
MySQL	Performance > Processes
ORCL	Performance > Processes
SQL SVR	Performance > Processes
SYB ASE	Performance > Processes

Toolbar or Right-click actions

The following object-specific action is available for this object type:

	DB2 LUW	DB2 z/OS	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
Kill (DBArtisan)	✓		✓	✓		✓	✓	

For a complete listing of object actions and detailed instructions, see Object actions.

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

Restricted Tablespaces

The **Restricted Tablespaces** node lists `-DISPLAY DATABASE(*) SPACENAM(*) RESTRICT` output, showing status codes for all tablespaces in an exception state

Platform availability and Explorer node location

DB2 z/OS**V8^**	Performance > Restricted Tablespaces

Toolbar or Right-click actions

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

Related Information

- Performance Databases Node
- Database Locks

Table Locks Node

This node displays `sp_iqlocks` procedure output. The following table shows the columns displayed, by Sybase IQ version, and where appropriate, shows additional details on values.

Version 15 and later	Versions prior to 15	Notes on values
`conn_name`		
`conn_id`	`connection`	
`user_id`	user_id	
`table_type`		**BASE**, **GLBTMP**, or **MVIEW**.
`creator`		
`table_name`	`table_name`	
`index_id`		
`lock_class`	`Lock_type`	**S** (share), **SW** (share and write), **EW** (exclusive and write), **E** (exclusive), **P** (phantom), **A** (antiphantom), or **W** (write) as well as the qualifiers **T**, *****, and *nnn*.
`lock_duration`		(**Transaction**, **Position**, or **Connection**
`lock_type`	`lock_name`	
`row_identifier`		

Platform availability and Explorer node location

SYB IQ Database Monitor > Table Locks

Toolbar or Right-click actions

The following object-specific action is available for this object type:

- Kill (DBArtisan)

For a complete listing of object actions and detailed instructions, see Object actions.

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

Traces

The Traces node displays a list of active traces. Traces let you monitor and collect DB2 for OS/390 monitoring, auditing, performance, accounting, statistics, and serviceability (global) data.

Note: This information is read-only.

Platform availability and Explorer node location

DB2 z/OS Performance > Traces

Toolbar or Right-click actions

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

Utilities

The Utilities node displays wizards that are typically run from the command line and do not apply to a particular type of schema object. Some of these wizards are also available via the command menus for objects. The wizards include:

- Check Data
- Check Index
- Copy
- Copy to Copy
- Load
- Merge Copy
- Modify Recovery
- Modify Statistics
- Quiesce Tablespace
- Rebuild Index
- Recover
- Reorg Index
- Reorg Tablespace
- Runstats Index
- Runstats Tablespace
- Unload
- Maintain Template

Tip: To run a wizard, select the command in the right pane of the application and click the **Execute** button on the Explorer toolbar.

Platform availability and Explorer node location

DB2 z/OS Performance > Utilities

Toolbar or Right-click actions

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

Schema Explorer Node (DBArtisan)

The Schema node is a container for object types supported by DBArtisan.

Platform availability

DB2 LUW	DB2 z/OS	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
✓	✓		✓	✓			✓

Explorer node location

DBArtisan places **Schema** as the first level of information under the datasource node in the Explorer.

Sub-nodes

When expanded, this node provides access to nodes corresponding to the object type nodes supported for the DBMS platform. For a support summary for each object type, see Supported Objects.

Toolbar or Right-click actions

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

Security Explorer Node (DBArtisan)

The **Security** node gives you quick access to related security functions and lets you administer and control security on your database from a single vantage point. Rather than having to sift through a large number of objects, you can simply create and alter security settings to be applied across the server from this top-level.

Explorer node location

DBArtisan places **Security** as the first level of information under the datasource node in the Explorer.

Sub-nodes

	DB2 LUW	DB2 z/OS	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
Groups	✓							✓
Logins						✓	✓	
Linked Servers (DBArtisan)						✓		
Profiles				✓				
Remote Servers (DBArtisan)						✓	✓	✓
Roles				✓	✓	✓	✓	
Users	✓	✓	✓	✓				✓

For a support summary for those object types, see Supported Objects.

Toolbar or Right-click actions

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

Server Explorer Node (DBArtisan)

The **Server** node provides access to server-centric information or resources.

Explorer node location

DBArtisan places **Server** as the first level of information under the datasource node in the Explorer.

Sub-nodes and Platform Availability

Expanding this node provides access to the following nodes:

	MySQL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
Version Explorer Node	✓	✓	✓	✓	✓
Databases Explorer Node	✓	✓	✓	✓	✓
Configuration, **Engine Configuration**, or **Database Configuration**. For details, see Working with Configuration/initialization Parameters (DBArtisan).	✓		✓	✓	✓
Scheduling	✓				
SQL Server Agent			✓		
Server Triggers. For a support summary for this object type, see Server Triggers (DBArtisan)			✓		
Error Logs. For a support summary for this object type, see Error Logs (DBArtisan)			✓		
Logs Explorer Node		✓			

Toolbar or Right-click actions

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

Version Explorer Node

DBArtisan places **Version** under the **Server** node of the Explorer. When you select **Version**, DBArtisan shows version information about your datasource in the right pane of the Explorer window.

Platform availability and Explorer node location

MySQL	Server> Version
PSTGRS *	Server> Version
SQL SVR	Server> Version
SYB ASE	Server> Version
SYB IQ	Server> Version

Toolbar or Right-click actions

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

SQL Server Agent

DBArtisan assists you in using the Microsoft SQL Server Agent service to automate administration of a Microsoft SQL server. Microsoft SQL Server Agent manages, and schedules, allowing programmed responses to server events or the execution of scheduled administrative tasks. Automating recurring administrative tasks and responses to server events can save considerable time for other administrative tasks.

Platform availability and Explorer node location

SQL SVR Server> SQL Server Agent

Sub-nodes

Expanding this node provides access to the following nodes:

- Alerts (DBArtisan)
- Jobs (Oracle)
- Logs (DBArtisan - SQL Server Agent)
- Operators (DBArtisan)

For a support summary for those object types, see Supported Objects.

Note: Before using SQL Agent elements from DBArtisan, consult the Microsoft SQL Server documentation for a detailed understanding of the prerequisites, operational requirements, and functionality provided by that service. For more information, see Accessing Third Party Documentation.

Logs Explorer Node

On PostgreSQL datasources, the Logs node provides access to logging output files.

Platform availability and Explorer node location

PSTGRS * Server> Logs

Toolbar or Right-click actions

For details on actions available from this node, see Rotate Log (DBArtisan).

Storage Explorer Node (DBArtisan)

Object types corresponding to storage mechanisms are stored under the **Storage** Explorer node.

Platform availability

DB2 LUW	DB2 z/OS	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
✓	✓	✓	✓	✓	✓	✓	✓

Sub-nodes

The following table lists the sub-nodes available, DBMS-by-DBMS, when the **Storage** node is expanded:

	DB2 LUW	DB2 z/OS	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
Bufferpools (DBArtisan)	✓	✓					
Control Files (DBArtisan)			✓				
Directories			✓				
Data Caches (DBArtisan)						✓	
Database Devices (DBArtisan)						✓	
Dump and Backup Devices (DBArtisan)					✓	✓	
Nodegroups (DBArtisan)	✓						
Redo Log Groups (DBArtisan)			✓				
Rollback Segments (DBArtisan)			✓				
Stogroups (DBArtisan)		✓					
Tablespaces	✓	✓	✓	✓			

For a support summary for those object types, see Supported Objects.

Subsystem Explorer Node (DBArtisan)

The Subsystem node displays detailed information about the DB2 subsystem.

Platform availability

DB2 LUW	DB2 z/OS	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
	✓						

Explorer node location

DBArtisan places the **Subsystem** node as the first level of information under the datasource node in the Explorer.

Sub-nodes

Expanding this node provides access to the following nodes:

- Connections
- DDF
- Parameters

Connections

Connections displays the current connections to the subsystem.

Note: The information is read-only.

Platform availability and Explorer node location

DB2 z/OS Subsystem > Connections

Toolbar or Right-click actions

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

DDF

DDF (distributed data facility) displays the information regarding the status and configuration of the DDF, a set of DB2 for OS/390 components through which DB2 for OS/390 communicates with another RDBMS.

Note: The information is read-only.

Platform availability and Explorer node location

DB2 z/OS Subsystem > DDF

Toolbar or Right-click actions

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

Parameters

Parameters displays the DB2 subsystem parameters.

Note: The information is read-only.

Platform availability and Explorer node location

DB2 z/OS Subsystem > Parameters

Toolbar or Right-click actions

For details on actions available from the Explorer, see Initiating Object Actions in the Navigator/Explorer.

Working with Supplementary Organizer Tabs

In addition to the commonly used **Explorer** tab, the Organizer window offers the following tabs:

- **Project** - when working with the Project Management feature, this tab provides a visual method for browsing, accessing, and manipulating your projects. For more information on working with projects, see Project Management.
- **VC Files** - when working with the Version Control feature, this tab provides access to version control files. For more information on working with projects, see Version Control.
- **Favorites** - lets you designate and access favorite scripts. This tab lets you view, navigate, save, recall, and execute scripts. Sample Favorite Scripts are installed for Microsoft SQL Server, Oracle, and Sybase Adaptive Server. For more information, see Using the Script Library.

Using the Script Library

The Script Library provides a drag-and-drop library interface of all supported DBMS syntax, SQL syntax, built-in functions, optimizer hints, and SQL-conditional syntax. Additionally, it provides the ability to create custom folders to store commonly-used code for quick and efficient access or execution, as needed.

To open the Script Library:

1 Select **View > Script Library**. The **Script Library** tab opens.

See the following topics for more information:

- Basic File and Folder Operations in the Script Library
- Execution Options in the Script Library
- Viewing and Setting File Properties in the Script Library

Basic File and Folder Operations in the Script Library

By default, the Script library contains a set of folders corresponding to supported DBMS platforms. These high level folders are preloaded with platform-specific subfolders, **System Diagnostics** and **Schema** for example, as well as files corresponding to DDL for object types supported on that platform.

The following basic options are provided.

- **Creating a subfolder** - right-click on a folder, select **New Folder**, and immediately provide a non-default name for the new folder.
- **Sorting the files in a folder to alphabetical order** - right-click on a file or folder and select **Sort**.
- **Renaming a file or folder** - right-click on a file or folder, select **Rename**, and provide a new name for the file or folder.
- **Deleting a file or folder** - right-click on a file or folder, select **Delete**, and select **Yes** when prompted to confirm.
- **Adding a SQL, JScript, or VBScript file to a folder** - right-click on a folder, select **New Item**, and use the controls on the **Script Library Item Properties** dialog to specify the file type, locate and select the file, and provide a hot key keystroke sequence to open the file.

Execution Options in the Script Library

The following execution are offered:

- **Opening a SQL, JScript, or VBScript file in the relevant editor** - right-click the file and select **Open**.
- **Executing a file** - right-click the file and select **Execute**, and then respond to any prompts regarding datasource or file type.

- **Opening a file in the Script Execution Facility** - right-click the file and select **Script Execution Facility**. For more information, see Script/File Execution Facilities.

Viewing and Setting File Properties in the Script Library

The following options are available:

- **Viewing the file/folder information, file type, and hot key sequence for an item** - right-click the file and select **Properties**.

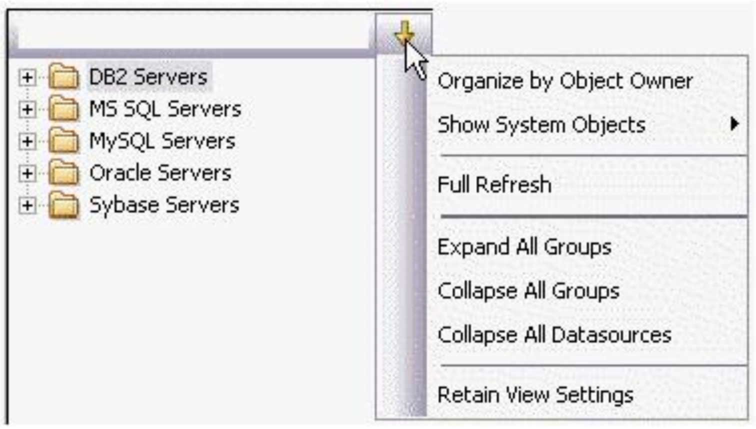
Customizing the Navigator/Explorer Display

DBArtisan offers flexibility in the elements displayed and in how elements are displayed in the Datasource Explorer and Datasource Navigator.

- Basic Viewing Options in the Navigator/Explorer
- Categorizing Datasources using Color-coding and Labeling
- Filtering in the Navigator/Explorer
- Working with Datasource Groups

Basic Viewing Options in the Navigator/Explorer

The dropdown menu, available above the Datasource Navigator/Explorer tree, provides basic organization and display options.



To change basic organization and viewing options

1 From the Options dropdown menu, select an option. Use the following table as a guide to your selection.

Option	Description
Organize by Object Owner or **Organize by Object Type**	Toggling these options has the following effect: **Owner -** the Navigator/Explorer displays objects for each user, categorized by object type. This display mode is most efficient if you are working with databases containing a high number of objects. **Type -** the Navigator/Explorer displays all objects, categorized by object type, for all users in the same list. This display mode speeds or slows performance in databases that contain many objects.
Show System Objects and **Show Only My Objects**	Lets you control display of system objects and objects that you own: NEVER, ALWAYS, or USE DATASOURCE FILTER. For information on filter options, see Simple, on-the-fly, Character-based Tree Filtering.
Full Refresh	Select to refresh.
Expand All Groups	Select to expand all groups.
Collapse All Groups	Select to collapse all groups.
Collapse All Datasources	Select to collapse all datasources.
Retain View Settings	Select to retain the current state of the Explorer so that the it opens the same way the next time you start DBArtisan.
Bookmarks	Provides access to bookmark resources.For more information, see Active Threads Node.
Bookmarks style="display: none;"	Provides access to bookmark resources.

Tip: You can also set these options on the Explorer Options tab of the Options editor.

Working with Datasource Groups

Datasource groups help you to organize the datasources in your enterprise. Datasource Groups behave as folders in the Windows Explorer, allowing you to group related datasources together. Anywhere that datasources are presented in a hierarchical tree format, datasource group folders expand to display one or more contained datasources.

By default, the Datasource Navigator and Datasource Explorer have one group per supported DBMS platform.

By default, the Datasource Navigator has one group per supported DBMS platform.

You can create custom groups or subgroups. This level of organization is useful if you want to group large numbers of registered datasources or have another need to organize datasources. For basic database group maintenance tasks, see the following topics:

- Removing, Adding, and Renaming Datasource Groups.
- Changing the Level of a Datasource Group in the Hierarchy.

When you manually register a datasource, you have the option of changing the default group to which the datasource will automatically be assigned. Subsequently, you can move a datasource from one folder to another. For details, see Moving Datasources Between Groups.

Moving Datasources Between Groups

There are two ways to change datasource groups:

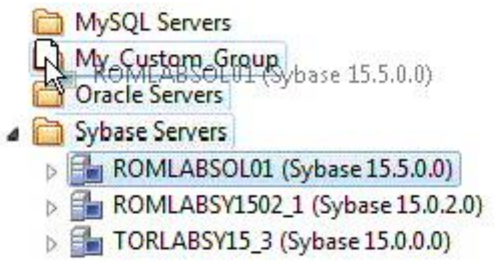
- Dragging the datasource between groups
- Invoking the **Change Group** dialog box.

Note: You should always disconnect from a datasource before changing datasource groups.

To move a datasource between groups by dragging and dropping

1 If necessary, expand groups until both the datasource and the target datasource group are visible.

2 Drag the datasource over the target folder and drop it there.


MySQL Servers
My Custom Group
ROMLABSOL01 (Sybase 15.5.0.0)
Oracle Servers
Sybase Servers
ROMLABSOL01 (Sybase 15.5.0.0)
ROMLABSY1502_1 (Sybase 15.0.2.0)
TORLABSY15_3 (Sybase 15.0.0.0)

To move a datasource using the Change Datasource Group dialog box

1 If necessary, expand groups until the datasource you want to move is visible.

2 Right-click the datasource you want to move, and select **Change Group**.

The **Change Datasource Group** dialog box opens.

3 Use the S**elect...** tree to locate and select the target group.

4 Click **OK**.

For more general information on datasource groups and associated tasks, see Categorizing Datasources using Color-coding and Labeling.

Removing, Adding, and Renaming Datasource Groups

A set of datasource group maintenance tasks are available. You can remove database groups that you no longer need to access, or that have otherwise become obsolete. Keep the following in mind when removing datasource groups:

- If a datasource group contains datasources or other datasource groups, those items are deleted when you delete the containing group. You should move any contained datasources or groups before preceding. For details on moving datasources, see Moving Datasources Between Groups.
- The default datasource groups correspond to the DBMS platforms supported. If you delete a default group, it will be recreated if you subsequently register a datasource for the associated platform.

To remove a datasource group

1 If necessary, expand datasource groups in the Explorer until the datasource you want to remove is visible.

2 Right-click the datasource group, and select **Remove**. You are prompted to verify.

3 Click **Yes**.

Similarly, you can define datasource groups to organize the datasources in your enterprise.

To create a new datasource group

1 If creating a folder within an existing folder, expand datasource groups in the Explorer until the target group is visible.

2 On the Explorer tree take one of the following actions:

- To create a new folder at the top level, right-click in empty space and select **New** from the context menu.
- To create a folder within an existing folder, right-click the target folder and select **New** from the context menu.

The **New Datasource Group** dialog opens.

3 In the **Datasource Group Name** box, type the name of the new datasource group.

4 Click **OK**.

Lastly, you can change the name of an existing datasource group.

To rename a datasource group

1 If necessary, expand datasource groups in the Explorer until the datasource you want to rename is visible.

2 On the Explorer, right-click the datasource group folder, and select **Rename**. The **Rename Datasource Group** dialog opens.

3 In the **Datasource Group Name** box, type the new name of the datasource group.

4 Click **OK**.

For related information, see the following topics:

- See Changing the Level of a Datasource Group in the Hierarchy for information on changing hierarchy levels.
- See Categorizing Datasources using Color-coding and Labeling for more general information on working with datasource groups.

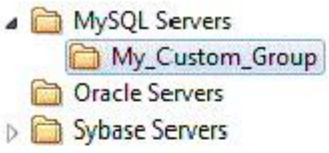
Changing the Level of a Datasource Group in the Hierarchy

You can reorganize the datasource group tree by moving individual groups up and down the hierarchy.

To move a datasource group to the top level of the hierarchy

1 If necessary, expand groups until the datasource group you want to move is visible.

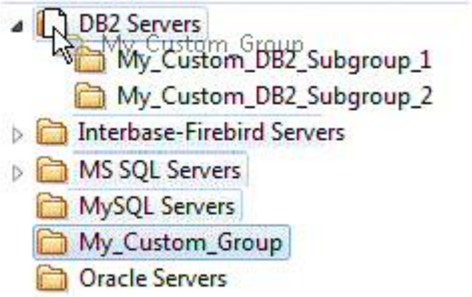
2 Drag the target group to open space in the Explorer pane.



To make a datasource group a subgroup within another group

1 If necessary, expand groups until the datasource group you want to move and the target group are visible.

2 Drag the target group and drop it on the datasource group that is to be the new parent.



For related information, see the following topics:

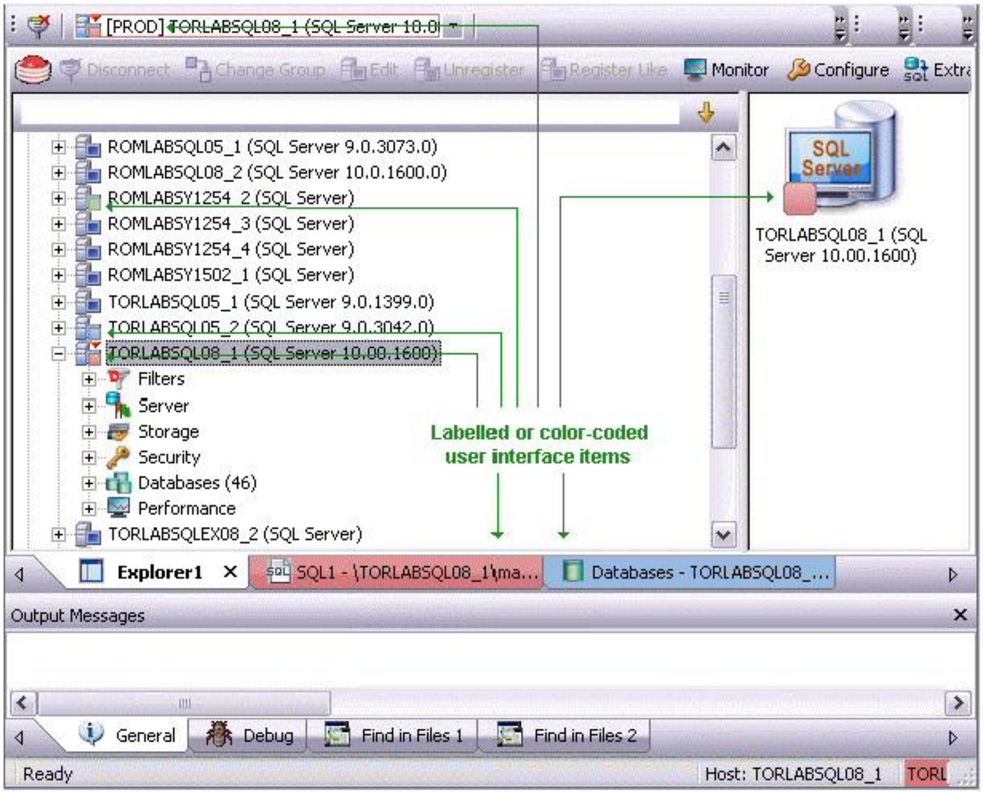
- See Removing, Adding, and Renaming Datasource Groups for information on basic datasource tree tasks.
- See Categorizing Datasources using Color-coding and Labeling for more general information on working with datasource groups.

Categorizing Datasources using Color-coding and Labeling

You can set up a means to visually distinguish between datasources used for different purposes in your organization. You can assign a category, such as development or production, when creating or editing a datasource. This color-codes or labels particular user interface elements associated with that datasource.

When you assign a category to a datasource, the Navigator/Explorer icons for the datasource show a distinctive color-coded category indicator. Optionally, you can customize your categorization scheme to include the following user interface elements:

- A short name label included in the Datasource toolbar combo/dropdown when a categorized datasource is selected
- Color-coded tabs on windows associated with actions against a categorized datasource, editor or ISQL windows for example
- Color-coding of the status bar at the bottom of the main window


[PROD] TORLABSQL08_1 (SQL Server 10.0
Disconnect
Change Group
Edit
Unregister
Register Like
Monitor
Configure
Extra
ROMLABSQL05_1 (SQL Server 9.0.3073.0)
ROMLABSQL08_2 (SQL Server 10.0.1600.0)
ROMLABSY1254_2 (SQL Server)
ROMLABSY1254_3 (SQL Server)
ROMLABSY1254_4 (SQL Server)
ROMLABSY1502_1 (SQL Server)
TORLABSQL05_1 (SQL Server 9.0.1399.0)
TORLABSQL05_2 (SQL Server 9.0.3042.0)
TORLABSQL08_1 (SQL Server 10.00.1600)
Filters
Server
Storage
Security
Databases (46)
Performance
TORLABSQLEX08_2 (SQL Server)
SQL Server
TORLABSQL08_1 (SQL Server 10.00.1600)
Labelled or color-coded user interface items
Explorer1
SQL1 - \TORLABSQL08_1\ma...
Databases - TORLABSQL08_...
Output Messages
General
Debug
Find in Files 1
Find in Files 2
Ready
Host: TORLABSQL08_1
TORL

The default categories [and short name labels] and the associated colors are as follows:

Default Category	Short Name	Label Color
Production	PROD	🟥
Development	DEV	🟦
QA	QA	🟧
Test	TEST	🟩

In addition to the defaults, you can create custom category/color code/label category combinations.

To work with color-coded datasource resources, you must be familiar with the following tasks:

- Enabling the specific user interface items that can be color-coded or labelled. For details, see Datasource Options.
- Customizing your category scheme. For details, see Customizing Datasource Categories.
- Assigning a category to a datasource when creating or editing it. For details, see Manually Registering or Editing Datasources.

Customizing Datasource Categories

You can customize your datasource category scheme. A datasource category has the following configurable components:

- **Category name** - The name displayed as a selection in the **Select a category** dropdown
- **Short Name** - The abbreviation shown in the Datasource toolbar dropdown when a categorized datasource is selected.
- **Color** - The color used to denote a categorized datasource in the Navigator/Explorer tree icons and window tabs.

Note: Display of specific user items used to denote categories is configurable. For more information, see Datasource Options.

To customize your datasource categories

1 On the **Datasource** menu select **Register Datasource** and then select the **Datasource Group** panel.

2 From the **Select a category** dropdown select **Customize**. The **Manage Datasource Categories** dialogs opens.

3 Take one of the following actions:

- Create a new category by clicking **New** and selecting or providing a **Full name**, **Short name**, and **Color** combination.

- Edit an existing category by selecting the category, clicking **Edit** and modifying the name and color combination.

Note: The short name for a category cannot be edited.

- Delete an existing category by selecting the category, clicking **Delete** and verifying the deletion at the prompt.

Filtering in the Navigator/Explorer

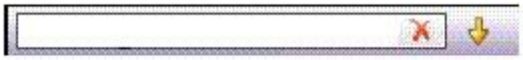
For ease of navigation, you can filter the Navigator/Explorer tree to display only selected nodes and objects. The following filtering options are available:

- Real-time modification of the tree to show only nodes with names matching a typed character string. For details, see Simple, on-the-fly, Character-based Tree Filtering.
- Permanent object filters, enabled and disabled for individual datasources, that restrict display based on object name and owning schema (and optionally, object type). For details, see Object Name/schema Filtering.
- The ability to restrict object types displayed for each DBMS platform. For details, see Node, or Object Type, Filtering.

Two default filters applying to owned and system objects are available. For details, see Using the Default Filters.

Simple, on-the-fly, Character-based Tree Filtering

The Filter box at the top of the Navigator/Explorer provides a quick, ad hoc way to restrict the nodes displayed.



To restrict the Explorer tree display to only user-defined nodes whose name contains a particular character string

- Type one or more characters in the Filter box.

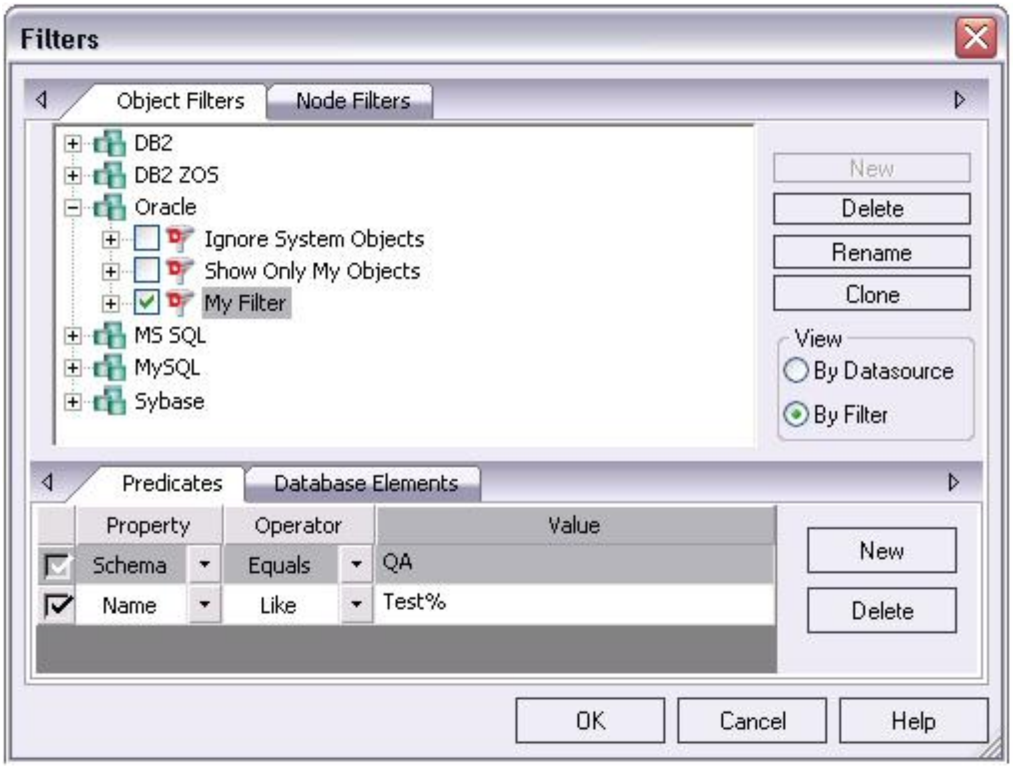
The display is updated to show only nodes whose name contains the typed character string and their parent nodes.

Note: The Clear button in the Filter box (X) deletes the contents of the Filter box and restores the unfiltered tree display.

Object Name/schema Filtering

Object filters let you restrict the Navigator/Explorer tree display for a datasource by name, owning schema, or name/schema combinations. These filters are defined at the DBMS platform level and are enabled or disabled on a datasource by datasource level. Multiple object filters can be created for a DBMS platform, letting you enable combinations of filters for a single datasource, ad hoc enabling and disabling of individual filters depending on your requirements, and so on.

Each filter consists of one or more ANDed conditions based on name or schema. For example, you could create a two-condition filter that restricts the Navigator/Explorer tree for a datasource to display only objects belonging to the Schema **QA** and whose name starts with the string **Test**.



	Property	Operator	Value
☑	Schema	Equals	QA
☑	Name	Like	Test%

Each filter condition is a simple expression that tests object names for equality (**Equals**, **Not Equals**), inclusion in a specified list (**In, Not In**), or using pattern matching (**Like, Not Like**). Optionally, an object filter can also be used to restrict display of specified object types on a datasource.

To create an object filter that can be enabled for all datasources for a DBMS platform

1 Right-click in the Navigator/Explorer tree and select **Filter** from the context menu.

The **Filters** dialog opens. The **Object Filters** tab opens by default, letting you create an object filter definition. The **Node Filters** tab, on the other hand, lets you perform real-time filtering of object types for all datasources on a DBMS platform. For details, see Node, or Object Type, Filtering.

2 In the **View** area, select **By Filter**.

Note: The current instructions create a filter at the DBMS level. To create a filter for a single datasource, select the **By Datasource** view, select a specific datasource, and then proceed to step4.

3 Expand the icon corresponding to the DBMS for which you want to create a filter and ensure that the icon is selected.

4 Click **New** and type a name for the new filter.

5 On the **Predicates** tab, create one or more name-based or schema-based conditions, all of which must be satisfied when this filter is enabled. Objects that do not satisfy the conditions will be filtered from view. Use the following guidelines in constructing conditions:

- The **Like** and **Not Like** operators take pattern-matching **Value**s. Wildcard characters and conventions supported on the target DBMS are supported.
- The **In** and **Not In** operators take a comma-delimited list such as **CO,CA,CU**, as a **Value**.
- Click the associated **New** button to add new predicates.

6 Optionally, on the **Database Elements** tab, select the check boxes for all object types to which this filter is to be applied. The filter does not affect the display of any deselected object types.

7 Click **OK**.

By default, when you create an object filter for a DBMS platform, it is enabled for each registered datasource of that DBMS type.

To disable or enable an object filter for a datasource

1 Right-click in the Navigator/Explorer tree and select **Filter** from the context menu. The **Filters** dialog opens.

2 In the **View** area, select **By Datasource**.

3 Expand the icon corresponding to the target DBMS and then expand the icon corresponding to the specific datasource for which you want to for enable or disable a filter.

4 Enable an object filter by selecting the associated check box or disable the object filter by deselecting the check box.

5 Click **OK**.

Note: You can also enable and disable filters using controls found under the **Filters** node for each datasource in the Navigator/Explorer tree.

You can also use the **Filters** dialog to select and edit an object filter, rename or create a clone of an existing object filter, and delete filters.

Node, or Object Type, Filtering

Node filtering lets you hide specified Navigator/Explorer tree nodes for a DBMS platform. For example, if you do not typically work with credential-related objects such as users, groups, profiles or logins, you may wish to filter those out of the Navigator/Explorer tree. You could either disable display of the individual object types or if practical, disable display of the **Security** node for the DBMS.

To change the hidden/displayed status of object types for a DBMS platform

1 Right-click in empty space in the Navigator/Explorer tree and select **Filter** from the context menu. The **Filters** dialog opens.

The **NodeFilters** tab opens by default, letting perform real-time filtering of object types for all datasources on a DBMS platform. The **Object Filters** tab, on the other hand, lets you create permanent, name-based or schema-based filters. For details, see Object Name/schema Filtering.

2 Click the **Node Filters** tab.

3 Expand the icon corresponding to the DBMS for which you want to create a node filter, and continue to expand nodes until all nodes for which you want to change the hidden/displayed status, are visible.

4 Ensure that all nodes corresponding to objects types that are to be displayed, are selected. Similarly, ensure that all nodes corresponding to objects types that are to be hidden, are deselected.

5 Click **OK**.

Hidden/displayed status is retained for the DBMS until you explicitly open the Filters dialog and change the current settings.

Using the Default Filters

The **Filters** node for each registered datasource includes two default Navigator/Explorer tree filters, **Ignore System Objects** and **Show Only My Objects**. These filters can be enabled and disabled for each datasource but cannot be edited or deleted.

Server Management Options (DBArtisan)

The following features are offered for operating and configuring servers:

- Starting, pausing, and resuming a server (DBArtisan - SQL Server)
- Shutting Down a Server (DBArtisan - SQL Server, Sybase ASE)
- Working with Configuration/initialization Parameters (DBArtisan)
- Session Recording (DBArtisan)

Starting, pausing, and resuming a server (DBArtisan - SQL Server)

You can start a server that has been shut down. As well, you can also pause and resume the SQL Server service.

To start a server, pause a server, or resume after pausing

- Use the **Datasource** menu's **Start Server**, **Pause Server**, and **Continue After Pause** commands.

Shutting Down a Server (DBArtisan - SQL Server, Sybase ASE)

You can perform an ordered or immediate shutdown of an SQL Server or Sybase ASE.

To shut down a server

1 In the Navigator/Explorer, select the target datasource.

2 On the **Datasource** menu, select **Shutdown**. The **Shutdown Server** dialog box opens.

The table below describes the options and functionality on the Shutdown Sever dialog.

Option	Description
Normal	Shuts down the server in an orderly fashion. Logins are disabled and currently executing Transact-SQL statements and stored procedures are allowed to finish. **Wait Time** (Sybase ASE only) - Lets you enable a grace period, an upper limit on how long to wait for normal operations to complete before forcing a shutdown. Wait time is specified in hours, minutes, and seconds, and if this option is not enabled, the Sybase ASE server will wait indefinitely before shutting down.
Immediate	Shuts down the server immediately. Checkpoints are not performed. The server terminates all user processes and rolls back any active transactions.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Working with Configuration/initialization Parameters (DBArtisan)

You can view, extract, and modify DBMS-specific configuration/parameters:

Platform	Parameter, options, or variables	For a listing, see:
DB2 LUW	Database systems configuration parameters	Supported DB2 LUW System Configuration Parameters (DBArtisan)
MySQL	Server System Variables	Supported MySQL Server System Variables (DBArtisan)
ORCL	Initialization parameters	Supported Oracle Initialization Parameters (DBArtisan)
SQL SVR	Server Configuration Options	Supported SQL Server Configuration Options (DBArtisan)
SYB ASE	Configuration parameters	Supported Sybase ASE Configuration Parameters (DBArtisan)

To display the server configuration/initialization parameters and options

1 Open the Datasource Explorer as configuration/parameter nodes are not visible in the Navigator. For more information, see Working with Bookmarks.

2 Connect to the datasource for which you want to view the server configuration/initialization parameters or options.

For details, see Connecting to Datasources.

3 Take one of the following actions:

- If working against a DB2 LUW datasource, select the **Database Configuration** node.
- If working against a SQL Server, MySQL, or Sybase ASE datasource, expand the **Server** node and then select the **Configuration** node.
- If working against an Oracle datasource, expand the **Instance** node and then select the **Configuration** node.

Server configuration/initialization parameters or options are displayed in the right pane of the Explorer.

See the following topics provide instructions on other available options:

- Extracting Configuration/Initialization Statements (DBArtisan)
- Modifying Configuration/Initialization Parameters and Options (DBArtisan)

Extracting Configuration/Initialization Statements (DBArtisan)

You can extract the following statement types corresponding to the DBMS-specific methods for setting server configuration/initialization parameters.

MySQL	SET... SET GLOBAL...
ORCL	ALTER SYSTEM SET
SQL SVR	RECONFIGURE/GO/EXEC sp_configure
SYB ASE	EXEC sp_configure

To extract configuration/initialization parameter statements

1 Open the Datasource Explorer as configuration/parameter nodes are not visible in the Navigator. For more information, see Working with Bookmarks.

2 Connect to the datasource for which you want to view the server configuration/initialization parameters or options. For details, see Connecting to Datasources.

3 Display the configuration/initialization parameters or options for the server. For details, see Working with Configuration/initialization Parameters (DBArtisan).

The server configuration parameters are displayed in the right pane of the Explorer.

4 In the right pane of the Explorer, right-click one or more selected parameters and select **Extract**. The DDL Editor opens. For more information, see Using the SQL Editor.

For related information, see Modifying Configuration/Initialization Parameters and Options (DBArtisan).

Modifying Configuration/Initialization Parameters and Options (DBArtisan)

You can modify the configuration/initialization parameters for each server.

Note: Before modifying configuration/initialization parameters consult the documentation for each platform. Many changes to these parameters require you to shutdown and to restart the server or have other effects. For access to third-party documentation, see Accessing Third Party Documentation.

Note: Not all available parameters are modifiable.

To modify a configuration/initialization for a datasource

1 Connect to the datasource for which you want to view the server configuration/initialization parameters or options.

For details, see Connecting to Datasources.

2 On the **Datasource** menu, click **Configure**.

- If you are working with DB2 LUW, Sybase ASE, or Oracle, a dialog opens, displaying each supported configuration/initilization parameter, its current value, and a description. Double-click a parameter to open an **Edit Configuration** dialog on the parameter. Provide a new value in the **New Value** box and then click **OK** to accept the new value and close the dialog. When finished editing parameters, click E**xecute**.
- If you are working with SQL Server or MySQL, a System Configuration tab opens, displaying each supported configuration/initilization parameter, its current value, and a description. Click in the **Value** field of a parameter to select it and then click in the **Value** field a second time to enable editing. Provide a new value in the **Value** field and then click anywhere outside the field to accept the new value. When finished editing parameters, close the tab and when prompted to save the system configuration, select **Yes**.

Notes on Oracle Managed Files (OMF)

For Oracle 91 only, Oracle Managed Files (OMF) suport is provided. OMF automatically manages files for tablespace, redo logs, and control files. Oracle automatically creates and removes operating system files as you create and drop storage objects.

You can enter the following query to set the db_create_file_dest parameter at the system level. You can set the parameter value for the session level by using Alter session clause:

DB_CREATE_FILE_DEST = '/u01/oradata/sample2'

Tip: You can also set the parameter at the system level in the Edit Configuration dialog box.

Session Recording (DBArtisan)

The session recording feature records changes to the server. The Session Recording feature lets you create a script file that you can execute as a batch at another time.

The session recording feature automates repetitive tasks that you routinely perform by issuing the commands in the collected script with a single command. Session recording lets you:

- Streamline time consuming, repetitive tasks.
- Execute multiple actions with from a single file.
- Automate a complex series of tasks by creating a batch file.
- Distribute automated tasks to multiple users with the session recording file.

The Session Recording feature is not available for operations that are destructive to your server. These operations include:

- Schema Migration
- Data Load
- ISQL and DDL Editors

When you activate session recording, these operations are grayed out and not available.

You can also use the session recording feature for Tivoli Integration. Schema change management sessions are recorded to instruction files, which are read by Tivoli Management Systems. Tivoli subsequently transports these to multiple managed endpoints for local execution.

For more information, see Recording a Session (DBArtisan) and Executing a Session File (DBArtisan).

Recording a Session (DBArtisan)

You can create a session file that includes a series of commands or actions using the **Session Recording Mode** dialog box.

Completing the Session Recording Mode Dialog Box

1 On the Datasource menu, click Session Recording.

The table below describes the options and functionality on the Enter Session Recording Mode dialog box.

Option	Description
Session Name	Lets you type the name of the session.
Session File	Lets you type the location and session file name or click the browse button. Session files have a *.ses file extension.

Note: Each task that you record is appended to the session file. In place of an Execute button, a Record button is displayed. To record a certain task, click the Record Button to open the Task Recording dialog box.

2 When you want to exit session recording mode, open an ISQL Window that lets you review the session file.

3 On the Datasources menu, click Session Recording.

An SQL window opens.

For more information, see Session Recording (DBArtisan) and Executing a Session File (DBArtisan).

Executing a Session File (DBArtisan)

You can execute a session file which includes a series of collected commands in a single batch from an ISQL Window.

1 On the File menu, click Open.

The Open Files dialog box opens.

2 Select the session file(s) (*.ses) from the file list, or in the File Name box, type a session file name.

3 Click OK.

Tthe session file opens into a SQL window.

4 On the SQL tool bar, click the Execute button.

5 On the SQL tool bar, click the Step-Execute button to see a graphical display to identify the execution path of the script, and then click the Execute button to execute the session file.

6 On the SQL tool bar, click the Query Plan button to see a graphical display to identify the execution path of the script.

7 Click the Execute button.

The script executes.

For more information, see Session Recording (DBArtisan).

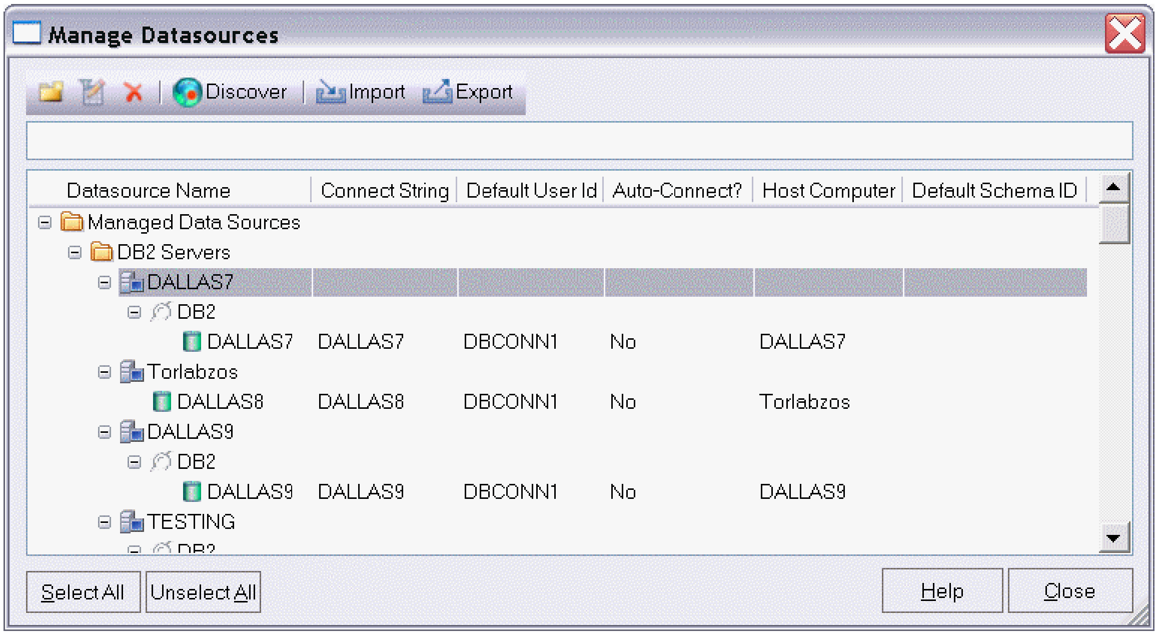
Using the Manage Datasources Utility

The Manage Datasources dialog box lets you manage datasources throughout your enterprise from a single vantage point. It provides concise, relevant information on your datasources in a simple grid format. It provides centralized access to common datasource operations such as adding, modifying, deleting, discovering, importing, and exporting datasources.

To open the Manage Datasources dialog box:

1 On the Datasource menu, click ManageDatasources.

The Manage Datasources dialog box opens.



Datasource Name	Connect String	Default User Id	Auto-Connect?	Host Computer	Default Schema ID
Managed Data Sources					
DB2 Servers					
DALLAS7					
DB2					
DALLAS7	DALLAS7	DBCONN1	No	DALLAS7	
Torlabzos					
DALLAS8	DALLAS8	DBCONN1	No	Torlabzos	
DALLAS9					
DB2					
DALLAS9	DALLAS9	DBCONN1	No	DALLAS9	
TESTING					

The grid format lets you access and view datasource specifications. datasources are grouped according to the default, DBMS-based folder structure and then within any custom datasource groups you have set up. The table below describes information available for each datasource entry:

Column	Description
Datasource Name	Uses an explorer-like interface to display all registered datasources and their groups. You can navigate this column in the same manner as the datasource explorer, by clicking on nodes to expand or collapse your view of the datasources.
Connect String	Displays the full connection string for the datasource.
Default User Id	Displays the Default User ID for the datasource.
Auto-Connect?	Indicates whether the Auto Connect feature is turned on or off.
Host Computer	Displays the name of the Host Computer if one has been configured.
Default Schema	Displays view default schemas for your DB2 datasources.

In addition, you can take the following actions:

- Click the New Datasource button to register a datasource. For more information, see Registering Datasources.
- Select a datasource and click the Edit Datasource button to edit that datasource definition.
- Select a datasource and click the Remove Datasource button to unregister that datasource definition. For more information, see Categorizing Datasources using Color-coding and Labeling.
- Click the **Discover** button to locate all configured datasources on your network and automate the process of registering them. For more information, see Moving Datasources Between Groups.
- Import or export datasource definitions. For details, see Importing and Exporting Datasource Definitions.

DB Team Server Support

DB Team Server is a data governance repository that can be used as an aid in managing enterprise data. It centralizes data resources, aids in resource-allocation planning, and fosters collaboration.

As an inventory, DB Team Server stores the following resources:

- Datasource definitions
- ER/Studio data models
- Glossaries and definitions
- User details.

Team Server is also a communications hub and information center. Activity streams provide the following:

- Datasources are monitored for DBMS alerts on conditions regarding storage and performance.
- Add, modify, and delete resource details are logged.
- Users can post comments on resources or to other users.

Common social media conventions such as *Like*s and *Follow*s are supported.

DBArtisan integrates with DB Team Server. For information on preliminary steps, see Setting up and Connecting to DB Team Server. Once configured and connected, the following options are provided:

Datasources	Datasource definitions on DB Team Server can be viewed in a dedicated Team Server Navigator. Local datasources created from the Team Server definitions can be permanently synchronized with the definitions on Team Server. A set of options and actions let you manage your local and DB Team Server datasource definitions and their relationships. An advanced search mechanism is also provided. For details, see Working with DB Team Server Datasources.
Data models	If a database or schema of a synchronized datasource has associated model data in Team Server, model data such as alerts and attribute/entity descriptions can be displayed in the SQL Editor as Code Assist proposals, as comments automatically added to scripts, as tooltips, and as the basis for warnings due to alerts. For details, see Viewing Embarcadero Team Server Model Metadata in the SQL Editor.
Activity Streams	Users can monitor their own activity stream or the activity stream for a datasource definition on the Team Server. For details, see Working with DB Team Server Streams.

In addition, DBArtisan lets you open a browser interface that lets you work directly with DB Team server. For details, see Working Directly with DB Team Server.

Setting up and Connecting to DB Team Server

Before you can make use of resources on DB Team Server, you must login to the server. See your Team Server administrator for a login URL, credentials, and permissions setup.

1 Ensure that DB Team Server usage is enabled. For details, see Team Server Options.

2 Select **Tools > Team Server > Login** and then use the **Embarcadero Team Server - Log In** dialog to provide credentials and complete the login.

Note: Use the **Keep me signed in** check box to have logins persist across startups and shutdowns of DBArtisan. The **Keep me signed** control can be set and disabled using a registry entry. For details, see Locking Down Features Using the Registry.

The Team Server Navigator opens. After logging in, you have access to resources on the DB Team Server. For a general description of available functionality, see Using the Team Server Navigator.

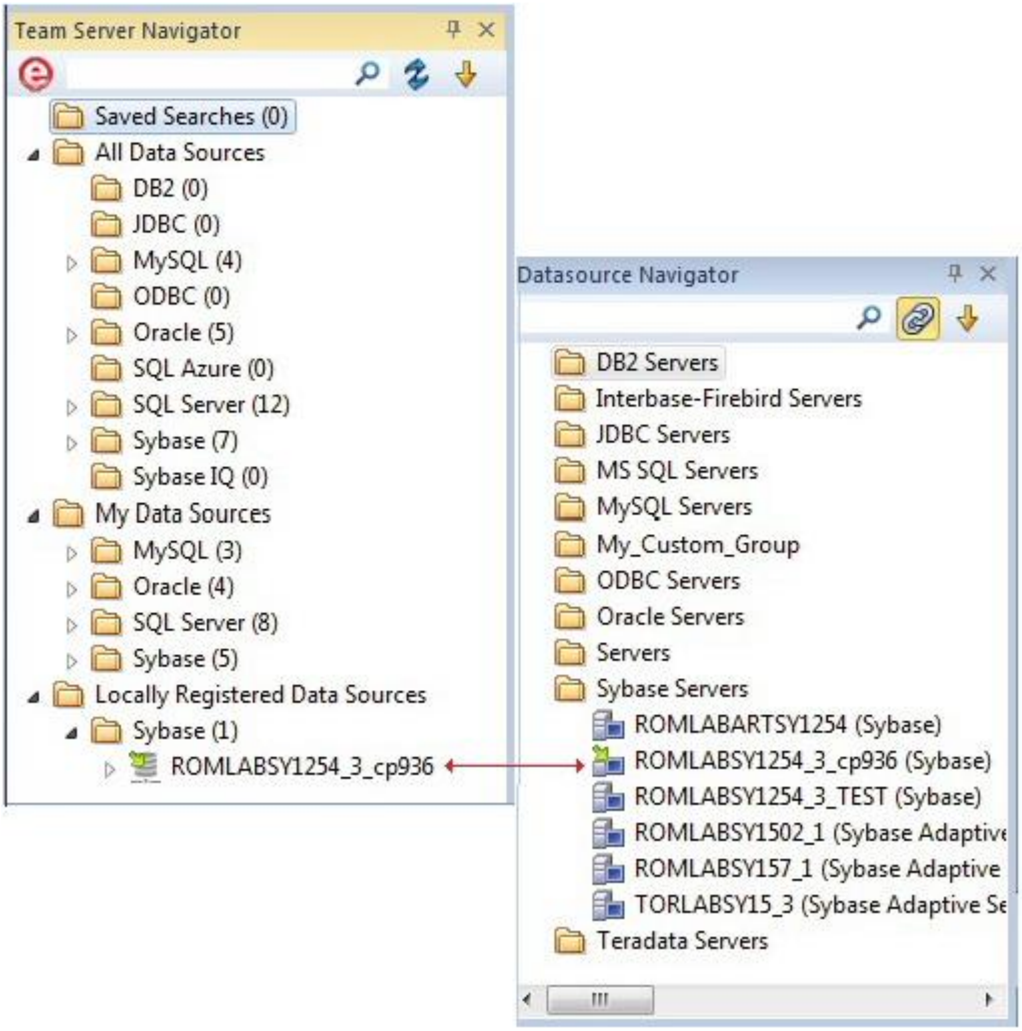
For related information, see the following topics:

- Viewing Embarcadero Team Server Model Metadata in the SQL Editor
- Working with DB Team Server Streams
- Working Directly with DB Team Server

Working with DB Team Server Datasources

With DBArtisan properly configured, and subsequent login to DB Team Server, a dedicated DB Team Server Navigator shows the available datasource definitions on the DB Team Server. A datasource definition on DB Team Server can be registered as a local datasource.

Registering a DB Team Server datasource as a local datasource creates a local copy of the datasource definition, accessible in the existing Datasource Navigator/Explorer.



Note: For access to DB Team Server documentation, see docs.embarcadero.com.

By default, when you register a DB Team Server datasource as a local datasource, the local definition is “attached” to the DB Team Server version. The local definition’s key connectivity information is synchronized with the DB Team Server definition on each attempt to connect to the datasource.

An attached datasource has a distinctive icon:

You can subsequently detach a local datasource definition from a DB Team Server definition. Settings for the detached, local datasource definition are those that were active on the DB Team Server definition when the datasource was detached. When no longer attached to a DB Team Server datasource definition, a detached datasource is functionally no different than any locally registered (manually registered or automatically discovered) datasource.

The Team Server Navigator lets you perform tasks such as attaching local datasources and managing datasource definitions residing on DB Team Server.

Basic steps in getting started with DB Team Server datasources are as follows:

1 **Enable and configure DB Team Server usage** - For details, see Team Server Options.

2 **Log in to DB Team Server** - For details, on how to establish a session with DB Team Server, see Setting up and Connecting to DB Team Server.

3 **Register DB Team Server datasources definitions as local datasources** - See Using the Team Server Navigator for information on how to use a simple Explorer interface to manage your local/DB Team Server datasources.

Note: Key connectivity information, the **Connection Information** tab settings (excluding **Datasource Name**) and a custom JDBC driver if supplied, is not editable in the resulting local datasource definition. **Security Parameters** tab information, such as **User ID** and **Password**) is not copied to the local datasource definition.

4 **Connect to a Team Server datasource that you have registered locally** - See Connecting to Datasources.

For related information, see:

- Searching DB Team Server for Datasources
- Reconciling Local and Team Server Datasources
- Viewing Embarcadero Team Server Model Metadata in the SQL Editor.

Using the Team Server Navigator

The Team Server Navigator displays datasources on DB Team Server and lets you work with relationships to local datasources.

Before working with the Team Server Navigator:

- Familiarize yourself with the Datasource Navigator/Explorer. It offers comparable functionality. For details, see Using the Datasource Explorer/Navigator.
- Understand the basics of DB Team Server datasource support. For details, see Working with DB Team Server Datasources.

To display the Team Server Navigator

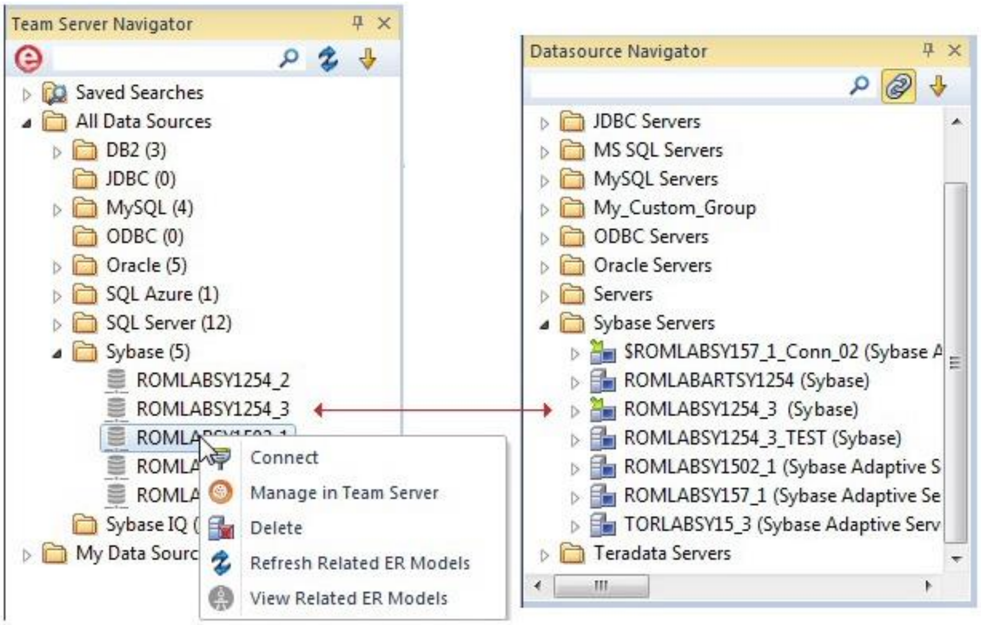
1 Ensure that DB Team Server usage is enabled and that you are logged in. For details, see Setting up and Connecting to DB Team Server.

2 Use **View > Team Server > Navigator** to toggle display of the Team Server Navigator.

At the topmost level, the **Saved Searches** folder contains search definitions you created using the advanced search feature. For details, see Searching DB Team Server for Datasources. The **All Data Sources** folder stores definitions for all available datasources on the DB Team Server. The **My Data Sources** folder provides access to those datasources that you are following, as set up on your DB Team Server account.

Within the two datasource folders, the next levels in the expandable tree are:

- Datasource groups
- Datasource definitions in the DB Team Server repository



Using the Datasource Navigator and the Team Server Navigator displayed, you can then:

- Create a local datasource from a definition on the repository - Register a local datasource as you would any static datasource but use the **Relate Team Server Source** control to attach the local definition to a specified Team Server datasource definition. For more information, see Manually Registering or Editing Datasources.

The icon for a datasource that is attached to a Team Server definition shows a green arrow in the upper left.

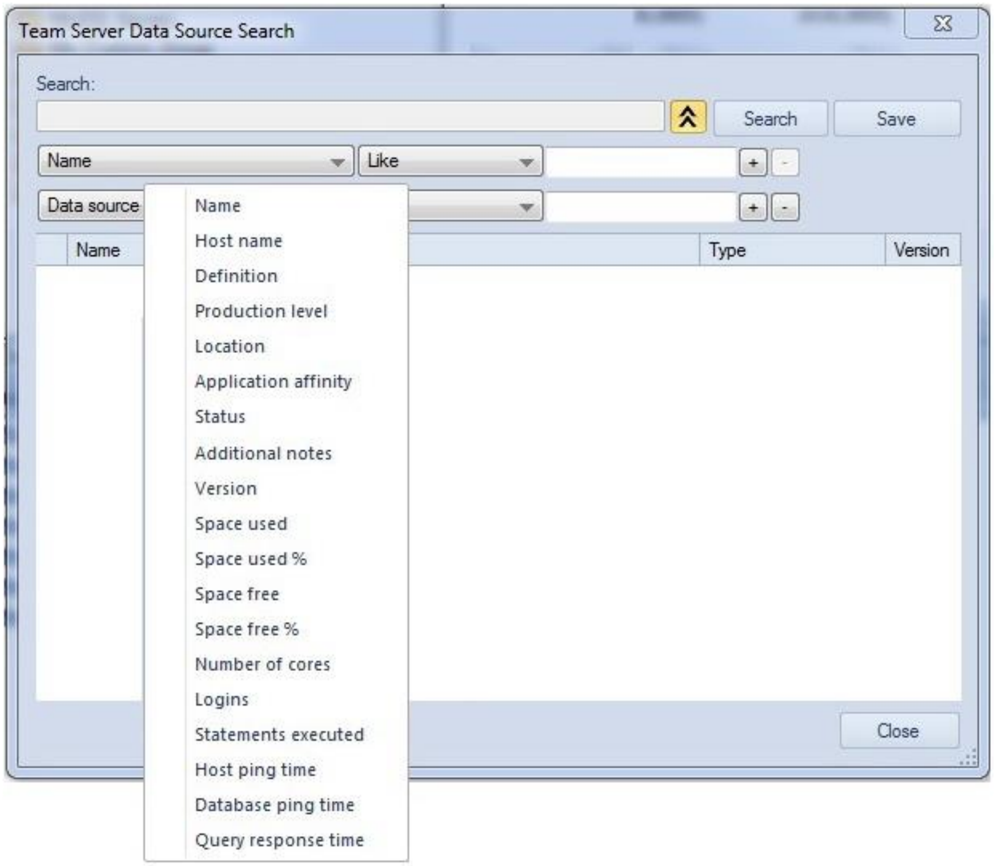
- Detach a currently attached local datasource definition so that it is no longer synchronized with the DB Team Server definition - right-click the attached datasource in the Datasource Navigator and select **Unrelate Team Server Data Source** and select **Yes** when prompted to verify.
- Work directly with datasource definitions in DB Team Server - Right-click a repository datasource definition in the Team Server Navigator, and select **Manage in Team Server**.
- Filter displayed nodes by name - Type one or more characters in the filter box at the top of the Explorer to show only nodes whose name contains the typed character string and their parent nodes.

- Display nodes by attached/unattached status - Use the **Filter** option on the dropdown menu at the top of the Team Server Navigator to display all datasource definitions, only definitions from which a local datasource has been registered, or only definitions from which no local datasource has been registered.
- Work with multiple local and Team Server datasources. For details, see Reconciling Local and Team Server Datasources.

Searching DB Team Server for Datasources

While the Team Server Navigator provides a useful browsing mechanism, you can also perform an advanced search of datasources on DB Team Server. The **Team Server Data Source Search** dialog lets you search for Team Server datasource definitions by name or by one or more ANDed conditions:

Name	**Datasource Type**	**Host Name**
Definition	**Production Level**	**Location**
Application Affinity	**Status**	**Additional Notes**
Version, **Space Used**	**Space Used %**	**Space Free**
Space Free %	**Number of Cores**	**Logins**, **Statements Executed**
Host Ping Time	**Database Ping Time**	**Query Response Time**



To initiate a name-based or criteria-based advanced search

1 Ensure that DB Team Server usage is enabled and that you are logged in. For details, see Setting up and Connecting to DB Team Server.

2 Select **Datasources > Team Server Data Sources**. The **Team Server Data Source Search** dialog opens.

To search by criteria

1 Use the expand/collapse control to expose the condition controls.

This deactivates the **Search** box.

2 Use the criteria, operator, and value controls to create a condition.

3 To add conditions, use the **+** control and repeat the previous step.

4 When ready, click **Search**.

For more general information on DB Team Server integration, see DB Team Server Support.

Reconciling Local and Team Server Datasources

While the Team Server Navigator provides separate, individual operations for working with the relationships between local and Team Server datasource definitions, the **Reconcile Team Server Data Sources** dialog lets you work simultaneously with multiple datasource definitions.

To reconcile multiple datasource definitions

1 Ensure that DB Team Server usage is enabled and that you are logged in. For details, see Setting up and Connecting to DB Team Server.

2 On the Team Server Navigator dropdown, select **Reconcile Data Sources**. The **Reconcile Team Server Data Sources** dialog opens.

3 Reconcile your Team Server and local datasources using the three list boxes as follows:

- **Matching data sources** - lists DB Team Server datasource definitions and local datasource definitions with identical connection information details but that are currently not attached. Selecting an entry from this list attaches the local datasource to the Team Server datasource, ensuring that any changes to the DB Team Server definition will be copied to the local datasource definition on the next attempt to connect to the datasource

Note: For more information on local, attached, and matching datasource definitions, see Working with DB Team Server Datasources.

- **Unmatched Team Server data sources** - lists DB Team Server datasource definitions for which there are no local datasource definitions with matching connection information. Selecting an entry from this list creates a matching local datasource definition and attaches the local definition to the matching Team Server datasource definition.
- **Unmatched local data sources** - lists local datasource definitions for which there are no DB Team Server datasource definitions with matching connection information. Selecting an entry from this list creates a matching Team Server datasource definition and attaches the local definition to the DB Team Server datasource definition.

4 Click **Reconcile**.

Working with DB Team Server Streams

Embarcadero DB Team Server activity streams include user Likes, user follows, and alerts associated with users or datasources. You can open a window to view your notification.

Note: Embarcadero DB Team Server is a licensed feature. For access to Team Server features and to obtain credentials, contact your Team Server administrator. For access to Team Server documentation, see http://docs.embarcadero.com/.

To view the activity stream for a Team Server resource

1 Ensure that Team Server usage is enabled and configured. For details, see Team Server Options.

2 Select **View > Team Server > Streams**. The **Team Server Streams** window opens.

3 Use the controls in the **Team Server Streams** window to display the activity stream for the current user or a DB Team Server datasource definition.

For more general information on DB Team Server integration, see DB Team Server Support.

Viewing Embarcadero Team Server Model Metadata in the SQL Editor

If a database or schema has associated model data in Team Server, model data such as alerts and attribute/entity descriptions can be displayed/included in the SQL Editor in the following ways:

- Attribute/Entity descriptions can be included in Code Assist proposals
- On explicitly formatting a script, metadata for tables and views referenced in SELECT, CREATE PROCEDURE, CREATE FUNCTION, CREATE TRIGGER, CREATE VIEW statements are added in a single comment block above the referencing statement
- When hovering the mouse over a table column name, table name or view name, a tooltip showing the entity/attribute description is displayed
- Warning icons are displayed in the window gutter beside lines referencing objects whose metadata includes alerts
- On execution of a script referencing objects whose metadata includes alerts, the error pane opens with an entry for each referenced object.

Note: Metadata for view columns is not available.

To use model metadata in the SQL Editor

1 Ensure that Team Server repository usage is enabled and that you are logged in to the Team Server. For details, see Team Server Options and Setting up and Connecting to DB Team Server.

2 Configure the **Metadata** features in the Options editor. For details, see ISQL Options - Code Assist Tab.

For related information, see DB Team Server Support.

Working Directly with DB Team Server

You can open a browser interface on DB Team Server to access its full range of functionality. This includes increased datasource and data model functionality, access to glossaries and terms, reporting, and access to other users.

To open a browser interface on DB Team Server

1 Ensure that access to DB Team Server has been enabled. For details, see DB Team Server Support.

2 Select **Tools > Team Server > View in Web browser**.

For access to the full set of DB Team Server documentation, see http://docs.embarcadero.com/.

Database Object Management

Extensive object management facilities are available for each supported DBMS platform. The following sets of topics describe the major areas of support:

Supported Objects	Describes the database objects and related elements supported. In addition to a listing of all supported objects by DBMS platform, a support summary is provided for each object type.
Creating objects	Tells you how to create new objects using graphical wizards.
Modifying objects using editors	Tells you how to edit existing objects.
Object actions	Provides details on the actions and operations that you can perform against database objects.

Supported Objects

Database object management features are available across different supported database platforms. The table below indicates the supported object types by platform:

	DB2 LUW	DB2 z/OS	ITB/FBD *	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
Alerts (DBArtisan)							✓	✓	
Aliases	✓	✓						✓	
Asymmetric Keys							✓		
Dump and Backup Devices (DBArtisan)							✓		
Bufferpools (DBArtisan)	✓	✓							
Certificates							✓		
Chains					✓				
Check Constraints	✓	✓	✓		✓	✓	✓	✓	
Clusters					✓				
Control Files (DBArtisan)					✓				
Database Devices (DBArtisan)							✓	✓	
Database Links					✓				
Database Triggers							✓		
Databases		✓			✓		✓	✓	
Data Caches (DBArtisan)								✓	
DBRM (DBArtisan)		✓							
Defaults							✓	✓	
Directories					✓				
Disk Groups (DBArtisan)					✓				
Domains			✓			✓			✓
Error Logs (DBArtisan)							✓		
Event Monitors (DBArtisan)	✓								
Events									✓
Exclusion Constraints						✓			

Extended Procedures								✓	
Foreign Keys	✓	✓	✓	✓	✓	✓	✓	✓	✓
Full-text Catalogs							✓		
Full-text Indexes							✓		
Functions	✓	✓		✓	✓	✓	✓	✓	✓
Groups	✓				✓		✓	✓	✓
Image Copies (DBArtisan)		✓							
Indexes	✓	✓	✓	✓	✓	✓	✓	✓	✓
Instance	✓				✓				
Java Classes					✓				
Java Resources					✓				
Java Sources					✓				
Jobs (Oracle)					✓				
Jobs (DBArtisan - SQL Server)							✓		
Job Queues					✓				
Join Indexes									✓
Libraries					✓				
Linked Servers (DBArtisan)							✓		
Logins							✓	✓	
Logs (DBArtisan - SQL Server Agent)							✓		
Materialized Query Tables	✓								
Materialized Views					✓				
Materialized View Logs					✓				
Nodegroups (DBArtisan)	✓								
Operators (DBArtisan)							✓		
Outlines					✓				

Package Bodies					✓				
Packages	✓	✓			✓				
Partition Functions							✓		
Partition Schemes							✓		
Plans		✓							
Primary Keys	✓	✓	✓	✓	✓	✓	✓	✓	✓
Procedures	✓	✓	✓		✓		✓	✓	✓
Profiles					✓				
Programs					✓				
Recycle Bin					✓			✓	
Redo Log Groups (DBArtisan)					✓				
Remote Servers (DBArtisan)							✓	✓	
Roles			✓		✓	✓	✓	✓	
Rollback Segments (DBArtisan)					✓				
Rules						✓	✓	✓	
Schedules					✓				
Schema						✓	✓		
Segments							✓	✓	
Sequences	✓	✓			✓				
Server Roles (DBArtisan)							✓		
Server Triggers (DBArtisan)							✓		
Stogroups (DBArtisan)		✓							
Structured Types	✓	✓							
Symmetric Keys							✓		
Synonyms		✓			✓		✓		
System Indexes							✓	✓	✓

System Tables							✓	✓	✓
System Triggers									✓
Tables	✓	✓	✓	✓	✓	✓	✓	✓	✓
Tablespaces	✓	✓			✓	✓			
Triggers	✓	✓	✓		✓	✓	✓	✓	✓
Type Bodies					✓				
Types					✓	✓			
Unique Keys	✓	✓	✓	✓	✓	✓	✓	✓	✓
User Datatypes	✓	✓					✓	✓	
User Messages	✓						✓	✓	
Users	✓	✓		✓	✓		✓	✓	✓
Views	✓	✓	✓		✓	✓	✓	✓	✓

Alerts (DBArtisan)

An alert is a definition that matches one or more Microsoft SQL Server events with a response. Errors and messages, or events, are generated by Microsoft SQL Server and entered into the Microsoft Windows application log. SQL Server Agent reads the application log and compares events to alerts that you defined. When SQL Server Agent finds a match, it fires an alert. DBArtisan lets you define alerts to respond to SQL Server events by notifying operators, forwarding the event to another server, or executing a job. These administrative tasks runs automatically if the SQL Server Agent service has been started.

Creating and editing

- Alerts Wizard (DBArtisan - SQL Server) and Alerts Editor (DBArtisan - SQL Server)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SQL SVR
Drop	✓
Extract	✓
Rename	✓

Aliases

Aliases let you assume the permissions of another database user without creating a separate user identity. You can use an alias when a user requires only temporary access to a database. You can also use an alias to mask a user's identity.

Creating and editing

See the following topics:

- Aliases Wizard (DB2 LUW) and Aliases Editor (IBM DB2 LUW)
- Aliases Wizard (DB2 Z/OS) and Aliases Editor (IBM DB2 Z/OS)
- Aliases Wizard (Sybase ASE) and Aliases Editor (Sybase ASE)

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	DB2 z/OS	SYB ASE
Drop	✓	✓	✓
Extract	✓	✓	✓
Migrate (DBArtisan)	✓	✓	✓
Report	✓	✓	✓
Transfer Ownership	✓		

Asymmetric Keys

Asymmetric keys, in addition to symmetric keys and certificates, provide support for Public Key Encryption. Asymmetric keys consist of a private key and corresponding public key and are used to secure symmetric keys.

Creating and editing

- Asymmetric Keys Wizard (SQL Server) and Asymmetric Keys Editor (SQL Server)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SQL SVR
Drop	✓
Extract	✓
Report	✓

Related information

- Certificates
- Symmetric Keys

Bufferpools (DBArtisan)

Bufferpools are areas of system memory to cache table and index data pages. DBArtisan retrieves table and index data pages from their physical storage location and places them in the bufferpool so that when you modify or access data, you can retrieve the data from memory rather than a physical device or file. Bufferpools improve database system performance because DBArtisan can retrieve the data quickly from memory.

Creating and editing

- Bufferpools Wizard for IBM DB2 (DBArtisan - DB2 LUW) and Bufferpool Editor (DBArtisan - IBM DB2 LUW)
- Buffer Pools Editor (DBArtisan - IBM DB2 Z/OS)

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	**DB2 z/OS**
Drop	✓	✓
Extract	✓	✓
Report	✓	✓

Certificates

Certificates, in addition to symmetric and asymmetric keys, provide support for Public Key Encryption. Certificates are digitally signed security objects containing a public key and optionally, a private key.

Creating and editing

- Certificate Wizard (SQL Server) and Certificates Editor (SQL Server)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SQL SVR
Add Private Key	✓
Backup Certificate	✓
Drop	✓
Extract	✓
Report	✓

Related information

- Asymmetric Keys
- Symmetric Keys

Chains

The following summarizes support for named series of tasks

Platform availability

DB2 LUW	DB2 z/OS	ITB/FBD *	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
				✓				

Creating and editing

No support is provided for creation or editing this object type.

DBMS platform availability and object actions/operations supported

No specific object actions are available for this object type/

Related Information

See details for the following, related object types:

- Jobs (Oracle)
- Programs
- Schedules

Check Constraints

Check constraints are data values that are acceptable in a column. They are logical expressions that verify column values meet defined acceptance criteria.

Creating and editing

See Add or Modify Check Constraint.

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	DB2 z/OS	ITB/FBD *	ORCL	PSTGRS *	SQL SVR	SYB ASE
Change Status (check constraints)				✓		✓	
Drop	✓	✓	✓	✓	✓	✓	✓
Extract	✓	✓	✓	✓	✓	✓	✓
Hide Text							✓
Migrate (DBArtisan)	✓	✓		✓		✓	✓
Rename	✓	✓	✓	✓	✓	✓	✓
Report	✓	✓	✓	✓		✓	✓
Transfer Ownership	✓						

Clusters

Clusters provide an optional method of storing table data. A cluster comprises of a group of tables that share the same data blocks, and which are grouped together because they share common columns and are often used together. The related columns of tables stored in a cluster are known as the cluster key.

There are two types of clusters:

- Index
- Hash

Index clusters store the cluster data together and index the cluster key, which should make them faster at retrieving a range of data rows.

Hash clusters apply hashing functions to the cluster key to determine the physical location of a data row, which should make them faster at retrieving specific data rows.

Note: To place a table on a cluster, 6 include the ON CLUSTER syntax within the CREATE TABLE statement. Placing a table on a cluster precludes you from placing it on a tablespace or defining the associated storage parameters.

Creating and editing

- Clusters Wizard (Oracle) and Clusters Editor (Oracle).

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Allocate Extent	✓
Analyze	✓
Deallocate Unused Space	✓
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Report	✓
Truncate	✓

Control Files (DBArtisan)

Control files map to the structures that make up the database including the location of datafiles. Control files are read-only because they are only constructed during database creation.

Creating and editing

No support is provide for creating or editing control files.

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Backup Control Files (DBArtisan - ORCL)	✓
Open Control Files (DBArtisan)	✓
Report	✓

Database Devices (DBArtisan)

A database device is a disk or a portion of a disk that stores databases and database objects.

Creating and editing

- Databases Device Wizard (DBArtisan - Sybase ASE) and Database Devices Editor (DBArtisan - Sybase ASE)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SYB ASE
Drop	✓
Extract	✓
Mirror (DBArtisan)	✓
Reinitialize (DBArtisan)	✓
Refit (DBArtisan)	✓
Report	✓
Resize (DBArtisan)	✓
Unmirror (DBArtisan)	✓

Database Links

Database links are named schema objects that describe a path from one database to another. Database links are implicitly used when a reference is made to a global object name in a distributed database. To use a database link, either it is public or you own it.

Note: Oracle syntax does not let you alter an existing database link. To change its definition, drop and re-create it.

Creating and editing

- Database Links Wizard (Oracle) and Database Links Editor (Oracle)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Rename	✓
Report	✓

Database Triggers

A database trigger implements a database-scope DDL trigger. These triggers fire in response to events associated with DDL statements.

Note: Before working with database triggers, consult Microsoft SQL Server documentation for a general understanding of DDL triggers. For more information, see Accessing Third Party Documentation.

Creating and editing

- Database Triggers Wizard (SQL Server) and Database Triggers Editor (SQL Server)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SQL SVR
Change Status	✓
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Report	✓

Related information

- Server Triggers (DBArtisan)

Databases

Databases are a collection of tables, or a collection of index spaces and tablespaces. The goals of a database system are straightforward but challenging. In general, a database aims to manage large amounts of data in a multi-user environment. It should achieve high performance while letting many users access the same information concurrently without compromising data integrity. A database also must protect against unauthorized access and provide reliable solutions for failure recovery.

Creating and editing

- Databases Wizard (DB2 LUW) and Databases Editor (IBM DB2 LUW)
- Databases Wizard (DB2 Z/OS) and Databases Editor (IBM DB2 Z/OS)
- Databases Wizard (SQL Server) and Databases Editor (SQL Server)
- Databases Wizard (Sybase ASE) and Databases Editor (Sybase ASE)

Note: Objects of this type cannot be created or edited against PostgreSQL datasources.

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	DB2 z/OS	PSTGRS *	SQL SVR	SYB ASE
Attach Database				✓	
Checkpoint				✓	✓
Copy Schema	✓				
DBCC				✓	✓
Detach Database				✓	
Drop	✓	✓	✓	✓	✓
Drop Automatic Storage Path(s)	✓				
Extract	✓	✓	✓	✓	✓
Hide Text					✓
Migrate (DBArtisan)	✓	✓		✓	✓
Move Log					✓
Quiesce (Database)	✓				
Rename				✓	✓
Report		✓		✓	✓
Resynchronize					✓
Set Online/Offline				✓	✓
Shrink				✓	
Start Database		✓			
Stop Database		✓			
Unquiesce	✓				
Update Statistics				✓	

DBMS-specific information

- IBM DB2 for Linux, Unix, and Windows Instances
- Microsoft SQL Server Databases
- IBM DB2 for OS/390 and z/OS Instances

  - Sybase ASE Databases

IBM DB2 for Linux, Unix, and Windows Instances

Databases are a collection of tables, or a collection of index spaces and tablespaces. The goals of a database system are straightforward but challenging. In general, a database aims to manage large amounts of data in a multi-user environment. It should achieve high performance while letting many users access the same information concurrently without compromising data integrity. A database also must protect against unauthorized access and provide reliable solutions for failure recovery.

Microsoft SQL Server Databases

Databases are a collection of tables, or a collection of index spaces and tablespaces. The goals of a database system are straightforward but challenging. In general, a database aims to manage large amounts of data in a multi-user environment. It should achieve high performance while letting many users access the same information concurrently without compromising data integrity. A database also must protect against unauthorized access and provide reliable solutions for failure recovery.

Note: Microsoft SQL Server recommends that you do not create any user objects, such as tables, views, stored procedures, or triggers, in the master database. The master database includes the system tables that store the system information used by SQL Server, such as configuration option settings.

IBM DB2 for OS/390 and z/OS Instances

Databases are a collection of tables, or a collection of index spaces and tablespaces. The goals of a database system are straightforward but challenging. In general, a database aims to manage large amounts of data in a multi-user environment. It should achieve high performance while letting many users access the same information concurrently without compromising data integrity. A database also must protect against unauthorized access and provide reliable solutions for failure recovery.

Sybase ASE Databases

Databases are a collection of tables, or a collection of index spaces and tablespaces. The goals of a database system are straightforward but challenging. In general, a database aims to manage large amounts of data in a multi-user environment. It should achieve high performance while letting many users access the same information concurrently without compromising data integrity. A database also must protect against unauthorized access and provide reliable solutions for failure recovery.

Data Caches (DBArtisan)

Data Caches are devices stored in memory. If configured properly, they can substantially boost performance by allowing users to read or write data in memory instead of to or from physical disk. With System 11, you can now “borrow” memory from the default data cache and allocate it to customized data caches.

Named data caches let you achieve substantially greater concurrency. Data caches also increase the likelihood that Sybase ASE reads data from memory and not disk. DBArtisan lets you bind particularly active databases (for example, tempdb), tables (for example, syslogs and sysindexes) or indexes to dedicated data caches. In addition, you can boost performance by creating memory pools, each configured for different I/O throughput (2 K to 16 K), within every named data cache.

In spite of their potential, named data caches are not a panacea for performance problems. You should still carefully optimize your queries and indexing strategy. Poor configuration of named data caches can lead to worse performance because precious memory resources are misallocated. To determine data cache performance, you should examine the cache hit ratio.

Creating and editing

- Data Caches Wizard (DBArtisan - Sybase ASE) and Data Cache Editor (DBArtisan - Sybase ASE)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SYB ASE
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Report	✓

DBRM (DBArtisan)

A Database Request Module (DBRM) is an application containing information on SQL statements extracted from the DB2 precompiler. A DBRM refers to a specific database and is stored outside of the DB2 directory.

Creating and editing

- DBRMs Editor (DBArtisan - IBM DB2 Z/OS)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 z/OS
Report	✓

Defaults

Defaults promote data integrity by supplying a default value to a table column if the user does not explicitly provide one. They are reusable objects that you can bind to table columns or user datatypes.

Creating and editing

- Defaults Wizard (SQL Server) and Defaults Editor (SQL Server)
- Defaults Wizard (Sybase ASE) and Defaults Editor (Sybase ASE)

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	SQL SVR	SYB ASE
Drop	✓	✓
Extract	✓	✓
Hide Text		✓
Migrate (DBArtisan)	✓	✓
Rename	✓	✓
Report	✓	✓

Directories

Directories create an alias to an external operating system directory to your database files, which can be used for storing large binary object files. When you create a directory, provide the full path name to the outside operating system where the BFILE's are stored. This object lets you store large files, such as video, outside of the database. The directory object lets you provide a simple alias to the full path name of an outside server's file system, which you can then use to point to the files when creating procedural logic objects. This saves the developer from having to type the full path name when coding.

To create a Directory object, you need CREATE ANY DIRECTORY system privileges. You also create or have in place a corresponding operating system directory to store the file. This directory must have the correct read permissions for Oracle processes.

Creating and editing

- Directories Wizard (Oracle) and Directories Editor (Oracle)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Report	✓

Disk Groups (DBArtisan)

When connected to an Oracle datasource with an ASM instance, the Explorer/Navigator's **Instance** node shows **ASM** and **ASM > Disk Groups** subnodes. DBArtisan lets you connect to the ASM instance and provides basic object management for disk groups.

DBArtisan lets you use the default datasource connection or a dedicated connection to the ASM instance. When using the default datasource connection, you can open a Disk Group Editor but you can only view limited properties of the disk group. With a dedicated connection to the ASM instance, you can create and edit disk groups and make use of the disk group-specific object actions: **Check Disk Group**, **Mount Disk Group**, **Rebalance Disk Group**, and **Undrop Disk**. After connecting to the datasource, you are prompted to choose between the default datasource connection and a dedicated ASM instance connection on the first object action against a disk group. The connection type persists and you can also explicitly change the connection type using the **Manage ASM Connection** and **Disconnect ASM Connection** actions.

Note: The **ASM** and **Disk Groups** nodes are only visible if the datasource instance has an ASM instance.

Creating and editing

- Disk Groups Wizard (DBArtisan - Oracle) and Disk Groups Editor (DBArtisan - Oracle)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Check Disk Group (DBArtisan)	✓
Disconnect ASM Connection (DBArtisan)	✓
Drop	✓
Extract	✓
Mount Disk Group (DBArtisan)	✓
Rebalance Disk Group (DBArtisan)	✓
Undrop Disk (DBArtisan)	✓

Connection details

When registering an Oracle datasource:

- As of Oracle 11gR1, you can specify the SYSASM privilege for the datasource connection. This provides full ASM administrative privilege for this connection.

  - Using the SYSDBA privilege, you can register the ASM instance directly using the Datasource Registration Wizard.

For more information, see Registering Datasources.

Domains

Support for domains is provided, letting you work with these type definitions.

Creating and editing

Domains Wizard (PostgreSQL) and Domains Editor (PostgreSQL).

Note: No support is provided for creation or editing of this object type against Sybase IQ datasources.

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions..

	ITB/FBD *	PSTGRS *	SYB IQ
Change Owner		✓	
Change Schema		✓	
Drop	✓	✓	✓
Extract	✓	✓	✓
Object Properties			✓
Rename	✓	✓	✓
Report			✓

Dump and Backup Devices (DBArtisan)

Dump devices store backups of databases and transaction logs. In Microsoft SQL Server, dump devices have been renamed to backup devices. Before you can dump a database or its transaction log, first create a dump device that stores the resulting backup file. Microsoft SQL Server and Sybase ASE lets you create tape and disk dump devices. Disk dump devices are a fast and convenient mechanism for backups. Tape dump devices facilitate off site storage for enhanced disaster recovery.

Creating and editing

- Backup Devices Wizard (DBArtisan - SQL Server) and Backup Devices Editor (DBArtisan - SQL Server)
- Dump Devices Wizard (DBArtisan - Sybase ASE) and Dump Devices Editor (DBArtisan - Sybase ASE)

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	SQL SVR	**SYB ASE**
Drop	✓	✓
Extract	✓	✓
Migrate (DBArtisan)	✓	✓
Report	✓	✓
View Contents (DBArtisan)	✓	

Error Logs (DBArtisan)

The following topics summarize support for archived error logs.

Creating and editing

Logs are generated and cannot be created or edited.

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SQL SVR
Open (DBArtisan - Logs and Error logs)	✓
Report	✓

Event Monitors (DBArtisan)

Event Monitors lets you monitor specified events that occur on the database including deadlocks, connections, and statements. Because monitoring is configurable by type, you can record activity against all trackable objects or only those you consider most important.

Note: Event Monitors cannot be altered using the ALTER command. An Event Monitor has to be dropped and re-created.

Creating and editing

- Event Monitors Wizard (DBArtisan - DB2 LUW) and Event Monitor Editor (DBArtisan - IBM DB2 LUW)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Report	✓
Transfer Ownership	✓

Events

These state changes events can be monitored or used as trigger conditions.

Creating and editing

No support is provided for creation or editing of this object type against a Sybase IQ datasource.

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SYB IQ
Drop	✓
Extract	✓
Object Properties	✓

Exclusion Constraints

An Exclusion Constraint guarantees that if any two rows are compared on the specified columns or expressions using the specified operators, not all of comparisons will return TRUE.

Creating and editing

Exclusion Constraints, Primary Keys, and Unique Keys Wizards (PostgreSQL) and Exclusion Constraints, Primary Keys, or Unique Keys Editors (PostgreSQL).

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	PSTGRS *
Drop	✓
Extract	✓
Rename	✓

Extended Procedures

Extended Procedures are dynamic link libraries that can be used to load and execute application routines written in other programming languages, such as C or Visual Basic. Extended Procedures function and appear in the same manner as normal stored procedures in that you can pass parameters to them and obtain results.

Note: Extended Procedures can only be accessed on the Master database.

Creating and editing

- Extended Procedures Wizard (Sybase ASE) and Extended Procedures Editor (Sybase ASE)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SYB ASE
Drop	✓
Describe	✓
Execute	✓
Extract	✓
Hide Text	✓
Migrate (DBArtisan)	✓
Rename	✓
Report	✓

Foreign Keys

Foreign keys enforce referential integrity between tables by verifying the existence of foreign key values in the parent table before letting you insert or update foreign key values in the child table.

Creating and editing

- Foreign Keys Wizard (DB2 LUW) and Foreign Keys Editor (IBM DB2 LUW)
- Foreign Keys Wizard (DB2 Z/OS)) and Foreign Keys Editor (IBM DB2 Z/OS)
- Foreign Keys wizard (MySQL) and Foreign Keys editor (MySQL)
- Foreign Keys Wizard (Oracle) and Foreign Keys Editor (Oracle)
- Foreign Keys Wizard (PostgreSQL) and Foreign Keys Editor (PostgreSQL)
- Foreign Keys Wizard (SQL Server) and Foreign Keys Editor (SQL Server)
- Foreign Keys Wizard (Sybase ASE) and Foreign Keys Editor (Sybase ASE)

Note: Creation and editing of objects of this type is not supported against Sybase IQ datasources.

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	DB2 z/OS	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
Change Status (check constraints)				✓		✓		
Drop	✓	✓		✓	✓	✓	✓	✓
Extract	✓	✓		✓	✓	✓	✓	✓
Migrate (DBArtisan)	✓	✓	✓	✓			✓	
Rename	✓	✓		✓	✓	✓	✓	
Report	✓	✓		✓		✓	✓	
Transfer Ownership	✓							

DBMS platform-specific notes

MySQL	MySQL recognizes two types of tables, MyISAM and InnoDB. MyISAM tables access data records using an index, where InnoDB tables allow transactions and foreign keys. Therefore, the discussion of foreign keys is limited to your InnoDB tables.

Full-text Catalogs

A full-text catalog is a set of operating system files that store full-text indexes. Full-text catalogs, along with full-text indexes and the object actions supported for these object types, provide full text search support.

Note: Before working with full-text catalogs, consult Microsoft SQL Server documentation for a general understanding of full-text searching. For more information, see Accessing Third Party Documentation.

Creating and editing

- Full-text Catalogs Wizard (SQL Server) and Full-text Catalogs Editor (SQL Server)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SQL SVR
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Rebuild (Full-text Catalogs)	✓
Reorganize (SQL Server Full-text Catalogs)	✓
Report	✓

Related information

- Full-text Indexes

Full-text Indexes

A full-text index stores information used by the Full-Text Engine to compile full-text queries that can quickly search a table for particular words or word combinations. A full-text index stores information about key words and their location within one or more columns of a table.

Full-text indexes, along with full-text catalogs and the object actions supported for these object types, provide full text search support.

Note: Before working with full-text indexes, consult Microsoft SQL Server documentation for a general understanding of full-text searching. For more information, see Accessing Third Party Documentation.

Creating and editing

- Full-text Indexes Wizard (SQL Server) and Full-text Indexes Editor (SQL Server)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SQL SVR
Change Status (Full-text Indexes)	✓
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Population status	✓
Report	✓

Related information

- Full-text Catalogs

Functions

Functions are subroutines that you define. Functions are useful for reusable application logic. You can use functions to determine the best methods for controlling access and manipulation of the underlying data contained in an object.

The table below describes the types of user-defined functions you can create:

Function	Description
Column or External Table Function	You can write in a host programming language, such as C. This function can act on a table and returns a table value rather than a scalar value.
External Scalar Function	You can write in a language other than SQL, such as C++ or Java and returns a scalar value to the program. This type of function is referenced by the CREATE FUNCTION statement and can be used to perform computations on data contained in the database but cannot directly reference the data.
OLEDB Function	Accesses OLE DB data in user-defined OLE DB external tables.
Sourced Function	Inherits the semantics of another function and can be an operator.
Template Function	Partial functions that do not contain any executable code. Mainly used in a federated database to map the template function to a data source function -Oracle, SQL Server, Sybase ASE, etc. A function mapping needs to be created in conjunction with the template function.

Creating and editing

- Functions Wizard (DB2 LUW) and Functions Editor (IBM DB2 LUW)
- Functions Wizard (DB2 Z/OS) and Functions Editor (IBM DB2 Z/OS)
- Functions wizard (MySQL) and Functions editor (MySQL)
- Functions Wizard (Oracle) and Functions Editor (Oracle)
- Functions Wizard (PostgreSQL) and Functions Editor (PostgreSQL)
- Functions Wizard (SQL Server) and Functions Editor (SQL Server)
- Functions Wizard (Sybase ASE) and Functions Editor (Sybase ASE)

Note: Creation and editing of objects of this type is not supported against Sybase IQ datasources.

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	DB2 z/OS	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
Change Owner					✓			
Change Owner					✓			
Compile				✓				
Create Synonym				✓				
Describe	✓	✓		✓				
Drop	✓	✓	✓	✓	✓	✓	✓	✓
Execute				✓				
Extract	✓	✓	✓	✓	✓	✓	✓	✓
Hide Text							✓	
Migrate (DBArtisan)	✓	✓	✓	✓		✓	✓	
Object Properties								✓
Rename					✓	✓	✓	
Report	✓	✓	✓	✓		✓	✓	
Transfer Ownership	✓							

Important Notes

With respect to functions and stored procedures, it can be necessary to have statements executed before and after creation of the procedure or function. This can be useful for example, if you need to create or drop temporary tables used by the function or procedure. Two tag pairs, ETStart and ETEnd, let you embed statements in the first comment block of a stored procedure or function. The following shows the expected syntax:

create procedure dbo.*procname*(@a numeric) as

/*

<ETStart>*SQL Statement*</ETStart>

<ETEnd>*SQL Statement*</ETEnd>

*/

begin

...

Groups

Groups are a defined collection of database users. The primary use of groups is to consolidate the management of permissions. By matching together similar users into groups, you can greatly reduce the number of commands required to set permissions.

Every user automatically belongs to the public group. To assign a user to another group, add the user to that group. Then the user belongs to that group and public.

Note: A user can only belong to one group at a time other than public.

Creating and editing

- Groups Wizard (DBArtisan - DB2 LUW) and Groups Editor (DBArtisan - IBM DB2 LUW)
- Groups Editor (DBArtisan - Oracle)
- Groups Wizard (Sybase ASE) and Groups Editor (Sybase ASE)

Note: Creation and editing of objects of this type is not supported against Sybase IQ datasources.

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	ORCL	SYB ASE	SYB IQ
Drop			✓	✓
Extract	✓	✓	✓	✓
Migrate (DBArtisan)	✓	✓	✓	
Report	✓	✓	✓	

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

Image Copies (DBArtisan)

Image Copies are a full image copy of all pages in a tablespace, partition, data set, or index space. Image Copies can also be an incremental image copy of only pages that have been modified since the last use of the Image Copy Wizard.

Creating and editing

No support is provide for creating or editing Image Copies..

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 z/OS
Copy to Copy (DBArtisan)	✓
Merge Copy (DBArtisan)	✓
Modify Recovery (DBArtisan)	✓

Indexes

Indexes are optional structures associated with tables. You can create indexes specifically to speed SQL statement execution on a table. When properly used, Indexes are the primary means of reducing disk I/O. Indexes are logically and physically independent of the data in the associated table. Unique Indexes guarantee that no two rows of a table have duplicate values in the columns that define the index.

Creating and editing

- Indexes Wizard (DB2 LUW) and Indexes Editor (IBM DB2 LUW)
- Indexes Wizard (DB2 Z/OS) and Indexes Editor (IBM DB2 Z/OS)
- Indexes, Primary Keys, or Unique Keys wizard (MySQL) and Indexes, Primary Keys, and Unique Keys editors (MySQL)
- Indexes Wizard (Oracle) and Indexes Editor (Oracle)
- Indexes Wizard (PostgreSQL) and Indexes Editor (PostgreSQL)
- Indexes Wizard (SQL Server) and Indexes Editor (SQL Server)
- Indexes Wizard (Sybase ASE) and Indexes Editor (Sybase ASE)

Note: Creation and editing of objects of this type is not supported against Sybase IQ datasources.

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	DB2 z/OS	ITB/FBD *	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ	SYB IQ
Allocate Extent					✓					
Analyze					✓					
Check Index (DBArtisan)		✓								
Copy to Copy (DBArtisan)		✓								
Deallocate Unused Space					✓					
Disable Index			✓				✓			
Drop	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DBCC							✓	✓		
Estimate Size					✓			✓		
Extract	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Migrate (DBArtisan)	✓	✓		✓	✓		✓	✓		
Modify Recovery (DBArtisan)		✓								
Modify Statistics (DBArtisan)		✓								
Place								✓		
Rebuild Index		✓	✓		✓			✓		
Recover (DBArtisan)		✓								
Rename	✓	✓	✓		✓	✓	✓	✓		
Reorganize /Reorg	✓	✓					✓	✓		
Report	✓	✓	✓	✓	✓		✓	✓		
Runstats Index (DBArtisan)		✓								
Shrink					✓					
Transfer Ownership	✓									

Update Statistics	✓			✓	✓

DBMS-specific information

- IBM DB2 for Linux, Unix, and Windows Indexes
- IBM DB2 for OS/390 and z/OS Indexes
- Microsoft SQL Server Indexes
- Oracle Indexes
- Sybase ASE Indexes

IBM DB2 for Linux, Unix, and Windows Indexes

IBM DB2 for Linux, Unix, and Windows offers two types of indexes:

- Unique
- Non-Unique

Unique Indexes guarantee that no two rows of a table have duplicate values in the columns that define the index.

Microsoft SQL Server Indexes

Microsoft SQL Server offers two types of indexes: clustered and non-clustered. Clustered indexes physically sort table data to match their logical order. Non-clustered indexes only order the table data logically. In a database, an index lets you speed queries by setting pointers that allow you to retrieve table data without scanning the entire table. An index can be unique or non-unique.

Microsoft SQL Server creates indexes as B-Trees, which are a series of pointers mapping index pages to their underlying data pages. As tables and, therefore, indexes grow, the number of levels in the B-Tree increases. The B-Tree of a clustered index is shorter than that of a non-clustered index because the leaf level of a clustered index is the data page.

A sound indexing strategy is critical to overall system performance. One pitfall to avoid is placing many indexes on a table without regard for their cumulative cost. Remember that indexes improve read but slow write performance because Microsoft SQL Server must update more information in the system catalog. Consequently, extra indexes can actually slow overall performance if data modification occurs frequently on the table. To determine the efficacy of indexes, you should tune your queries using SHOWPLAN and IO STATISTICS and analyze the selectivity of indexes using DBCC SHOW_STATISTICS.

System indexes and user-defined indexes are handled separately in the Datasource Navigator/Explorer to ensure that system indexes are not accidentally altered or dropped.

Oracle Indexes

Oracle offers two types of indexes. The table below describes these indexes:

Index	Description
Table	A table index is defined on an individual table.
Cluster	A cluster index is defined on a set of tables physically stored together in a cluster. In an Oracle database, both table and cluster indexes use a B-tree structure.

The indexing strategy, particularly with large, active tables, is critical to overall system performance. The optimal definition and number of indexes for a given table is determined by the mix of access paths to that table performing insert, update, delete and select operations. For example, adding or changing an index can speed up your selects but slow your inserts, updates and deletes. Careful tuning and testing helps you achieve the best overall performance.

Tip: Indexes generally improve read operations in a database, but you should not place too many indexes on some tables. Since Oracle must maintain each index along with its referenced table, placing too many indexes on a table that is the object of much insert, update, and delete activity, can actually degrade performance.

Even when an index exists on a table, the way a SQL statement is coded can actually disallow the use of the index. To prevent this from happening, follow these rules of thumb:

- Try not to use SQL statements that include the NOT IN, NOT LIKE, <>, IS NULL operators because they typically suppress the use of indexes.
- When referencing concatenated indexes with queries, be sure the leading column in the index is used. If it isn't, the index won't be used at all.
- Avoid using functions in WHERE predicates.

If you must use functions, investigate the use of function-based indexes.

Index Partitions

Index partitions are similar to table partitions. There are three types of partitioned indexes that Oracle supports:

1 Local prefixed

2 Local nonprefixed

3 Global prefixed

Note: An index cannot be partitioned if it is a cluster index or if the index is defined on a clustered table.

Local prefixed and nonprefixed indexes

A local partitioned index has keys that refer to rows in a single table partition. A local partitioned index is automatically partitioned to mirror the underlying table. The number of partitions or subpartitions and the partition bounds for the partitioned index correspond with the partitions on the table. Oracle maintains this correspondence. If the table partitions are altered, the index partitions are altered accordingly.

A local partitioned index is prefixed if it is partitioned on the same column as the underlying table. The local partitioned index is nonprefixed if it is partitioned on a different column.

Global prefixed indexes

A global partitioned index can refer to rows in more than one table partition or subpartition. Global partitioned indexes are more difficult to manage than local partitioned indexes because any change in the underlying table partition affects all partitions in a global index. As a result, there is increased partition maintenance.

Note: A global index can only be range partitioned but it can be defined on any kind of partitioned table.

IBM DB2 for OS/390 and z/OS Indexes

IBM DB2 for OS/390 and z/OS offers two types of indexes:

- Unique
- Non-Unique

Unique Indexes guarantee that no two rows of a table have duplicate values in the columns that define the index.

Non-Unique indexes let table rows have duplicate values in the columns that define the indexes.

Sybase ASE Indexes

Sybase ASE offers two types of indexes: clustered and non-clustered. Clustered indexes physically sort table data to match their logical order. Non-clustered indexes only order the table data logically. In a database, an index lets you speed queries by setting pointers that let you retrieve table data without scanning the entire table. An index can be unique or non-unique.

Sybase ASE creates indexes as B-Trees, which are a series of pointers mapping index pages to their underlying data pages. As tables and, therefore, indexes grow, the number of levels in the B-Tree increases. The B-Tree of a clustered index is shorter than that of a non-clustered index because the leaf level of a clustered index is the data page.

A sound indexing strategy is critical to overall system performance. One pitfall to avoid is placing many indexes on a table without regard for their cumulative cost. Remember that indexes improve read but slow write performance because Sybase ASE must update more information in the system catalog. Consequently, extra indexes can actually slow overall performance if data modification occurs frequently on the table. To determine the efficacy of indexes, you should tune your queries using SHOWPLAN and IO STATISTICS and analyze the selectivity of indexes using DBCC SHOW_STATISTICS.

System indexes and user-defined indexes are handled separately in the Datasource Navigator/Explorer to ensure that system indexes are not accidentally altered or dropped.

Instance

Note: This object is supported for IBM DB2 and Oracle.

Instance is placed as the first level of information under the Datasource node in the Datasource Navigator/Explorer. Instance includes:

- DB Manager Configuration
- Data sources

Available Functionality

The following functionality is available for this object type:

Databases Wizard (DB2 LUW) Quiesce (Database) Unquiesce

Java Classes

The Java Classes contain compiled Java code. Java Classes are made up of a group of data items, with associated functions that perform operations. The data items are called fields or variables; the functions are referred to as methods.

Tip: Oracle is shipped with a JVM (Java Virtual Machine). The JVM provided by Oracle sits atop the Oracle RDBMS and interacts directly with the RDBMS instead of the operating system.

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Compile	✓
Create Synonym	✓
Drop Java	✓
Load Java	✓
Report	✓

Java Resources

Note: This object is supported by Oracle only.

The Java Resources node offers support for browsing Java resources.

Available Functionality

	ORCL
Drop	✓

Java Sources

Java Sources contain the uncompiled Java source code.

Tip: Oracle is shipped with a JVM (Java Virtual Machine). The JVM provided by Oracle sits atop the Oracle RDBMS and interacts directly with the RDBMS instead of the operating system.

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Compile	✓
Drop	✓
Extract	✓
Load Java	✓
Migrate (DBArtisan)	✓
Rename	✓
Report	✓

Jobs (Oracle)

The following summarizes support for this object type, making up a combination of a schedule a program, and arguments required by the program.

Creating and editing

- Jobs wizard (Oracle) and Jobs Editor (Oracle)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Drop	✓
Enable/Disable (Oracle Jobs)	✓
Extract	✓
Run Job (Oracle Jobs)	✓
Stop Job	✓

Related Information

See details for the following object types

- Chains
- Programs
- Schedules

Jobs (DBArtisan - SQL Server)

Jobs define administrative tasks that can execute on a local server according to schedules or in response to alerts. A job is comprised of job steps that can be executable programs, Windows commands, Transact-SQL statements, Microsoft ActiveX script, or Replication Agents.

Creating and editing

- Jobs Wizard (DBArtisan - SQL Server) and Jobs Editor (DBArtisan - SQL Server)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SQL SVR
Change Job's Owner (DBArtisan)	✓
Drop	✓
Enable/Disable Job (DBArtisan - SQL Agent Job)	✓
Extract	✓
Rename	✓
Start/Stop Job (DBArtisan - SQL Agent Job)	✓
View Job History (DBArtisan)	✓

Job Queues

Job Queues are built-in mechanisms that let you schedule a variety of SQL-based or command-line driven tasks.

Creating and editing

- Oracle Job Queue Wizard (Oracle) and Job Queue Editor (Oracle)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Disable Job	✓
Drop	✓
Enable Job (Job Queue)	✓
Extract	✓
Migrate (DBArtisan)	✓
Report	✓
Run Job (job queues)	✓

Join Indexes

Limited support is offered for these multi-table indexes.

Creating and editing

No support is offered for creation or editing of objects this type.

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	SYB IQ
Drop	
Extract	✓

Libraries

Libraries are an object type that represent a call to an operating system shared library. After the call is made, libraries can be used by SQL or PL/SQL to link to external procedures or functions. Libraries are only to be used on operating systems that support shared libraries and dynamic linking. Libraries serve as pointers or aliases to physical operating system shared library files and do not have existence as a physical object on their own, rather they rely on the physical existence of the files in the external operating system library to which they refer. To access the function or procedures stored in the library, you need execute privileges at the operating system level where the shared library resides.

Creating and editing

- Libraries Wizard (Oracle) and Libraries Editor (Oracle)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Rename	✓
Report	✓

Linked Servers (DBArtisan)

Linked servers let you create links to OLE DB data sources. With a linked server you can execute commands against OLE DB data sources on different servers. Using linked servers you can:

- Access remote servers
- Issue enterprise-wide distributed queries, updates, commands, and transactions.

Distributed Queries

Before you run a distributed query, write the following as the first statement of your SQL script:

```
SET ANSI_NULLS ON

GO

SET ANSI_WARNINGS ON

GO
```

Creating and editing

- Linked Servers Wizard (DBArtisan - SQL Server) and Linked Servers Editor (DBArtisan - SQL Server)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SQL SVR
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Report	✓

Logins

Logins let you access your account. Your login account controls access to the server and all of the databases within it. Only the System Administrator or System Security Officer can create logins. Once you can log into a server, you need additional privileges to access user databases. Specifically, each database owner adds the login as a user or alias to the database.

Creating and editing

- Logins Wizard (SQL Server) and Logins Editor (SQL Server)
- Logins Wizard (Sybase ASE) and Logins Editor (Sybase ASE)

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	SQL SVR	SYB ASE
Add/Modify Login Trigger		✓
Bind To Temporary Database		✓
Change Password	✓	
Create Like	✓	
Drop	✓	✓
Drop Login Trigger		✓
Extract	✓	✓
Migrate (DBArtisan)	✓	✓
Report	✓	✓
Unbind From Temporary Database		✓

Explorer Details

Depending on the DBMS platform, the icon associated with a login differs according to the login's locked or unlocked state, when viewed in the Explorer.

DBMS Platform	Icon	Login Accont State
SQL SVR or SYB ASE		Locked
		Unlocked

DBMS-specific notes

  - Microsoft SQL Server Logins
  - Sybase ASE Logins

Microsoft SQL Server Logins

Logins let you access your account. Your login account controls access to the server and all of the databases within it. Only the System Administrator or System Security Officer can create logins. Once you can log into a server, you need additional privileges to access user databases. Specifically, each database owner adds the login as a user or alias to the database.

Sybase ASE Logins

Logins let you access your account. Your login account controls access to the server and all of the databases within it. Only the System Administrator or System Security Officer can create logins. Once you can log into a server, you need additional privileges to access user databases. Specifically, each database owner adds the login as a user or alias to the database.

Logs (DBArtisan - SQL Server Agent)

The following summarizes support for archived SQL Server Agent logs.

Creating and editing

Logs are generated and cannot be created or edited.

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SQL SVR
Open (DBArtisan - Logs and Error logs)	✓
Report	✓

Materialized Query Tables

A materialized query table is a table whose definition is based on the result of a query. The materialized query table typically contains pre-computed results based on the data existing in the table or tables that its definition is based on. If the SQL compiler determines that a query will run more efficiently against a materialized query table than the base table or tables, the query quickly executes against the materialized query table.

Creating and editing

- Materialized Query Tables Wizard (DB2 LUW) and Materialized Query Tables Editor (IBM DB2 LUW)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW
Compile	✓
Build Query	✓
Create Alias	✓
Create Like	✓
Create View	✓
Describe	✓
Drop	✓
Drop Materialized Query Table	✓
Extract	✓
Lock	✓
Migrate (DBArtisan)	✓
Quiesce Tablespaces (DBArtisan)	✓
Refresh Table	✓
Reorganize /Reorg	✓
Report	✓
Schema (Object Action)	✓
Select * From	✓
Set Integrity	✓
Transfer Ownership	✓
Update Statistics	✓

Materialized Views

Note: This object is supported by Oracle only.

Materialized views are used to dynamically copy data between distributed databases. There are two types of materialized views:

- Complex
- Simple

Complex materialized views copy part of a master table or data from more than one master table. Simple materialized views directly copy a single table. You cannot directly update the underlying data contained in materialized views.

Creating and editing

- Materialized Views Wizard (Oracle) and Materialized Views Editor (Oracle)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Compile	✓
Create Synonym	✓
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Rename	✓
Report	✓
Shrink	✓

Materialized View Logs

Materialized View logs are tables that maintain a history of modifications to the master table, and they are used to refresh simple materialized views. When you create a materialized view log, Oracle automatically creates a log table to track data changes in the master table and a log trigger to maintain the data in the log table.

Creating and editing

- Materialized View Logs Wizard (Oracle) and Materialized View Logs Editor (Oracle)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Report	✓
Shrink	✓

Nodegroups (DBArtisan)

Nodegroups are a named subset of one or more partitions in a database. DBArtisan places Nodegroup under the Storage node in the Datasource Navigator/Explorer.

Creating and editing

- Nodegroups Wizard (DB2 LUW) and Nodegroups Editor (DBArtisan - IBM DB2 LUW)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Redistribute (DBArtisan)	✓
Report	✓

Operators (DBArtisan)

Operators are individuals responsible for the maintenance of one or more computers running Microsoft SQL Server. E-mail, pager, and Net Send can notify these operators.

Creating and editing

- Operators Wizard (DBArtisan - SQL Server) and Operators Editor (DBArtisan - SQL Server)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SQL SVR
Drop	✓
Extract	✓
Rename	✓

Outlines

Outlines are a set of results for the execution plan generation of a particular SQL statement. When you create an outline, plan stability examines the optimization results using the same data used to generate the execution plan. That is, Oracle uses the input to the execution plan to generate an outline, and not the execution plan itself.

Creating and editing

- Outlines Wizard (Oracle) and Outlines Editor (Oracle)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Change Category	✓
Drop	✓
Drop By Category	✓
Drop Unused	✓
Disable Keys	✓
Extract	✓
Migrate (DBArtisan)	✓
Reassign By Category	✓
Rebuild Outlines	✓
Rename	✓
Report	✓

Packages

Note: This object is supported by IBM DB2 for Linux, Unix, and Windows, IBM DB2 for OS/390 and z/OS, and Oracle only.

Packages contain all the information needed to process SQL statements from a single source file. You can use packages to process and call batches of SQL. Depending on the platform, packages can include:

- Procedures
- Functions
- Types
- Variables
- Constants
- Exceptions
- Cursors
- Subprograms

Packages offer a number of important advantages over using standalone procedures and functions, including the ability to:

- Modify package objects without recompiling dependent database objects.
- Declare global variables and cursors that can be shared within the package.
- Grant privileges more efficiently.
- Load multiple package objects into memory at once.

Packages usually have two parts: a header or specification and a body, although sometimes the body is unnecessary. The package header declares the members of the package while the body details the logic underlying each of the package components.

Note: The Datasource Navigator/Explorer splits out package headers and package bodies; however, you create both a package header and body from the packages node.

Creating and editing

- Packages Editor (IBM DB2 LUW)
- Packages Editor (IBM DB2 Z/OS)
- Packages Wizard (Oracle) and Packages Editor (Oracle)

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	DB2 z/OS	ORCL
Bind Package		✓	
Compile			✓
Create Synonym			✓
Drop	✓		✓
Extract			✓
Flush Cache	✓		
Free (Packages)		✓	
Migrate (DBArtisan)			✓
Rebind Packages	✓	✓	
Report	✓	✓	✓

Package Bodies

Package Bodies implement the package specification in that the package body includes the definition of every cursor and subprogram declared in the package specification.

Creating and editing

While Package Bodies are listed as a separate object in the Datasource Navigator/Explorer, they are created on the Packages Editor in conjunction with Packages. For more information, see Package Bodies Editor (Oracle).

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Compile	✓
Create Synonym	✓
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Report	✓

Partition Functions

Partition functions are functions that maps the rows of a table or index into partitions based on the values of a specified column. Partition functions are referenced in partition scheme definitions in partitioning indexes or tables.

Creating and editing

See the following topics:

- Partition Functions Wizard (SQL Server) and Partition Functions Editor (SQL Server)

Available Functionality

	SQL SVR
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Report	✓

For related information, see the following topics:

- Partition Schemes and Partition Schemes Wizard (SQL Server)
- Indexes Wizard (SQL Server)
- Tables Wizard (SQL Server)

Partition Schemes

Partition schemes map the partitions of a partitioned table or index to filegroups. The number and domain of the partitions for a partitioned table or index are specified in a partition function definition. Partition scheme definitions are referenced when partitioning indexes or tables.

Creating and editing

See the following topics:

- Partition Schemes Wizard (SQL Server) and Partition Schemes Editor (SQL Server)

Available Functionality

	SQL SVR
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Next Used Filegroup	✓
Report	✓

For related information, see the following topics:

- Partition Functions and Partition Functions (SQL Server) - Properties
- Indexes Wizard (SQL Server)
- Tables Wizard (SQL Server)

Plans

A Plan is an executable application created in the bind process. It can include one or more packages or debris.

Creating and editing

- Plans Wizard (DB2 Z/OS) and Plans Editor (IBM DB2 Z/OS)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 z/OS
Drop	✓
Free Plan	✓
Rebind Plans	✓
Report	✓

Primary Keys

Note: This object is supported by all platforms.

Primary Keys are a set of table columns that can uniquely identify every row of a table.

Creating and editing

- Primary Keys Wizard (DB2 LUW) and Primary Keys Editor (IBM DB2 LUW)
- Primary Keys Wizard (DB2 Z/OS) and Primary Keys Editor (IBM DB2 Z/OS)
- Indexes, Primary Keys, or Unique Keys wizard (MySQL) and Indexes, Primary Keys, and Unique Keys editors (MySQL)
- Primary Keys Wizard (Oracle) and Primary Keys Editor (Oracle)
- Exclusion Constraints, Primary Keys, and Unique Keys Wizards (PostgreSQL) and Exclusion Constraints, Primary Keys, or Unique Keys Editors (PostgreSQL)
- Primary Keys Wizard (SQL Server) and Primary Keys Editor (SQL Server)
- Primary Keys Wizard (Sybase ASE) and Primary Keys Editor (Sybase ASE)

Note: Creation and editing of objects of this type is not supported against Sybase IQ datasources.

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	DB2 z/OS	ITB/FBD *	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
Allocate Extent					✓				
Analyze					✓				
Change Status (check constraints)					✓				
Deallocate Unused Space					✓				
Disable Index							✓		
Drop	✓	✓	✓	✓	✓	✓	✓	✓	✓
Extract	✓	✓	✓	✓	✓	✓	✓	✓	✓
Migrate (DBArtisan)	✓	✓		✓	✓		✓	✓	
Rebuild Index					✓		✓		
Rename	✓	✓	✓	✓	✓	✓	✓	✓	
Reorganize /Reorg							✓		
Report	✓	✓	✓	✓	✓		✓	✓	✓

Procedures

Procedures are a reusable block of PL/SQL, stored in the database, that applications can call. Procedures streamline code development, debugging, and maintenance by being reusable. Procedures enhance database security by letting you write procedures granting users execution privileges to tables rather than letting them access tables directly.

Creating and editing

- Procedures Wizard (DB2 LUW) and Procedures Editor (IBM DB2 LUW)
- Procedures Wizard (DB2 Z/OS) and Procedures Editor (IBM DB2 Z/OS)
- Procedures Wizard (Oracle) and Procedures Editor (Oracle)
- Procedures Wizard (SQL Server) and Procedures Editor (SQL Server)
- Procedures Wizard (Sybase ASE) and Procedures Editor (Sybase ASE)

Note: Creation and editing of objects of this type is not supported against Sybase IQ datasources.

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	DB2 z/OS	ITB/FBD *	ORCL	SQL SVR	SYB ASE	SYB IQ
Build		✓					
Compile	✓			✓			
Create Synonym				✓			
Describe	✓	✓		✓	✓	✓	
Drop	✓	✓	✓	✓	✓	✓	✓
Execute			✓	✓		✓	
Extract	✓	✓	✓	✓	✓	✓	✓
Hide Text						✓	
Migrate (DBArtisan)	✓	✓		✓	✓	✓	
Object Properties							✓
Rename	✓	✓	✓	✓	✓	✓	
Report	✓	✓	✓	✓	✓	✓	
Transfer Ownership	✓						

For more specific information, see the following topics:

- DB2 z/OS Procedures
- Sybase ASE Procedures
- Executing Statements Before and After Procedure or Function Creation

DB2 z/OS Procedures

Only IBM DB2 for OS/390 and z/OS SQL stored procedures created by DBArtisan, or IBM's Stored Procedure Builder can be retrieved by DBArtisan.

Sybase ASE Procedures

Procedures perform procedural logic in your Sybase ASE applications. They are batches of SQL statements that are compiled and stored in the system catalog. Procedures execute faster than embedded SQL statements because they are pre-compiled and have execution plans for use by the optimizer. When you create a procedure, Sybase ASE builds a query tree, which it stores in a system table. When you execute a procedure for the first time, Sybase ASE loads it from the system table, compiles, and optimizes it. Sybase ASE places the resulting query plan in the procedure cache where it remains on a most recently used basis. In addition to better performance, procedures yield other benefits, including easier code maintenance, additional security and reduced network traffic.

Note: Extended procedures are also supported. For details, see Extended Procedures.

Executing Statements Before and After Procedure or Function Creation

With respect to functions and stored procedures, it can be necessary to have statements executed before and after creation of the procedure or function. This can be useful for example, if you need to create or drop temporary tables used by the function or procedure. Two tag pairs, ETStart and ETEnd, let you embed statements in the first comment block of a stored procedure or function. The following shows the expected syntax:

```
create procedure dbo.procname(@a numeric) as

/*

<ETStart>SQL Statement</ETStart>

<ETEnd>SQL Statement</ETEnd>

*/

begin

...
```

Profiles

Profiles are a mechanism for allocating system and database resources to users. Profiles let you specify limits on:

- Number of sessions
- CPU time
- Connect time
- Idle time
- Logical reads and space in the SGA's shared pool

You can assign profiles to one or more users. The database's default profile and all of its resource limits are assigned to users without a specific profile assignment.

Creating and editing

- Profiles Wizard (Oracle) and Profiles Editor (Oracle)

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Drop	✓
Extract	
Migrate (DBArtisan)	✓
Report	✓

Programs

The following summarizes support for this object type, making up a collection of metadata about a particular task.

Creating and editing

- Programs wizard (Oracle) and Programs Editor (Oracle)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Drop	✓
Enable/Disable (Oracle Jobs)	✓
Extract	✓

Related Information

See details for the following object types

- Chains
- Jobs (Oracle)
- Schedules

Recycle Bin

DBArtisan provides basic support for Oracle's Recycle Bin feature. When enabled, dropped tables, indexes, and associated objects are stored in the Recycle Bin until explicitly purged.

Creating and editing

The Recycle Bin cannot be created or edited.

DBMS platform availability and object actions/operations supported

On the Datasource Navigator/Explorer, selecting the **Recycle Bin** node has the same effect as issuing a SHOW RECYCLEBIN statement. The contents of the Recycle Bin are displayed.

Related actions include:

	ORCL
Enable Recycle Bin	✓
Flashback Recycle Bin Entry	✓
Purge Recycle Bin	✓
Purge Recycle Bin Entry	✓

Note:Select * From functionality is supported as a right-click option for individual tables in the Recycle Bin. For details on this feature, see Select * From.

Redo Log Groups (DBArtisan)

Redo Logs save information on the state of the database after an update. If the database fails before the update is written to the datafiles, the update can be re-done, bringing the changes back after a crash, so it appears the update did happen.

Tip: Have sufficient redo logs available so that other Oracle threads do not have to wait until one is free before performing an update.

Redo logs are arranged in groups, and when a particular group is being written to, each member receives the same data.

Tip: To avoid crashes, have two logfiles in a group, each one being on a different (physical) disk.

Creating and editing

- Redo Log Groups Wizard (DBArtisan - Oracle) and Redo Log Groups Editor (Oracle)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Report	✓
Switch Logfile (DBArtisan)	✓
Take Checkpoint (DBArtisan)	✓

Remote Servers (DBArtisan)

Remote Servers are servers on a network that you can access through your local server. Instead of logging directly into a remote server, you can log into your local server and execute a remote procedure call to it.

The local server routes all commands to the remote server. To communicate with a remote server, you log into a local server and submit your remote procedure calls for remote servers. The remote server receives remote procedure calls from the local server and processes them accordingly.

Creating and editing

- Remote Servers Wizard (DBArtisan - SQL Server) and Remote Servers Editor (DBArtisan - SQL Server)
- Remote Servers Wizard (DBArtisan - Sybase ASE) and Remote Servers Editor (DBArtisan - Sybase ASE)

Note: Against Sybase IQ datasources, DBArtisan does not offer support for creating or editing objects of this type.

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SQL SVR	SYB ASE	SYB IQ
Drop	✓	✓	✓
Extract	✓	✓	✓
Migrate (DBArtisan)	✓	✓	
Report	✓	✓	

Roles

Roles are sets of user privileges you associate with access to objects within a database. Roles streamline the process of granting permissions. You can use roles to grant sets of permissions and privileges to users and groups.

Creating and editing

- Roles Wizard (SQL Server) and Roles Editor (SQL Server)
- Roles Wizard (Oracle) and Roles Editor (Oracle)
- Roles Wizard (PostgreSQL) and Roles Editor (PostgreSQL)
- Role Wizard (DBArtisan - Sybase ASE) and Roles Editor (DBArtisan - Sybase ASE)

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL	ITB/FBD *	PSTGRS *	SQL SVR	SYB ASE
Create Like			✓		
Drop	✓	✓	✓	✓	✓
Extract	✓	✓	✓	✓	✓
Migrate (DBArtisan)	✓			✓	✓
Rename			✓		
Report	✓	✓		✓	✓
Role Activation	✓				✓

For related information

- Microsoft SQL Server Roles
- Oracle Roles
- Sybase ASE Roles (DBArtisan)
- Server Roles (DBArtisan)

Microsoft SQL Server Roles

Roles let you manage permissions for users to perform a set of activities based on job functions. Roles are a mechanism for streamlining the assignment of object and system privileges to multiple users. Instead of granting sets of privileges to individual users, you can create a role, grant system and object privileges to that role, and then grant that role to all the users who should share the same privileges. You can grant one or more roles to the same user. Therefore, if you need to change the permissions for a certain role you can do so, and not have to grant or revoke the permissions for each individual user.

For related information, see Server Roles (DBArtisan).

Oracle Roles

Roles are a mechanism for streamlining the assignment of object and system privileges to multiple users. Instead of granting sets of privileges to individual users, you can create a role, grant system and object privileges to that role, then simply grant that role to all the users who should share the same privileges. You can grant one or more roles to the same user.

DBMS-specific functionality

- For information on activating and deactivating roles for the current login in the current session, see Role Activation.

Sybase ASE Roles (DBArtisan)

Roles are a mechanism for streamlining the assignment of object and system privileges to multiple users. Instead of granting sets of privileges to individual users, you can create a role, grant system and object privileges to that role, then simply grant that role to all the users who should share the same privileges. You can grant one or more roles to the same user.

DBMS-specific functionality

- For information on activating and deactivating roles for the current login in the current session, see Role Activation.

Rollback Segments (DBArtisan)

Rollback segments manage all transactions in your Oracle databases. They maintain read consistency among concurrent users in a database and can rollback transactions. They are transaction logs that maintain discrete records of changes to data. By maintaining a history of data changes, rollback segments can rollback uncommitted transactions so that data is rolled back to their prior state.

Tip: Databases with large user bases and intensive data manipulation characteristics can benefit from having multiple rollback tablespaces that are placed on different server drives or file systems. This strategy aids in the reduction of I/O and rollback segment contention.

Because all changes are logged to a rollback segment, they can become a performance bottleneck as the number of concurrent users and transaction activity increases. You can tune rollback segment performance by creating additional ones to handle increased traffic. To determine whether rollback segment performance is satisfactory, you should monitor the ratio of gets to waits. If you see that the number of waits is climbing, create another rollback segment to further spread concurrency.

Note: The analysis for determining when to add a rollback segment can be quite complicated. As a rule-of-thumb, however, many DBAs simply plan on allocating a rollback segment for every four concurrent users.

The proper sizing of rollback segments is critical to their overall performance. Performance degrades whenever a rollback segment must extend, wrap or shrink in response to transaction loads. Ideally, you want to make their extents as small as possible while ensuring that each transaction fits into a single extent, which in practice is very difficult to achieve.

Note: You should carefully monitor each rollback segments for the occurrence of extends, wraps and shrinks. If you see a high level of extends and wraps, then the rollback segment is probably too small. If you see a high level of shrinks, then you should increase the rollback segment's optimal size.

Creating and editing

- Rollback Segments Wizard (Oracle) and Rollback Segments Editor (Oracle)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Report	✓
Set Online/Offline	✓
Shrink	✓

Rules

Rules promote data integrity by allowing you to validate the values supplied to a table column. They are reusable objects that you can bind to table columns or user datatypes. For example, you can create a rule, bind it to a column in a table and have it specify acceptable values that can be inserted into that column.

Creating and editing

- Rules Wizard (PostgreSQL) and Rules Editor (PostgreSQL)
- Rules Wizard (SQL Server) and Rules Editor (SQL Server)
- Rules Wizard (Sybase ASE) and Rules Editor (Sybase ASE)

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	PSTGRS *	SQL SVR	SYB ASE
Create Like	✓		
Drop	✓	✓	✓
Extract	✓	✓	✓
Hide Text			✓
Migrate (DBArtisan)		✓	✓
Rename		✓	✓
Rename	✓		
Report		✓	✓

Schedules

The following summarizes support for the object type that implements a schedule defining when a job should be run or when a window should open.

Creating and editing

- Schedules Wizard (Oracle) and Schedules Editor (Oracle)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Drop	✓
Extract	✓

Related Information

See details for the following object types

- Chains
- Jobs (Oracle)
- Programs

Schema

A Schema is a container of objects that can be owned by any user.

Note: The **Schema** nodes available under DB2 LUW, DB2 z/OS, and Oracle datasources do not own a specific set of objects. They are simply an organizational container for all database objects for a datasource or database.

Creating and editing

- Schemas Wizard (PostgreSQL) and Schemas Editor (PostgreSQL)
- Schema Wizard (SQL Server) and Schemas Editor (SQL Server)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	PSTGRS *	SQL SVR
Change Owner	✓	
Drop	✓	✓
Extract	✓	✓
Migrate (DBArtisan)		✓
Rename	✓	
Report		✓

Segments

Segments are a mechanism for placing tables and indexes on specific logical partitions. You create segments on one or more fragments of a database. You can map segments to specific database fragments, which in turn reside on specific hard disks; and, mapping segments lets you increase i/o throughput by placing intensively used tables and indexes on different physical devices. You can allocate tables and indexes to segments by including placement statements at the end of CREATE TABLE or CREATE INDEX statements.

Creating and editing

- Segment Wizard (Sybase ASE) and Segments Editor (Sybase ASE)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SYB ASE
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Report	✓

Sequences

Sequences are programmable database objects that generate a definable sequence of values.

Creating and editing

- Sequences Wizard (DB2 LUW) and Sequences Editor (IBM DB2 LUW)
- Sequences Wizard (Oracle) and Sequences Editor (Oracle)
- Sequences Wizard (SQL Server) and Sequences Editor (SQL Server)

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	**ORCL**	**SQL SVR**
Create Alias	✓		
Create Synonym		✓	
Drop	✓	✓	✓
Extract	✓	✓	✓
Migrate (DBArtisan)	✓	✓	✓
Rename	✓	✓	✓
Report	✓	✓	✓
Restart Sequence(s)			✓
Run Job (job queues)	✓		
Transfer Ownership	✓		

Server Roles (DBArtisan)

You can manage the login members for each of SQL Server's fixed server roles: bulkadmin, dbcreator, diskadmin, processadmin, securityadmin, serveradmin, setupadmin, and sysadmin.

Platform availability

DB2 LUW	DB2 z/OS	ITB/FBD *	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ	TERA
						✓			

Creating and editing

- Server Roles Editor (DBArtisan - SQL Server)

Note: Server roles are fixed and cannot be created.

Server Triggers (DBArtisan)

A server trigger implements a server-scope DDL trigger. These triggers fire in response to events associated with DDL statements or specifically to logons.

Note: Before working with server triggers, consult Microsoft SQL Server documentation for a general understanding of DDL triggers. For more information, see Accessing Third Party Documentation.

Creating and editing

- Server Triggers Wizard (DBArtisan - SQL Server) and Server Triggers Editor (DBArtisan - SQL Server)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SQL SVR
Change Status	✓
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Report	✓

Related information

- Database Triggers

Stogroups (DBArtisan)

Stogroups are storage groups, named sets of volumes on which DB2 for OS/390 data is stored.

When you select the Stogroups node of the Datasource Navigator/Explorer, DBArtisan lists pertinent information about each of the.

Creating and editing

See the following topics:

- Stogroups Wizard (DB2 Z/OS) and Stogroups Editor (IBM DB2 Z/OS)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 z/OS
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Report	✓

Structured Types

Structured types define an abstract data type or object composed of a collection of similar types of data. For example, create an structured type that defines a full address rather than the pieces of an address, such as city, state and postal code. An structured type stores the pieces of an address in a single type, storing them in the same location and allowing the full address to be accessed and manipulated as single unit rather than multiple units.

Structured types are useful for ensuring uniformity and consistency as they are defined as single encapsulated entity that can be reused in other structured types and objects. They also offer flexibility by allowing for the creation of objects that represent real-world situations which is limited in relational objects.

Creating and editing

- Structured Types Wizard (DB2 LUW) and Structured Types Editor (IBM DB2 LUW)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Report	✓
Transfer Ownership	✓

Symmetric Keys

Symmetric keys, in addition to asymmetric keys and certificates, provide support for Public Key Encryption. Symmetric keys consist of a private key and corresponding public key and are used to secure symmetric keys.

Creating and editing

- Symmetric Keys Wizard (SQL Server) and Symmetric Key Editor (SQL Server)

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	SQL SVR
Drop	✓
Extract	✓
Report	✓

Related information

- Certificates
- Asymmetric Keys

Synonyms

The function and usage of synonyms differs by DBMS platform

DB2 z/OS	An alternate name for an existing table, view, or alias.
Oracle	An alternative name for a table, view, sequence, procedure, function, package, materialized view, java class, type, or another synonym.
SQL Server	A single-part name used to reference multi-part names in SQL statements. While synonyms can be created for functions, procedures, tables, and views, consult Microsoft SQL Server documentation for restrictions on function and procedure support. For more information, see Accessing Third Party Documentation.

Creating and editing

- Synonyms Wizard (DB2 Z/OS) and Synonyms Editor (IBM DB2 Z/OS)
- Synonyms Wizard (Oracle) and Synonyms Editor (Oracle)
- Synonyms Wizard (SQL Server) and Synonyms Editor (SQL Server)

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 z/OS	ORCL	SQL SVR
Create Synonym		✓	
Describe		✓	
Drop	✓	✓	✓
Extract	✓	✓	✓
Migrate (DBArtisan)	✓	✓	✓
Rename		✓	✓
Report	✓	✓	✓

System Indexes

On some DBMS platforms, system indexes are treated separately from standard indexes.

Platform availability

DB2 LUW	DB2 z/OS	ITB/FBD *	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
						✓	✓	✓

Creating, Editing, and Object Actions Supported

Against SQL Server and Sybase ASE datasources, functionality offered for system indexes is identical to that offered for standard indexes. For details, see Indexes.

Against Sybase IQ datasources, no support is offered for creation or editing against this object type. The followng object actions are available:

- Drop
- Extract

System Tables

On some DBMS platforms, system tables are listed separately from standard tables.

Platform availability

DB2 LUW	DB2 z/OS	ITB/FBD *	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
						✓	✓	✓

Creating, Editing, and Object Actions Supported

Against SQL Server and Sybase ASE datasources, functionality offered for system indexes is identical to that offered for standard indexes. For details, see Indexes.

Against Sybase IQ datasources, no support is offered for creation or editing against this object type. The followng object actions are available:

- Drop
- Extract
- Object Properties
- Select * From

System Triggers

Against Sybase IQ datasources, system triggers are listed separately from standard triggers.

Platform availability

DB2 LUW	DB2 z/OS	ITB/FBD *	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
								✓

Creating, Editing, and Object Actions Supported

No support is offered for creation, editing, or object actions against this object type.

Tables

Tables are a the basic unit of data storage. Tables store all the data accessible to users in rows and columns. Each column has a name, datatype and other associated properties. After you define a table, users can insert valid data into the table, which you can later query, update and delete.

Note: System tables are treated separately from user-defined tables in the Navigator/Explorer to ensure that system tables are not accidentally altered or dropped.

Creating and editing

- Tables Wizard (DB2 LUW) and Tables Editor (IBM DB2 LUW)
- Tables Wizard (DB2 Z/OS) and Tables Editor (IBM DB2 Z/OS)
- Tables wizard (MySQL) and Tables editor (MySQL)
- Tables Wizard (Oracle) and Tables Editor (Oracle)
- Tables Wizard (PostgreSQL) and Tables Editor (PostgreSQL)
- Tables Wizard (SQL Server) and Tables Editor (SQL Server)
- Tables Wizard (Sybase ASE) and Tables Editor (Sybase ASE)

Note: Creation and editing of objects of this type is not supported against Sybase IQ datasources.

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	DB2 z/OS	ITB/FBD *	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
Allocate Extent					✓				
Analyze					✓				
Analyze Tables				✓					
Build Query	✓	✓	✓	✓	✓	✓	✓	✓	
Change Access Status					✓				
Change Owner						✓			
Change Schema						✓			
Check Tables				✓					
Checksum Tables				✓					
Convert Tables				✓					
Create Alias	✓	✓							
Create Clone		✓							
Create Insert Statements	✓	✓	✓	✓	✓	✓	✓	✓	
Create Like	✓	✓	✓	✓	✓	✓	✓	✓	
Create Synonym					✓				
Create View	✓	✓	✓		✓		✓	✓	
DBCC							✓	✓	
Deallocate Unused Space					✓				
Delete Statistics								✓	
Describe	✓	✓		✓	✓		✓	✓	
Disable Keys				✓					
Disable/Enable Triggers							✓	✓	
Drop	✓	✓	✓		✓	✓	✓	✓	✓
Drop Clone		✓							
Drop Unused					✓				

Edit Data	✓	✓	✓	✓	✓	✓	✓	✓	
Enable Keys				✓					
Enable/Disable Filetable(s)							✓		
Estimate Size					✓			✓	
Exchange Data With Clone		✓							
Extract	✓	✓	✓		✓	✓	✓	✓	✓
Extract Data as XML					✓		✓	✓	
Flashback Table					✓				
Flush Tables				✓					
Hide Text								✓	
Import Data From File	✓	✓	✓	✓	✓		✓	✓	
Lock	✓	✓							
Migrate (DBArtisan)	✓	✓		✓	✓		✓	✓	
Move Table	✓								
Object Properties									✓
Optimize Tables				✓					
Place								✓	
Quiesce Tablespaces (DBArtisan)	✓								
Rebuild Table				✓					
Recompile							✓	✓	
Rename	✓	✓		✓	✓	✓	✓	✓	
Reorganize /Reorg	✓				✓			✓	
Repair Tables				✓					
Report	✓	✓	✓		✓		✓	✓	
Runstats Tablespace (DBArtisan)		✓							

Schema (Object Action)	✓	✓	✓	✓	✓		✓	✓	
Select * From	✓	✓	✓	✓	✓	✓	✓	✓	✓
Set Integrity	✓								
Shrink					✓				
Transfer Ownership	✓								
Truncate	✓	✓		✓	✓		✓	✓	
Update Statistics	✓						✓	✓	

DBMS platform-specific notes

MySQL	MySQL servers can store tables in multiple formats, including MyISAM and InnoDB. MyISAM tables (ISAM is the acronym for indexed sequential access method) are used most often for read operations. The read operation is very fast, but you cannot include any referential integrity, such as a foreign key. Also, MyISAM tables only issue table-level locks. *InnoDB tables, on the other hand, do permit transactions and foreign key constraints. InnoDB tables also lock data at the row level, which is appropriate for high transaction tables. Additional table types available are MERGE, MEMORY, FEDERATED, and ARCHIVE among others.For a complete discussion of table types, go to the MySQL documentation of table and engine types.* For more information, see Accessing Third Party Documentation.

Tablespaces

Tablespaces are storage structures that act as partitions for the database. You can create a tablespace to store table data and other objects related to table performance such as indexes or large object data. Tablespaces are used to manage large complex databases. Once you have created a tablespace, you can place objects on it.

Tip: Create separate tablespaces for your tables and indexes and put each tablespace on a different drive or file system. Segmenting tables and their corresponding indexes in this manner helps eliminate I/O contention at the server level.

Note: IBM DB2 for Linux, Unix, and Windows lets you assign a location for table or index data directly to physical storage devices. Each tablespace can also be broken down into a collection of containers which are the actual physical storage files or devices. You can then spread the data or database objects across multiple file systems, thereby giving you the necessary space for objects that require it.

Once you have created a tablespace, you can place individual tables and indexes on it. Because tablespaces map to physical drives, you can improve i/o performance by placing tables and their indexes on physically separated table spaces.

Creating and editing

- Tablespaces Wizard (DB2 LUW) and Tablespaces Editor (IBM DB2 LUW)
- Tablespaces Wizard (DB2 Z/OS) and Tablespaces Editor (IBM DB2 Z/OS)
- Tablespaces Wizard (Oracle) and Tablespaces Editor (Oracle)
- Tablespaces Wizard (PostgreSQL) and Tablespaces Editor (PostgreSQL)

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	DB2 z/OS	ORCL	PSTGRS *
Change Owner				✓
Change Status			✓	
Check Data (DBArtisan)		✓		
Check Index (DBArtisan)		✓		
Coalesce			✓	
Copy to Copy (DBArtisan)		✓		
Drop	✓	✓	✓	✓
Extract	✓	✓	✓	✓
Lower High Water Mark	✓			
Merge Copy (DBArtisan)		✓		
Migrate (DBArtisan)	✓	✓	✓	
Modify Recovery (DBArtisan)		✓		
Modify Statistics (DBArtisan)		✓		
Quiesce Tablespaces (DBArtisan)		✓		
Rebalance	✓			
Rebuild Index		✓		
Recover (DBArtisan)		✓		
Rename	✓		✓	✓
Reorganize /Reorg		✓		
Report	✓	✓	✓	
Runstats Index (DBArtisan)		✓		
Runstats Tablespace (DBArtisan)		✓		
Set Default			✓	
Set UNDO			✓	
Start Database		✓		

Stop Database		✓
Switch Online	✓	
Transfer Ownership	✓	

Triggers

Triggers are a special type of procedure that automatically fire when defined data modification operations (insert, update, or delete) occur on a target table or view. Trigger support differs by DBMS type, but in general offers options such as:

- Standard INSERT/DELETE/UPDATE event support
- BEFORE/AFTER timing control
- Row/statement trigger type control
- INSTEAD OF triggers
- Compound triggers
- Enabled/disabled status options

Creating and editing

- Triggers Wizard (DB2 LUW) and Triggers Editor (IBM DB2 LUW)
- Triggers Wizard (DB2 Z/OS) and Triggers Editor (IBM DB2 Z/OS)
- Triggers Wizard (Oracle) and Triggers Editor (Oracle)
- Triggers Wizard (PostgreSQL) and Triggers Editor (PostgreSQL)
- Triggers Wizard (SQL Server) and Triggers Editor (SQL Server)
- Triggers Wizard (Sybase ASE) and Triggers Editor (Sybase ASE)

Note: Creation and editing of objects of this type is not supported against Sybase IQ datasources.

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	DB2 z/OS	ITB/FBD *	ORCL	PSTGRS *	SQL SVR	SYB ASE
Change Status			✓			✓	✓
Compile				✓			
Drop	✓	✓	✓	✓	✓	✓	✓
Extract	✓	✓	✓	✓	✓	✓	✓
Hide Text							✓
Migrate (DBArtisan)	✓	✓		✓		✓	✓
Report							
Rename	✓	✓	✓	✓	✓	✓	✓
Transfer Ownership	✓						

Types

Types define an abstract data type or object composed of a collection of similar types of data. For example, create an object type that defines a full address rather than the pieces of an address, such as city, state and postal code. An object type stores the pieces of an address in a single type, storing them in the same location and allowing the full address to be accessed and manipulated as single unit rather than multiple units.

Object types are useful for ensuring uniformity and consistency as they are defined as single encapsulated entity that can be reused in other object types and objects. They also offer flexibility by allowing for the creation of objects that represent real-world situations which is limited in relational objects.

You can choose to create a type that is incomplete, complete, a VARRAY, or a nested table or any combination of the above. An incomplete type specifies no attributes and can be used for circular references such as person - female. It lets the type be referenced before it is complete. The VARRAY type can be used to store small sets of related data. For example, if you have ten offices (each one with a different description) at a particular division in your company, you could create a VARRAY of 10 to hold the details of these offices. The values for a VARRAY type must be fixed and known and small values as they are stored in RAW format. A nested table type can be used when data is repeated for the same entity an unknown number of times and storage is a concern.

Creating and editing

- Object Types Wizard (Oracle) and Types Editor (Oracle)

Note: Objects of this type cannot be created or edited against PostgreSQL datasources.

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL	**PSTGRS ***
Change Owner		✓
Change Schema		✓
Compile	✓	
Drop	✓	✓
Extract	✓	✓
Migrate (DBArtisan)	✓	
Rename		✓
Report	✓	

Type Bodies

Type Bodies implement object type specification by containing the definition of every cursor and subprogram declared in the object type specification. While Type Bodies are listed as a separate object in the Datasource Navigator/Explorer, they are created on the Types editor in conjunction with Types.

Creating and editing

While listed as separate objects in the Datasource Navigator/Explorer, type bodies are created and edited on the **Body** tab of the Types Editor. For details, see Types Editor (Oracle).

DBMS platform availability and object actions/operations supported

The table below lists the platforms on which this object type is available. It also lists the object actions available against objects of this type For an introduction to object actions and details on usage of specific actions, see Object actions.

	ORCL
Compile	✓
Drop	✓
Extract	✓
Migrate (DBArtisan)	✓
Report	✓

Unique Keys

Unique keys can enforce logical keys that are not chosen as the primary key. They enforce uniqueness for specified columns in a table.

Creating and editing

- Unique Keys Wizard (DB2 LUW) and Unique Keys Editor (IBM DB2 LUW)
- Unique Keys Wizard (DB2 Z/OS) and Unique Keys Editor (IBM DB2 Z/OS)
- Indexes, Primary Keys, or Unique Keys wizard (MySQL) and Indexes, Primary Keys, and Unique Keys editors (MySQL)
- Unique Keys Wizard (Oracle) and Unique Keys Editor (Oracle)
- Exclusion Constraints, Primary Keys, and Unique Keys Wizards (PostgreSQL) and Exclusion Constraints, Primary Keys, or Unique Keys Editors (PostgreSQL)
- Unique Keys Wizard (SQL Server) and Unique Keys Editor (SQL Server)
- Unique Keys Wizard (Sybase ASE) and Unique Keys Editor (Sybase ASE)

Note: Creation and editing of objects of this type is not supported against Sybase IQ datasources.

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	DB2 z/OS	ITB/FBD *	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
Allocate Extent					✓				
Analyze					✓				
Change Status					✓				
Deallocate Unused Space					✓				
Disable Index							✓		
Drop	✓	✓	✓	✓	✓	✓	✓	✓	✓
Extract	✓	✓	✓	✓	✓	✓	✓	✓	✓
Migrate (DBArtisan)	✓	✓		✓	✓		✓	✓	
Rebuild Index					✓		✓		
Rename	✓	✓	✓		✓	✓	✓	✓	
Reorganize /Reorg							✓		
Report	✓	✓	✓	✓	✓		✓	✓	
Transfer Ownership	✓								

User Datatypes

User-defined datatypes promote domain consistency by streamlining the definition of commonly used table columns in a database. You can build a customized datatype from system datatypes and bind defaults and rules to it to enhance integrity. When you reference the user datatype in a column, the column assumes all of the properties of the user datatype.

Creating and editing

- User Datatypes (DB2 LUW) and User Datatypes Editor (IBM DB2 LUW)
- User Datatypes Wizard (DB2 Z/OS) and User Datatypes Editor (IBM DB2 Z/OS)
- User Datatypes Wizard (SQL Server) and User Datatypes Editor (SQL Server)
- User Datatypes Wizard (Sybase ASE) and User Datatypes Editor (Sybase ASE)

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	**DB2 z/OS**	**SQL SVR**	**SYB ASE**
Drop	✓	✓	✓	✓
Extract	✓	✓	✓	✓
Migrate (DBArtisan)	✓	✓	✓	✓
Rename			✓	✓
Report	✓	✓	✓	✓
Transfer Ownership	✓			

User Messages

User Messages lets you catalog error messages that your database applications can re-use. Microsoft SQL Server stores your error messages in a system table, sysmessages. To return error messages from stored procedures and triggers, you need to call a system stored procedure and pass an error number as a parameter.

Creating and editing

- User Messages Wizard (SQL Server) and User Messages Editor (SQL Server)
- User Messages Wizard (Sybase ASE) and User Messages Editor (Sybase ASE)

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	SQL SVR	SYB ASE
Drop	✓	✓
Extract	✓	✓
Migrate (DBArtisan)	✓	✓
Report	✓	✓

Users

A user is an individual with access to the DBMS.

Creating and editing

- Users Wizard (DB2 LUW) and Users Editor (IBM DB2 LUW)
- Users Wizard (DB2 Z/OS) and Users Editor (IBM DB2 Z/OS)
- Users wizard (MySQL) and Users editor (MySQL)
- Users Wizard (Oracle) and Users Editor (Oracle)
- Users Wizard (SQL Server) and Users Editor (SQL Server)
- Users Wizard (Sybase ASE) and Users Editor (Sybase ASE)

Note: Creation and editing of objects of this type is not supported against Sybase IQ datasources.

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	DB2 z/OS	ITB/FBD *	MySQL	ORCL	SQL SVR	SYB ASE	SYB ASE
Analyze					✓			
Change Password		✓		✓	✓			
Compile					✓			
Create Like	✓	✓		✓	✓	✓	✓	
Drop	✓	✓		✓	✓	✓	✓	✓
Extract	✓	✓		✓	✓	✓	✓	✓
Hide Text							✓	
Migrate (DBArtisan)	✓	✓		✓	✓	✓	✓	
Report	✓	✓		✓	✓	✓	✓	
Transfer Ownership	✓							

Explorer Details

Depending on the DBMS platform, the icon associated with a user differs according to the account's locked or unlocked state, when viewed in the Explorer.

DBMS Platform	Icon	User Accont State
ORCL or SYB IQ	👤🔒	Locked
	👤	Unlocked

Related information

- IBM DB2 for Linux, Unix, and Windows Users (DBArtisan)
- IBM DB2 for OS/390 and z/OS Users (DBArtisan)
- Microsoft SQL Server Users
- MySQL Users
- Oracle Users
- Sybase ASE Users

IBM DB2 for Linux, Unix, and Windows Users (DBArtisan)

IBM DB2 for Linux, Unix, and Windows relies on your machine's operating system to define users for the server. You create and maintain users through your operating system.

Note: DBArtisan lets you set a default SQLID user for a DB2 data source. This lets the user set the implicit schema for the unqualified object references to a schema different from the user's login id.

IBM DB2 for OS/390 and z/OS Users (DBArtisan)

IBM DB2 relies on the operating system you are running to define users for the server. This means that users are created and maintained through your operating system.

Note: DBArtisan lets you set a default SCHEMA user for a DB2 data source. This lets the user set the implicit schema for the unqualified object references to a schema different from the user's login id.

Microsoft SQL Server Users

Microsoft SQL Server controls access at the database level by requiring the System Administrator or Database Owner to add a login as a database user or alias. After you create a database user, you can implement further security by the granting or revoking the privileges for that user on specific database objects. To consolidate the process of granting or revoking permissions to many users, the database owner can assign users to groups.

MySQL Users

Unlike adding new users to other platforms, MySQL doesn’t rely solely on a userid/password combination. The MySQL server must not only validate the user and password, but also the host from which the connection is being requested before the new user is admitted to the server’s inner sanctum. The User Wizard and User Editor enable these validations.

Oracle Users

To access an Oracle database, you need a user account authenticated with a password. A user account is what Oracle uses to permit access by the user. You can assign the following optional properties to the user:

- Default tablespace
- Temporary tablespace
- Quotas for allocating space in tablespaces
- Profile containing resource limits

Sybase ASE Users

Sybase ASE controls access at the database level by requiring the System Administrator or Database Owner to add a login as a database user or alias. After you create a database user, you can implement further security by granting or revoking the privileges for that user on specific database objects. To consolidate the process of granting or revoking permissions to many users, the database owner can assign users to groups.

Views

Views are SQL queries stored in the system catalog that customize the display of data contained in one or more tables. Views behave like tables because you can query views and perform data manipulation operations on them. However, views do not actually store any data. Instead, they depend on data contained in their base tables. Views let you:

- View a customized selection of data from one or more tables. As a result, you can display data more cogently to different sets of users, even though the underlying data is the same.
- Restricting access to a defined set of rows and columns.

Creating and editing

- Views Wizard (DB2 LUW) and Views Editor (IBM DB2 LUW)
- Views Wizard (DB2 Z/OS) and Views Editor (IBM DB2 Z/OS)
- Views Wizard (Oracle) and Views Editor (Oracle)
- Views Wizard (PostgreSQL) and Views Editor (PostgreSQL)
- Views Wizard (SQL Server) and Views Editor (SQL Server)
- Views Wizard (Sybase ASE) and Views Editor (Sybase ASE)

Note: Creation and editing of objects of this type is not supported against Sybase IQ datasources.

DBMS platform availability and object actions/operations supported

The following table lists object actions available for this object type. For an introduction to object actions and details on usage of specific actions, see Object actions.

	DB2 LUW	DB2 z/OS	ITB/FBD *	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ	SYB IQ
Build Query	✓	✓	✓	✓		✓	✓		✓
Change Owner					✓				
Change Schema					✓				
Compile				✓					
Create Alias	✓	✓							
Create Synonym		✓		✓					
Describe	✓			✓	✓	✓	✓		
Drop	✓	✓	✓	✓	✓	✓	✓	✓	✓
Extract	✓	✓	✓	✓	✓	✓	✓	✓	✓
Hide Text							✓		
Migrate (DBArtisan)	✓	✓		✓		✓	✓		
Object Properties								✓	✓
Refresh Materialized View					✓				
Rename				✓	✓	✓	✓		✓
Report	✓	✓	✓	✓		✓	✓		
Schema (Object Action)	✓	✓	✓	✓		✓	✓		✓
Select * From	✓	✓	✓	✓	✓	✓	✓	✓	✓
Transfer Ownership	✓								
Update Statistics	✓								

Creating objects

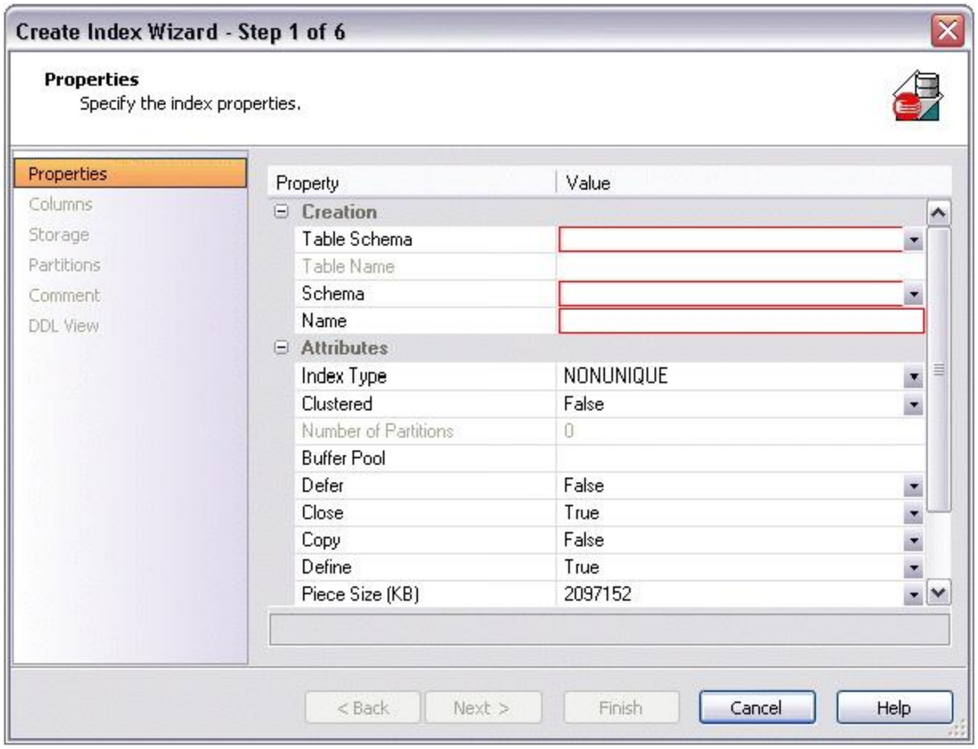
A set of easy-to-use wizards are available for creating objects. For an introduction to object creation wizards, see Overview and Common Usage of Object Wizards.

Platform-by-platform discussions of wizards are discussed as follows:

- IBM DB2 for Linux, Unix, and Windows Object Wizards
- IBM DB2 for z/OS Object Wizards
- Microsoft SQL Server Object Wizards
- MySQL Object Wizards
- Oracle Object Wizards
- PostgreSQL Object Wizards
- Sybase ASE Object Wizards

Overview and Common Usage of Object Wizards

An object wizard lets you create new database and server objects on a datasource. For example, a Tables wizard lets you create columns, set up permissions to work with that table, provide details on physical storage for the table, and so on. Each pane on an object wizard lets you perform a logical task or collection of logical tasks for that object type. For example:



Properties
Specify the index properties.

Property	Value
Creation	
Table Schema	
Table Name	
Schema	
Name	
Attributes	
Index Type	NONUNIQUE
Clustered	False
Number of Partitions	0
Buffer Pool	
Defer	False
Close	True
Copy	False
Define	True
Piece Size (KB)	2097152

In order to work with object wizards, you must be familiar with the following:

- Opening an Object Wizard
- Navigating and Setting Properties in an Object Wizard
- Adding a Comment to an Object
- Setting Permissions or Privileges for an Object
- Previewing the DDL Generated to Create the New Object.

Opening an Object Wizard

Object wizards are accessed from the Datasource Navigator/Explorer pane of the main window.

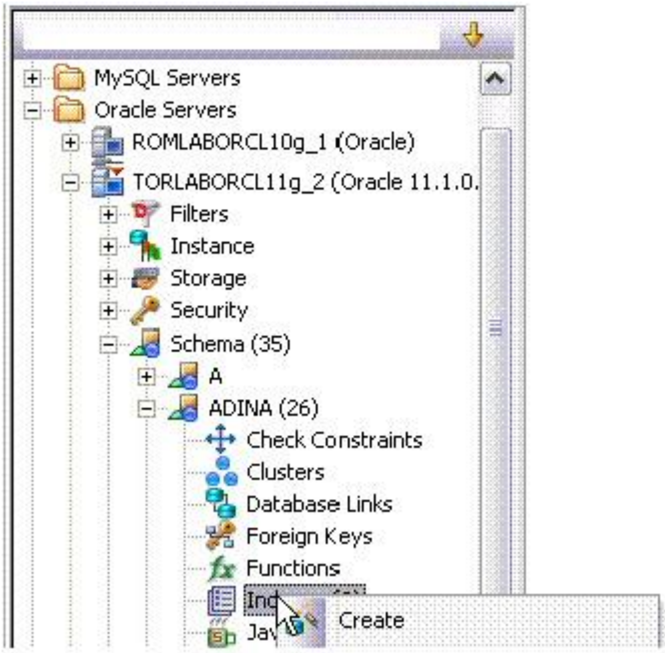
To open an Object wizard for a server or database object

1 Connect to the datasource where you want to create a new resource. For more information, see Datasource and Server Management.

2 On the Datasource Explorer, expand the target datasource.

3 Continue to expand folders under the datasource until the type of object you want to create is visible.

4 On the Datasource Navigator/Explorer, right-click on the specific type of object that you want to create and select **Create** from the context menu.



The object wizard for that object type opens.

Navigating and Setting Properties in an Object Wizard

When you invoke an object wizard, it opens on the first pane of the wizard, typically a **Properties** pane. As you select options or provide details, you use the navigation buttons at the bottom of the window and the pane controls at the left of the window to navigate through the wizard.

- Use the **Next** button to move to the next pane of the wizard.

Note: In some cases, the **Next** button is not activated until required information is provided. Similarly, some panes of a wizard do not become accessible until activated by choice of an option or property on a previous pane. For example, a **Partitions** tab in a table wizard may only become available if the **Clustered** and **Number of Partitions** options on a prior tab are set accordingly.

- Use the **Back** button to move to the previous pane of the wizard.
- Use the pane controls at the left of the window to move to a specific pane of the wizard.

Adding a Comment to an Object

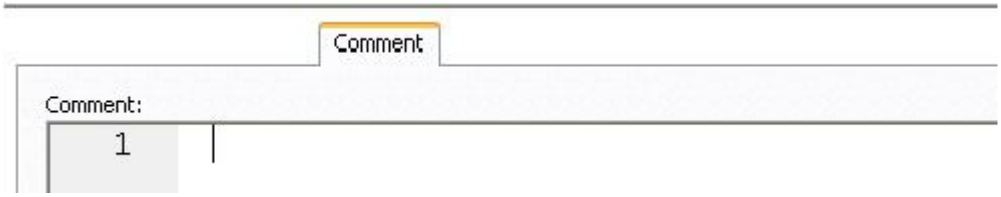
The object wizards for certain object types feature a **Comment** tab that lets you add an explanatory note to specific object definitions. Comments are stored in the REMARKS column of the object's system-catalog.

To add a comment to an object:

1 Open an object wizard on an object type that permits comments. For details, see Opening an Object Wizard.

See the topics for specific Object editors later in this chapter for information on whether that object type supports comments.

2 Click the **Comment** panel when enabled.



::

3 In the **Comment** area, type an explanatory note of up to 254 characters long.

Setting Permissions or Privileges for an Object

When you open an Object wizard to create an object with associated privileges, the **Permissions** (or **Privileges** or **Object Permissions** or **System Permissions**) panel for that editor displays the relevant privileges and lets you make changes accordingly:

To set permissions for an object:

1 Open an object wizard on an object type with associated permissions or privileges. For details, see Opening an Object Wizard.

1 object type with associated permissions or privileges.

See the topics for specific Object wizards later in this chapter for information on whether that object type supports permissions/privileges.

2 Click the **Permissions** (**Privileges**, **Object Permissions** or **System Permissions**) panel when enabled.

3 For each specific permission to be granted to an entity such as a user, login, or group, select the cell corresponding to the entity and specific permission, and click the **Grant** button. To revoke a privilege, select a cell showing a Granted permission and click **Revoke**.

Previewing the DDL Generated to Create the New Object

The final pane or tab of a wizard or editor is most commonly designated **DDL View** or **Definition**.

- A **DDL View** tab/panel lets you view the DDL generated by your selections on the other tabs/panels of the wizard or editor.
- A **Definition** tab/panel is only available for objects such as triggers and procedures, for which you must provide the additional SQL statements that provide the actions taken for that object.

In most cases you arrive at **DDL View** or **Definition** by navigating through panes or tabs, choosing options as you proceed. Alternatively, if you have provided all required information for that object type, you can click **Finish** or **Execute** to create the object.

IBM DB2 for Linux, Unix, and Windows Object Wizards

You create DB2 LUW objects using the following wizards:

- Aliases Wizard (DB2 LUW)
- Bufferpools Wizard for IBM DB2 (DBArtisan - DB2 LUW)
- Databases Wizard (DB2 LUW)
- Event Monitors Wizard (DBArtisan - DB2 LUW)
- Foreign Keys Wizard (DB2 LUW)
- Functions Wizard (DB2 LUW)
- Groups Wizard (DBArtisan - DB2 LUW)
- Indexes Wizard (DB2 LUW)
- Materialized Query Tables Wizard (DB2 LUW)
- Nodegroups Wizard (DB2 LUW)
- Primary Keys Wizard (DB2 LUW)
- Procedures Wizard (DB2 LUW)
- Schemas Wizard (DB2 LUW)
- Sequences Wizard (DB2 LUW)
- Structured Types Wizard (DB2 LUW)
- Tables Wizard (DB2 LUW)
- Tablespaces Wizard (DB2 LUW)
- Triggers Wizard (DB2 LUW)
- Unique Keys Wizard (DB2 LUW)
- User Datatypes (DB2 LUW)
- Users Wizard (DB2 LUW)
- Views Wizard (DB2 LUW)

In addition, see Create Synonym.

Aliases Wizard (DB2 LUW)

An alias offers you security and convenience so that you can refer to an object without revealing who owns it or what database it belongs to. You can create aliases for tables, views, and even other aliases. The Alias Wizard lets you create an alias without knowing the underlying commands. As you complete the Alias Wizard process, a CREATE ALIAS statement is generated based on the information that you supply.

To create a new alias using a wizard:

1 Open a creation wizard for an alias. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details see Aliases (DB2 LUW) - Properties.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Aliases (DB2 LUW) - Properties

When creating or editing an alias, this tab/panel lets you work with the following settings:

Setting	Description
Schema	Select the schema that is to own the alias.
Name	Provide a name for the alias
Target Owner	Select the owner of the object to which you are creating an alias.
Target Type	Select the type of object to which you want to create an alias.
Target Name	Select the specific object to which you are creating an alias.

For context information on the wizard and editor that include this tab/panel, see **Aliases Wizard (DB2 LUW)** and Aliases Editor (IBM DB2 LUW).

Bufferpools Wizard for IBM DB2 (DBArtisan - DB2 LUW)

The Bufferpool Wizard lets you create a bufferpool (an area of memory where data pages are read, altered, and held during an operation) without knowing the underlying commands. As you complete the Bufferpool Wizard, DBArtisan constructs the necessary CREATE BUFFERPOOL statement based on the information that you supply. Note that the bufferpool you're creating to store the data from the tables in selected database won't be active until you restart the database.

The Bufferpool Wizard lets you:

- Name the tablespace and specify the size of the bufferpool.
- Set storage options for UNIX systems.
- Set Bufferpool disbursement parameters.
- Override the default bufferpool size for partitions (or nodes).

To Open the Bufferpool Wizard

1 On the Navigator/Explorer, find the database where you want to add the new bufferpool.

2 On the **Buffer Pools** branch of the Performance node, right-click and select **New**.

3 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see BufferPools (DB2 LUW) - Properties.
- **Nodegroups** panel - for details, see BufferPools (DB2 LUW) - Nodegroups.
- **Partitions** panel - for details, see BufferPools (DB2 LUW) - Partitions.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

4 Finally, use the **Execute** button to create the object.

BufferPools (DB2 LUW) - Properties

When creating or editing a buffer pool, this tab/panel lets you work with the following settings:

Setting	Description
Create deferred	Lets you defer changes until the next database activation.
Pagesize	The page size, in KB, is available in the increments 4, 8, 16, or 32.
Automatic Size	Enables automatic size adjustment to account for workload.
Number of Pages	Indicate the number of pages you want to include in the new bufferpool. If the database is partitioned database, the default size is of all the partitions (or nodes) where the bufferpool exists. A value of -1, the default, means the size will be taken from the database configuration parameter (BUFFPAGE).
No. of Block Pages and **Block Size**	Lets you specify NUMBLOCKPAGES and BLOCKSIZE values used when creating or modifying buffer pools.

BufferPools (DB2 LUW) - Nodegroups

Lets you select the Nodegroups on which to create the buffer pool. For more information, see Nodegroups Wizard (DB2 LUW).

BufferPools (DB2 LUW) - Partitions

Lets you modify the default size of assocaited partitions.

Databases Wizard (DB2 LUW)

The Database Wizard lets you create a database (a structured collection of data that can be updated or queried) without knowing the underlying commands. Databases can be simple, that is one file with many records and the same fields, or much more complicated with multiple files with different fields.

To open the IBM DB2 for Linux, Unix, and Windows Database Wizard

On the **Registration** tool bar, click **New UDB Database**.

OR

On the Navigator/Explorer, right-click an instance node or the **Datasources** node, and then click **New UDB Database**.

The tables below describe the fields you may encounter as you complete the wizard.

Naming the New Database

Required Information	Description
What is the name of the database?	Each database requires a unique name. The database name should be between 1 - 8 characters and must begin with a letter A-Z. It's wise to avoid the symbols @, #, and $ if the database will exist in a communications environment or be summoned from an area where those characters do not exist in the language (and hence, on a keyboard).
What is the datasource name for the new database?	Each datasource should have a unique name. This is the name that will appear on the Navigator/Explorer.
What is the alias name of the database.	OPTIONAL: A database alias is the local synonym given to the database you are creating, and must be unique. If you don't assign an alias, the database name becomes the default alias. Note that a DB2 client can have connections to two different databases with the same name if those databases live on different servers and each database has its own alias.
What is the comment used for the database?	OPTIONAL: Lets you type a comment up to 30 characters. Any comment you enter can be changed later.

Drive Path and Parameters

Required Information	Description
On what drive/path will the database reside?	OPTIONAL: Leave blank if you want to create the database using the DFTBPATH (default database path configuration) parameter.
What default tablespace parameters should be used?	OPTIONAL Extent size: This is the number of pages of table data that will be written before data is written to the next container. Number of segments: The default is 0. The number you designate specifies the number of extended memory segments available for the database to use. You should only set this number if you have a large amount of real memory.
What global default parameters should the database use?	Territory: The territory where the database was created. This is purely informational, for example, en_US and is related to region-specific support. Codeset: IBM-1252 or UTF-8 are the options for language support. Collating Sequence: Compatibility, Identity, System are your choices for comparing character data. Note: The collation sequence can't be changed after the database has been created.
Finish	Opens the **Preview SQL** dialog.

Add Catalog/Temporary/User Containers

There are three distinct wizard panels, one for each container type, that enable you to customize your database. Click **Next** or **Finish** depending on how you want to proceed.

Required Information	Description
What container(s) will be used to store the catalog/temporary tables?	Use System Managed Space: If you select this option, the operating system's file system manager allocates and manages the space where the catalog tables are stored. This is the default. Use Database Managed Space: If you select this option, the database manager controls the storage space for catalog tables. Add or Edit - For details, seeAdd/Edit Container for Tablespace.
What optional default storage parameters should be used.	Optionally identify values for extent size, pre-fetch size, transfer rate, and overhead.
Finish	Opens the **Preview SQL** dialog.

Add/Edit Container for Tablespace

The table below describes the options and functionality on this dialog:

Required Information	Description
Container Parameters	These parameters are enabled for modification only if you have opted to use database managed space. When enabled, you can either indicate file size parameters or specify device information.
Directory	Identify the directory where the catalog table container will be stored.
File Name	Identify a name for the file where the catalog table container will be stored.
Device Information	Enable either raw partition and type a value, or raw drive, and select a value.

Event Monitors Wizard (DBArtisan - DB2 LUW)

An event monitor is used to collect information about a particular database and connected appliances when one or more events you specify take(s) place. Where a snapshot monitor captures information for a given moment in time, the event monitor lets you trace the activity you specify.

The Event Monitor Wizard lets you provide basic properties, select event types, and specify conditions where appropriate, for an event monitor.

To create a new event monitor using a wizard:

1 Open a creation wizard for an event monitor. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Event Monitors (DB2 LUW) - Properties.
- **Definition** panel - for details, see Event Monitors (DB2 LUW) - Definition.
- **Conditions** panel - for details, see Event Monitors (DB2 LUW) - Conditions.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Event Monitors (DB2 LUW) - Properties

When creating or editing an event monitor, this tab/panel lets you work with the following settings:

Setting	Description
Name	The name used to reference the object.
Auto Start	Indicates whether the event monitor is activated each time the database starts.
Target Type	Indicates where the output is written: FILE, PIPE, or TABLE.
Buffer Size	For a **Target Type** of TABLE or FILE, indicates the buffer size
Blocked	For a **Target Type** of TABLE or FILE, indicate whether the event monitor is to be blocked.
Target	For a **Target Type** of PIPE, a named pipe identifier.
Write Mode	For a **Target Type** of FILE, the value can be APPEND or REPLACE.
Max Files	For a **Target Type** of FILE, indicate the maximum number of files.
Max FIle Size	For a **Target Type** of FILE, indicates the maximum file size.

Event Monitors (DB2 LUW) - Definition

This tab/panel lets you specify the **Event Types** used to construct the CREATE EVENT MONITOR statement FOR clause.

The base set of event types is BUFFERPOOLS, CONNECTIONS, DATABASE, DEADLOCKS, DEADLOCK WITH DETAILS, STATEMENTS, TABLES, TABLESPACES, and TRANSACTIONS.

For IBM DB2 for Linux, Unix, and Windows version 9.0, DEADLOCK WITH DETAILS HISTORY and DEADLOCKS WITH DETAILS HISTORY VALUES event types are available.

For IBM DB2 for Linux, Unix, and Windows version 9.5, ACTIVITIES, THRESHOLD VIOLATIONS, and STATISTICS event types are available.

Note: If your choices are all from the DEADLOCKS... types, you can also specify **Deadlock Monitoring** of **Global** or **Local**. If you select Local, specify the **Database Partitions** the event monitor is to run against.

Event Monitors (DB2 LUW) - Conditions

Available only for CONNECTION, STATEMENT, and TRANSACTION **Event Type**s, this tab lets you identify a condition used to determine which CONNECTION, STATEMENT, or TRANSACTION the event monitor will collect data for.

Foreign Keys Wizard (DB2 LUW)

Foreign keys are unique values that refer to specific columns of other tables. Thus, a foreign key links two tables together. The Foreign Key Wizard makes it easy for you to create a relational link between two tables, thereby speeding queries and giving you faster access to data.The Foreign Key Wizard lets you create a foreign key without knowing the underlying commands.

To create a new foreign key using a wizard:

1 Open a creation wizard for a foreign key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Foreign Keys (DB2 LUW) - Properties
- **Column Mapping** panel - for details, see Foreign Keys (DB2 LUW) - Column Mapping
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Foreign Keys (DB2 LUW) - Properties

When creating or editing a foreign key, this tab/panel lets you work with the following settings:

Setting	Description
Table Schema	The schema owning the table where the foreign key is being created.
Table Name	This is the table where the foreign key link originates--the child table.
Name	Lets you select a constraint name. System Generated Name - DB2 automatically generates a name. User Specified Constraint Name - You type the name.
Enabled	TRUE enables the Foreign Key while FALSE disables the Foreign Key.
Delete Rule	Select an action: **NO ACTION** - ensures that referenced values cannot be updated or deleted if to do so would violate referential integrity. **CASCADE** permits a referenced row in a child table to be deleted/updated if it is deleted/updated in the parent table. A row in the child table is SET NULL when rows in the parent table are deleted/updated. **RESTRICT** prevents a command from executing if changes to referential data prompts an error.
Update Rule	Select an action. **NO ACTION** - ensures that referenced values cannot be updated or deleted if to do so would violate referential integrity. **RESTRICT** - prevents a command from executing if changes to referential data prompts an error

Foreign Keys (DB2 LUW) - Column Mapping

1 Under **Referenced Table**, choose the **Owner** and then the **Name** of the referenced, or parent, table.

2 Under the **Main Table**, select check boxes corresponding to the columns that are to reference columns in the referenced table.

Functions Wizard (DB2 LUW)

The Functions Wizard lets you create a relationship between one set of values and another. You can develop reusable subroutines so you can control, access, and manipulate the data that underlies an object. As you complete the Function Wizard process, a CREATE FUNCTION statement is generated based on the information that you supply. The Function Wizard lets you create a function without knowing the underlying commands.

Note: To create a user-defined function, you need CREATE ANY privileges or IMPLICIT_SCHEMA authority on the database if the schema does not already exist.

To create a new function using a wizard:

1 Open a creation wizard for a function. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Functions (DB2 LUW) - Properties.
- **Advanced** panel - for details, see Functions (DB2 LUW) - Advanced.
- **Source** panel (only available for a **Function Type** of SOURCED) - for details, see Functions (DB2 LUW) - Source.
- **Parameters** panel (only available for a **Function Type** of SQL, SOURCED, or TEMPLATE) - for details, see Functions (DB2 LUW) - Parameters.
- **Return Scalar** panel - for details, see Functions (DB2 LUW) - Return Scalar.
- **Return Columns** panel (only available when you choose a **Function Type** of EXTERNAL TABLE) - for details, see Functions (DB2 LUW) - Return Columns.
- **Body** panel - for details, see Functions (DB2 LUW) - Body.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Functions (DB2 LUW) - Properties

When creating or editing a foreign key, this tab/panel lets you work with the following settings:

Setting	Description
Schema, **Name**, and **Specific Name**	Let you select the owner of the function, provide a name for the function, and provide the Specific name to be used by some SQL statements and DB2 commands for this function.
Function Type	Select the type of function: External Scalar - written in a programming language and returns a scalar value. External Table - written in a programming language and returns a complete table. OLEDB - accesses OLE DB data in user-defined OLE DB external tables. Sourced - another function is invoked to implement the function you are creating. SQL Language - written in SQL and returns a table, scalar value, or single row. Template: - this is a partial function and can only be invoked from a federated datasource. For more information, see About function types.
Language	If you chose a **Function Type** of EXTERNAL SCALAR or EXTERNAL TABLE, specify a language of C, JAVA, or OLE. For more information, see About function types.
Return Type	For a **Function Type** of SQL, select ROW, TABLE, or SCALAR. For other **Function Type** choices, this indicates the default return type for that choice.
External Name	Provide the External Name of the function.
SQL Access Level	Indicates whether the function can execute SQL statements. CONTAINS SQL: Statements that don't read or modify SQL can be executed. NO SQL: No SQL statements can be executed. READS SQL: Statements that cannot modify SQL can be executed.

Functions (DB2 LUW) - Advanced

This tab is only available after clicking the **Advanced** button on the Function wizard's **Properties** panel. It lets you work with the **Threadsafe**, **Fenced**, **Scratchpad**, **Scratchpad Length**, **Allow Parallel**, **Final Call**, **Parameter Style**, **Inherit Special Registers**, **DBINFO**, **Deterministic**, **External Action**, **Called on Null Input**, and **Parameter CCSID** properties.

== Functions (DB2 LUW) - Source ==

Note: This panel is only available for a **Function Type** of SOURCED.

Select the **Schema**, **Name**, and **Specific Name** of the source function. Function **Parameters** and **Return Type** for the selected function are displayed. For more information, see About function types.

Functions (DB2 LUW) - Parameters

Note: This panel is only available for a **Function Type** of SQL, SOURCED, or TEMPLATE.

For each parameter for this function, use the New button to add a new parameter, provide a name for the parameter, and in the **Attributes** area, select a **Type**, and if appropriate, the **Precision**, **Scale**, and **Size** options.

Functions (DB2 LUW) - Return Scalar

Under **Return Datatype**, select a Type and depending on your choice, provide or select **Precision**, **Scale**, **Size**, and **As Locator** options.

To make use of a CAST FROM clause, under **Cast Datatype** set **Enabled** to True, select a **Type**, and if appropriate, the **Scale**, **Size**, and **As Locator** options

Functions (DB2 LUW) - Return Columns

Note: This panel is only available when you choose a **Function Type** of EXTERNAL TABLE.

For each column returned by this function, use the New button to add a new parameter, provide a name for the parameter, and in the **Attributes** area, select a **Type**, and if appropriate, the **Precision**, **Scale**, **Size**, and **As Locator** options.

Functions (DB2 LUW) - Body

Enter the return statement for the function.

About function types

External Scalar/Table/OLE DB Function

External scalar user-defined functions are implemented in an external programming language. The functions are executed on the server and can read SQL data but cannot make changes to the data. These functions are often used to extend the set of built-in functions for DB2, perform logic inside a SQL query that SQL can't perform on its own, and, encapsulate a scalar query that is often used as a subquery in SQL statements, for example, if given an ingredient, search a table for a recipe that uses that ingredient.

When specifying the **Specific Name** for one of these function types:

- If you are using C language, specify the full library path and the function name, otherwise IBM DB2 Database Manager assumes the function is under the IBM DB2 library.
- If you are using Java script, specify the Class ID and the function name, otherwise IBM DB2 Database Manager assumes the function is under the IBM DB2 library.
- If you are using OLE language, specify the full library path and the function name, otherwise IBM DB2 Database Manager assumes the function is under the IBM DB2 library.

Sourced Functions

When you create a sourced function, the new function you are creating will be implemented by a preexisting (source) function that's known to the database manager. The source function can be a built-in function or a previously created user-defined scalar function.

When you select the appropriate schema, you are really specifying an implicit schema privilege. In other words, you're selecting a schema/function that belongs to a user with DBAdmin privileges. If you want to use a built-in function, you must specify the function's specific name.

SQL Language Function

The function you are creating is written in SQL. A table, scalar value, or single row is returned.

Template Function

A template function is better thought of as a template for a function. It's a partial function that has no executable code. You create a function template for mapping it to a datasource function. When the mapping is created, you can specify the function template be used in queries submitted to the federated server. When the query is processed, the federated server invokes datasource function where the template is mapped and returns the appropriate values.

Groups Wizard (DBArtisan - DB2 LUW)

The Group Wizard lets you define new groups and to assign the appropriate object and system level privileges. Any user who is a part of the group you create will inherit the authorities and privileges assigned to the group.

To Open the Group Wizard

1 On the Navigator/Explorer, find the datasource where you want to add the new group.

2 Expand the **Security** branch, right-click **Groups**, and select **New**.

The table that follows describes the fields you may encounter as you complete the Group Wizard:

Required Information	Description
What is the name of the group?	Lets you enter the name of the group.
What database authority(ies) do you want to give this group?	Select all or some of the authorities: BINDADD: User can create new database packages CONNECT: Gives users access to the database CREATETAB: A user can create new tables in the database CREATE_NOT_FENCED: Allows a user to create user-defined functions IMPLICIT_SCHEMA: Enables any user to implicitly create a schema by creating an object with a CREATE statement using a schema name that doesn’t already exist. DBADM: Gives users administrative privileges. LOAD: Users can load data into a table. NOTE: The authorities listed here are the system privileges.

Indexes Wizard (DB2 LUW)

Comparable to an index in a book, an index gives you speedy access to particular records in a table. The Index Wizard lets you create an index without knowing the underlying commands.

To create a new index using a wizard:

1 Open a creation wizard for an index. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Indexes (DB2 LUW) - Properties.
- **Columns** and **Include Columns** panels - for details, see Indexes (DB2 LUW) - Columns and Include Columns.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Indexes (DB2 LUW) - Properties

When creating or editing an index, this tab/panel lets you work with the following settings:

Setting	Description
Parent Type, **Parent Schema** and **Parent Name**	Choose the type (TABLE or MATERIALIZED QUERY TABLE) owning schema, and name of the parent object in which the index is being created.
Schema and **Name**	Choose the owner and name of the index being created.
Index Type	Index enforces uniqueness on the values of the table's index key.
Clustered	Specifies that the index is the clustering index of the table. The cluster factor of a clustering index is maintained or improved dynamically as data is inserted into the associated table, by attempting to insert new rows physically close to the rows for which the key values of this index are in the same range.
Allow Reverse Scans	If selected, a ALLOW REVERSE SCAN clause is included with the CREATE/ALTER INDEX statement, specifying that the index supports both forward and reverse scans.
Compress (9.7^)	YES - Selecting this option adds a COMPRESS YES clause, enabling index compression. NO - Selecting this option adds a COMPRESS NO clause, disabling index compression. DEFAULT - Selecting this option specifies default compression handling to that of the parent object.
Percent Free	Lets you provide a PCTFREE value, specifying the percentage of each index page to leave as free space when building the index, from 0 to 99.
Minimum Percent Used	Lets you provide a MINPCTUSED value, specifyinf whether index leaf pages are merged online, and the minimum percentage threshold of space used on index leaf pages.

Indexes (DB2 LUW) - Columns and Include Columns

Index columns can be segregated into unique key columns (**Columns** pane) and Include columns that are to be part of the index but do not form part of the unique key.

The steps in completing the panes for the two column types are identical.

- From the **Column** dropdown, select a column for the index and specify a **Sort** option. To add more columns, click the **New** button and then follow the steps in the last instruction. Use the Delete button to drop columns.

Materialized Query Tables Wizard (DB2 LUW)

A materialized query table (MQT) is a table based on the result of a query. An MQT contains information that is summarized from other tables and can save time when processing dynamic SQL queries. The Materialized Query Table Wizard lets you create a table without knowing the underlying commands.

To Open the Materialized Query Wizard

1 On the Navigator/Explorer, find the datasource where you want to add the new materialized query table.

2 Expand the **Schema** branch, right-click **Materialized Query Tables**, and select **New**.

The table that follows describes the fields you may encounter as you complete the Materialized Query Table Wizard.

Note: These options are only available if the tablespace you selected is a database managed tablespace.

Required Information	Description
Who owns the table?	Choose the owner of the table you're creating from the drop-down list.
What is the name of the table?	Type the name of the materialized query table you are creating.
Select a tablespace on which to place the table:	OPTIONAL: No selection is the default. But you can select a tablespace that belongs to the new table's owner if you want.
Specify separate tablespaces for index and long data	OPTIONAL: Lets you separate indexes or long data from the table. Indexes Long data
Specify the query on which this table is based	Write the query you want to use to create the table. Note: Every select list element must have a name (use the AS clause for expressions)
Add the columns belonging to this table Add the columns belonging to the partition key	Click **Add**, **Insert** or **Edit** to add or modify table columns. Click **Drop** to delete a selected column.
Do you want the table replicated across database partitions?	The default is no, but check the box if you do want to replicate the table. Replicated materialized query tables can help you improve query performance by enabling collocation between tables. Replication is especially helpful when you have joins between large fact tables and small dimension tables. It's best if tables that are to be replicated are infrequently updated.
Definition Only	Lets you select definition options, Include Column Defaults and Include Identity Column Attributes. When you select the Definition Only option, the new table is treated as if it was a base table rather than a MQT. After you complete the wizard, the Tables Editor opens.
Refreshable	Lets you select refresh options: Immediate: The table you are creating will be refreshed automatically when changes are made to the base table(s). Deferred: Static SQL will not be optimized. Changes to the base table(s) will not be reflected. Query Optimization: Enabled: Queries will be routed to the MQT. Disabled: This is the default. Maintained by: System: This is the default User After you complete the wizard, the Material Query Tables Editor opens.
Would you like extra information regarding SQL changes to this table to be written to the log?	If you opted to replicate the table, you must make a selection here. Yes means you want to receive error messages issued by the DB2 replication programs. Include Longvar Columns means you want to receive error messages for these columns if long varchar data is a part of the table you're replicating. No
Would you like to skip logging changes made to this table by Insert... and Alter Table operations in the same unit of work in which this table is created?	Initially Not Logged: This is an option that can reduce logging and increase performance, but also means that you cannot recover the table when rolling forward. Logged Initially. This is the default.

What type of table lock would you like when it is being accessed?	Row: This is the default. During a table or index scan, DB2 locks each row that is scanned before determining whether that row is relevant to the query. Table: During a table scan, DB2 locks the table so no data can be added or altered while the query is executed.
What percent of free space to leave for load and reorganization?	-1 is the default.
Do you want data to append to the end of the table?	Yes No: This is the default
Do you want the access plan to this table to be based on existing statistics and optimization level?	Volatile: A volatile table's contents can vary from empty to huge at run time and can render collected statistics inaccurate. Not Volatile: This is the default.
Enter a comment	Optional

Nodegroups Wizard (DB2 LUW)

A nodegroup is a named subset of one or more database partitions. The Node Group Wizard lets you create a node group without knowing the underlying commands. When you create a nodegroup, the wizard simply asks you to name the nodegroup and select the partitions to include in the nodegroup.

To Open the Nodegroup Wizard

1 On the Navigator/Explorer, find the datasource where you want to add the new Nodegroup.

2 Expand the **Storage** branch, right-click **Nodegroups**, and select **New**.

Primary Keys Wizard (DB2 LUW)

A primary key is a column or group of columns that you can use to identify or access one or more specific rows in a table. A primary key is 'constrained' in that no values can be null and no two values are equal. You can only create one primary key for any table. The **Create Primary Key Constraint** dialog lets you create a primary key without knowing the underlying commands.

When you create a primary key, specify the table owner and the table on which you want to place the primary key constraint. You name the constraint and select the column(s) you want to include.

To create a new primary key using a wizard:

1 Open a creation wizard for a primary key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Primary Keys (DB2 LUW) - Properties.
- **Columns** panel - for details, see Primary Keys (DB2 LUW) - Columns.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object. For more information, see Previewing the DDL Generated to Create the New Object.

Primary Keys (DB2 LUW) - Properties

When creating or editing a primary key, this tab/panel lets you work with the following settings:

Setting	Description
Table Schema and **Table Name**	Choose the owner and name of the table in which the primary key is being created.
Name	Choose the owner and name of the primary key being created.
Time period policy	This setting is for use with tables set up as business-time tables. For more information, see Tables (DB2 LUW v10) - Temporal Properties. When selected, a BUSINESS_TIME WITHOUT OVERLAPS option is specified against the column or columns defining the constraint, in the generated DDL. This ensures that values for the constraint columns are unique with respect to the time for the BUSINESS_TIME period. When you select this option, columns specified as business-time begin period and end period are not available as candidates on the **Columns** tab/panel.

Primary Keys (DB2 LUW) - Columns

From the **Column** dropdown, select a column for the primary key and specify a **Sort** option. To add more columns, click the **New** button and then follow the steps in the last instruction. Use the Delete button to drop columns.

Procedures Wizard (DB2 LUW)

Procedures are a reusable block of PL/SQL, stored in the database, that applications can call. Procedures streamline code development, debugging, and maintenance by being reusable. Procedures enhance database security by letting you write procedures granting users execution privileges to tables rather than letting them access tables directly.

To create a new procedure using a wizard:

1 Open a creation wizard for a procedure. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Procedures (DB2 LUW) - Properties.
- **Advanced** panel - for details, see Procedures (DB2 LUW) - Advanced.
- **Parameters** panel - for details, see Procedures (DB2 LUW) - Parameters.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Procedures (DB2 LUW) - Properties

When creating or editing a procedure, this tab/panel lets you work with the following settings:

Setting	Description
Schema	Select the owner for the procedure.
Name	Provide the name of the function.
Specific Name	Optionally, provide the unique name of the procedure.
Language	Select among C, JAVA, COBOL, OLE, or SQL. The database manager will call the procedure accordingly.
External Name	Provide the external name of the procedure.
SQL Access Level	Select an option: MODIFIES SQL DATA - the procedure can support any SQL statement except those that cannot be supported in procedures. CONTAINS SQL DATA - only SQL statements that neither modify nor read SQL data can be executed in the procedure. READS SQL DATA - some SQL statements that don’t modify SQL data can be included in the procedure.

Procedures (DB2 LUW) - Advanced

When creating or editing a procedure, this tab/panel lets you work with the following settings:

Setting	Description
Results Sets	Indicate the estimated upper bound of returned result sets. 0 is the default.
External Action	Select the External Action option.
New Save Point	Lets you specify a NEW SAVEPOINT LEVEL clause for the procedure.
Threadsafe	Specify whether the procedure is safe to run within the same process as other routines.
Fenced	If you select yes, you are saying you do not want the procedure to run in the manager operating system environment. This means the database management system will protect its internal resources from the procedure. This option can affect the procedure's operation. To run a procedure as not fenced, or a No selection, you must have SYSADMIN or DBADMIN privileges because of the potential to compromise data if the procedure has not been adequately tested.
Parameter Style	Lets you select an option: DB2DARI, DB2GENERAL, DB2SQL, GENERAL, GENERAL WITH NULLS, JAVA, and SQL. DB2GENERAL is for Java Language only. DB2SQL is for C, COBOL, or OLE Language only. GENERAL is for C Language only. GENERAL WITH NULLS is for C or COBOL Language only. JAVA is for Java Language only. SQL is for C, COBOL, or OLE Language only.
Program Type	MAIN: valid for C or COBOL Language and Parameter Style GENERAL, GENERAL WITH NULLS, SQL, or DB2SQL only. In this case, parameters will be passed as an argument counter or argument vector. SUBROUTINE: the procedure expects the parameters to be passed as separate arguments.
Inherit Special Registers	Lets you specify this optional clause dictating that the procedure will inherit initial values from the environment of the invoking statement.
DBINFO	Specific information contains such information as the database name, application ID, database code page, and so on.
Deterministic	Enabling this feature specifies the procedure will always return the same result for given argument values. Disabling it means there are state values that affect the results and so the same result will not always be returned when you use identical inputs to invoke the procedure.
Parameter CCSID	Select an encoding scheme of ASCII, UNICODE, or NONE for character or graphic string parameters.

Procedures (DB2 LUW) - Parameters

For each parameter for this function, use the New button to add a new parameter, provide a name for the parameter, and in the **Attributes** area, select a **Type**, specify a **Parameter Mode** of INPUT, OUTPUT, or INPUT_OUTPUT, and if appropriate, the **Precision**, **Scale**, and **Size** options.

Sequences Wizard (DB2 LUW)

A sequence allows the automatic generation of values, something well-suited to the generation of unique key values. Sequences are not tied to particular table columns. The Sequence Wizard lets you create a sequence without knowing the underlying commands. As you complete the Sequence Wizard, a CREATE SEQUENCE statement is generated from the information that you have supplied. When finished, you can compile the sequence on the target database or to write a script file containing the CREATE SEQUENCE statement.

The Sequence Wizard lets you:

- Specify the name and owner of the sequence.
- Set both the value of the sequence, and an interval and ranges for incrementing it.
- Cache the sequence, cycle the sequence when it reaches its minimum or maximum values, and guarantee that sequence numbers are generated in the order of request.

Note: To create a sequence, it must belong to your schema or you need CREATE SEQUENCE privileges.

To Open the Sequence Wizard

1 On the Navigator/Explorer, find the datasource where you want to add the new Sequence.

2 Expand the **Schema** branch, right-click **Sequences**, and select **New**.

The table that follows describes what you may encounter as you complete the Sequence Wizard:

Required Information	Description
Who owns the sequence?	You decide.
What is the sequence name?	Your choice.
What numeric datatype should the Sequence use?	Choose among BIGINT (big integer), decimal (choose width as well), integer, small integer.
What is the first sequence number to be generated?	Starting with 1 is the default.
What is the interval between sequence numbers?	Increment by 1 is the default.
What is the sequence's minimum value?	Choose none or set a value
What is the sequence's maximum value?	Choose none or set a value
Should DB2 preallocate sequence numbers and cache them for faster access?	Preallocating and storing values in the cache reduces synchronous I/O to the log when values are generated for the sequence. If Yes, give number of values No
Should the sequence continue to generate values after reaching either its maximum or minimum value?	Lets you make the sequence cycle and continue to generate numbers. Yes or No
Should the sequence numbers be generated in the order of request?	Select to generate sequence numbers in the order of request. The ORDER option is useful when you are using the sequence number as a timestamp. Yes or No

Schemas Wizard (DB2 LUW)

The Schema Wizard lets you create the structure of a database system including database objects.

To Open the Schema Wizard

1 On the Navigator/Explorer, find the datasource where you want to add the new Schema.

2 Right-click **Schema** and select **New**.

All you need to do when the single-panel wizard opens is to give a unique name to the schema you're creating.

Structured Types Wizard (DB2 LUW)

The Structured Type wizard lets you create the object type specification and define attributes for a structured type. You then use the Structured Type editor to define the methods and body for the type.

To Open the Type Wizard

1 On the Navigator/Explorer, find the datasource where you want to add the new Structured Type.

2 Expand the Schema branch, right-click **Structured Type**, and select **Create**.

3 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Structured Types (DB2 LUW) - Properties.
- **Attributes** panel - for details, see Structured Types (DB2 LUW) - Attributes.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

4 Finally, use the **Execute** button to create the object.

On completion, the Structured Type editor opens, letting you provide method and body specifications for the structured type. For details, see Structured Types Editor (IBM DB2 LUW).

Structured Types (DB2 LUW) - Properties

When creating or editing a structured type, this tab/panel lets you work with the following settings:

Setting	Description
Attributes	**Instantiable** - if disabled, no constructor function can be generated, nor can a non-instantiable type be used as the type of a table or view. **Final Type** - enabled, indicates that the structured type cannot be used as a supertype. **With Function Access** - If enabled, all methods of the type you are creating, and those you will create, can be accessed using functional notation. Some applications prefer functional notation over method invocation. **Without Comparisons** - If enabled, indicates that there are no comparison functions supported for instances of the structured type. **Inline Length** - Specifies the maximum size of an instance of a structured type in a column. If the size of a structured type instance is less than the defined maximum, the data is stored inline with the other values in the row. If the size of the structured type is larger than the defined maximum, the structured type data is stored outside of the table (like LOBs).
Supertype	**Supertype Schema** and **Supertype Name** - let you designate this structured type as a subtype by providing details of the owning supertype.
Reference	**Cast (Source as Ref) With** and **Cast (Ref as Source) With** - Lets you name the cast function, although one will be created with the default name of the structured type you are creating. The cast function “casts” a value between the reference type and the representation type in both directions. **Reference Using**, **Size**, **Precision**, and **Scale** - Define the built-in data type used as the underlying data type for the structured type you are creating and all its subtypes.

Structured Types (DB2 LUW) - Attributes

For each attribute to be added to the structured type, click the **New** button, and provide a name for the attribute.

With an attribute selected, you can modify the **Datatype**. Depending on the datatype you choose you can provide the following datatype qualifiers:

- **Width** (decimal only)
- **Scale** (decimal only)
- **Size** (blob, character, clob, dbclob, graphic, varchar, vargraphic only)
- **Allow log** (blob, clob, dbclob only)
- **Allow compact** (blob, clob, dbclob only)
- **For bit data** (character, long varchar, varchar, only)

Tables Wizard (DB2 LUW)

The Table Wizard lets you create a table without knowing the underlying commands.

To create a new table using a wizard:

1 Open a creation wizard for a table. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Tables (DB2 LUW) - Properties.
- **Columns** panel - for details, see Tables (DB2 LUW) - Columns.
- **Partition** panel - for details, see Tables (DB2 LUW) - Partition.
- **Tablespaces** panel - for details, see Tables (DB2 LUW) - Tablespaces.
- **Dimension** panel - for details, see Tables (DB2 LUW) - Dimensions.
- **Distribution Key Columns** panel - for details, see Tables (DB2 LUW) - Distribution Key Columns.
- **Indexes** panel - for details, see Tables (DB2 LUW) - Indexes.
- **Temporal Properties** panel - for details, see Tables (DB2 LUW v10) - Temporal Properties.
- **Constraints** panel - for details, see Tables (DB2 LUW) - Constraints.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **Permissions** panel - Setting Permissions or Privileges for an Object
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Tables (DB2 LUW) - Properties

When creating or editing a table, this tab/panel lets you work with the following settings:

Setting	Description
Schema	Select the schema that is to own the table.
Name	Provide a name for the table.
Percent Free	
Lock Size	The table-level lock prevents concurrently operating applications from using or changing a table. When scanning a table for a query, a row-level lock locks the row when it is being assessed.
Append Data	Enable or Disable appending data to the end of the table.
Volatile	Enable this feature if a table contents may fluctuate from empty to very large. The access plan will not depend on the existing statistics for that table.
Compress	Enable or disable value compression
Row Compression	Enable or disable row compression.
Security Policy	Lets you add a security policy to a table.
RestrictDrop	Corresponds to the DB2 Restrict on Drop attribute.
Log Index Build	Enables this level of logging when creating, recreating, or reorganizing an index.
CCSID	Specify ASCII or UNICODE or leave unspecified. If specified, this is the encoding scheme for string data. If unspecified, CCSID default encoding is used.
Tablespace, Index Tablespace, and Long Data Tablespace	Select a tablespace, an index tablespace, and a tablespace for Long or LOB table columns.
Do not initially log	If enabled, all changes to the table will be flushed out at commit time. This also means that if a statement fails, the unit of work will rollback. If you are concerned about recoverability, disable this feature.
Data Capture	Specify additional information logged by selecting DATACAPTURE NONE, DATA CAPTURE CHANGES, or DATA CAPTURE CHANGES INCLUDE LONGVAR.

Tables (DB2 LUW) - Columns

For each column in the table, click the **New** button to create a column and provide a name for the column. Then, in the **Column Attributes** area, provide details for the column.

Use the Delete button to drop a selected column.

Row Properties (V10) settings are available when you select a **Type** of TIMESTAMP. These settings are used in conjunction with settings on the **Temporal Properties** tab in setting up system and application temporal tables. For more information, see Tables (DB2 LUW v10) - Temporal Properties.

Specific settings are:

- **As Row Begin** - adds a GENERATED ALWAYS AS ROW BEGIN option to the column specification, used to track when a row is current.
- **As Row End** - adds a GENERATED ALWAYS AS ROW END option to the column specification.
- **Transaction Start ID** - adds a GENERATED ALWAYS AS TRANSACTION START ID option to the column specification indicating that this column captures the start times for transactions that affect rows.

A column can have only one of the three settings selected.

When you select a **Row Properties** setting, the **Scale** value and if appropriate, the **Allow Nulls**, **Default Value**, and **Expression** settings are automatically adjusted for compatibility with row generation. Similarly, if one of the three **Row Properties** settings is selected for a column, and any of the **Scale**, **Allow Nulls**, **Default Value**, or **Expression** settings are modified, then the selected **Row Properties** setting is cleared.

Tables (DB2 LUW) - Partition

Under **Partition Columns**, for each partition column, click the **New** button and then choose a column name from the dropdown. To add a **Data Partition**, click the **New** button to open a dialog that lets you add a partition.

Tables (DB2 LUW) - Tablespaces

For each Data Tablespace or Long Tablespace, click the **New** button and then choose a tablespace from the dropdown. To specify an Index Tablespace, select a tablespace from the dropdown.

Tables (DB2 LUW) - Dimensions

For each column that is to make up a dimension, click the **New** button to open a dialog that lets you add a column.

Tables (DB2 LUW) - Distribution Key Columns

For each column that is to make up the distribution key, click the **New** button and then select a column from the dropdown

Tables (DB2 LUW) - Indexes

Click **Add** to open the Index wizard. For more information, see Indexes Wizard (DB2 LUW).

Tables (DB2 LUW v10) - Temporal Properties

This tab/panel lets you specify temporal properties to a table, designating it as a table that records the period of time when a row is valid.

Note: Before using this object action, consult DB2 LUW documentation for details on setup and use of the temporal tables feature. For more information, see Accessing Third Party Documentation.

This tab/panel has the following controls:

- **System-time table** - selecting this option creates a table with the PERIOD SYSTEM_TIME clause, designating it as a system-period temporal table. This option is available if the table has three columns of type TIMESTAMP, and the following prerequisites have been met:
- At least one TIMESTAMP-typed column has the **As Row Begin** property selected
- At least one TIMESTAMP-typed column has the **As Row End** property selected
- At least one TIMESTAMP-typed column has the **Transaction Start ID** property selected

For details on column definitions, see Tables (DB2 LUW) - Columns.

The **Row-begin column**, **Row-end column**, and **Transaction Start ID column** controls are automatically populated with the columns having the corresponding property.

When you select this option, in the generated DDL, the **Row-begin column** and **Row-end column** values are used as the PERIOD SYSTEM_TIME column parameters, defining the system period.

- **Business-time table** - selecting this option creates a table with the PERIOD BUSINESS_TIME clause, designating it as an application-period temporal table. This option is available if the table has two columns of type DATE or TIMESTAMP, for which no **Row Properties** have been assigned.

The **Begin column** and **End Column** controls are automatically populated with the valid candidate columns from the table's column list and the first distinct column pair is automatically chosen for convenience.

- **Use history table** - this set of controls is only active when **Business-time table** is selected. It lets you associate a history table. When you select **Use history table**, the Owner and Name controls are activated, letting you identify the history table to associate with this table.

Note that a constraint created on a business-time table can enforce the uniqueness of the data for any point in business time via the **Time period policy** property that has been added in the constraint wizards. For details, see the following topics:

- Primary Keys (DB2 LUW) - Properties
- Unique Keys (DB2 LUW) - Properties

Tables (DB2 LUW) - Constraints

Selecting a constraint type and clicking **Add** opens the object wizard for that object type. For details see:

- Primary Keys Wizard (DB2 LUW)
- Unique Keys Wizard (DB2 Z/OS)
- Foreign Keys Wizard (DB2 Z/OS))

Tablespaces Wizard (DB2 LUW)

Tablespaces establish connections between the physical storage devices of your database system and the logical containers or tables being use to store data. In essence, a tablespace is a storage structure that can hold tables, indexes, large objects, and long data. The Tablespace Wizard lets you create a tablespace without knowing the underlying commands.

To create a new tablespace using a wizard:

1 Open a creation wizard for a tablespace. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Tablespaces (DB2 LUW) - Properties.
- **Container** panel - for details, see Tablespaces (DB2 LUW) - Container.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **Permissions** panel - Setting Permissions or Privileges for an Object
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Tablespaces (DB2 LUW) - Properties

When creating or editing a tablespace, this tab/panel lets you work with the following settings:

Setting	Description
Tablespace properties	**Type** - Select REGULAR, LARGE, TEMPORARY, or USER TEMPORARY. **Use Automatic Storage** and **Managed By** let you specify whether storage is managed automatically, by the database, or by the system. **Database Partition Group** - lets you select a database partition group. **Buffer Pool** - lets you select a buffer pool. **Drop Recovery** - For REGULAR type tablespaces, lets you enable/disable drop recovery.
Performance properties	This group lets you specify or select the **Page SIze**, **Extent Size**, **Prefetch Automatic**, **Prefetch Size**, **Overhead**, **Transfer Rate**, and **File System Caching** properties.
Automatic Storage properties	This group lets you specify or select the **AutoResize**, **Initial Size**, **Increase Size**, **Max Size Unlimited**, and **Max Size** attributes.

Tablespaces (DB2 LUW) - Container

For each container in the tablespace, in the **Container Properties** area, provide the following container properties: **Database Partitions**, **Type** (FILE or DEVICE), **Name**, and **Size**, and then click the **New** button.

Use the **Delete** button to drop a selected container.

Triggers Wizard (DB2 LUW)

A trigger defines a set of actions that take place in conjunction with, or are triggered, for a particular base table, with an insert, update, or delete statement. Triggers are dandy ways to validate input data, read from other tables for cross-referencing purposes, and other similar purposes. The Trigger Wizard lets you create a trigger without requiring you to know any of the underlying commands.

To create a new trigger using a wizard:

1 Open a creation wizard for a trigger. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Triggers (DB2 LUW) - Properties.
- **Column Selection** panel - for details, see Triggers (DB2 LUW) - Column Selection.
- **Definition** panel - for details, see Triggers (DB2 LUW) - Definition.

3 Finally, use the **Execute** button to create the object.

Triggers (DB2 LUW) - Properties

When creating or editing a trigger, this tab/panel lets you work with the following settings:

Setting	Description
Table Schema and **Table Name**	Choose the owner and name of the table for which the trigger is being created.
Schema and **Name**	Choose the owner and name of the trigger being created.
Trigger Timing	BEFORE: These triggers serve as extensions to the constraint subsystem and are most often used to validate input data, generate values for newly inserted rows, and read from other tables for cross-reference purposes. Note: Before triggers must be created as a For Each Row. AFTER: Such a trigger is run after the integrity constraint validations; they can be used to modify operations in the database or be used for activities beyond the database, like supporting an alert notification.
Trigger Events	An INSERT trigger must be associated with an INSERT statement. For example, if a data load operation doesn't include an INSERT statement, the trigger will not be invoked. An UPDATE trigger can be associated with specific columns of the base table and will only be activated if those columns are updated.
Trigger Type	**STATEMENT**: (only fires once). **ROW** (fires for each affected row): The trigger runs as many times as there are rows in the affected section. If the set of affected rows is empty, the trigger doesn't run.
Old Table Alias	Type the name of a temporary table of rows as they exist before they're updated or deleted.
New Table Alias	Type a name for a temporary table of rows as they exist after they're inserted or updated.
Old Row Alias	Type a name for the rows as they are before they've been deleted or updated.
New Row Alias	Type a name for the rows as they are after they've been inserted or updated.

Triggers (DB2 LUW) - Column Selection

If you chose UPDATE as the **Trigger Event**, select the columns, select the check box beside each column that is to fire the trigger.

Triggers (DB2 LUW) - Definition

Complete the CREATE TRIGGER outline provided by typing or pasting the body of the trigger.

Unique Keys Wizard (DB2 LUW)

A unique key constraint is a key for which no two of its values can be equal and no values can be null. A table can have a number of unique constraints, but it cannot have more than one unique constraint on the same set of columns. If you are creating a unique key constraint on a table that already exists (as opposed to creating a unique key at the time the table is first generated), a unique index must already exist on the columns of the unique key you want to constrain. If no unique index exists, the Index Wizard will open as you complete the **Create Unique Key Constraint** dialog.

To create a new unique key using a wizard:

1 Open a creation wizard for a unique key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Unique Keys (DB2 LUW) - Properties.
- **Columns** panel - for details, see Unique Keys (DB2 LUW) - Columns.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Unique Keys (DB2 LUW) - Properties

When creating or editing a unique key, this tab/panel lets you work with the following settings:

Setting	Description
Table Schema and **Table Name**	Choose the owner and name of the table in which the unique key is being created.
Name	Provide a name for the unique key being created.
Time period policy	This setting is for use with tables set up as business-time tables. For more information, see Tables (DB2 LUW v10) - Temporal Properties. When selected, a BUSINESS_TIME WITHOUT OVERLAPS option is specified against the column or columns defining the constraint, in the generated DDL. This ensures that values for the constraint columns are unique with respect to the time for the BUSINESS_TIME period. When you select this option, columns specified as business-time begin period and end period are not available as candidates on the **Columns** tab/panel.

Unique Keys (DB2 LUW) - Columns

From the **Column** dropdown, select a column for the primary key and specify a **Sort** option. To add more columns, click the **New** button and then follow the steps in the last instruction. Use the Delete button to drop columns.

User Datatypes (DB2 LUW)

User defined datatypes allow column definitions to be consistent across a database. They let you associate frequently used datatype information to be associated with a specified function and take advantage of performance features available to built-in datatypes including indexing and parallel database queries.

To create a new user datatype using a wizard:

1 Open a creation wizard for a user datatype. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see User Datatypes (DB2 LUW) - Properties.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

User Datatypes (DB2 LUW) - Properties

When creating or editing a user datatype, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the owner of the user datatype.
Datatype	Provide a name for the datatype.
Type	Select the base datatype.
Size	Provide the size of the datatype.
Allow Bit Data	The option is only available for certain datatypes. A check means you want to store the data in a bit format.

Users Wizard (DB2 LUW)

Users have authorization to use a database and its objects, and the User Wizard gives you an easy way to add new ones.

To create a new user using a wizard:

1 Open a creation wizard for a user. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Users (DB2 LUW) - Properties.
- **Object Permissions** and **System Permissions** panels - Setting Permissions or Privileges for an Object
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Users (DB2 LUW) - Properties

When creating or editing a user datatype, this tab/panel lets you provide the user **Name**.

Views Wizard (DB2 LUW)

A view gives you an alternative way to look at data in one or more tables. You can customize how you see a table's contents.

To create a new view using a wizard:

1 Open a creation wizard for a view. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Views (DB2 LUW) - Properties.
- **Definition** panel - for details, see Views (DB2 LUW) - Definition.

3 Finally, use the **Execute** button to create the object.

Views (DB2 LUW) - Properties

When creating or editing a view, this tab/panel lets you work with the following settings:

Setting	Description
Schema	Select the owner of the view. The owner of the view must have SELECT privileges for the tables in the CREATE view statement or DBADM authority on the database that contains the table.
Name	Provide a name for the view.
Check Type	**CHECK_NONE** - No search conditions must be satisfied for insert or update operations. CHECK_LOCAL - Update and insert operations on view must satisfy the search conditions of the view and underlying views that are defined with a check option. Furthermore, every updatable view that is directly or indirectly defined on view inherits those search conditions (the search conditions of view and all underlying views of that are defined with a check option) as a constraint on insert or update operations. CHECK_CASCADED - Update and insert operations on the view must satisfy the search conditions of view and all underlying views, regardless of whether the underlying views were defined with a check option. Furthermore, every updatable view that is directly or indirectly defined on view inherits those search conditions (the search conditions of view and all underlying views) as a constraint on insert or update operations.

Views (DB2 LUW) - Definition

Complete the CREATE VIEW statement by typing or pasting in the relevant query.

IBM DB2 for z/OS Object Wizards

You create DB2 OS390 objects using the following wizards:

- Aliases Wizard (DB2 Z/OS)
- Databases Wizard (DB2 Z/OS)
- Foreign Keys Wizard (DB2 Z/OS))
- Functions Wizard (DB2 Z/OS)
- Indexes Wizard (DB2 Z/OS)
- Plans Wizard (DB2 Z/OS)
- Primary Keys Wizard (DB2 Z/OS)
- Procedures Wizard (DB2 Z/OS)
- Stogroups Wizard (DB2 Z/OS)
- Synonyms Wizard (DB2 Z/OS)
- Tables Wizard (DB2 Z/OS)
- Tablespaces Wizard (DB2 Z/OS)
- Triggers Wizard (DB2 Z/OS)
- Unique Keys Wizard (DB2 Z/OS)
- User Datatypes Wizard (DB2 Z/OS)
- Users Wizard (DB2 Z/OS)
- Views Wizard (DB2 Z/OS)

In addition, see Create Synonym.

Aliases Wizard (DB2 Z/OS)

An alias offers you security and convenience so that you can refer to an object without revealing who owns it or what database it belongs to. You can create aliases for tables or views. The Alias Wizard lets you create an alias without knowing the underlying commands. As you complete the Alias Wizard process, a CREATE ALIAS statement based on the information that you supply is generated. To create an alias, you must have CREATE ALIAS privileges or sysadmin or sysctrl authority.

To create a new alias using a wizard:

1 Open a creation wizard for an alias. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Aliases (DB2 z/OS) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Aliases (DB2 z/OS) - Properties

When creating or editing a view, this tab/panel lets you work with the following settings:

Setting	Description
Schema	Select the schema that is to own the alias.
Name	Provide a name for the alias.
Target Owner	Select the owner of the object to which you are creating an alias.
Target Type	Select the type of object (TABLE, VIEW) to which you are creating an alias.
Target Name	Select the specific object to which you are creating an alias.

Databases Wizard (DB2 Z/OS)

The Database Wizard lets you create a database (a structured collection of data that can be updated or queried) without knowing the underlying commands. Databases can be simple, that is one file with many records and the same fields, or much more complicated with multiple files with different fields.

To create a new database using a wizard:

1 Open a creation wizard for a database. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Databases (DB2 z/OS) - Properties.
- **Permissions** panel - for details, see Setting Permissions or Privileges for an Object
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Databases (DB2 z/OS) - Properties

When creating or editing a database, this tab/panel lets you work with the following settings:

Setting	Description
Name	Provide a unique name for the database.
Type	Workfile: This option in only available if the server is configured in IBM DB2 for OS/390 and z/OS to allow sharing. For more information, contact your System administrator. Temp: This option indicates the database is only for declared temporary tables. A temp database cannot be shared.
Group Member	Specifies the member for which this database is being created. Use this only in a shared environment.
Tablespace Buffer Pool	Select the default buffer pool to be used for tablespaces created within the database.
Index Buffer Pool	Select the default buffer pool name to be used for indexes created within the database.
Storage Group	Select the default storage group to support the DASD space requirements for tablespaces and indexes within the database.
Encoding Scheme	Select an encoding schema of DEFAULT, ASCII, EBCDIC, or UNICODE. NOTE: To change the encoding scheme for a database after it has been created to use a different coded character set identifier (CCSID) that supports the Euro symbol, all data must be unloaded and reloaded. For more information regarding the encoding scheme, contact your System administrator.

Foreign Keys Wizard (DB2 Z/OS))

Foreign keys are unique values that refer to specific columns of other tables. Thus, a foreign key links two tables together. The Foreign Key Wizard makes it easy for you to create a relational link between two tables, thereby speeding queries and giving you faster access to data.The Foreign Key Wizard lets you create a foreign key without knowing the underlying commands.

To create a new foreign key using a wizard:

1 Open a creation wizard for a foreign key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Foreign Keys (DB2 z/OS) - Properties.
- **Column Mapping** panel - for details, see Foreign Keys (DB2 z/OS) - Column Mapping.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Foreign Keys (DB2 z/OS) - Properties

When creating or editing a foreign key, this tab/panel lets you work with the following settings:

Setting	Description
Table Schema	Select the owner of the referencing, or child, table.
Table Name	Select the name of the referencing, or child, table.
Name	If you do not want to use the system-generated name, provide a new one.
Delete Rule	Select the action to be taken (NO ACTION, RESTRICT, CASCADE, or SET NULL) when a row of the referenced, or parent, table is deleted.

Foreign Keys (DB2 z/OS) - Column Mapping

Under **Referenced Table**, choose the **Owner** and then the **Name** of the referenced, or parent, table.

Under the **Main Table**, select check boxes corresponding to the columns that are to reference columns in the referenced table. Then, under **Referenced Table**, select the corresponding column check boxes.

Functions Wizard (DB2 Z/OS)

The Functions Wizrd lets you create a relationship between one set of values and another. You can develop reusable subroutines so you can control, access, and manipulate the data that underlies an object. As you complete the Function Wizard process, a CREATE FUNCTION statement is generated based on the information that you supply.

Note: To create a user-defined function, you need CREATE ANY privileges or IMPLICIT_SCHEMA authority on the database if the schema does not already exist.

To create a new function using a wizard:

1 Open a creation wizard for a function. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, seeFunctions (DB2 z/OS) - Properties.
- **Source** panel - (Only available for a Function Type of SOURCED.) for details, see Functions (DB2 z/OS) - Source.
- **Parameters** panel - for details, see Functions (DB2 z/OS) - Parameters.
- **Return Scalar** panel - for details, see Functions (DB2 z/OS) - Return Scalar.
- **Return Columns** panel - (Only available when you choose a Function Type of EXTERNAL TABLE.) for details, see Functions (DB2 z/OS) - Return Columns.
- **Body** panel - for details, see Functions (DB2 z/OS) - Body.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Functions (DB2 z/OS) - Properties

When creating or editing a function, this tab/panel lets you work with the following settings:

Setting	Description
Schema, **Name**, and **Specific Name**	Select the owner of the function, provide a name for the function, and provide the Specific name to be used by some SQL statements and DB2 commands for this function.
Function Type	External Scalar: This allows you to extend the function by adding your own or another party's definition for the function. External Table: Use this to create a function that is written in ASSEMBLE, C, COBOL, or PLI to return a table after it is deployed. Sourced: Here you are creating a function that is based on an existing scalar or table function with an application server. SQL: This type of function returns a single value when the function is invoked if the SQL statement that defines it is valid.
Language	If you chose a Function Type of EXTERNAL SCALAR or EXTERNAL TABLE, specify a language of ASSEMBLE, C, COBOL, or PLI.
Return Type	Identifies the return type of the function.
External Name	Provide the External Name of the function.
SQL Access Level	Indicates whether the function can execute SQL statements. CONTAINS SQL: Statements that don't read or modify SQL can be executed. NO SQL: No SQL statements can be executed. READS SQL: Statements that cannot modify SQL can be executed.
WLM Environment	Specify a Workload Management Environment (Required if Language is JAVA/COMPJAVA/REXX, the Procedure contains a LOB parameter, Security is set to 'USER' or 'DEFINER', or program type is 'SUB').
WLM For Nested	Self-explanatory

Functions (DB2 z/OS) - Source

Note: Only available for a **Function Type** of SOURCED.

Select the **Schema**, **Name**, and **Specific Name** of the source function.

Functions (DB2 z/OS) - Parameters

For each parameter for this function, use the New button to add a new parameter, provide a name for the parameter, and in the **Attributes** area, select a **Type**, and if appropriate, the **Precision**, **Scale**, **Size**, and **As Locator** options.

Functions (DB2 z/OS) - Return Scalar

Under **Return Datatype**, select a Type and depending on your choice, provide or select **Precision**, **Scale**, **Size**, and **As Locator** options.

To make use of a CAST FROM clause, under **Cast Datatype** set **Enabled** to True, select a **Type**, and if appropriate, the **Scale**, **Size**, and **As Locator** options

Functions (DB2 z/OS) - Return Columns

Note: Only available when you choose a **Function Type** of EXTERNAL TABLE.

For each column returned by this function, use the New button to add a new parameter, provide a name for the parameter, and in the **Attributes** area, select a **Type**, and if appropriate, the **Precision**, **Scale**, **Size**, and **As Locator** options. == Functions (DB2 z/OS) - Body ==

Enter the return statement for the function.

Indexes Wizard (DB2 Z/OS)

Like the index in a book, a database index makes it easier for you to access information quickly. An index lists the location of rows, sorted by the contents of one or more columns.

To create a new index using a wizard:

1 Open a creation wizard for an index. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Indexes (DB2 z/OS) - Properties
- **Columns** panel - for details, see Indexes (DB2 z/OS) - Columns
- **Indexes** panel - for details, see Indexes (DB2 z/OS) - Storage
- **Partitions** panel - for details, see Indexes (DB2 z/OS) - Partitions
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Indexes (DB2 z/OS) - Properties

When creating or editing an index, this tab/panel lets you work with the following settings:

Setting	Description
Table Schema and **Table Name**	Choose the owner and name of the table in which the index is being created.
Schema and **Name**	Choose the owner and name of the index being created.
Index Type	Unique: Prevents the selected table from having two or more rows with the same value of the index key. The uniqueness is enforced at the end of the SQL statement update. Also, null values are treated like any other values, so a column cannot contain more than one null. If you later elect to partition the index, the columns specified for the unique key must include all columns for the partitioning key. Non-Unique (default)
Clustered	Enable or disable clustering. Unless you specifically select the CLUSTER option when you create an index, the first index you create on a table will be bestowed with that distinction. Each table can only have one clustering index at a time. The clustering index tells DB2 to insert rows in the table in the order of the clustering key values. Inserted rows will be stored contiguously in sequence when possible. Columns frequently searched with a range of values BETWEEN, less than, greater than, and LIKE, or GROUP BY, etc., are good candidates for a clustering index.
Number of Partitions	If you enabled Clustering, specify the number of partitions.
Buffer Pool	Provide the buffer pool in which this index should reside
Defer, **Close**, **Copy**, and **Define**	Enable or disable these DB2 options as required.
Piece Size	The maximum addressability of the dataset for a non-partitioned index.
Compress	Enabling this setting specifies a COMPRESS=YES parameter when creating the table, resulting in a compressed index. This feature is available if the size of the specified **Buffer Pool** is 8k, 16k, or 32k.
Index on expression	Setting the check box enables the Index on Expression feature. The key expression is then provided on the **Columns** tab/panel. For details, see Indexes (DB2 z/OS) - Columns.
Padded	Setting this check box specifies that the table is to be created with the PADDED keyword. This feature must be enabled if you are using index keyrandomization. The RANDOM setting is applied on a column by column basis on the **Columns** tab/panel.

Indexes (DB2 z/OS) - Columns

The actions you take on this tab/panel depend on whether you set the **Index on Expression** check box on the Properties panel. For more information, see Indexes (DB2 z/OS) - Properties.

- If Index on Expression is disabled:

From the **Column** dropdown, select a column for the index and specify a **Sort** option.

Note: If you want to use the RANDOM sort option, you must first set the **Padded** check box on the **Properties** tab/panel.

- If Index on Expression is enabled:

Enter a valid key-expression in the **Expression** field and choose a **Sort** order.

To add more columns/key-expressions, click the **New** button and then follow the steps in the last instruction. Use the Delete button to drop columns.

Indexes (DB2 z/OS) - Storage

This tab/panel lets you perform the following tasks:

- Select a dataset management scheme
- Provide associated attribute values

To select a data set management scheme:

1 Click the **Edit** button. The **Data Set Management** dialog opens.

2 Set one of the following data set management options:

- **DB2 will define and manage the data sets on a volume of the default storage group of the database**
- **DB2 will define and manage the data sets on a volume of the specified storage group** - Select a storage group (a storage group is made up of disk device volumes): Each data set will be defined on a volume listed in the storage group you select.

Minimum primary space allocation: 12 kilobytes is the default.

Minimum secondary space allocation: 12 kilobytes is the default.

Note: NOTE: If the primary and (118 x) secondary space allocations combine to be greater than 2 gigabytes, more than one data set may eventually be used.

Erase data sets when index dropped? If choose this option, DB2 will overwrite all data with zeros before they are deleted as a security measure.

- **User will manage the data sets on a specified VCAT catalog-name** - Enter or select the VCAT. Do not select this option for an index on a declared temporary table.

3 Click **OK**.

In addition to the attributes specific to your data set management choice, this tab/panel also offers the following settings:

Free Page	One free page exists for every x pages. The x specifies how often to leave a page of free space when index entries are created from executing a DB2 utility or creating an index for a table with pre-existing rows. (0-255)
Percent Free	The percentage of free space you want to leave in every page when entries are added to an existing index. The default is 10%.
GBP Cache	This option is available only in a data-sharing environment. ALL: As pages are read, all of them will be cached in the group buffer pool. CHANGED: Updated pages are cached to the group buffer pool. NONE: No pages will be cached.

Indexes (DB2 z/OS) - Partitions

Displays the default settings for the number of partitions you specified on the **Properties** pane. Select a partition and click the **Edit** button to modify details for that partition.

Plans Wizard (DB2 Z/OS)

A plan, also known as the application plan, is a control structure that is used to process the SQL statements DB2 encounters when it is executing those SQL statements. The Plan Wizard, really, the Bind Plan Wizard, creates the structure that is used in the bind process--the process by which the output from the SQL precompiler is converted into usable form. Some authorization checking is necessary.

To Open the Bind Plan Wizard

1 On the Navigator/Explorer, find the database where you want to add the new bind plan.

2 On the **Plan** branch, right-click and select **Create**.

The Bind Plan Wizard lets you set plan parameters, add packages, and set bind properties. The table below describes the options and functionality on the Bind Plan wizard:

Panel		Settings and tasks
1	Plan Name	Lets you select the plan name.
	Qualifier	OPTIONAL: Lets you select a qualifier, the plan creator.
	Action	OPTIONAL: Lets you select an action.
	Sql Rules	OPTIONAL: Determines whether you can execute a type 2 CONNECT statement to an existing SQL connection, according to DB2 rules. Lets you select DB2 or STD.
	Cache Size	OPTIONAL: Lets you select or type the cachesize in bytes, the authorization cache acquired in the EDM pool for the plan. At run time, the authorization cache stores user IDs authorized to run. Consulting the cache can avoid a catalog lookup for checking authorization to run the plan.
	Plan Owner	OPTIONAL: Determines the authorization ID of the owner of the plan.
	Current Server	OPTIONAL: Determines the location to connect to before running the plan.
	Resource Acquire	OPTIONAL: Use - Acquires table space locks only when first used by a bound application program. Allocate - Acquires all table space locks when the plan is allocated. The value has no effect on dynamic SQL statements, which always use ACQUIRE(USE).
	Disconnect	OPTIONAL: Determines which remote connections to destroy during commit operations. The option applies to any application process that uses the plan and has remote connections of any type. Regardless of the value of this option, a commit operation destroys all connections in the release pending state. Explicit - Destroy only connections in the release pending state. This value allows you maximum flexibility for controlling remote connections. Automatic - Destroy all remote connections. Conditional - Destroy all remote connections unless an open cursor defined as WITH HOLD is associated with the connection.

2	Lets you select the **Member Name**, **PDS Name** (partitioned data set) and click **Add** to enter each member and PDS name.	
3	Lets you select the **Location** to connect to, the **Collection** (location of the DBMS where the plan binds and where the description of the plan resides.) and a **Package**.	
4	Isolation	Determines how far to isolate an application from the effects of other running applications.
	Keep Dynamic	Specifies that DB2 keeps dynamic SQL statements after commit points. The application does not need to prepare an SQL statement after every commit point. DB2 keeps the dynamic SQL statement until the application process ends, a rollback operation occurs or the application executes an explicit PREPARE statement with the same statement identifier. If the prepared statement cache is active, DB2 keeps a copy of the prepared statement in the cache. If the prepared statement cache is not active, DB2 keeps only the SQL statement string past a commit point. DB2 then implicitly prepares the SQL statement if the application executes an OPEN, EXECUTE, or DESCRIBE operation for that statement.
	Current Data	Determines whether to require data currency for read-only and ambiguous cursors when the isolation level of cursor stability is in effect. It also determines whether block fetching can be used for distributed, ambiguous cursors.
	Degree	Determines whether to attempt to run a query using parallel processing to maximize performance. Lets you select an option.
	Dynamic Rules	Determines what values apply at run time for the following dynamic SQL attributes: The authorization ID that is used to check authorization The qualifier that is used for unqualified objects The source for application programming options that DB2 uses to parse and semantically verify dynamic SQL statements Whether dynamic SQL statements can include GRANT, REVOKE, ALTER, CREATE, DROP, and RENAME statements
	Release	Determines when to release resources that a program uses. Options are at each **Commit** point or **Deallocate** when the program terminates.
	Validate	Determines whether to recheck, at run time, errors found during bind. The option has no effect if all objects and needed privileges exist. Bind - If not all objects or needed privileges exist at bind time, the wizard displays an error messages, and does not bind the package. Run - If not all objects or privileges exist at bind time, the process issues warning messages, but the bind succeeds. DB2 checks existence and authorization again at run time for SQL statements that failed those checks during bind. The checks use the authorization ID of the plan owner.
5	Explain	Obtains information about how SQL statements in the member list of the plan, are to execute, and then inserts that information into the table owner.PLAN_TABLE, where owner is the authorization ID of the owner of the plan or package. This option does not obtain information for statements that access remote objects.
	Reopt(VARS)	Re-determines the access path at run time.
	Prepare	Prepares dynamic SQL statements that refer to remote objects.

ImmedWrite	Immediate writes will be done for updates made to group buffer pool dependent pagesets or partitions.
Opthint	Query optimization hints are used for static SQL.
Encoding	Lets you select type of language for the package.
Path	Lets you select a path that DB2 uses to resolve unqualified user-defined distinct types, functions, and stored procedure names (in CALL statements).
Flag	Lets you select all message types or a specified subset to display: informational, warning, error, and completion messages.

6 **Enable** or **Disable** system connection types that can use the plan or package, select a **System** or **Cname** option.

Primary Keys Wizard (DB2 Z/OS)

A primary key is a unique key that is part of a table's definition. There can be only one primary key for each table, and the columns contained in a primary key cannot have null values. A primary key constraint forbids duplicate values in one or more columns. A table with a primary key will be considered the parent table to a table with a foreign key, which becomes the dependent table.

Note: A nullable column cannot be a part of a primary key.

To create a new primary key using a wizard:

1 Open a creation wizard for a primary key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Primary Keys (DB2 z/OS) - Properties.
- **Columns** panel - for details, see Primary Keys (DB2 z/OS) - Columns.
- **Storage** panel - for details, see Primary Keys - Storage - Edit button (manage datasets) and Primary Keys (DB2 z/OS) - Storage - Attributes.
- **Partitions** panel - for details, see Primary Keys (DB2 z/OS) - Partitions.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Primary Keys (DB2 z/OS) - Properties

When creating or editing a primary key, this tab/panel lets you work with the following settings:

Setting	Description
Table Schema and **Table Name**	Choose the owner and name of the table in which the index is being created.
Schema and **Name**	Choose the owner and name of the index being created.
Clustered	Enable or disable clustering.
Number of Partitions	If you enabled Clustering, specify the number of partitions.
Buffer Pool	Provide the buffer pool in which this index should reside
Defer, **Close**, **Copy**, and **Define**	Enable or disable these DB2 options as required.
Piece Size	The maximum addressability of the dataset for a non-partitioned index.
Time period policy	This setting is for use with tables set up as business-time tables. For more information, see Tables (DB2 z/OS v10) - Temporal Properties. When selected, a BUSINESS_TIME WITHOUT OVERLAPS option is specified against the column or columns defining the constraint, in the generated DDL. This ensures that values for the constraint columns are unique with respect to the time for the BUSINESS_TIME period. When you select this option, columns specified as business-time begin period and end period are not available as candidates on the **Columns** tab/panel.

Primary Keys (DB2 z/OS) - Columns

From the **Column** dropdown, select a column for the primary key and specify a **Sort** option. To add more columns, click the **New** button and then follow the steps in the last instruction. Use the Delete button to drop columns.

Primary Keys - Storage - Edit button (manage datasets)

Note: Availability differs according to the dataset management options you chose

Choose a data set management option:

DB2 will define and manage the data sets on a volume of the default storage group of the database

DB2 will define and manage the data sets on a volume of the specified storage group

Select a storage group (a storage group is made up of disk device volumes): Each data set will be defined on a volume listed in the storage group you select.

Minimum primary space allocation: 12 kilobytes is the default.

Minimum secondary space allocation: 12 kilobytes is the default.

Note: If the primary and (118 x) secondary space allocations combine to be greater than 2 gigabytes, more than one data set may eventually be used.

Erase data sets when index dropped? If you choose this option, DB2 will overwrite all data with zeros before they are deleted as a security measure.

User will manage the data sets on a specified VCAT catalog-name Enter or select the VCAT. Do not select this option for an index on a declared temporary table.

Primary Keys (DB2 z/OS) - Storage - Attributes

When creating or editing a primary key, this tab/panel lets you work with the following settings:

Setting	Description
Storage Group, **Primary Space Allocation**, **Secondary Space Allocation**, **Erase**, and **VCAT catalog**	The ability to set these options depends on the dataset management options you chose.
Free Page	One free page exists for every x pages. The x specifies how often to leave a page of free space when index entries are created from executing a DB2 utility or creating an index for a table with pre-existing rows. (0-255)
Percent Free	The percentage of free space you want to leave in every page when entries are added to an existing index. The default is 10%.
GBP Cache	This option is available only in a data-sharing environment. ALL: As pages are read, all of them will be cached in the group buffer pool. CHANGED: Updated pages are cached to the group buffer pool. NONE: No pages will be cached.

Primary Keys (DB2 z/OS) - Partitions

Displays the default settings for the number of partitions you specified on the **Properties** pane. Select a partition and click the **Edit** button to modify details for that partition.

Procedures Wizard (DB2 Z/OS)

Procedures are a reusable block of PL/SQL, stored in the database, that applications can call. Procedures streamline code development, debugging, and maintenance by being reusable. Procedures enhance database security by letting you write procedures granting users execution privileges to tables rather than letting them access tables directly.

To create a new procedure using a wizard:

1 Open a creation wizard for a procedure. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Procedures (DB2 z/OS) - Properties.
- **Parameters** panel - for details, see Procedures (DB2 z/OS) - Parameters.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Important Notes

- If you are creating a SQL routine procedure, you must have the latest DB2 fixpack installed on your OS/390 Client. If you do not have the latest fixpack installed, the SQL routine procedure creation will fail.

Procedures (DB2 z/OS) - Properties

When creating or editing a procedure, this tab/panel lets you work with the following settings:

Setting	Description
Schema, **Name**, and **Specific Name**	Select the schema that is to own the procedure, provide a name for the procedure, and provide the Specific Name for the procedure.
Language	The database manager will call the procedure accordingly assuming the program is designed to run in the server's environment. Assemble: a stored procedure written in Assembler C: a stored procedure written in C or C++ COBOL: a stored procedure written in COBOL CompJAVA: CompJAVA is no longer supported. Stored procedures should alternatively be written in JAVA. JAVA PLI: A stored procedure written in PL/I. REXX (Restructured Extended Executor Language) - Don't use this language when SQL parameter style is in effect. To specify REXX, the general parameter style or general with nulls. SQL
SQL Access Level	MODIFIES SQL DATA (Default): The procedure can support any SQL statement except those that cannot be supported in procedures. CONTAINS SQL DATA: Only SQL statements that neither modify nor read SQL data can be executed in the procedure. READS SQL DATA Some SQL statements that don't modify SQL data can be included in the procedure NO SQL: Only SQL statements with a data access classification of NO SQL can be executed. Don't select this option for a JAVA procedure that uses a .jar.
WLM Environment	Specify a Workload Management Environment (Required if Language is JAVA/COMPJAVA/REXX, the Procedure contains a LOB parameter, or Security is set to 'USER' or 'DEFINER').
WLM For Nested	Self-explanatory

Procedures (DB2 z/OS) - Parameters

For each parameter for this function, use the New button to add a new parameter, provide a name for the parameter, and in the **Attributes** area, select a **Type**, specify a **Parameter Mode** of INPUT, OUTPUT, or INPUT_OUTPUT, and if appropriate, the **Precision**, **Scale**, and **Size** options.

Stogroups Wizard (DB2 Z/OS)

Stogroups are storage groups. You create them on the current server. Storage from the named sets of volumes you create can be allocated at a later date for tablespaces or index spaces.

To Open the Stogroup Wizard

1 On the Navigator/Explorer, find the datasource where you want to add the new Storage Group.

2 Expand the Storage branch, right-click **Stogroup**, and select **New**.

The table below describes the fields you will encounter as you complete the Stogroup Wizard.

Required Information	Description
What is the name of the Stogroup?	Enter a name for the storage group.
VCAT	This is the integrated catalog facility catalog, or volume catalog. Name the catalog or choose one from the drop-down list if it is available.
Select the volumes in the Stogroup	Specify a set of volumes that may exist on the system but may not be in use by other storage groups.
Select All	Selects all listed volumes.
Unselect All	Unselects all listed volumes.
Add	Opens the Add Volume Dialog Box.
Remove	Deletes all selected volumes from the list.

Add Volume Dialog Box

The table below describes the options and functionality on the **Add Volume** dialog:

Required Information	Description
Enter one or more volumes to add to the stogroup	Type the names of the volumes (separated by spaces) to add to the stogroup.
Check	Click to see if any additional information is available about the volumes you typed. Opens the Volumes Info Dialog Box.
Or select volumes for the list	Lets you select volumes.

Volumes Info Dialog Box

The table below describes the options and functionality on the **Volume Info** dialog:

Required Information	Description
Volumes	Lets you select volumes.
OK	Click to add volumes to Stogroup Wizard.

Synonyms Wizard (DB2 Z/OS)

A synonym is an alternative name you can create for a table or view. Using a synonym can make it easier for you to remember that table or view instead of having to think about the possibly cumbersome formal name (for example, a show dog may have a formal name that incorporates all his ancestors, but will answer to the name Spot around the house).

To create a new synonym using a wizard:

1 Open a creation wizard for a synonym. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Synonyms (DB2 z/OS) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Synonyms (DB2 z/OS) - Properties

When creating or editing a synonym, this tab/panel lets you work with the following settings:

Setting	Description
Schema	Select the schema that is to own the synonym.
Name	Provide a name for the synonym.
Referenced Object Owner	Select the owner of the object to which you are creating a synonym.
Referenced Object Type	Select the type of object (TABLE, VIEW) to which you are creating a synonym.
Referenced Object Name	Select the specific object to which you are creating a synonym.

Tables Wizard (DB2 Z/OS)

All data in a database is stored in a tabular format, that is a collection of rows and columns. Tables, therefore are fundamental to whatever database you are administering.

To create a new table using a wizard:

1 Open a creation wizard for a table. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Tables (DB2 z/OS) - Properties.
- **Columns** panel - for details, see Tables (DB2 z/OS) - Columns.
- **Indexes** panel - for details, see Tables (DB2 z/OS) - Indexes.
- **Temporal Properties** panel - for details, see Tables (DB2 z/OS v10) - Temporal Properties.
- **Constraints** panel - for details, see Tables (DB2 z/OS) - Constraints.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **Permissions** panel - Setting Permissions or Privileges for an Object
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

The table that follows describes the fields you may encounter as you complete the Table Wizard.

Tables (DB2 z/OS) - Properties

Select the **Schema**, provide a **Name**, and provide or select other table properties.

Tables (DB2 z/OS) - Columns

For each column in the table, click the **Add Column** button to create a column, provide a **Name** for the column and provide or select the remaining column attributes.

Use the **Delete** button to drop a selected column.

Use the arrow buttons to reposition the columns.

Row Properties (V10) settings are available when you select a **Type** of TIMESTAMP. These settings are used in conjunction with settings on the **Temporal Properties** tab in setting up system and application temporal tables. For more information, see Tables (DB2 z/OS v10) - Temporal Properties.

Specific settings are:

- **As Row Begin** - adds a GENERATED ALWAYS AS ROW BEGIN option to the column specification, used to track when a row is current.

- **As Row End** - adds a GENERATED ALWAYS AS ROW END option to the column specification.
- **Transaction Start ID** - adds a GENERATED ALWAYS AS TRANSACTION START ID option to the column specification indicating that this column captures the start times for transactions that affect rows.

A column can have only one of the three settings selected.

When you select a **Row Properties** setting, the **Scale** value and if appropriate, the **Allow Nulls**, **Default Value**, and **Expression** settings are automatically adjusted for compatibility with row generation. Similarly, if one of the three **Row Properties** settings is selected for a column, and any of the **Scale**, **Allow Nulls**, **Default Value**, or **Expression** settings are modified, then the selected **Row Properties** setting is cleared.

Tables (DB2 z/OS) - Indexes

Click **Add** to open the Indexes Wizard (DB2 Z/OS).

Tables (DB2 z/OS v10) - Temporal Properties

This tab/panel lets you specify temporal properties to a table, designating it as a table that records the period of time when a row is valid.

Note: Before using this object action, consult DB2 LUW documentation for details on setup and use of the temporal tables feature. For more information, see Accessing Third Party Documentation.

This tab/panel has the following controls:

- **System-time table** - selecting this option creates a table with the PERIOD SYSTEM_TIME clause, designating it as a system-period temporal table. This option is available if the table has three columns of type TIMESTAMP, and the following prerequisites have been met:
- At least one TIMESTAMP-typed column has the **As Row Begin** property selected
- At least one TIMESTAMP-typed column has the **As Row End** property selected
- At least one TIMESTAMP-typed column has the **Transaction Start ID** property selected

For details on column definitions, see Tables (DB2 z/OS) - Columns.

The **Row-begin column**, **Row-end column**, and **Transaction Start ID column** controls are automatically populated with the columns having the corresponding property.

When you select this option, in the generated DDL, the **Row-begin column** and **Row-end column** values are used as the PERIOD SYSTEM_TIME column parameters, defining the system period.

- **Business-time table** - selecting this option creates a table with the PERIOD BUSINESS_TIME clause, designating it as an application-period temporal table. This option is available if the table has two columns of type DATE or TIMESTAMP, for which no **Row Properties** have been assigned.

The **Begin column** and **End Column** controls are automatically populated with the valid candidate columns from the table's column list and the first distinct column pair is automatically chosen for convenience.

- **Use history table** - this set of controls is only active when **Business-time table** is selected. It lets you associate a history table. When you select **Use history table**, the Owner and Name controls are activated, letting you identify the history table to associate with this table.

Note that a constraint created on a business-time table can enforce the uniqueness of the data for any point in business time via the **Time period policy** property that has been added in the constraint wizards. For details, see the following topics:

- Primary Keys (DB2 z/OS) - Properties
- Unique Keys (DB2 z/OS) - Properties

Tables (DB2 z/OS) - Constraints

Selecting a constraint type and clicking **Add** opens the object wizard for that object type. For details see:

- Foreign Keys Wizard (DB2 Z/OS))
- Primary Keys Wizard (DB2 Z/OS)
- Unique Keys Wizard (DB2 Z/OS)
- Create Synonym

Tablespaces Wizard (DB2 Z/OS)

Tablespaces establish connections between the physical storage devices of your database system and the logical containers or tables being used to store data. In essence, a tablespace is a storage structure that can hold tables, indexes, large objects, and long data.

Note: To change the encoding scheme for a database after it is created to utilize a different coded character set identifier (CCSID) that supports the Euro symbol, all data must be unloaded and reloaded. For more information regarding the encoding scheme, contact your System administrator.

To Open the Tablespace Wizard

1 On the Navigator/Explorer, find the datasource where you want to add the new Tablespace.

2 Expand the Storage branch, right-click **Tablespaces**, and select **New**.

3 Use the following table as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Tablespaces (DB2 z/OS) - Properties.
- **Permissions** panel - Setting Permissions or Privileges for an Object
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

4 Finally, use the **Execute** button to create the object.

Tablespaces (DB2 z/OS) - Properties

When creating or editing a tablespace, this tab/panel lets you work with the following settings:

Group	Settings	Description
Creation	**Name and Database**	Provide a name and select an assocaited database.
	Database type	Lets you select a REGULAR or WORKFILE database type.
	Creation type	Lets you select among PARTITIONED, SEGMENTED, RANGE-PARTITIONED UNIVERSAL, PARTITION-BY-GROWTH UNIVERSAL, and LOB options. Refer to DB2 z/OS product documentation for detailed information on these types and other required settings. For more information, see Accessing Third Party Documentation.

Tablespace management	**Management type**	Lets you select how you want the tablespace managed: DEFAULT STORAGE GROUP, STORAGE GROUP, or VCAT CATALOG-NAME.
	Storage group, Minimum primary space allocation, Minimum secondary space allocation, and Erase rule	If you selected a **Management type** of STORAGE GROUP, select a storage group, specify minimum space allocations, and specify whether data sets are to be erased when the tablespace is dropped.
	VCatName	If you selected a **Management type** of VCAT catalog-name, select a catalog name.
Bufferpool	**Buffer Pool**	Select the buffer pool where the tablespace is to reside.
Partitions and size	**Number of partitions**	If you selected a **Creation Type** of PARTITIONED or RANGE-PARTITIONED UNIVERSAL, specify the number of partitions.
	Partition size (DSSIZE)	If you did not select a **Creation Type** of SEGMENTED, select a partition size.
	Segment size	If you selected a **Creation Type** of SEGMENTED, PARTITION-BY GROWTH, or RANGE-PARTITIONED UNIVERSAL, select the number of pages that are to be assigned to each segment of the tablespace.
	Max rows per page	Specify the number of rows (1-255) that can be placed on each data page.
Space Management	**Free space portion of each page (%)**	If you did not select a **Creation Type** of LOB, specify the percentage of each page to be left as free space when a tablespace is reloaded or reorganized.
	Free page frequency	If you did not select a **Creation Type** of LOB, specify how often to leave a free page when a tablespace is reloaded or reorganized.
Other parameters	**GBPCACHE**	Select a group buffer pool cache scheme of CHANGED, ALL, SYSTEM, or NONE.
	Compress	Enable this setting to allow compression.
	Track modified pages	Enable this setting to track modified pages in the space map pages.
	Encoding scheme	Select an encoding scheme of EBCDIC, UNICODE, ASCII, or NONE.
	Log	Enabling this check box specifies that changes to a LOB column are to be written to the log.
	Define	Enabling this check box specifies that tablespace data sets are defined when the tablespace is created. Otherwise, data sets are not created until data is inserted.

Member Cluster	Enable this setting to manage space for inserts on a member-by-member basis.
Close rule	Enable this setting to specify CLOSE YES, dictating the data set close priority.
Lock Size	Specify a lock size of ANY, TABLESPACE, PAGE, or ROW.
Maximum locks	Specify the number of locks that are allowed before escalating.

Triggers Wizard (DB2 Z/OS)

All data in a database is stored in a tabular format, that is a collection of rows and columns. Tables, therefore are fundamental to whatever database you are administering.

To create a new trigger using a wizard:

1 Open a creation wizard for a trigger. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Triggers (DB2 z/OS) - Properties.
- **Column Selection** panel - for details, see Triggers (DB2 z/OS) - Column Selection.
- **Definition** panel - for details, see Triggers (DB2 z/OS) - Definition.

3 Finally, use the **Execute** button to create the object.

Triggers (DB2 z/OS) - Properties

When creating or editing a trigger, this tab/panel lets you work with the following settings:

Setting	Description
Table Schema and **Table Name**	Choose the owner and name of the table for which the trigger is being created.
Schema and **Name**	Choose the owner and name of the trigger being created.
Trigger Timing	BEFORE: These triggers serve as extensions to the constraint subsystem and are most often used to validate input data, generate values for newly inserted rows, and read from other tables for cross-reference purposes. Note: Before triggers must be created as a For Each Row. AFTER: Such a trigger is run after the integrity constraint validations; they can be used to modify operations in the database or be used for activities beyond the database, like supporting an alert notification.
Trigger Events	An INSERT trigger must be associated with an INSERT statement. For example, if a data load operation doesn't include an INSERT statement, the trigger will not be invoked. An UPDATE trigger can be associated with specific columns of the base table and will only be activated if those columns are updated.
Trigger Type	**STATEMENT**: (only fires once). **ROW** (fires for each affected row): The trigger runs as many times as there are rows in the affected section. If the set of affected rows is empty, the trigger doesn't run.
Old Table Alias	Type the name of a temporary table of rows as they exist before they're updated or deleted.
New Table Alias	Type a name for a temporary table of rows as they exist after they're inserted or updated.
Old Row Alias	Type a name for the rows as they are before they've been deleted or updated.
New Row Alias	Type a name for the rows as they are after they've been inserted or updated.

Triggers (DB2 z/OS) - Column Selection

If you chose UPDATE as the **Trigger Event**, select the columns, select the check box beside each column that is to fire the trigger.

Triggers (DB2 z/OS) - Definition

Complete the CREATE TRIGGER outline provided by typing or pasting the body of the trigger.

Unique Keys Wizard (DB2 Z/OS)

A unique key constraint is a key for which no two of its values can be equal and no values can be null. A table can have a number of unique constraints, but it cannot have more than one unique constraint on the same set of columns. If you are creating a unique key constraint on a table that already exists (as opposed to creating a unique key at the time the table is first generated), a unique index must already exist on the columns of the unique key you want to constrain. If no unique index exists, the Index Wizard will open as you complete the **Create Unique Key Constraint** dialog.

To create a new unique key using a wizard:

1 Open a creation wizard for a unique key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Unique Keys (DB2 z/OS) - Properties.
- **Columns** panel - for details, see Unique Keys (DB2 z/OS) - Columns.
- **Storage** panel - for details, see Unique Keys (DB2 z/OS) - Storage - Edit button (manage datasets) and Unique Keys (DB2 z/OS) - Storage - Attributes.
- **Partitions** panel - for details, see Unique Keys (DB2 z/OS) - Partitions.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Unique Keys (DB2 z/OS) - Properties

When creating or editing a primary key, this tab/panel lets you work with the following settings:

Setting	Description
Table Schema and **Table Name**	Choose the owner and name of the table in which the unique key is being created.
Schema and **Name**	Choose the owner and name of the index being created.
Clustered	Enable or disable clustering.
Number of Partitions	If you enabled Clustering, specify the number of partitions.
Buffer Pool	Provide the buffer pool in which this unique key should reside
Defer, **Close**, **Copy**, and **Define**	Enable or disable these DB2 options as required.
Piece Size	The maximum addressability of the dataset for a non-partitioned index.
Time period policy	This setting is for use with tables set up as business-time tables. For more information, see Tables (DB2 z/OS v10) - Temporal Properties. When selected, a BUSINESS_TIME WITHOUT OVERLAPS option is specified against the column or columns defining the constraint, in the generated DDL. This ensures that values for the constraint columns are unique with respect to the time for the BUSINESS_TIME period. When you select this option, columns specified as business-time begin period and end period are not available as candidates on the **Columns** tab/panel.

Unique Keys (DB2 z/OS) - Columns

From the **Column** dropdown, select a column for the unique key and specify a **Sort** option. To add more columns, click the **New** button and then follow the steps in the last instruction. Use the **Delete** button to drop columns.

Unique Keys (DB2 z/OS) - Storage - Edit button (manage datasets)

Note: Availability differs according to the dataset management options you chose

Choose a data set management option:

DB2 will define and manage the data sets on a volume of the default storage group of the database

DB2 will define and manage the data sets on a volume of the specified storage group

Select a storage group (a storage group is made up of disk device volumes): Each data set will be defined on a volume listed in the storage group you select.

Minimum primary space allocation: 12 kilobytes is the default.

Minimum secondary space allocation: 12 kilobytes is the default.

Note: If the primary and (118 x) secondary space allocations combine to be greater than 2 gigabytes, more than one data set may eventually be used.

Erase data sets when index dropped? If you choose this option, DB2 will overwrite all data with zeros before they are deleted as a security measure.

User will manage the data sets on a specified VCAT catalog-name Enter or select the VCAT. Do not select this option for an index on a declared temporary table.

Unique Keys (DB2 z/OS) - Storage - Attributes

Note: Availability differs according to the dataset management options you chose

When creating or editing a unique key, this tab/panel lets you work with the following settings:

Setting	Description
Storage Group, **Primary Space Allocation**, **Secondary Space Allocation**, **Erase**, and **VCAT catalog**	The ability to set these options depends on the dataset management options you chose.
Free Page	One free page exists for every x pages. The x specifies how often to leave a page of free space when index entries are created from executing a DB2 utility or creating an index for a table with pre-existing rows. (0-255)
Percent Free	The percentage of free space you want to leave in every page when entries are added to an existing index. The default is 10%.
GBP Cache	This option is available only in a data-sharing environment. ALL: As pages are read, all of them will be cached in the group buffer pool. CHANGED: Updated pages are cached to the group buffer pool. NONE: No pages will be cached.

Unique Keys (DB2 z/OS) - Partitions

Displays the default settings for the number of partitions you specified on the **Properties** pane. Select a partition and click the **Edit** button to modify details for that partition.

User Datatypes Wizard (DB2 Z/OS)

A datatype is a named set of valid values that can be manipulated by a set of operations. There are intrinsic datatypes, which are predefined and always available, and derived datatypes. A derived datatype is a user-defined datatype, which can include both intrinsic and previously derived datatypes. The User Datatype Wizard lets you create a derived datatype without knowing the underlying commands.

To create a new user datatype using a wizard:

1 Open a creation wizard for a user datatype. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see User Datatypes (DB2 z/OS) - Properties.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

User Datatypes (DB2 z/OS) - Properties

When creating or editing a user datatype, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the owner of the user datatype.
Datatype	Provide a name for the datatype.
Type	Select the base datatype.
Size	Provide the size of the datatype.
For Data	Select the MIXED, SBCS, or BIT option for the datatype.
CCSID	Select the NONE, ASCII, EBCDIC, or UNICODE option for the datatype.

Users Wizard (DB2 Z/OS)

Users have authorization to use a database and its objects, and the User Wizard gives you an easy way to add new ones.

To create a new user using a wizard:

1 Open a creation wizard for a user. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Users (DB2 z/OS) - Properties.
- **Object Permissions** and **System Permissions** panels - Setting Permissions or Privileges for an Object
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Users (DB2 z/OS) - Properties

When creating or editing a user datatype, this tab/panel lets you provide the user **Name**.

Views Wizard (DB2 Z/OS)

A view gives you a new way of looking at data in a results table. Views behave like tables because you can query views and perform data manipulation operations on them. However, views do not actually store any data. Instead, they depend on data contained in their base tables. Columns added to the base table(s) after the view is created are not included in the result set. Views are thus handy tools for controlling access to a table. You can allow someone to see portions of data without allowing that user to see the table in its entirety. For example, you can create a view that will permit a user to see employee names in a table without allowing access to the Social Security numbers of that same table.

The wizard itself is a single panel. After you complete the wizard, the View Editor opens so you can complete the definition of the view, choose the columns to show in the view, the dependencies, and access privileges to the view.

To create a new view using a wizard:

1 Open a creation wizard for a view. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Views (DB2 z/OS) - Properties.
- **Definition** panel - for details, see Views (DB2 z/OS) - Definition.

3 Finally, use the **Execute** button to create the object.

Views (DB2 z/OS) - Properties

When creating or editing a view, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the owner of the view. The owner of the view must have SELECT privileges for the tables in the CREATE view statement or DBADM authority on the database that contains the table.
Name	Provide a name for the view.
Check Type	CHECK_NONE - No search conditions must be satisfied for insert or update operations. CHECK_LOCAL - Update and insert operations on view must satisfy the search conditions of the view and underlying views that are defined with a check option. Furthermore, every updatable view that is directly or indirectly defined on view inherits those search conditions (the search conditions of view and all underlying views of that are defined with a check option) as a constraint on insert or update operations. CHECK CASCADED - Update and insert operations on the view must satisfy the search conditions of view and all underlying views, regardless of whether the underlying views were defined with a check option. Furthermore, every updatable view that is directly or indirectly defined on view inherits those search conditions (the search conditions of view and all underlying views) as a constraint on insert or update operations.

Views (DB2 z/OS) - Definition

Complete the CREATE VIEW statement by typing or pasting in the relevant query.

Microsoft SQL Server Object Wizards

You create SQL Server objects using the following wizards:

- Alerts Wizard (DBArtisan - SQL Server)
- Asymmetric Keys Wizard (SQL Server)
- Backup Devices Wizard (DBArtisan - SQL Server)
- Certificate Wizard (SQL Server)
- Databases Wizard (SQL Server)
- Database Triggers Wizard (SQL Server)
- Defaults Wizard (SQL Server)
- Extended Procedures Wizard (SQL Server)
- Foreign Keys Wizard (SQL Server)
- Full-text Catalogs Wizard (SQL Server)
- Full-text Indexes Wizard (SQL Server)
- Functions Wizard (SQL Server)
- Indexes Wizard (SQL Server)
- Jobs Wizard (DBArtisan - SQL Server)
- Linked Servers Wizard (DBArtisan - SQL Server)
- Logins Wizard (SQL Server)
- Operators Wizard (DBArtisan - SQL Server)
- Partition Functions Wizard (SQL Server)
- Partition Schemes Wizard (SQL Server)
- Primary Keys Wizard (SQL Server)
- Procedures Wizard (SQL Server)
- Remote Servers Wizard (DBArtisan - SQL Server)
- Roles Wizard (SQL Server)
- Rules Wizard (SQL Server)
- Schema Wizard (SQL Server)
- Sequences Wizard (SQL Server)
- Server Triggers Wizard (DBArtisan - SQL Server)

  - Symmetric Keys Wizard (SQL Server)
  - Synonyms Wizard (SQL Server)
  - Tables Wizard (SQL Server)
  - Triggers Wizard (SQL Server)
  - Unique Keys Wizard (SQL Server)
  - User Messages Wizard (SQL Server)
  - Users Wizard (SQL Server)
  - User Datatypes Wizard (SQL Server)
  - Views Wizard (SQL Server)

Alerts Wizard (DBArtisan - SQL Server)

The Alerts Wizard lets you build and submit an **sp_add_alert** call to create a SQL Agent Alert. You can create event-based, performance-based, or WMI-based alerts. Job-based and operator-based responses are available.

Note: For an overview of SQL Agent service support, see SQL Server Agent.

To create a new alert using a wizard:

1 Open an object creation wizard for an alert. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details see Alerts (SQL Server) - Properties.
- **Response** panel - for details see Alerts (SQL Server) - Response.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

For information on related SQL Agent elements, see the following topics:

- Jobs Wizard (DBArtisan - SQL Server)
- Operators Wizard (DBArtisan - SQL Server)

Alerts (SQL Server) - Properties

When creating or editing an alert, this tab/panel lets you select an alert type and provide details on the type selected. It has the following settings:

Category	Property	Description
Creation	**Name**	Provide a name for the alert.
	Enabled	Select this check box to enable the alert.
	Type	Select an alert type of SQL SERVER EVENT ALERT, SQL SERVER PERFORMANCE CONDITION ALERT, or WMI EVENT ALERT
Alert definition	If you selected a **Type** of SQL SERVER EVENT ALERT, select or provide the following details	
	Error #	Type a valid, native or user-defined error number.
	Error Description	Displays the textual description for the **Error #** you specified.
	Severity	Select a severity to be associated/displayed for this alert.

Database	If you want to restrict this alert to be reported on a specific database, select a specific database.
Event Description	Provide a textual description to be displayed with the alert.

If you selected a **Type** of SQL SERVER PERFORMANCE CONDITION ALERT, select or provide the following details

Object	Select the performance object to be monitored. Available performance objects differ by the version of SQL Server you are using. They MAY include: **Access Methods**, Backup Device, **Broker Activation**, **Broker Statistics**, **Broker/DBM Transport**, **Buffer Manager**, **Buffer Node**, **Buffer Partition**, Cache Manager, **Catalog Metadata**, **CLR**, **Cursor Management by Type**, **Cursor Management Total**, Database Mirroring, **Databases**, **Deprecated Features**, **Exec Statistics**, **General Statistics**, **Latches**, **Locks**, **Memory Manager**, **Plan Cache**, **Resource Pool Stats**, **SQL Errors**, **SQL Statistics**, **Transactions**, **User Settable**, **Wait Statistics**, and **Workload Group Stats**.
Counter	Select a counter specific to the performance **Object** you selected.
Instance	If a specific instance is available for the **Object** and **Counter** you selected, select an instance.
Alert if counter and Value	Use these settings to select the relational operator (<, =,or >) and value that make up the condition for which the alert is generated.

If you selected a **Type** of WMI EVENT ALERT, select or provide the following details

Namespace	Type the WMI namespace for the WMI Query Language (WQL) statement that identifies which WMI event will trigger this alert.
Query	Type the WQL statement that identifies the event that this alert responds to.

Alerts (SQL Server) - Response

When creating or editing an alert, this tab/panel lets you specify details of responses to the alert. It lets you specify a job to be executed as well as details of email, pager or net send notifications. You work with the following settings:

Category	Property	Description
Job Properties	**Execute Job**	Select this check box to have a job executed on this alert.
	Job Name	If you selected **Execute Job**, select the specific job to execute.

Include alert error text in	Use the **Email**, **Pager**, and **Net send** controls to specify whether a textual description of the SQL Server error is to be included as part of the notification message for the target media.	
Additional Properties	**Notification message**	Use this control to add an optional additional message that will be sent to the operator as part of the notification.
	Delay Between Responses	The wait period, in seconds, between responses to the alert.
Operators to notify	The list displays all defined operators. Use the **Email**, **Pager**, and **Netsend** controls to select the operators and the media by which they will be contacted for this alert.	

Note: The **New Job** and **New Operator** buttons let you open the Jobs Wizard and Operators Wizard, respectively. For details, see Jobs Wizard (DBArtisan - SQL Server) and Operators Wizard (DBArtisan - SQL Server).

Asymmetric Keys Wizard (SQL Server)

This wizard lets you build and submit a CREATE ASYMMETRIC KEY statement with options corresponding to the contained private and public keys.

To create a new asymmetric key using a wizard:

1 Open a creation wizard for an asymmetric key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Asymmetric Keys (SQL Server) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Asymmetric Keys (SQL Server) - Properties

When creating or editing an asymmetric key, this tab/panel lets you work with the following settings:

Setting	Availability and description	
Authorization Owner	The name of the user that will own the certificate.	
Name	The name for the certificate.	
Use Existing Keys	Selected, a WITH ALGORITHM option variation of the CREATE ASYMMETRIC KEY statement is generated. Deselected, a FROM FILE, FROM EXECUTABLE FILE, FROM ASSEMBLY, or FROM PROVIDER option variation of the CREATE ASYMMETRIC KEY statement is generated.	
Provider Name	Available if **Use Existing Keys** is not selected.	Lets you provide a FROM PROVIDER argument value, specifying an Extensible Key Management provider and name.
Key Algorithm	Available if **Use Existing Keys** is not selected.	Lets you select a WITH ALGORITHM value of RSA_512, RSA_1024, or RSA_2048, specifying the key length.
Key Name In Provider	Available if **Use Existing Keys** is not selected, after **Provider Name** is entered.	Lets you provide a PROVIDER_KEY_NAME value, specifying the key name from the external provider.
Creation Disposition		Lets you select a CREATION_DISPOSITION value of OPEN_EXISTING or CREATE_NEW, specifying whether the asymmetric key is mapped to an existing EKM key or a new key is created on the EKM device.
File	Available if **Use Existing Keys** is selected.	Let you provide FROM FILE, FROM EXECUTABLE FILE, FROM ASSEMBLY, or FROM PROVIDER option values, respectively.
Executable File		
Assembly		
Provider Name		
Encryption Password	Available if **Use Existing Keys** is not selected and after **Subject** is provided.	Lets you provide an ENCRYPTION BY PASSWORD = clause value, specifying the password used to encrypt the private key.

Backup Devices Wizard (DBArtisan - SQL Server)

The Backup Device Wizard lets you add a disk or tape backup device to a SQL Server instance. Backup devices store backups of databases and transaction logs.

Before you can dump a database or its transaction log, first link to a backup device that stores the resulting backup file.

To create a new backup device using a wizard:

1 Open a creation wizard for a backup device. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Backup Devices (SQL Server) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Backup Devices (SQL Server) - Properties

When creating or editing a backup device, this tab/panel lets you work with the following settings:

Setting	Description
Name	Lets you provide a logical name for the backup device.
Device Type	DISK and TAPE are supported.
Physical Name	The physical name of the device. This value can be any simple file name (sqltable.dat for example) or full path (a:\sqltable.dat for example) that is recognized by the operating system.

Certificate Wizard (SQL Server)

This wizard lets you build and submit a CREATE CERTIFICATE statement, adding a certificate to a database.

To create a new certificate using a wizard:

1 Open a creation wizard for a certificate. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Certificates (SQL Server) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Certificates (SQL Server) - Properties

When creating or editing a certificate, this tab/panel lets you work with the following settings:

Setting	Availability and description	
Owner	The name of the user that will own the certificate.	
Name	The name for the certificate.	
Use Existing Keys	Lets you choose between EXISTING KEYS and GENERATE NEW KEYS options.	
Active For Begin Dialog	Lets you select ON/OFF values for a ACTIVE FOR BEGIN_DIALOG = option.	
Subject	Available if **Use Existing Keys** is not selected.	Lets you provide a WITH SUBJECT = clause value, referring to a field in the certificate metadata.
StartDate and **ExpiryDate**	Available if **Use Existing Keys** is not selected and after **Subject** is provided.	Lets you provide START_DATE = and EXPIRY_DATE = values, defining the period that the certificate is valid
Assembly ExecutableFile File	Available if **Use Existing Keys** is selected.	These controls let you select among FROM ASSEMBLY =, FROM FILE =, FROM EXECUTABLE FILE = clauses and provide the relevant signed assembly, path to a DER-encoded certificate file, or path to a private key file.
PrivateKeyFile	Available if **Use Existing Keys** is selected.	Lets you provide a WITH PRIVATE KEY (FILE =... clause value, specifying that the certificate's private key is loaded.
Encryption Password	Available if **Use Existing Keys** is not selected and after **Subject** is provided.	Lets you provide an ENCRYPTION BY PASSWORD = clause value, specifying the password used to encrypt the private key.
DecryptionPassword	Available if **Use Existing Keys** is not selected. Also available if **Use Existing Keys** is selected and after an **Assembly** or **PrivateKeyFile** value is provided.	Lets you provide a DECRYPTION BY PASSWORD = clause value, specifying the password used to decrypt a private key retrieved from file.

Databases Wizard (SQL Server)

The Database Wizard presents you with a different set of options based on your server version to create the database accurately on each platform.

Tip: Microsoft SQL Server recommends that you do not create any user objects, such as tables, views, stored procedures, or triggers, in the master database. The master database includes the system tables that store the system information used by SQL Server, such as configuration option settings.

To create a new database using a wizard:

1 Open a creation wizard for a database. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Options** panel - for details, see Databases (SQL Server) - Options.
- **Placement** panel - for details, see Databases (SQL Server) - Placement.
- **Transaction Log** panel - for details, see Databases (SQL Server) - Transaction Log.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Databases (SQL Server) - Options

When creating or editing a database, this tab/panel lets you work with the following settings:

Setting	Description
Name	Provide a name for the database.
Attach existing OS files	To create a database from an existing set of operating system files, there must be a <filespec> entry for the first PRIMARY file. The PRIMARY filegroup contains all the database system tables. Primary files have a .mdf extension.
Compatible Level	Select a version compatibility level.
Properties group	Select the following settings: **ANSI null default**, **ANSI nulls**, **ANSI padding**, **ANSI warnings,auto create statistics**, **auto update statistics**, **autoclose**, **autoshrink**, **concat null yields null**, **cursor close on commit**, **arithabort**, **db chaining**, **dbo use only**, **default to local cursor**, **merge publish**, **numeric roundabout**, **offline**, **published**, **quoted identifier**, **read only**, **recursive triggers**, **select into/bulkcopy/pllsort**, **single user**, **subscribed**, **torn page detection**, and **trunc log on chkpt**.

Databases (SQL Server) - Placement

Indicate the file where you want the database to live. For example, a new Books database could include author and title filegroups.

By default, when you open the Wizard and click the **Placement** tab, a filegroup definition, using the name you provided for the database and default settings, is displayed. For each filegroup to be added, click the **New** button, provide a **Device File Name** for the filegroup, and use the **File Group Properties** and **Device File Properties** groups to provide the attributes of the filegroup.

Use the Delete button to delete a selected filegroup.

Databases (SQL Server) - Transaction Log

The transaction log file is a required file for each database. This file holds the log information to recover the database. There can be multiple log files for a database, but there has to be at least one. Traditionally the logfile extension has been .ldf.

By default, when you open the Wizard and click the **Transaction Log** tab, a transaction log file definition, using the name derived from the name you provided for the database and with default settings, is displayed. For each file to be added, click the **New** button, provide a **Device File Name**, and use the **Log Device Properties** group to provide the attributes of the file.

Use the Delete button to delete a selected file.

Note: As you complete the wizard, be aware that the Primary file contains startup information for the database and is also used to store data. The transaction log files hold the information used to recover the database.

Database Triggers Wizard (SQL Server)

This wizard builds and submits a CREATE TRIGGER... ON DATABASE statement, letting you create a DDL trigger with database scope, that fires in response to selected DDL events. The wizard builds basic syntax only, letting you provide the actions and conditions SQL making up the main body of the trigger.

Note: Before working with database triggers, consult Microsoft SQL Server documentation for general information on DDL triggers as well as specifics such as FOR/AFTER argument details, valid actions and conditions, and supported events. For more information, see Accessing Third Party Documentation.

To create a new database trigger using a wizard:

1 Open a creation wizard for a database trigger. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Database Triggers (SQL Server) - Properties.
- **Trigger Events** panel - for details, see Database Triggers (SQL Server) - Trigger Events.
- **Definition** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Note: In order to fire in response to specified events, a database trigger must be enabled. For information on enabling and disabling a database trigger, see Database Triggers Editor (SQL Server) and Change Status (SQL Server DDL triggers).

Database Triggers (SQL Server) - Properties

When creating or editing a database trigger, this tab/panel lets you work with the following settings:

Setting	Description
Name	Provide a name for the DDL trigger.
Trigger Timing	Use this control to specify a FOR/AFTER argument option. The specific events for the argument are provided on the **Trigger Events** tab/panel.
Encrypted	Select this check box to add a WITH ENCRYPTION argument to the generated DDL. This prevents the trigger from being published as part of a SQL Server replication.

Database Triggers (SQL Server) - Trigger Events

When creating or editing a database trigger, this tab/panel lets you specify the FOR/AFTER argument appearing in the generated CREATE TRIGGER... ON DATABASE syntax. For each event that is to fire this trigger, select the associated check box.

Defaults Wizard (SQL Server)

When bound to a column or user-defined object, a default ensures that a specific value will be inserted into the column where the object will be bound if no explicit value is given.

The Default Wizard lets you name the default and specify its value.

To create a new default using a wizard:

1 Open a creation wizard for a default. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Defaults (SQL Server) - Properties.
- **Dependencies** panel - Defaults (SQL Server) - Dependencies.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Defaults (SQL Server) - Properties

When creating or editing a default, this tab/panel lets you work with the following settings:

Setting	Description
Schema	Select the schema that is to own the default.
Name	Provide a name for the default.
Value	Provide the value of the default.

Defaults (SQL Server) - Dependencies

From the **Type** dropdown, choose Column or Datatype, and if you chose **Column**, choose a Table from the **Table** dropdown. The list on the left is populated with candidate columns or datatypes. To move a candidate from the list on the left to the dependencies column on the right, select the candidate and click **Add**. Remove columns or datatypes from the dependencies list on the right by selecting the column or datatype and clicking **Remove**.

Extended Procedures Wizard (SQL Server)

Extended Procedures are dynamic link libraries that can be used to load and execute application routines written in other programming languages, such as C or Visual Basic. Extended Procedures function and appear in the same manner as SQL Server stored procedures in that you can pass parameters to them and obtain results.

Extended stored procedures provide a method for calling procedural language functions from within the Adaptive Server.

Note: Extended procedures can only be created in the Master database.

To create a new Extended procedure using a wizard:

1 Open a creation wizard for an extended procedure. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Extended Procedures (SQL Server) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Extended Procedures (SQL Server) - Properties

When creating or editing an extended procedure, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the owner of the extended procedure.
Name	Provide a name for the extended procedure.
Library	Provide the name of the DLL containing the extended procedure.

Foreign Keys Wizard (SQL Server)

Foreign keys are unique values that refer to specific columns of other tables. Thus, a foreign key links two tables together. The Foreign Key Wizard makes it easy for you to create a relational link between two tables, thereby speeding queries and giving you faster access to data. The column in the initial table, the parent table, is called the primary key. The corresponding column in the (child) table that references the primary key, is the foreign key. Foreign keys can also refer to columns within the same table.

To create a new Foreign Key using a wizard:

1 Open a creation wizard for a foreign key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Foreign Keys (SQL Server) - Properties.
- **Column Mapping** panel - Foreign Keys (SQL Server) - Column Mapping.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Foreign Keys (SQL Server) - Properties

When creating or editing a foreign key, this tab/panel lets you work with the following settings:

Setting	Description
Table Owner and **Table Name**	Select the owner and name of the table for which the foreign key is being created.
Name	Provide a name for the foreign key.
Enabled	Enables or disables the foreign key.
With NOCHECK	Specifies whether the data in the table is validated against a newly added or re-enabled foreign key.
Not For Replication	Replication copies and distributes data and database objects from one database to another and then synchronizes information between databases for consistency.
Delete Rule	If you choose the CASCADE option, all rows containing data involved with the foreign key will be deleted after a delete operation.
Update Rule	If you choose the CASCADE option, all rows containing data involved with the foreign key will be deleted after an update operation.

Foreign Keys (SQL Server) - Column Mapping

Under **Referenced Table**, choose the **Owner** and then the **Name** of the referenced, or parent, table.

Under the **Main Table**, select check boxes corresponding to the columns that are to reference columns in the referenced table. Then, under **Referenced Table**, select the corresponding column check boxes.

Full-text Catalogs Wizard (SQL Server)

This wizard builds and submits a CREATE FULLTEXT CATALOG statement. This lets you add a full-text catalog to a database, specifying accent sensitivity and an owner, and letting you define the catalog as the default full-text catalog. You can also provide filegroup and root directory details.

Along with full-text indexes, full-text catalogs facilitate full-text searching. For related information, see Full-text Indexes Wizard (SQL Server).

Note: Full-text catalogs cannot be created in the master catalog.

Note: Before working with full-text catalogs, consult Microsoft SQL Server documentation for general information on full-text searching and detailed information on topics such as restrictions on where these objects can be created, required permissions, virtual object and filegroup considerations, and the relationship with full-text indexes and the default collation. For more information, see Accessing Third Party Documentation.

To create a new full-text catalog using a wizard:

1 Open a creation wizard for afull-text catalog. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Full-text Catalogs (SQL Server) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Full-text Catalogs (SQL Server) - Properties

When creating or editing a full-text catalog, this tab/panel lets you work with the following settings:

Setting	Description
Name	Provide a name for the catalog.
Filegroup	Selecting a filegroup adds an explicit ON FILEGROUP clause, specifying the selected filegroup to which the catalog will belong. If not specified, the new catalog will be part of the default filegroup used for all full-text catalogs.
Path	Specifying a path adds an explicit IN PATH clause, specifying the root directory for the catalog. If no path is specified, the new catalog will be located in the default directory specified during setup.
Accent Sensitive	Lets you specify a WITH ACCENT_SENSITIVITY = ON/OFF clause, reflecting the selected/deselected status of the check box.
Default	Select this check box to add an AS DEFAULT argument, specifying that this catalog is the default full-text catalog. If an existing full-text catalog is currently specified as the default, this catalog becomes the new default.
Authorization Owner	Select a user or role to add an AUTHORIZATION argument that sets the owner of the full-text catalog.

Full-text Indexes Wizard (SQL Server)

This wizard lets you build and submit a CREATE FULLTEXT INDEX statement, adding a full-text index for a table or indexed view. Along with full-text catalogs, full-text indexes facilitate full-text searching.

Note: Full-text indexes cannot be created in the master catalog.

Note: Before working with full-text indexes, consult Microsoft SQL Server documentation for general information on full-text searching and detailed information on topics such as required permissions, column specification details, recommended datatype, and XML considerations. For more information, see Accessing Third Party Documentation.

To create a new full-text index using a wizard:

1 Open a creation wizard for a full-text index. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Full-text Indexes (SQL Server) - Properties.
- **Columns** panel - for details, see Full-text Indexes (SQL Server) - Columns.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

For related information, see Full-text Catalogs Wizard (SQL Server).

Full-text Indexes (SQL Server) - Properties

When creating or editing a full-text index, this tab/panel lets you work with the following settings:

Setting	Description
Parent Type	Select table or indexed view.
Parent Schema	The parent's owning schema.
Parent Name	Select the name of the table or indexed view on which the index is to be created.
Key Index	Select the unique key index on the table or view that will be used as the value for the KEY INDEX argument.
Full-text Catalog Name	Select an existing full-text catalog. For information on creating full-text catalogs, see Full-text Catalogs Wizard (SQL Server). If you do not select a catalog, SQL Server will assign the index to the default full-text catalog, generating an error if no default exists.
Filegroup Name (SQL Server 2008 ^)	Select the filegroup that the index is to be created on. Specifying no filegroup results in SQL Server default behavior, placing the index in the same filegroup as the base table or view for a non-partitioned table or in the PRIMARY filegroup for a partitioned table.
Change Tracking	Select a CHANGE_TRACKING argument value of MANUAL, AUTO, OFF, or OFF, NO POPULATION to specify whether changes to columns covered by the full-text index will be propagated to the full-text index.
Stoplist (SQL Server 2008 ^)	Select a STOPLIST argument value of OFF to specify that no stoplist be associated with the index or SYSTEM to specify that the default full-text system STOPLIST be used for this index.

Full-text Indexes (SQL Server) - Columns

When creating or editing a full-text index, this tab/panel lets you specify the columns that make up the index:

- For each column to include in the full-text index, select the associated **Indexed** check box.
- Optionally, select a language from the associated **Language** column.
- For varbinary(max) and image columns, provide a TYPE COLUMN argument filename extension.

Functions Wizard (SQL Server)

Functions are subroutines that you define so you can reuse code without having to reinvent the wheel each time. You can use functions to determine the best methods for controlling access and manipulation of the underlying data contained in an object. A function returns a value, unlike a stored procedure, which does not.

- To create a user-defined function, you need CREATE ANY privileges or IMPLICIT_SCHEMA authority on the database if the schema does not already exist.

To create a new function using a wizard:

1 Open a creation wizard for a function. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Functions (SQL Server) - Properties.
- **Definition** panel - for details, see Functions (SQL Server) - Definition.

3 Finally, use the **Execute** button to create the object.

Functions (SQL Server) - Properties

When creating or editing a function, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the owner of the function.
Name	Provide a name for the function.
Schema Binding	Choose whether the function is bound to database objects that it references.
Encryption	Choose whether SQL Server encrypts table columns that contain the text of the CREATE FUNCTION statement.

Functions (SQL Server) - Definition

Complete the CREATE FUNCTION outline provided by typing or pasting the body of the function.

Indexes Wizard (SQL Server)

Like an index in a book, a table index helps you get at the data you want without having to read through the whole table. Indexes can exist on single column or on multiple columns. Indexes appear in the form of B-trees. And, as for books, you can have multiple indexes for a single table. You can also create indexes for a view.

To create a new index using a wizard:

1 Open a creation wizard for an index. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Indexes (SQL Server) - Properties.
- **Columns** panel - Indexes (SQL Server) - Columns.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Indexes (SQL Server) - Properties

When creating or editing an index, this tab/panel lets you work with the following settings:

Setting	Description
Parent Type	Select TABLE or VIEW.
Parent Owner	Select the owner of the table or view.
Parent Name	Select the specific table or view containing the columns you want to index.
Name	Provide a name for the index.
Build Online	Enabling this feature specifies that the ONLINE=ON clause is used when creating this object and can subsequently be used when rebuilding or dropping this object.
Max degree of parallelism	Lets you specify a MAXDOP index operation value, limiting the number of processors used in parallel plan execution.
Index Type	Select UNIQUE or NONUNIQUE. An index is unique when no two rows are permitted to have the same index value. (Note: A clustered index on a view must be unique.) If an INSERT or UPDATE statement creates a duplicate value, the operation may fail.
Clustered	A clustered index is one in which the physical order of rows matches the order of indexed rows. A table or view can only have one clustered index at a time. In a nonclustered index, the physical order of rows is independent of the indexed order of rows. For an indexed view, you can only create a nonclustered index if there is already a clustered index extant. With this check box enabled, subsequent rebuild and drop operations offer an online option.
Ignore Duplicate Key	This option controls what happens when an attempt is made to insert a duplicate key value into a column that is part of a unique clustered index. If the option is selected and an INSERT statement that creates a duplicate key is executed, SQL Server issues a warning and ignores the duplicate row. If not selected, SQL Server issues an error message and rolls back the entire INSERT statement.
Statistics Recompute	Enabling this feature means queries involving the table run at the optimal level as distribution statistics are updated automatically when the index is created. If you disable this option, you can compromise query performance.
Partitioned	When selected, an ON clause is added to the CREATE TABLE statement, letting you specify a **Partition Scheme**, partitioning this table.
Partition Scheme	This property is only available if the **Partitioned** check box is selected. Select the partition scheme that specifies the filegroup mapping for this table. For information on creating partition schemes, see Partition Schemes Wizard (SQL Server).
Partition Column	This property is only available if the **Partitioned** check box is selected. Select the column that the index will be partitioned against.
Filegroup	This property is only available if the **Partitioned** check box is not selected. Select an existing, named filegroup to have the index stored on the specified filegroup. Select PRIMARY to have the index stored on the default database filegroup.
Fill Factor	Specifies the percentage that the Database Engine should make the leaf level of each index page when the index is created or rebuilt.

Pad Index	Enable or disable padding of index pages.
Sort in TempDB	Select to store the intermediate index sort results in tempdb. This option may reduce the time needed to create an index if tempdb is on a different set of disks than the user database, but it increases the amount of disk space used to create an index. In addition to the space required in the user database to create the index, tempdb must have about the same amount of additional space to hold the intermediate sort results.
Allow Row Locks and **Allow Page Locks**	Lets you enable locking granularity at the page and row level (Default=TRUE) **NOTE:** You cannot reorganize an index (primary key, or unique key) that has an **Allow Page Locks** property set to FALSE. For information on reorganizing indexes, see Reorganize (SQL Server indexes, primary keys, and unique keys).

Indexes (SQL Server) - Columns

From the **Column** dropdown, select a column for the index and specify a **Sort** option. To add more columns, click the **New** button and then follow the steps in the last instruction. Use the Delete button to drop columns.

Jobs Wizard (DBArtisan - SQL Server)

This wizard lets you create SQL Agent jobs. It lets you specify a category, set up associated notifications, create a step structure, set up schedules, work with associated alerts, and define a single or multiple server targets for the job.

Note: For an overview of SQL Agent service support, see SQL Server Agent.

To create a new job using a wizard:

1 Open an object creation wizard for a job. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details see Jobs (SQL Server) - Properties.
- **Steps panel** - for details see Jobs (SQL Server) - Steps.
- **Schedules** panel - for details see Jobs (SQL Server) - Schedules.
- **Alerts** panel - for details see Jobs (SQL Server) - Alerts.
- **Target** panel - for details see Jobs (SQL Server) - Target.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

For information on related SQL Agent elements, see the following topics:

- Alerts Wizard (DBArtisan - SQL Server)
- Operators Wizard (DBArtisan - SQL Server)

Jobs (SQL Server) - Properties

When creating or editing a SQL Agent job, this tab panel provides most settings of an **sp_add_job** call, letting you specify an enabled/disabled status and select a category, as well as specify notify and delete levels for the job. This tab has the following settings:

Category	Setting	
Creation	**Owner** and **Name**	Let you provide basic identification details.
	Category	Lets you select the category to which the job belongs.
	Description	Lets you provide a textual description for the job.
	Enabled	Lets you specify whether the job is enabled or disabled.

Notifications	Each subcategory lets you specify when notifications should be sent or jobs deleted (NEVER, WHEN JOB SUCCEEDS, WHEN JOB FAILS, or WHEN JOB COMPLETES). The **Email**, **Page**, and **Net send** subcategories let you select the specific operator to which notifications are directed. For information on creating operators, see Operators Wizard (DBArtisan - SQL Server). The **Windows application event log** subcategory lets you select when an entry is placed in the Microsoft Windows application log for this job. The **Job Deletion** subcategory lets you select when a job is deleted.

Jobs (SQL Server) - Steps

When creating or editing a job, this tab/panel lets you work with the steps for the job. You can use it to create one or more **sp_add_jobstep** calls that will be included as part of the job definition DDL.

To add a step to the job

1 Click the **Add Step** button, provide a **Name** for the step and press TAB or ENTER. The step is added to the step list with a set of default properties.

2 Modify the default properties using the table below as a guide.

Note: Properties available differ by the subsystem **Type** you select.

Setting	Description
Is start step	Lets you designate this step as the first step in the job. Note that at least one step must have this check box selected. If you deselect this check box, the change does not actually take effect until you select the check box in another step.
Type	Lets you select the subsystem used by the SQL Server Agent service to execute the step: ACTIVESCRIPTING, CMDEXEC, DISTRIBUTION, SNAPSHOT, LOGREADER, MERGE, QUEUEREADER, ANALYSISQUERY, 'ANALYSISCOMMAND, DTS, POWERSHELL (SQL Server 2008 only), or TSQL.
Database	The name of the database in which to execute the step.
Database user name	The user account to use when executing the step.
Command	The commands to be executed by the SQLServerAgent service through subsystem.
On success action	The job action to perform (QUIT WITH SUCCESS, QUIT WITH FAILURE, GO TO NEXT STEP, GO TO STEP: [*N*] *STEPNAME*) if the step succeeds.
Retry attempts	The number of retry attempts to use if the step fails.
Retry interval	The duration, in minutes, between retry attempts.
On fail action	The job action to perform (QUIT WITH SUCCESS, QUIT WITH FAILURE, GO TO NEXT STEP, GO TO STEP: [*N*] *STEPNAME*) if the step fails.
Output file	The name of the file in which the output of this step is saved
Append output to file, **Log to table (overwrite)**, **Log to table (append)**, and **Output to history**	Lets you set the flags that control how the output file is written.

To modify a job step

1 Select a step from the step list, and modify properties accordingly.

To delete a step from the job

1 Select a step from the step list and click **Delete**. If you delete the step that currently has the **Is start step** check box selected, the next lower step in the list becomes the first step. If there was no step below the deleted step, the first step in the list has its **Is start step** check box selected.

To change the order for a step

1 Select a step from the step list, and use the arrow buttons to move the step up or down one position.

Jobs (SQL Server) - Schedules

When creating or editing a job, this tab/panel lets you manage the schedules for a job. Specifically, you can:

- Create new schedules for this job
- Add schedules that were defined in other job definitions
- Delete schedules currently associated wit the current job definition
- Display all jobs using a selected schedule

To add a schedule

1 Click **New**, provide a **Name** for the schedule and press TAB or ENTER. DBArtisan adds the schedule to the schedule list with a set of default properties.

2 Select a frequency **Type**: START AUTOMATICALLY WHEN SQL SERVER AGENT STARTS, START WHENEVER THE CPUS BECOME IDLE, ONE TIME, DAILY, WEEKLY, MONTHLY, or MONTHLY RELATIVE TO FREQUENCY INTERVAL.

3 Use the **Enabled** check box to enable or disable the schedule.

4 If you selected a **Type** of ONE TIME ONLY, use the **One-time occurrence** controls to provide the **Date** and **Time** of the one-tome occurrence.

5 If you selected a **Type** of DAILY, WEEKLY, MONTHLY, or MONTHLY RELATIVE TO FREQUENCY INTERVAL, use the table below as a guide to providing details in the **Frequency**, **Daily Frequency**, and **Duration** property groups.

Property group	Type	Description
Frequency	**Daily**	Lets you specify the recurrence frequency, in days.
	Weekly	Lets you specify the recurrence frequency, in weeks., and the specific weekdays on which the job is to execute.
	Monthly	Lets you specify the recurrence frequency, in months, and the specific day of the month on which the job will execute.
	Monthly relative to frequency interval	Lets you specify the recurrence frequency, in months, and a relative weekday (such as the second Sunday) on which the job is to execute.

Daily Frequency Daily	For the specific days on which the job is to execute, this group lets you specify whether the job is to execute once per day or several times per day. If you select the **Occurs once** check box, use the **at** control to specify the schedule time. If you deselect the **Occurs once** check box, use the remaining controls to specify an hourly recurrence frequency, and a start and end time within the day when the schedule is active.
Duration	Use these controls to specify the start and end dates for a fixed duration schedule. For an open-ended schedule, provide a start date and select the **No end date** check box.

To delete a schedule from the schedule list

1 Select a schedule from the schedule list and click **Remove**.

To copy an already existing schedule definition to the current job

1 Click **Pick**. A **Pick schedule for job** dialog opens. The **Available schedules** list shows all schedules defined on this datasource.

2 Select the schedule that is to be added to the schedule list for this job definition and click **OK.**

To display all jobs currently using a schedule in the schedule list

1 Select a schedule from the schedule list and click **Jobs in schedule**. A **Jobs in schedule** dialog opens, showing all jobs currently using the selected schedule.

Jobs (SQL Server) - Alerts

When creating or editing a job, this tab/panel provides a convenient way to work with SQL Agent alerts associated with this job. On opening, this tab provides a list of all alerts that specify this job as the target job to be executed in response to those alerts. When you execute the job that creates or edits this job, DDL to create all alerts on this tab is submitted.

Note: Alerts cannot be created for multi-target jobs. They will be rejected by the server.

Manage the alerts list as follows:

- Click **Add** to create a new alert to be added to the list. This opens the Alert Wizard, with the **Execute Job** check box selected and the **Job Name** box specifying the current job, both uneditable. For information on using the Alert Wizard, see Alerts Wizard (DBArtisan - SQL Server).
- Select an alert and click **Edit** to open an editor on the selected alert. The **Execute Job** check box and the **Job Name** box specifying the current job are both uneditable. For information on using the Alert Wizard, see Alerts Editor (DBArtisan - SQL Server).
- Select an alert and click **Remove** to drop the selected alert from the alert list. DDL to create or alter the alert is not submitted when you execute the job that creates or edits this job. For existing alert definitions, this does not delete the definition nor does it modify the **Execute Job** or **Job Name** controls.

Jobs (SQL Server) - Target

When creating or editing a job, this tab/panel lets you specify a target server or servers for the job. The default setting, **Target Local Server**, specifies that the job will be executed on the local server. The **Target Multiple Servers** setting is only available in multiserver operation domains, with the selectable server names populated from the **systargetservers** table.

Note: Alerts cannot be created for multi-target jobs. They will be rejected by the server.

To target a job at one or more servers in a multiple server environment

1 Select **Target Multiple Servers**.

2 Select the check box associated with each server on which the job will be executed.

Linked Servers Wizard (DBArtisan - SQL Server)

A linked server object represents a database system other than Microsoft SQL Server. The links are created through OLE DB datasources. OLE DB is a COM-based application programming interface (API) for accessing data. OLE DB supports accessing data stored in any format (databases, spreadsheets, text files, and so on) for which an OLE DB provider is available. With a linked server you can execute commands against OLE DB datasources on different servers. Linked servers let you issue distributed queries, commands, and so on.

The wizard prompts you to provide much of the same information you give when you initially register a server.

To create a new linked server using a wizard:

1 Open a creation wizard for a linked server. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Linked Servers (SQL Server) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Linked Servers (SQL Server) - Properties

When creating or editing a linked server, this tab/panel lets you work with the following settings:

Group	Settings and descriptions
General	If defining a linked server defined through Microsoft SQL Server, set the **Is SQL Server** check box. The name must be the server's network name. Any tables retrieved from the server are from the default database defined for the login on the linked server. If defining a linked server not defined through Microsoft SQL Server, clear the **Is SQL Server** check box and use the following options to provide OLE DB details: **Product Name** - The product name of the OLE DB datasource to which you are linking. **Datasource** - The name of the OLE DB provider managing access to the linked server. **OLE DB Provider Name** - The name of the OLE DB provider managing access to the specified linked server. **Provider Location** - The OLE DB location property corresponding to the linked server. **Provider Connection String** - The OLD DB provider-specific connection string that identifies a unique datasource. **Catalog** - The name of the catalog to be used when making a connection to the OLE DB provider.
Server Options	**Data Access Server** - Lets you specify whether queries against this server are enabled. **RPC Out Server** - Lets you select the server with RPC encryption. **RPC** - Enables RPC encryption from the server. **Collation Compatible Server** - Indicates Change Distributed Query execution against linked servers. If you select this option, Microsoft SQL Server assumes that all characters in the linked server are compatible with the local server, with regard to character set and collation sequence (or sort order). This option is for Microsoft SQL Server 8.0 or later. **Lazy Schema Validation** - This option lets you optimize performance by ensuring the query processor does not request metadata for any of the linked tables until data is actually needed from the remote member table. This option is for Microsoft SQL Server 8.0 or later. **Use Remote Collation** and **Collation Name** - Let you indicate that the linked server should use remote collation and select the name of the collation. This option is for Microsoft SQL Server 8.0 or later. **Query Timeout** - Type the query timeout value. This option is for Microsoft SQL Server 8.0 or later. **Connection Timeout** - Type the connection timeout value. This option is for Microsoft SQL Server 8.0 or later.

Logins Wizard (SQL Server)

The Login wizard lets you build and submit a CREATE LOGIN statement, adding a new SQL Server, Windows, certificate-based, or asymmetric key-based login. The Login wizard also lets you:

- Add the login to fixed server roles
- Create user accounts for the login on selected databases

To create a new login using a wizard:

1 Open a creation wizard for alogin. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Logins (SQL Server) - Properties.
- **Server Roles** panel - for details, see Logins (SQL Server) - Server Roles.
- **Users** panel - for details, see Logins (SQL Server) - Users.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Logins (SQL Server) - Properties

When creating or editing a login, this tab/panel lets you work with the following properties:

Setting	Description
Name	Specifies a name for this login.
Default Database	Lets you specify a DEFAULT_DATABASE argument, specifying a default database for this login.
Default Language	Lets you specify a DEFAULT_LANGUAGE argument, specifying a default language for this login.
Account Type	Lets you select one of the following login types: STANDARD - (SQL Server logins) if you select this type, you must also provide **Password** and **Check Policy** values. NTGROUP or NTUSER - (Windows logins) if you select one of these types, you must also provide a **Domain** value. CERTIFICATE - (certificate-mapped logins) if you select this type, you must also provide a **Certificate** value. ASYMMETRIC KEY - (asymmetric key-mapped logins) if you select this type, you must also provide an **Asymmetric Key** value.
Password	This is only available for an **Account Type** of STANDARD. It lets you provide a password for the login.
Check Policy	This is only available for an **Account Type** of STANDARD. If selected, login policies defined on the server apply to this login.
Check Expiration	This is only available if **Check Policy** is selected. If selected, password expiration policies are applied to this login.
Must Change	This is only available if **Check Expiration** is selected. If selected, the user will be prompted to change the password on the first login.
Domain	This is only available for an **Account Type** of NTUSER or NTGROUP. Lets you specify the domain name value that will be specified with the loginName argument ([<domainName>\<loginName>]).
Certificate	This is only available for an **Account Type** of CERTIFICATE. It lets you specify the name of a certificate that is to be associated with this login. It lets you specify the name of a certificate that is to be associated with this login.
Asymmetric Key	This is only available for an **Account Type** of ASYMMETRIC KEY. It lets you specify the name of an asymmetric key that is to be associated with this login.

Logins (SQL Server) - Server Roles

When creating or editing a login, this tab/panel builds sp_addsrvrolemember procedure calls that will be submitted along with the CREATE LOGIN statement issued by this wizard/editor. This lets you add the login as a member of one or more fixed server roles (or common user-defined roles when selecting multiple databases) by selecting the check boxes associated with those roles.

For information on editing fixed server roles, see Roles Editor (SQL Server).

Logins (SQL Server) - Users

When creating or editing a role, this tab/panel lets you build CREATE USER... FOR LOGIN statements that will be submitted with the CREATE LOGIN statement issued with this wizard/editor. Maintain database user accounts for a login as follows:

- Add a user account for this login to a database by selecting a database from the **Databases where the Login does NOT have a User Account** and clicking the **Add** button. This opens the SQL Server User wizard, letting you define a new user on that database. For more information, see Users Wizard (SQL Server).
- Remove a user account for this login from a database by selecting a database from the **Databases where the Login HAS a User Account** and clicking the **Remove** button.

Operators Wizard (DBArtisan - SQL Server)

This wizard lets you create an SQL Agent operator. It also lets you assign a set of notifications sent to the operator for selected SQL Agent alerts.

Note: For an overview of SQL Agent service support provided by DBArtisan, see SQL Server Agent.

To create a new operator using a wizard:

1 Open an object creation wizard for an operator. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details see Operators (SQL Server) - Properties.
- **Notifications** panel - for details see Operators (SQL Server) - Notifications.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

For information on related SQL Agent elements, see the following topics:

- Jobs Wizard (DBArtisan - SQL Server)
- Alerts Wizard (DBArtisan - SQL Server)

Operators (SQL Server) - Properties

When creating or editing a SQL Agent operator, this tab/panel lets you identify the operator, set an enabled/disabled status, specify the operators contact information, and set a pager duty schedule. Offering controls to build a basic **sp_add_operator** call, it has the following settings:

Property group	Setting	Description
Creation	**Name**	Provide a name for the operator.
	Enabled	Set the status of this operator to enabled or disabled.
	Email	Type the operator's email address.
	Pager Address	Type the operator's pager address.
	Net Send Address	Type the operator's net send address.

Pager on duty schedule Monday through **Sunday** Select the days that this operator is available for pages.

Operators (SQL Server) - Notifications

When creating or editing an operator, this tab lets you select the SQL Agent alerts for which this operator will be notified. The selections are submitted as a set of **sp_add_notification** calls made after the **sp_add_operator** call used to create the operator or submit operator definition changes.

To assign one or more alerts to the operator currently being defined

1 Select one or more of the **Email**, **Pager**, or **Net Send** check boxes associated with an alert.

Note: When editing an operator, this tab displays the status of the most recent attempts to notify this operator by email, pager and net send.

Partition Functions Wizard (SQL Server)

The Partition Functions Wizard lets you build and submit a CREATE PARTITION FUNCTION statement, specifying a range, input parameter type, and boundary values. Once created, the partition function can be referenced by the partition scheme used to create a partitioned table or index.

Note: Before creating a partition function, consult Microsoft SQL Server documentation for detailed information on topics such as type restrictions and boundary value usage. For more information, see Accessing Third Party Documentation.

To create a new partition function using a wizard:

1 Open a creation wizard for a partition function. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties panel** - for details, see Partition Functions (SQL Server) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

For related information, see the following topics:

- Partition Schemes Wizard (SQL Server)
- Indexes Wizard (SQL Server)
- Tables Wizard (SQL Server)

Partition Functions (SQL Server) - Properties

When creating or editing a partition function, this tab/panel lets you work with the following settings:

Setting	Description
Name	Provide a name for the partition function.
Input Parameter Type	Select the data type of the column used for partitioning.
Range	Select to which side of each functional value interval (LEFT, RIGHT) the functional value belongs, when interval values are sorted in ascending order from left to right.
Function Values	Lets you specify the boundary values to include as FOR VALUES arguments to the CREATE PARTITION FUNCTION statement. Each boundary value is a constant expression that can reference variables. The set of values provided define the partition points for an index or table partitioned using this function. Use the New, Delete, Up, and Down buttons to create and maintain the value list.

Partition Schemes Wizard (SQL Server)

The Partition Scheme Wizard builds and submits a CREATE PARTITION SCHEME statement, letting you specify a partition function and a file group mapping. Once created, the partition scheme can be referenced when creating a partitioned table or index.

Note: Before creating a partition scheme, consult Microsoft SQL Server documentation for detailed information on topics such as the relationship between the partition function and filegroups specified. For more information, see Accessing Third Party Documentation.

To create a new partition wizard using a wizard:

1 Open a creation wizard for a partition scheme. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Partition Schemes (SQL Server) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

For related information, see the following topics:

- Partition Functions Wizard (SQL Server)
- Indexes Wizard (SQL Server)
- Tables Wizard (SQL Server)

Partition Schemes (SQL Server) - Properties

When creating or editing a partition scheme, this tab/panel lets you work with the following settings:

Setting	Description
Name	Provide a name for the partition scheme.
Partition Function	Select the partition function that specifies the range, input parameter type, and boundary values for this partition scheme.
Filegroup Name	Lets you specify the names of the filegroups to store the partitions specified by the selected **Partition Function**. Consult Microsoft SQL Server documentation for details on the relation ship between filegroups and partitions. For more information, see Accessing Third Party Documentation. Use the New, Delete, Up, and Down buttons to create and maintain the value list.

Primary Keys Wizard (SQL Server)

Primary key constraints make sure that no duplicate values or NULLS are entered in the columns you specify. You can use primary key constraints to enforce uniqueness and referential integrity. A table can only have a single primary key constraint.

The dialog box lets you specify the owner and table on which you want to place the primary key constraint.

To create a new primary key using a wizard:

1 Open a creation wizard for a primary key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Primary Keys (SQL Server) - Properties.
- **Columns** panel - Primary Keys (SQL Server) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Primary Keys (SQL Server) - Properties

When creating or editing a primary key, this tab/panel lets you work with the following settings:

Setting	Description
Table Owner and **Table Name**	Choose the owner and name of the table in which the primary key is being created.
Name	Provide a name of the primary key being created.
Build Online	Enabling this feature specifies that the ONLINE=ON clause is used when creating this object and can subsequently be used when rebuilding or dropping this object.
Max degree of parallelism	Lets you specify a MAXDOP index operation value, limiting the number of processors used in parallel plan execution.
Clustered	Enable or disable clustering. With this check box enabled, subsequent rebuild and drop operations offer an online option.
File Group	If you do not specify a filegroup, Microsoft SQL Server creates the index in the default filegroup.
Fill Factor	This specifies how full each index page that's storing data should be. The fill factor is a percentage value between 0 and 100.

Primary Keys (SQL Server) - Columns

From the **Column** dropdown, select a column for the primary key and specify a **Sort** option. To add more columns, click the **New** button and then follow the steps in the last instruction. Use the Delete button to drop columns.

Procedures Wizard (SQL Server)

Procedures are a reusable block of PL/SQL, stored in the database, that applications can call. Procedures streamline code development, debugging, and maintenance by being reusable. Procedures enhance database security by letting you write procedures granting users execution privileges to tables rather than letting them access tables directly.

The Procedure Wizard lets you:

- Name the procedure and specify its body.
- Specify any execution options and you can encrypt the stored procedure text in syscomments.

To create a new procedure using a wizard:

1 Open a creation wizard for a procedure. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Procedures (SQL Server) - Properties.
- **Definition** panel - for details, see Procedures (SQL Server) - Definition.

3 Finally, use the **Execute** button to create the object.

Procedures (SQL Server) - Properties

When creating or editing a procedures, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the owner of the procedure.
Name	Provide a name for the procedure
Procedure Number	Optionally, provide a procedure number. By using a number, you can group procedures of the same name together. This also enables you to drop them using only one DROP PROCEDURE statement. So, the procedures bill;1, bill;2, bill;3, etc. will be dropped simultaneously when the time comes.
Replication	This option creates a procedure that is used as a stored procedure filter and is executed only during replication.
Recompile	The plan for this procedure will not be cached and the procedure is recompiled when it is run. This option is appropriate when you're using atypical or temporary values and you don't want to override the execution plan cached in memory.
Encryption	If you select this option, SQL Server will encrypt the syscomments table entry containing the text of the CREATE PROCEDURE statement. It keeps the procedure from being published as part of replication.

Procedures (SQL Server) - Definition

Complete the CREATE PROCEDURE outline provided by typing or pasting the body of the procedure.

Remote Servers Wizard (DBArtisan - SQL Server)

Remote Servers are servers on a network that you can access through your local server. Instead of logging directly into a remote server, you can log into your local server and execute a remote procedure call to it.

Note: Only stored procedures are allowed against remote servers. You can use a linked server to execute distributed queries and stored procedures.

Before beginning

Before you can set up a remote server, you first configure both the local and remote servers to accept remote access. To configure the Microsoft SQL Servers for remote access, you need to log into each server. The table below describes the parameters you need to configure on each server:

Parameter	Description
Remote Access	1(enabled)
Remote Connections	Number of remote connections required
Remote Logins	Number of remote logins required
Remote Sites	Number of remote sites required

After setting these server configuration parameters, shutdown and restart the server so the new values can take effect.

To create a new remote server using a wizard:

1 Open a creation wizard for a remote server. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Remote Servers (SQL Server) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Remote Servers (SQL Server) - Properties

When creating or editing a remote server, this tab/panel lets you work with the following settings:

Setting	Description
Name	The name for the remote server.
Server Type	Lets you specify a server type of LOCAL or REMOTE. The local server routes all commands to the remote server. To communicate with a remote server, you log into a local server and submit your remote procedure calls for remote servers. The remote server receives remote procedure calls from the local server and processes them accordingly.
Publication Server	Select if the server is to publish data for replication.
Distribution Server	Select If the server is to manage the distribution database.
Subscription Server	This option applies if you want to deploy a subscription management application from the remote server.
Publisher/Subscriber Server	Select if the server is both a subscriber and publisher of replicated data.
Collation Compatible Server	Select if the server is to be collation compatible.
RPC Out Server	Select if the server is to serve as a RPC Out server.

Roles Wizard (SQL Server)

Roles are sets of user privileges you associate with access to objects within a database. Roles streamline the process of granting permissions. You can use roles to grant sets of permissions and privileges to users and groups. Roles can help you comply with Sarbanes Oxley regulations by limiting which users can have access to what privileges, for example a Human Resources Role versus an Accounting Role.

To create a new role using a wizard:

1 Open a creation wizard for a role. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Roles (SQL Server) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

After you click **Finish** and the role has been created, the Roles Editor open. You can then assign object and system privileges to the role and determine which users can take part in the role. For more information, see Roles Editor (SQL Server).

Roles (SQL Server) - Properties

When creating or editing a role, this tab/panel lets you work with the following settings:

Setting	Description
Name	The name used to reference the role.
RoleType	The value can be NONE, APPLICATION, or STANDARD.
Authorization Owner	Only required with a **Role Type** of STANDARD.
Password	Only required with a **Role Type** of APPLICATION.

Rules Wizard (SQL Server)

Rules promote data integrity by allowing you to validate the values supplied to a table column. They are reusable objects that you can bind to table columns or user datatypes. Check constraints are similar to rules, and are in fact the preferred way of restricting data. A column or user-defined data type can have only one rule bound to it, but a column can have both a rule and one or more check constraints associated with it. Not that a rule cannot apply to data already existing in the database at the time you're creating the rule and can't be bound to a system-created data type. If you create a new rule when one already exists, the new rule will override the previous one.

To create a new rule using a wizard:

1 Open a creation wizard for a rule. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Rules (SQL Server) - Properties.
- **Dependencies** panel - Rules (SQL Server) - Dependencies.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Rules (SQL Server) - Properties

When creating or editing a rule, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the Owner of the rule.
Name	Provide a name for the rule.
Restriction	Type the condition. The rule restriction is the condition that defines the rule and can be any expression valid in a WHERE clause and can include such elements as arithmetic operators, relational operators, and predicates (for example, IN, LIKE, BETWEEN).

Rules (SQL Server) - Dependencies

From the **Type** dropdown, choose Column or Datatype, and if you chose **Column**, choose a Table from the **Table** dropdown. The list on the left is populated with candidate columns or datatypes. To move a candidate from the list on the left to the dependencies column on the right, select the candidate and click **Add**. Remove columns or datatypes from the dependencies list on the right by selecting the column or datatype and clicking **Remove**.

Schema Wizard (SQL Server)

This wizard lets you create a schema on Microsoft SQL Server.

To create a new schema using a wizard:

1 Open a creation wizard for a schema. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Schema (SQL Server) - Properties.
- **Permissions** panel - Setting Permissions or Privileges for an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Schema (SQL Server) - Properties

When creating or editing a schema, this tab/panel lets you provide a **Name** and select the **Owner** of the schema.

Schema (SQL Server) - Permissions

For each specific permission to be granted, select the cell corresponding to the name and specific permission, and click the **Grant** button. To revoke a privilege, select a cell showing a Granted permission and click **Revoke**.

Sequences Wizard (SQL Server)

This wizard builds and submits a CREATE SEQUENCE statement.

To create a new full-text catalog using a wizard:

1 Open a creation wizard for a sequence. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Full-text Catalogs (SQL Server) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Sequences (SQL Server) - Properties

When creating or editing a sequence, this tab/panel lets you work with the following settings:

Setting	Description
Schema and **Name**	Lets you provide identification for the sequence.
Start With	Lets you provide a START WITH argument value, specifying the first value to be returned by the sequence.
Increment By	Lets you provide an INCREMENT BY argument value, specifying the value by which to increment (or decrement if negative) the sequence object for each call to the NEXT VALUE FOR function.
Minimum Value and **Maximum Value**	Lets you provide MINVALUE and MAXVALUE argument values, specifying the bounds for this sequence.
Cycle	If selected, a CYCLE argument is generated, specifying that sequence values should restart from the minimum value (or maximum for descending sequences) or throw an exception when the minimum/maximum value is exceeded. If unselected, the default NO CYCLE is assumed.
Cache and Cache Size	If **Cache** is unselected, a NO CACHE argument is generated. If **Cache** is selected, a CACHE argument is generated. A CACHE argument value can be provided using the Cache Size control, and if no Cache Size value is provided, SQL Server will generate a value.
Data Type Schema, **Data Type Name**, and **Data Type Precision**	These controls let you build an AS argument, specifying datatype details for the new sequence. If no values are selected/provided, SQL Server will, by default, generate a sequence object of type BIGINT. After selecting a **Data Type Schema**, the **Data Type Name** dropdown is populated with user datatypes created from a base datatype of BIGINT, TINYINT, SMALLINT, INT, BIGINT, DECIMAL, or NUMERIC. **Data Type Precision** is a read-only field displaying the precision of the selected user datatype. **NOTE:** You can use system types if you select a **Data Type Schema** of SYS.

After creation, the Sequences Editor lets you view the current value of the sequence and whether the maximum value has been exceeded. For details. see Sequences Editor (SQL Server).

Server Triggers Wizard (DBArtisan - SQL Server)

This wizard builds and submits a CREATE TRIGGER... ON ALL SERVER statement, letting you create a DDL trigger with server scope, that fires in response to either a logon or selected DDL events. The wizard builds basic syntax only, letting you provide the actions and conditions SQL making up the main body of the trigger.

Note: Before working with server triggers, consult Microsoft SQL Server documentation for general information on DDL triggers as well as specifics such as FOR/AFTER argument details, valid actions and conditions, and supported events. For more information, see Accessing Third Party Documentation.

To create a new server trigger using a wizard:

1 Open an object creation wizard for a server trigger. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Server Triggers (SQL Server) - Properties.
- **Trigger Events** panel - for details, see Server Triggers (SQL Server) - Properties.
- **Definition** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Note: In order to fire in response to logons or specified events, a server trigger must be enabled. For information on enabling and disabling a server trigger, see Server Triggers Editor (DBArtisan - SQL Server) and Change Status (SQL Server DDL triggers).

Server Triggers (SQL Server) - Properties

When creating or editing a server trigger, this tab/panel lets you work with the following settings:

Setting	Description
Name	Provide a name for the DDL trigger.
Trigger Timing	Use this control to specify a FOR/AFTER argument option. If the Fire On Logon check box is not selected, the specific events for the argument are provided on the **Trigger Events** tab/panel.
Fire On Logon	If selected, the CREATE TRIGGER statement is created with a FOR/AFTER LOGON argument, associating the DDL trigger with a user logon. If deselected, the trigger is associated with the event(s) selected on the **Trigger Events** tab/panel.
Encrypted	Select this check box to add a WITH ENCRYPTION argument to the generated DDL. This prevents the trigger from being published as part of a SQL Server replication.

Server Triggers (SQL Server) - Trigger Events

When creating or editing a database trigger, this tab/panel lets you select the FOR/AFTER argument DDL events that will fire this trigger.

Note: This tab/panel is only available if the **Fire On Logon** check box is deselected. For more information, see Server Triggers (SQL Server) - Properties.

Symmetric Keys Wizard (SQL Server)

This wizard lets you build and submit a CREATE SYMMETRIC KEY statement.

To create a new symmetric key using a wizard

1 Open a creation wizard for a symmetric key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Symmetric Keys (SQL Server) - Properties.
- **Encryption Mechanisms** panel - Symmetric Keys (SQL Server) - Encryption Mechanisms.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Symmetric Keys (SQL Server) - Properties

When creating or editing an asymmetric key, this tab/panel lets you work with the following settings:

Setting	Description
Authorization Owner	The name of the user that will own the symmetric key.
Name	The name for the symmetric key
Provider Name	Lets you provide a FROM PROVIDER argument value, specifying an Extensible Key Management provider and name.
Key Algorithm	Lets you select a WITH ALGORITHM value of DES, AES_192, RC2, RC4_128, DESX, AES_128, TRIPLE_DES, or TRIPLE_DES_3KEY, specifying the encryption algorithm.
Pass Phrase	Lets you provide a KEY_SOURCE argument value, specifying a pass phrase from which to derive the key.
Identity Phrase	Lets you provide an IDENTITY_VALUE argument value, specifying an identity phrase from which to generate a GUID that tags data encrypted with a temporary key.
Key Name In Provider	Lets you provide a PROVIDER_KEY_NAME value, specifying the key name from the external provider.
Creation Disposition	Lets you select a CREATION_DISPOSITION value of OPEN_EXISTING or CREATE_NEW, specifying whether the symmetric key is mapped to an existing EKM key or a new key is created on the EKM device.

For context information such as opening the wizard or editor for this object type, see Synonyms Wizard (SQL Server).

Symmetric Keys (SQL Server) - Encryption Mechanisms

When creating or editing a symmetric key, this tab/panels lets you provide an ENCRYPTION BY... argument, and indicate the specific asymmetric key, symmetric keys, or certificate used to encrypt the key being created or the password from which to derive a TRIPLE_DES key.

The **Used Encryption Mechanisms** lists the symmetric key, asymmetric key, certificate, or password options currently included in the ENCRYPTION BY argument. Selecting ASYMMETRIC KEYS, CERTIFICATES, PASSWORD, or SYMMETRIC KEYS from the **Encryption Type** dropdown displays the list of available elements of that type in the **Existing Encryption Mechanisms** list.

To specify encrypting mechanisms for a symmetric key

1 From the **Encryption Type** dropdown, select an option among ASYMMETRIC KEYS, CERTIFICATES, PASSWORD, or SYMMETRIC KEYS.

2 Take one of the following actions:

- Select CERTIFICATE or ASYMMETRIC KEY from the **Encryption Type** dropdown to display elements of that type in the **Existing Encryption Mechanisms** list. Select a specific certificate or asymmetric key and click **Add** to move that element to the **Used Encryption Mechanisms** list.
- Select SYMMETRIC KEY from the **Encryption Type** dropdown to display elements of that type in the **Existing Encryption Mechanisms** list. Select a specific symmetric key and click **Add** to open the **Open Key** dialog. For more information on opening and decrypting the key, see Opening a Symmetric Key.
- Select PASSWORD from the **Encryption Type** dropdown and click **Add** to open the **Add Password** dialog. That dialog lets you add an ENCRYPTION BY PASSWORD argument. Use the **Password** control to provide the ENCRYPTION BY PASSWORD value. Click **OK** to add the password to the **Used Encryption Mechanisms** list.

3 Repeat step1 and step2 to add more encrypting mechanisms for this symmetric key.

To remove an encryption mechanism from an asymmetric key

1 Select an item from the **Used Encryption Mechanisms** list.

2 Click **Remove**.

For context information such as opening the wizard or editor for this object type, see Synonyms Wizard (SQL Server).

For information on creating asymmetric keys or certificates to be used with the encryption method for this key, see Asymmetric Keys Wizard (SQL Server) and Certificate Wizard (SQL Server).

Opening a Symmetric Key

When specifying a symmetric keys as the encryption mechanism, you must decrypt the symmetric key and make it available for use. The **Open Key** dialog lets you build an OPEN SYMMETRIC KEY that will precede the CREATE SYMMETRIC KEY statement submitted to create or edit a symmetric key.

Use the **Certificate** control to provide a DECRYPTION BY CERTIFICATE value and the **Password** control to provide a WITH PASSWORD value. Click **Open Key** to add the symmetric key to the **Used Encryption Mechanisms** list.

Note: This functionality is available for Microsoft SQL Server only.

To open a symmetric key

1 Open a creation wizard or editor for a symmetric key. For details, see Opening an Object Wizard and Opening an Object Editor.

2 Select the **Encryption Mechanisms** tab.

3 Select **Symmetric Keys** from the **Encryption Type** dropdown, select a symmetric key from the **Existing Encryptions Mechanisms** list, and then click **Add**.

The **Open Key** dialog opens.

4 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks	
Action options	Lets you work with the following settings:	
	Decryption Mechanism and **Certificate/ Symmetric Key/Asymmetric Key/Password**	These controls let you build the DECRYPTION BY option, specifying the decryption mechanism and the specific certificate, asymmetric key, symmetric key or password to be used.
	Password	Lets you provide the BY PASSWORD or WITH PASSWORD values for a **Decryption Mechanism** of CERTIFICATE, ASYMMETRIC KEY, or PASSWORD.
Preview	Displays the DDL that will execute the object action. For details, see Preview.	

5 Click **Open Key** to dismiss the dialog and return to the Symmetric Key wizard or editor.

Synonyms Wizard (SQL Server)

This wizard builds and submits a CREATE SYNONYM statement, letting you build a one-part name that can be used in SQL statements instead of a fully-qualified, multi-part name.

To create a new synonym using a wizard:

1 Open a creation wizard for a synonym. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Synonyms (SQL Server) - Properties.
- **Permissions** panel - Setting Permissions or Privileges for an Object
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Synonyms (SQL Server) - Properties

When creating a synonym, this tab/panel lets you provide a synonym name and fully qualify the object name that it references. When editing a synonym, this tab/panel lets you view the synonym definition.

Note: Before creating or editing a synonym, consult Microsoft SQL Server documentation for information on topics such as support for specific function and procedure types, and three-part and four-part name restrictions. For assistance, see Accessing Third Party Documentation.

The following table describes the available settings:

Setting	Description
Schema and **Name**	Use these setting to select an owner and type the one-part name that will be used to reference
Server, **Database**, **Referenced Object Type**, **Referenced Object Owner**, and **Referenced Object Name**	Use these setting to select the specific object to which the synonym is to refer. Optionally, you can use the server, database, and owner settings to construct a partially-qualified to fully-qualified name for the object. The user interface will allow you to construct all valid, multi-part name variations (for example: *server*.*database*.*owner*.object, *database*.*owner*.*object*, *owner*.*object*, *server*...*object*).

Tables Wizard (SQL Server)

A table is a column-based arrangement of data in which the content of one column has a bearing on the other column(s). So, for example, a table might have a column for authors, another column for the books each author has written, and a third for the number of copies each title by a given author has sold. The data moves across the columns in rows.

You must have CREATE TABLE permissions to generate a new table.

Note: Before working with tables, consult Microsoft SQL Server documentation for general information on filegroups as well as specifics on CREATE TABLE... FILESTREAM and FILETABLE arguments. For more information, see Accessing Third Party Documentation.

To create a new table using a wizard:

1 Open a creation wizard for a table. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Dependencies** panel - Tables (SQL Server) - Properties.
- **Columns** panel - Tables (SQL Server) - Columns.
- **Indexes** panel - Tables (SQL Server) - Indexes.
- **Constraints** panel - Tables (SQL Server) - Constraints.
- **Permissions** panel - Setting Permissions or Privileges for an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Tables (SQL Server) - Properties

When creating or editing a table, this tab/panel lets you work with the following settings:

Setting	Description
Schema	Select the owner of the table.
Name	Provide a name for the table
ANSI_NULLS option	By setting this option, you are setting ANSI_NULLS, ANSI_PADDING, ANSI_WARNINGS, and QUOTED_IDENTIFIER to on so the table can be used in an Indexed View.
Partitioned	When selected, an ON clause is added to the CREATE TABLE statement, letting you specify a **Partition Scheme**, partitioning this index.
Partition Scheme	This property is only available if the **Partitioned** check box is selected. Select the partition scheme that specifies the filegroup mapping for this index. For information on creating partition schemes, see Partition Schemes Wizard (SQL Server).
Filegroup	This property is only available if the **Partitioned** check box is not selected. Select an existing, named filegroup to have the table stored on the specified filegroup. Select PRIMARY to have the table stored on the default database filegroup.
Text Image Filegroup	Select an existing, named filegroup to have text, ntext, image, xml, varchar(max), nvarchar(max), varbinary(max), and CLR user-defined type columns stored on the specified filegroup. Select PRIMARY to have columns of those types stored on the default database filegroup.
Filestream On (Version 2008^)	This lets you specify the filegroup for FileStream data, specifying a FILESTREAM_ON argument to the generated DDL. This control is only available if a filegroup that can contain filestreams is available in the database. For more on filegroups, see Databases Wizard (SQL Server).
As Filetable and other **Filetable...** properties (Version 2012^)	**As Filetable** lets you add an AS FILETABLE argument to the generated DDL, specifying that the table is created as a FileTable. The **Columns**, **Indexes**, and **Constraints** tabs/panels are disabled when this control is selected. **Filetable Collate Filename** lets you provide a FILETABLE_COLLATE_FILENAME argument value, specifying the name of the collation used. If no value is provided, the default collation for the database is used. **Filetable Directory Name** lets you provide a FILETABLE_DIRECTORY argument value, specifying a FileTable directory name. If no value is provided, the name of the filetable is used. **Filetable PK Constraint Name** lets you provide a FILETABLE_PRIMARY_KEY_CONSTRAINT_NAME argument value, specifying the name used for the primary key constraint automatically created on the FileTable. If this value is not provided, a system-generated name is used. **Filetable StreamID Unique Constraint Name** lets you provide a FILETABLE_STREAMID_UNIQUE_CONSTRAINT_NAME argument value, specifying the name used for the unique key constraint automatically created on the FileTable. If this value is not provided, a system-generated name is used. **Filetable Fullpath Unique Coonstraint Name** lets you provide a FILETABLE_FULLPATH_UNIQUE_CONSTRAINT_NAME argument value, specifying the name used for the unique constraint automatically created on the parent_path_locator and name columns in the FileTable. If this value is not provided, a system-generated name is used.

Tables (SQL Server) - Columns

Use the **Add Column** button to add the columns for the table. After providing a **Name** for a new column, you can modify column properties in the **Property/Value** list. Available properties depend on the datatype you choose as well as on the property values you select:

Computed and **Computed Expression** - Let you define a column as a computed column and provide the computed column expression.

Type - Lets you select a datatype (depending on the type, additional properties such as **Size**, **Width**, and **Scale** may be available).

Identity Column - Select this check box to define the column as an identity column.

Allow Nulls - Select this check box to allow nulls in this column.

Default Collation - available for text/character datatypes, lets you specify a default collation.

Default Value - Lets you type a constant value or select a function returning a constant value to serve as the default for the column

Default Binding and **Rule Binding** - Let you bind a rule or default to a column.

Is Sparse - Available to columns that allow NULL values, optimizes storage of the column for null values. This property does apply to the following data types: text, ntext, image, timestamp, user-defined data type, geometry, or geography. Columns with default values, default or rule bindings, cannot be defined as sparse. Computed columns cannot be defined as Sparse, but the columns in the computed expression can be Sparse columns.

Optionally, you can select a column and modify its values or select a column and **Delete** it.

Note: If you create a table with a Column datatype = text., you can set the storage and image values on the **Storage** tab of the Tables Editor **Storage** tab. When you have a text datatype, the **Storage** tab displays a Text In Row box where you can specify the maximum size to be stored.

Note: Because the smalldatetime datatype stores dates and time with less precision than the datetime datatype, before outputting you use the CAST or CONVERT functions to convert any boxes with the smalldatetime datatype to either VARCHAR or datetime datatypes. For more information, see SQL Server Books Online, Transact-SQL Reference.

Note: This tab/panel is not available if **As Filetable** is selected. For details, see Tables (SQL Server) - Properties.

Tables (SQL Server) - Indexes

Click **Add** to open the Index Wizard. For details, see Indexes Wizard (SQL Server).

Note: This tab/panel is not available if **As Filetable** is selected. For details, see Tables (SQL Server) - Properties.

Tables (SQL Server) - Constraints

Selecting a constraint type and clicking **Add** opens the object wizard for that object type. For details see:

  - Primary Keys Wizard (SQL Server)
  - Unique Keys Wizard (SQL Server)
  - Foreign Keys Wizard (SQL Server)
  - Create Synonym

Note: This tab/panel is not available if **As Filetable** is selected. For details, see Tables (SQL Server) - Properties.

Triggers Wizard (SQL Server)

Triggers are a special type of procedure that automatically fire when defined data modification operations (insert, update, or delete) occur on a target table or view. Triggers fire after an insert, update or delete, but belong to the same transaction as the data modification operation. Triggers can be implemented to enforce business rules or referential data integrity.

Important Notes

- For more information on the syntax for Trigger bodies, consult the Microsoft SQL Server Transact-SQL Documentation.

To create a new trigger using a wizard:

1 Open a creation wizard for a trigger. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Triggers (SQL Server) - Properties.
- **Definition** panel - for details, see Triggers (SQL Server) - Definition.

3 Finally, use the **Execute** button to create the object.

Note: You can use the Trigger Editor opens to create dependencies or alter the trigger statement.

Triggers (SQL Server) - Properties

When creating or editing a trigger, this tab/panel lets you work with the following settings:

Setting	Description
Parent Type	Select the TABLE or VIEW on which the trigger is to be created.
Parent Schema	Select the owner of the table or view on which the trigger is to be created.
Parent Name	Select the specific table or view in which the trigger is to be created.
Name	Provide a name for the trigger.
Trigger Timing	INSTEAD OF: This is the only option for a View trigger. An INSTEAD OF trigger fires in place of the triggering statement and will not make changes to the data unless the conditions of the INSTEAD OF statement are met first. So, your UPDATE execution statement is replaced by an INSTEAD OF UPDATE statement as a way to enforce particular business rules you establish. AFTER: An AFTER trigger fires following the successful completion of the triggering action. So, for example, the trigger would fire after an UPDATE statement has executed and after constraints have been checked and verified.
Fire On Insert	An INSERT trigger must be associated with an INSERT statement. For example, if a data load operation doesn't include an INSERT statement, the trigger won't be invoked.
Fire On Update	An UPDATE trigger can be associated with specific columns of the base table and will only be activated if those columns are updated.
Fire On Delete	A **DELETE** trigger is associated with a DELETE operation.
Encrypted	If you choose to encrypt the trigger, the trigger can't be published as part of SQL Server replication.

4 Finally, use the **Execute** button to create the object.

Triggers (SQL Server) - Definition

Complete the CREATE TRIGGER outline provided by typing or pasting the body of the trigger.

Unique Keys Wizard (SQL Server)

Unique keys can enforce logical keys that are not chosen as the primary key. In other words, you can use a unique key to ensure no duplicate values are entered in specific columns that are not a part of the primary key. Although you can only attach one primary key to a table, you can attach multiple unique keys. Also, you can use unique keys on columns that allow null values.

To create a new unique key using a wizard:

1 Open a creation wizard for a unique key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Unique Keys (SQL Server) - Properties.
- **Columns** panel - Unique Keys (SQL Server) - Columns.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Unique Keys (SQL Server) - Properties

When creating or editing a unique key, this tab/panel lets you work with the following settings:

Setting	Description
Table Owner and **Table Name**	Choose the owner and name of the table in which the unique key is being created.
Name	Provide a name of the unique key being created.
Build Online	Enabling this feature specifies that the ONLINE=ON clause is used when creating this object and can subsequently be used when rebuilding or dropping this object.
Max degree of parallelism	Lets you specify a MAXDOP index operation value, limiting the number of processors used in parallel plan execution.
Clustered	Enable or disable clustering. With this check box enabled, subsequent rebuild and drop operations offer an online option.
File Group	If you do not specify a filegroup, Microsoft SQL Server creates the unique key in the default filegroup.
Fill Factor	This specifies how full each index page that's storing data should be. The fill factor is a percentage value between 0 and 100.

Unique Keys (SQL Server) - Columns

From the **Column** dropdown, select a column for the primary key and specify a **Sort** option. To add more columns, click the **New** button and then follow the steps in the last instruction. Use the Delete button to drop columns.

User Messages Wizard (SQL Server)

The User Message wizard lets you create multiple-language versions of user messages, such as errors and warnings, associated with a single message number. Key properties include the severity level and whether the message is automatically written to the NT Event Log.

Note: The user messages node only displays under the master database.

To create a new user message using a wizard:

1 Open a creation wizard for a user message. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see User Messages (SQL Server) - Properties.
- **Information** panel - for details, see User Messages (SQL Server) - Information.
- **Object Permissions** and **System Permissions** panels - Setting Permissions or Privileges for an Object
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

User Messages (SQL Server) - Properties

When creating or editing a user message, this tab/panel lets you work with the following settings:

Setting	Description
Message Number	Lets you specify a message number. The number must be greater than 50,000. The wizard automatically loads a value one higher than the currently highest used message number, but you can override the value.
Severity	Lets you select a predefined severity level between 001 and 025. The levels and their meanings are indicated in the User Message editor.
Write to NT Event Log	Lets you specify that the message is always written to the Windows NT Event Log.

User Messages (SQL Server) - Information

Lets you create the different language versions of the text for the message. The first version must be created in us_english. Click the **Add new text for the user message** button, and in the dialog box that opens, select a **Language** of us_english and provide the **Message Text**.

You can subsequently use the same process to create each different language version of the us_english message.

Note: You cannot create two versions of a message for the same language.

This tab/panel also lets you edit and delete messages.

Users Wizard (SQL Server)

The User Wizard lets you create a user who will then have access to the database where you are registering him or her. You can also identify the appropriate user group and the system privileges you want to assign to the new user.

To create a new user using a wizard:

1 Open a creation wizard for a user. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Users (SQL Server) - Properties.
- **Roles** panel - Users (SQL Server) - Roles.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Users (SQL Server) - Properties

When creating or editing a user, this tab/panel lets you work with the following settings:

Setting	Description
Login Name	Select the server login associated with this user.
Name	Provide the user name.

Users (SQL Server) - Roles

For each role to be assigned to the user, select the check box beside that role.

User Datatypes Wizard (SQL Server)

User datatypes promote domain consistency by streamlining the definition of commonly used table columns in a database. You can build a customized datatype from system datatypes and bind defaults and rules to it to enhance integrity. When you reference the user datatype in a column, the column assumes all of the properties of the user datatype.

To create a new user datatype using a wizard:

1 Open a creation wizard for a user datatype. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - User Datatypes (SQL Server) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

User Datatypes (SQL Server) - Properties

When creating or editing a user datatype, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the owner of the user datatype.
Datatype	Provide a name for the datatype.
Type	Select the base datatype.
Size	Provide the size of the datatype.
Allow Nulls	Null has no explicitly assigned value. Null is not equivalent to zero or blank. A value of null is not considered to be greater than, less than, or equivalent to any other value, including another value of null.
Default Binding	Defaults promote data integrity by supplying a default value to a column if the user does not explicitly provide one. They are reusable objects that you can bind to user datatypes.
Rule Binding	Rules promote data integrity by allowing you to validate the values supplied to a column. They are reusable objects that you can bind to user datatypes.

Views Wizard (SQL Server)

Views are SQL queries stored in the system catalog that customize the display of data contained in one or more tables. Views behave like tables because you can query views and perform data manipulation operations on them. However, views do not actually store any data. Instead, they depend on data contained in their base tables.

To create a new view using a wizard:

1 Open a creation wizard for a view. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Views (SQL Server) - Properties.
- **Definition** panel - for details, see Views (SQL Server) - Definition.

3 Finally, use the **Execute** button to create the object.

Views (SQL Server) - Properties

When creating or editing a view, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the owner of the view. The owner of the view must have SELECT privileges for the tables in the CREATE view statement or DBADM authority on the database that contains the table.
Name	Provide a name for the view.
Encryption	If you select this option, the view will not be published with SQL Server replication.
Schema Binding	When you specify this option, the base table or tables cannot be modified in a way that would affect the view definition. The view definition itself must first be modified or dropped to remove dependencies on the table that is to be modified
Check Condition	When a row is modified through a view, this option makes sure the data remains visible through the view after the modification is committed.
Owner	Select the owner of the view. The owner of the view must have SELECT privileges for the tables in the CREATE view statement or DBADM authority on the database that contains the table.

Views (SQL Server) - Definition

Complete the CREATE VIEW statement by typing or pasting in the relevant query.

MySQL object wizards

You create MySQL objects using the following wizards:

- Databases wizard (MySQL)
- Foreign Keys wizard (MySQL)
- Functions wizard (MySQL)
- Indexes, Primary Keys, or Unique Keys wizard (MySQL)
- Tables wizard (MySQL)
- Users wizard (MySQL)

Databases wizard (MySQL)

The MySQL Database Wizard lets you build and execute a basic CREATE DATABASE statement and grant privileges on the database.

To create a new database using a wizard:

1 Open a creation wizard for a database. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Databases - Properties
- **Privileges** panel - Setting Permissions or Privileges for an Object
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Databases - Properties

When creating or editing database, this tab/panel lets you work with the following settings:

Setting	Description
Database Name	The database name you choose can be up to 64 characters long. Feel free to use all alphabetic characters or mix in some special or numeric characters. But a name cannot be exclusively special or numeric characters.
Default Character Set	This is the character set for a language or alphabet. If you do not make a selection from the drop-down list (click the arrow to see the choices), the database will use the server’s default character set.
Collation	The collation encodes the rules governing character used for a language (like Greek) or an alphabet. The database will use the server’s default collation unless you specify otherwise.

Note: It’s possible to create databases with different character sets and collations on the same MySQL server.

Foreign Keys wizard (MySQL)

The MySQL Foreign Keys Wizard lets you build and execute an ALTER TABLE statement with an ADD CONSTRAINT option implementing a foreign key.

The wizard makes it easy for you to create a relational link between two tables, thereby speeding queries and giving you faster access to data. By using the Create Foreign Key Wizard you obviate the need for remembering the code underlying the creation process.

To create a new foreign key using a wizard:

1 Open a creation wizard for a foreign key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Foreign Keys - Properties
- **Column Mapping** panel - for details, see Foreign Keys - Column Mapping
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Foreign Keys - Properties

When creating or editing a foreign key, this tab/panel lets you work with the following settings:

Group	Setting	Description
Constraint Name	**System Generated** and **User Specified**	The constraint name must be unique. You can either rely on MySQL to assign a unique name or you can specify one.
Constraint State	**On Update** and **On Delete**	Lets you specify ON UPDATE and ON DELETE values of NO ACTION, CASCADE, RESTRICT, or SET NULL.

Foreign Keys - Column Mapping

1 Under **Referenced Table**, choose the **Database** and then the referenced, or parent, **Table**.

2 Under the **Main Table**, select the referencing **Table**.

3 Under the **Main Table**, select the check box corresponding to the referring column and then under **Referenced Table**, select the check box corresponding to the referenced column.

Functions wizard (MySQL)

The MySQL Functions Wizard lets you build and submit a CREATE FUNCTION declaration, specifying a return type and the owning library.

Note: Functions must be written in C or C++, your operating system must support dynamic loading, and you must have compiled mysqld dynamically (not statically).

To create a new function using a wizard:

1 Open a creation wizard for a function. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Functions - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Functions - Properties

When creating or editing a function, this tab/panel lets you work with the following settings:

Required Information	Description
Function Name	In 64 characters or less, you can name your function.
Return Value	STRING, REAL (also known as Double), INTEGER
Shared Object Library	Identify the file that holds the library where your functions are stored.
Aggregate	Check the box if the function you are creating will collapse a large amount of data into a single output, or aggregate the data.

Indexes, Primary Keys, or Unique Keys wizard (MySQL)

The Indexes Wizard, Primary Keys Wizard, and Unique Keys wizard offer the same steps in creating these MySQL objects.

To create a new index, primary key, or unique key using a wizard:

1 Open a creation wizard for an index, primary key, or unique key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Indexes, Primary Keys, or Unique Keys - Properties
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object. For more information, see Previewing the DDL Generated to Create the New Object.

Indexes, Primary Keys, or Unique Keys - Properties

When creating or editing an index, primary key, or unique key, this tab/panel lets you work with the following settings:

Setting	Description
Table Name	Select the table on which you want to create the object.
Index Name	Provide a name of up to 64 alphanumeric characters.
Constraint Type	Primary: Each record of a table is identified as uniquely and relates to a foreign key in another table. Unique: Ensures that there is no duplication between values in columns where you place this constraint. Full Text: Enables the search for several words in arbitrary order in a table. Spatial: Allows you to find records that are defined by location, or geometry types.
Index Storage Type	Hash: Used for equality comparisons. Only whole keys can be used to search for a row. BTree: Tree data structure that keeps data sorted. BTrees grow from the bottom up as elements are inserted. RTree: Tree data structure used for spatial indexes and access to multidimensional information.
Specify Columns in Index	Columns are listed by name and datatype and whether or not they are nullable. As you check selections, the sort order is identified.

Tables wizard (MySQL)

The MySQL Table Wizard lets you create a basic table definition.

To create a new table using a wizard:

1 Open a creation wizard for a table. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Tables - Properties
- **Columns** panel - for details, see Tables - Columns
- **Indexes** panel - for details, see Tables - Indexes
- **Foreign Keys** panel - for details, see Tables - Foreign Keys
- **MERGE tables** panel - for details, see Tables - MERGE Tables
- **Privileges** panel - for details, see Setting Permissions or Privileges for an Object
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object

3 Finally, use the **Execute** button to create the object.

Tables - Properties

When creating or editing a table, this tab/panel lets you work with the following properties:

Note: Prior to working with MySQL table properties, you should have a detailed understanding of MySQL table creation options. For access to MySQL documentation, see Accessing Third Party Documentation.

Setting	Description
Table Name	Provide a unique name of up to 64 characters.
Storage Type	Lets you select a storage engine value of MyISAM, InnoDB, BerkeleyDB, ISAM, MRG_MyISAM, HEAP, or MEMORY.
Row Format	Lets you select a row format of DEFAULT (returns the default value if there is one), FIXED (each row is stored with a fixed number of bytes), DYNAMIC (data records have variable length), or COMPRESSED (each record is compressed separately).
Create via SELECT	When you create a table using a SELECT command, the individual columns take their data types from the SELECT command, and don't have to be declared explicitly. Attributes, however, are not carried over.
Unique Key Violations	When you try to insert or update a row that causes a violation, you can choose to IGNORE the violation and continue processing the next row. Or, you can choose REPLACE, which is equivalent to an insert statement if a violation occurs. The row will be replaced with the new data.
Default Character Set	This is the character set for a language or alphabet.
Default Collation	The collation encodes the rules governing character used for a language (like Greek) or an alphabet.
Auto-increment	You can specify the initial value used for the auto-increment sequence. This is possible only for MyISAM tables and for InnoDB tables built in MySQL versions 5.0.3 or greater. For InnoDB tables built in versions prior to 5.0.3, you can manually insert a dummy row with a value one less than the desired value after creating the dummy table; you then delete the dummy row.
Comment	Enter a descriptive comment.
Min Rows and **Max Rows**	Note that if you set the value for either parameter at 0, MySQL removes the setting. MySQL will take care of the row settings in this case.
Average Row Length	The average length (compressed or uncompressed) of table rows in the table space.
Pack Keys	This setting is only enabled for a **Storage Type** of **MyISAM**. It lets you specify a PACK KEYS option value of 0 (keys packing disabled), 1 (smaller indexes), or DEFAULT (pacl long columns only).
Check Sum	This setting is only enabled for a **Storage Type** of **MyISAM**. When selected the CREATE TABLE statement is issued with a CHECKSUM = 1 option. When deselected, the CREATE TABLE statement is issued with a CHECKSUM = 0 option.
Delay Key Write	This setting is only enabled for a **Storage Type** of **MyISAM**. When selected, the CREATE TABLE statement is issued with a DELAY_KEY_WRITE = 1 option. When deselected, the CREATE TABLE statement is issued with a DELAY_KEY_WRITE = 0 option.

Tables - Columns

When creating or editing a table, this tab/panel lets you modify the table’s column setup, as follows:

- Use the **Add Column** button to add the columns to the table. After providing a **Name** for a new column, you can select a **Type** and modify other **Datatype** properties corresponding to the type you selected. When selected, the **Allow Nulls** property corresponds to a CREATE TABLE with a NULL attribute, while unselected it corresponds to a NOT NULL attribute. When available the **Default Value** property corresponds to a DEFAULT attribute.
- Modify a column definition by selecting the column from the column list and modifying values under **Column Attributes**.
- Delete a column definition by selecting the column from the column list and clicking **Delete**.
- Use the arrow buttons to change the position of a selected column.

Tables - Indexes

When creating or editing a table, this tab/panel lets you build an associated PRIMARY KEY, UNIQUE KEY, FULL TEXT, or SPATIAL clause that will be included with the CREATE TABLE statement.

- Use the Insert a New Index button to add an index. In the **Index** field, type a name for the index, select a **Constraint Type** of PRIMARY KEY, UNIQUE KEY, FULL TEXT, or SPATIAL, and select an **Index Type** of BTREE, HASH, or RTREE as appropriate to the constraint type. Under **Specify Columns in Index**, select the column or columns that are to make up the index.
- Modify an index definition by selecting the index from the list and modifying the **Index** name, **Constraint Type**, **Index Type** or the columns making up the index.
- Delete an index by selecting the index from the list and clicking the **Remove** button.

Tables - Foreign Keys

When creating or editing a table, this tab/panel lets you work with foreign keys for a table:

Note: This tab/panel is only available for a table with a **Storage Type** property value of INNODB.

- Use the Insert a New Foreign Key button to add a foreign key. Provide a **Foreign Key Name**, select the referenced database and table from the **Ref. Database** and **Ref. Table** dropdowns, and select **On Delete** and **On Update** actions of NONE, CASCADE, DEFAULT, SET NULL, NO ACTION, or RESTRICT. Under **Main Table**, select the foreign key column and then under **Referenced Table**, select the column that the foreign key is to reference.
- Modify a foreign key definition by selecting the foreign key from the list and changing the name, referenced database or table, delete or update actions, or the column specifications.

- Delete a foreign key by selecting the key from the list and clicking the **Remove the selected foreign key** button.

Tables - MERGE Tables

When creating or editing a table, this tab/panel lets you work with a collection of identical MyISAM tables that are to be used as a single table.

Note: This tab/panel is only available for a table with a **Storage Type** property value of MRG_MyISAM.

- Use the New button to add a MyISAM table to the collection, selecting a table from the associated dropdown.
- Use the Delete button to remove a selected MyISAM table from the collection.
- Use the arrow buttons to change the ordering position of a selected table.

Users wizard (MySQL)

The MySQL Users Wizard lets you provide basic identification, grant privileges, and specify valid host information for a user definition.

To create a new user using a wizard:

1 Open a creation wizard for a user. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **User Information** panel - for details, see Users - User Information
- **User Hosts** panel - for details, see Users - User Hosts
- **System Privileges** and **Object Privileges** panels - for details, see Setting Permissions or Privileges for an Object
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object

3 Finally, use the **Execute** button to create the object.

Users - User Information

When creating or editing a user definition, this tab/panel lets you work with the following settings:

Setting	Description
User Name	The name of the new user. At this time, the name cannot exceed 16 characters, although the field accepts 34. The discrepancy cannot be enforced, so we urge you to show a little restraint. This is the name that is displayed in the User branch of the Navigator/Explorer tree. If you leave this field blank and complete the process, you create an <anonymous> user. Only one anonymous user per datasource is allowed.
Full Name	OPTIONAL. At this time, the name cannot exceed 16 characters, although the field accepts 34. If you entered a nickname or an alias in the user name field, you can display the user's true identity here.
Description	OPTIONAL. Write a brief description of the user if you want.
Email	OPTIONAL. The new user's email address.
Contact Information	OPTIONAL. The new user's title, address, phone number or whatever contact information you want to include.
User Icon	OPTIONAL. If you want to identify your new user by color or pattern, you can use the icon editor to make a distinguishing mark. You can assign groups of users the same icon or assign each user his or her own icon. This tool helps you distinguish users at a glance. Note: If you don't want to use the head that's preloaded, simply click the red X and start designing your own with the other tools.

Users - User Hosts

When creating or editing a user definition, this tab/panel lets you specify the valid hosts from which the user can connect to the server.

- Add a new host to the list by clicking the **New** button. Use the associated dropdown to select a valid host-name value (such as localhost or %), overwriting the text as necessary to provide a specific quoted identifier. Provide a password in the **Password** and **Confirm** fields and click **Apply.**

Note: Click **Apply To All** to assign the current password for each host currently in the list.

- Remove a selected host from the list by clicking the **Delete** button.

Oracle Object Wizards

You create Oracle objects using the following wizards:

- Clusters Wizard (Oracle)
- Database Links Wizard (Oracle)
- Directories Wizard (Oracle)
- Disk Groups Wizard (DBArtisan - Oracle)
- Foreign Keys Wizard (Oracle)
- Functions Wizard (Oracle)
- Indexes Wizard (Oracle)
- Oracle Job Queue Wizard (Oracle)
- Jobs wizard (Oracle)
- Libraries Wizard (Oracle)
- Materialized Views Wizard (Oracle)
- Materialized View Logs Wizard (Oracle)
- Outlines Wizard (Oracle)
- Packages Wizard (Oracle)
- Primary Keys Wizard (Oracle)
- Procedures Wizard (Oracle)
- Programs wizard (Oracle)
- Profiles Wizard (Oracle)
- Redo Log Groups Wizard (DBArtisan - Oracle)
- Roles Wizard (Oracle)
- Rollback Segments Wizard (Oracle)
- Schedules Wizard (Oracle)
- Sequences Wizard (Oracle)
- Synonyms Wizard (Oracle)
- Tables Wizard (Oracle)
- Tablespaces Wizard (Oracle)
- Triggers Wizard (Oracle)

- Object Types Wizard (Oracle)
- Unique Keys Wizard (Oracle)
- Users Wizard (Oracle)
- Views Wizard (Oracle)

Clusters Wizard (Oracle)

The Cluster Wizard lets you create a cluster. A cluster is a schema object that has one or more tables that all have one or more common columns. Rows of one or more tables that share the same value in these common columns are stored together in the database. The related columns of tables stored in a cluster are known as the cluster key.

Important Notes

- To create a cluster, you need the CREATE CLUSTER or CREATE ANY CLUSTER system privilege.

To Open the Cluster Wizard

1 On the Navigator/Explorer, find the schema where you want to add the new cluster.

2 On the **Cluster** branch, right-click and select **New**.

OR

1 On the main toolbar, click Datasource and scroll to Objects

2 Click **Clusters** and then click **New** from the toolbar.

The tables below describe the fields you may encounter as you complete the wizard.

Required Information	Description
Who owns the cluster?	Pick an owner
What is the name of the cluster?	Type a unique name
On which tablespace do you want to create the cluster?	Self-explanatory
Add columns that are in this cluster These are the columns that are common between the tables you are "clustering"	**Add** or **Edit** Button - For more information, see Adding or modifying a cluster column. **Drop** Button - Drops the column.
What is the size of this cluster?	This is the estimated number of bytes/KB/MB required by an average cluster key and its associated rows. Do not exceed the size of a data block.
What is the cluster type?	Index: Rows having the same cluster key value are stored together. Each separate cluster key is stored only once in each data block. An indexed cluster is helpful if your clustered tables might grow unpredictably. Hash: Rows with the same hash key value are stored together. This is helpful if the tables are static.
If this is a hash cluster, what is the number of hash keys?	Type the number of hash keys. Oracle will round the value up to the nearest prime number.
If this is a hash cluster, what is the hash function?	Oracle uses a hash function to generate a distribution of numeric values, called hash values, which are based on specific cluster key values. The key of a hash cluster, like the key of an index cluster, can be a single column or composite key (multiple column key). To find or store a row in a hash cluster, Oracle applies the hash function to the row's cluster key value. The resulting hash value corresponds to a data block in the cluster, which Oracle then reads or writes on behalf of the issued statement. Default is the Oracle internal hash function, otherwise specify the hash expression you want to use.
How many transaction entries are allowed for each data block in the cluster?	Each transaction that updates a data block requires a transaction entry. Initial (1-255): The initial parameter ensures that a minimum number of concurrent transactions can update a data block, avoiding the overhead of allocating a transaction entry dynamically. Maximum (1-255): The maximum parameter limits concurrency on a data block.
What is the percent of space reserved for future updates?	Percent Free (0-99): This sets the percentage of a data block to be reserved for possible row updates that are included in the block. The value you set is the percent kept free.
What is the minimum percentage of used space that Oracle maintains for each data block?	The storage parameter lets you tune performance by minimizing the occurrence of row migration and chaining caused by update operations that extend the length of rows stored on the data block. Percent Used (1-99)

How large are the cluster's extents?	The unit of space allocated to an object whenever the object needs more space. Initial Extent - The initial space extent (in bytes) allocated to the object. Next Extent - The next extent (in bytes) that the object will attempt to allocate when more space for the object is required.
Specify the number of free lists	Free lists are lists of data blocks that have space available for inserting rows. Identifying multiple free lists can reduce contention for free lists when concurrent inserts take place and potentially improve the performance of the cluster. Free Lists: The default and minimum value is 1; this option should be set higher if multiple processes access the same data block.
Specify the number of free list groups (specify only if you are using the parallel server option)	This is the number of groups of free lists for the database objects being created.
Define a default buffer pool for this cluster	Default - Select to retain the default. Keep - Select to retain the object in memory to avoid I/O conflicts.
Oracle's parallel query option	The parallel server query option lets you process queries using many query server processes running against multiple CPUs, which provides substantial performance gains such as reducing the query completion time.
Choosing Cache	Cache: This keeps the data block in memory by placing it at the most recently used end. This option is useful for small lookup tables. No Cache

Adding or modifying a cluster column

The **Add or Modify Cluster Column** dialog lets you manage cluster columns. You can open the dialog in the Oracle Cluster Wizard.

The table below describes the options and functionality on the **Add or Modify Cluster Column** dialog

Option	Description
Column Name	Lets you type the column name.
Datatype	Lets you select the datatype for the cluster. If you select CHAR, RAW or VARCHAR2, in the Width box, type the width value. If you select NUMBER, in the Width box, type the width value and in the Scale box, type the scale value.

Completing the Add or Modify Cluster Column Dialog Box

To complete this dialog, do the following:

1 In the **Add Cluster Column** dialog, in the **ColumnName** box, type the column name.

2 Click the **Datatype** list, click the datatype for the cluster.

- If you clicked CHAR, RAW or VARCHAR2, in the **Width** box, type the width value.

- If you clicked NUMBER, in the **Width** box, type the width value and in the **Scale** box, type the scale value.

3 Click the **Add** button.

4 To continue adding columns to the cluster, repeat steps 1-3.

5 When you finish adding columns, click **Close**.

The **Add Cluster Column** dialog closes.

For more information, see Adding or modifying a cluster column.

Database Links Wizard (Oracle)

A database link specifies a communication path from one database to another. If you're creating a link to a remote database, a database session is established in the remote database on behalf of the local application request. By creating either a public or private database link, you can determine which schema on the remote database the link will establish connections to by creating fixed, current, and connected database links. By creating a link you can reuse connectivity instructions each time you connect to the remote database.

To Open the Database Link Wizard

1 On the Navigator/Explorer, find the schema where you want to add the new database link.

2 On the Database Links branch, right-click and select **New**.

OR

1 On the main toolbar, click Datasource and scroll to Objects

2 Click **Database Links** and then click **New** from the toolbar.

The table that follows describes the fields you will encounter as you complete the wizard.

Note: To create a public database link, you need CREATE PUBLIC DATABASE LINK privileges.

Required Information	Description
What is the name of the database link?	Create a unique name
Should the database link be made public?	A public link is a link on a local database that's accessible to all users on that database. If you keep the database link private by selecting No, a link is created in a specific schema of the local database and only the owner of the link can use it to access database objects in the corresponding remote database.
What is the name of the remote user?	Self-explanatory
What is the remote user's password?	Create a password for the remote user.
What is the connection string?	Self-explanatory.

Disk Groups Wizard (DBArtisan - Oracle)

This wizard builds a CREATE DISKGROUP statement with redundancy and state properties, the associated disks and their properties, and attribute templates.

Note: This functionality is available as of Oracle 10g.

Note: Before working with disk groups, consult Oracle documentation for general information on disk groups and ASM instances, and details on ASM operation and related permissions requirements. For more information, see Accessing Third Party Documentation.

To create a disk group using a wizard

1 On a connected Oracle datasource, expand the **Instance** and **ASM** nodes.

2 Right-click the **Disk Groups** node and select **Create** from the context menu.

Note: If this is the first time you have initiated a disk group action since connecting, you are prompted to choose between a connection to the ASM instance or to continue using the default datasource connection. For details on the implications, see Manage ASM Connection (DBArtisan).

3 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Disk Groups (Oracle) - Properties.
- **Disks** panel - for details, see Disk Groups (Oracle) - Disks.
- **Templates** panel - for details, see Disk Groups (Oracle) - Templates.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

4 Finally, use the **Finish** button to create the object.

The Disk Groups Editor has properties and options over and above those in the Disk Groups Wizard. For details, see Disk Groups Editor (DBArtisan - Oracle).

Disk Groups (Oracle) - Properties

When creating or editing a disk group, this tab/panel lets you work with the following settings:

Setting	Description
Name	Provide a name for the disk group
Redundancy Type	Lets you select a REDUNDANCY clause value of NORMAL, EXTERNAL, or HIGH,
Group State After Creation (labelled **Group State** when editing an existing disk group)	Lets you select an initial disk group state of MOUNTED or DISMOUNTED. By default the disk group is MOUNTED after creation. If you select DISMOUNTED, an ALTER statement is executed to change the state of the disk group to DISMOUNTED after creation.
Allocation Unit Size (available as of Oracle 11gR1)	Specify the size of the allocation unit for the disk group.
Compatibility (available as of Oracle 11gR1)	Specify the ASM instance minimum software version.
Database Compatibility (available as of Oracle 11gR1)	Specify the database instance minimum software version.
ADVM Compatibility (available as of Oracle 11gR2)	Lets you add an ATTRIBUTE 'COMPATIBLE.ADVM' =... clause, specifying whether the disk group can contain ASM columns.

Disk Groups (Oracle) - Disks

When creating or editing a disk group, this tab/panel lets you manage the disks of the disk group and their properties.

To add a disk to the group

1 Click **Add**, select an entry from the **Disks** dropdown, optionally provide a **Name** for the disk, and press TAB or ENTER.

The disk is added to the **Path** list on the left, with default attribute values.

2 Proceed to edit the disk attributes.

3 Add more disks as necessary.

To edit disk attributes

1 Select the disk in the **Path** list on the left. The controls in the**Property**/**Values**list are updated with values of the selected disk.

2 Use the following table as a guide to providing additional property values.

Property or group	Description
Size	Specify a disk size in KB, MB, or GB.
Failure Group	Specify the failure group to which the disk is to be added.
Type (available as of Oracle 11GR1)	Lets you select a REGULAR or QUORUM failure group or disk qualifier. If QUORUM is specified at creation, the QUORUM keyword will be specified in any subsequent ALTER statements that will affect the disk or the failgroup.
Force	If selected, the disk is added to the disk group even if the disk is already a member of a different disk group.

To delete a disk from the disk group

1 Select a disk from the **Path** list.

2 Click **Delete**.

Disk Groups (Oracle) - Templates

When creating or editing a disk group, this tab/panel lets you manage disk group template clauses submitted with the CREATE/ALTER DISKGROUP statement. Each template is a named set of attributes applied to files on the disk group.

Note: Only custom templates need to be added when creating a disk group. The System Templates (CONTROL, DATAFILE, ONLINELOG, ARCHIVELOG, TEMPFILE, BACKUPSET, PARAMETERFILE, DATAGUARDCONFIG, FLASHBACK, CHANGETRACKING, DUMPSET, XTRANSPORT, and AUTOBACKUP) are automatically created for the disk group when the group is created. For more information, see Disk Groups Editor (DBArtisan - Oracle).

To add a template to the list of disk template clauses for this disk group

1 Click the **New** button. The **Add new template** dialog opens.

2 Use the following table as a guide to providing information on this dialog:

Setting	Description
Name	Provide a name for the template.
Stripe	Select COARSE to stripe files every 1MB. Select FINE to stripe files every 128KB.
Redundancy	Select a redundancy level of UNPROTECTED, MIRROR, PARITY, or HIGH.
Primary Region and **Mirror Region** (available as of Oracle 11GR2)	Let you add an ATTRIBUTE disk region clause of HOT/COLD and MIRRORHOT/MIRRORCOLD values to the template definition.

3 Click **OK**.

To modify a template already in the list

1 Select the template and click the **Edit** button. The **Edit template** dialog opens.

2 Use the table above as a guide to providing information on this dialog.

3 Click **OK**.

To delete a template from the list

1 Select the template and click the **Drop** button. You are prompted to verify.

2 Click **Yes**.

Directories Wizard (Oracle)

A directory object specifies an alias for a directory on the server file system where external binary file LOBs and external table data are located. The wizard completes a CREATE DIRECTORY statement from the information you supply. The Directory Wizard prompts you to name the directory and provide the full-qualified directory path.

To Open the Directories Wizard

1 On the Navigator/Explorer, find the Datasource where you want to create a directory and expand the Storage node.

2 Right-click **Directories** and select **New**.

The table that follows describes the fields you will encounter as you complete the wizard.

Required Information	Description
What is the name of the directory?	Type a meaningful name for the directory.
What is the directory path?	Type the full path name of the outside operating system directory that you want to alias in the directory (for example, /Video/Library/G_Rated). NOTE: Oracle trusts that directory you're specifying exists. The onus is on you to make sure it's valid and in the correct format (as required by your operating system).

After you have created the Directory, you can give other users Read or Write privileges by opening the new Directory on the Directories node, and making changes at the Privileges tab.

Foreign Keys Wizard (Oracle)

A foreign key value in one table (child table) refers to a primary key value in another table (parent table). For example, the Author Name column in a publisher's database may be the primary key for a table of addresses that includes the Author Name column. If an author isn't included in the parent table, you can't add the address to the dependent address table. So foreign keys enforce referential integrity between tables by verifying the existence of foreign key values in the parent table before letting you insert or update foreign key values in the child table. In other words, a foreign key is an integrity constraint that requires each value in one table's column to match a value in a related table's data.

To create a new foreign key using a wizard:

1 Open a creation wizard for a foreign key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Foreign Keys (Oracle) - Properties.
- **Column Mapping** panel - Foreign Keys (Oracle) - Column Mapping.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Foreign Keys (Oracle) - Properties

When creating or editing a foreign key, this tab/panel lets you work with the following settings:

Setting	Description
Table Owner	The owner of the table where the foreign key is being created.
Table Name	This is the table where the foreign key link originates--the child table.
Name	Lets you select a constraint name. System Generated Name - DB2 automatically generates a name. User Specified Constraint Name - You type the name.
Enabled	Enable or disable the foreign key. Enabled ensures that all data modifications to a given table (or tables) satisfy the conditions of the constraints. When disabled, the constraint is temporarily not operational.
Delete Rule	Select an action: **NO ACTION** - ensures that referenced values cannot be updated or deleted if to do so would violate referential integrity. **CASCADE** permits a referenced row in a child table to be deleted/updated if it is deleted/updated in the parent table. A row in the child table is SET NULL when rows in the parent table are deleted/updated.

Foreign Keys (Oracle) - Column Mapping

Under **Referenced Table**, choose the **Owner** and then the **Name** of the referenced, or parent, table.

Under the **Main Table**, select check boxes corresponding to the columns that are to reference columns in the referenced table. Then, under **Referenced Table**, select the corresponding column check boxes.

Functions Wizard (Oracle)

Functions are subroutines that you define and are useful for reusing application logic. You can use functions to determine the best methods for controlling access and manipulation of the underlying data contained in an object. A function returns a value, unlike a procedure, which does not.

To create a new function using a wizard:

1 Open a creation wizard for a function. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Functions (Oracle) - Properties.
- **Definition** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Functions (Oracle) - Properties

When creating or editing a function, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the owner of the function.
Name	Provide a name for the function.

Functions (Oracle) - Definition

Complete the CREATE FUNCTION outline provided by typing or pasting the body of the function.

Indexes Wizard (Oracle)

Indexes are optional structures associated with tables. You can create indexes specifically to speed SQL statement execution on a table. When properly used, indexes are the primary means of reducing disk I/O. Indexes are logically and physically independent of the data in the associated table. Unique Indexes guarantee that no two rows of a table have duplicate values in the columns that define the index.

Note: The Index Wizard varies slightly in content based on the version of Oracle to which you are connected.

Note: To create indexes in your own schema, you need INDEX privileges on the target table. To create indexes in other schema, you need CREATE ANY INDEX privileges.

Note: You can place a unique key constraint on an Index-Organized table.

Tip: Index-organized tables take up less storage space and quickly access table rows. Index-organized tables stores rows in primary key order reducing the amount of storage space needed.

Tip: An advantage of using index-organized tables is that the tables use less memory because key columns are not duplicated in the table and index. The remaining non-key columns are stored in the index structure.

To create a new index using a wizard:

1 Open a creation wizard for an index. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Indexes (Oracle) - Properties.
- **Columns** panel - Indexes (Oracle) - Columns.
- **Storage** panel - Indexes (Oracle) - Storage.
- **Partition** panel - Indexes (Oracle) - Partition.
- **Definition** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Indexes (Oracle) - Properties

When creating or editing an index, this tab/panel lets you work with the following settings:

Setting	Description
Table Owner and **Table Name**	Choose the owner and name of the table in which the index is being created.
Owner and **Name**	Choose the owner and provide the name of the index being created.
Index Type	NONUNIQUE - In a non-unique index, the ROWID is treated as part of the key. Oracle treats a constraint as deferrable. UNIQUE - Select if the index is a unique constraint.The values in the indexed columns must be distinct. BITMAP - Widely used in data warehousing environments. The environments typically have large amounts of data and ad hoc queries, but a low level of concurrent DML transactions.
No Sort	Enable this feature if the rows in the table already stored in ascending order. This increases the speed of the index creation process. Oracle does not sort the rows.
Logging	Enabling logs this operation to the redo file.
Reverse	Enabling this feature stores the bytes of the index block in reverse order and excludes the ROWID. The ROWID is a globally unique identifier for a row in a database. It is created at the time the row is inserted into a table, and destroyed when it is removed from a table.
Function-Based	Permits the results of known queries to be returned much more quickly. When you select this option, you are asked for the expression that governs the function-based index you are creating.
Invisible (Oracle 11g)	Adds an INVISIBLE keyword to the DDL generated to create or edit this index. The optimizer ignores an invisible index unless the OPTIMIZER_USE_INVISIBLE_INDEXES is set to TRUE at the session or system level.
No Parallel Execution	The parallel server query option lets you process queries, using many query server processes, running against multiple CPUs. This option provides substantial performance gains such as reduction of the query completion time. After creation, ALTER INDEX for NOPARALLEL execution - when you use multiple query servers and you select this option, the parallel query option remains in place, but parallel processing will be removed. If, for example, multiple users on numerous nodes are modifying the same small set of data, the cost of synchronization from the parallel processing may have an unnecessarily large drag on throughput.
Parallel Degree	The value you select indicates the number of query server processes that should be used in the operation.
Parallel Instances	The value you select indicates how you want the parallel query partitioned between the Parallel Servers.

Indexes (Oracle) - Columns

From the **Column** dropdown, select a column for the index and specify a **Sort** option. To add more columns, click the **New** button and then follow the steps in the last instruction. Use the Delete button to drop columns.

Indexes (Oracle) - Storage

When creating or editing an index, this tab/panel lets you work with the following settings:

Setting	Description
Data Block Storage group	Select the DEFAULT **Tablespace** only if you are creating a local partitioned index and want the partitions in the same tablespace as the partitions in the underlying table. (Each partition of a local index is associated with one partition of the table. Oracle can then keep the index partitions in synch with table partitions.) A transaction entry is needed for each INSERT, UPDATE, DELETE, etc. statement that accesses one or more rows in the block. Transaction entries in many operating systems require approximately 23 bytes. **Percent Free** identifies how much space you want to allocate for new rows or updates to existing rows. **Initial Transactions** ensures that a minimum number of concurrent transactions can update an index block, avoiding the overhead of allocating a transaction entry dynamically. **Maximum Transactions** limits concurrency on an index block.
Extents group	An extent is the unit of space allocated to an object whenever the object needs more space. **Initial Extent** - The initial space extent (in bytes) allocated to the object. **Next Extent** - The next extent (in bytes) that the object will attempt to allocate when more space for the object is required. **Percentage Increase** - Lets you type the percentage. NOTE: You should be careful when setting Percent Increase because it magnifies how an object grows and, therefore, can materially affect available free space in a tablespace. **Minimum Extents** - For a dictionary managed tablespace, this is the total number of extents to be allocated when the index is first created. For a locally managed tablespace, this is simply the initial amount of space allocated. **Maximum Extents** - For a dictionary managed tablespace, this is the total number of extents that can ever be allocated to the index. In a locally managed tablespace, the database will automatically manage the extents.
Freelists group	**Free lists** let you manage the allocation of data blocks when concurrent processes are issued against the index. You can potentially improve the performance of the index by identifying multiple free lists, which can reduce contention for free lists when concurrent inserts take place. The default and minimum value is 1. You should increase this number if multiple processes access the same data block. **Free List Groups** is the number of groups of free lists. NOTE: This option is only applicable for the parallel server option.
Buffer Pool	DEFAULT - Choose this if you want to use the default bufferpool. KEEP - Use this to retain the object in memory to avoid I/O conflicts. This type of bufferpool stores frequently referenced data blocks in a separate cache. RECYCLE - Select this option to save cache space by ridding data blocks from memory as soon as they are no longer in use.

Indexes (Oracle) - Partition

Clicking **Create Partition** opens a wizard that lets you create a partition.

Oracle Job Queue Wizard (Oracle)

This wizard lets you build and submit a DBMS_JOB.SUBMIT call, creating a once-only or scheduled job, optionally submitted as broken.

To create a new job queue using a wizard:

1 Open a creation wizard for a job queue. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Job Queues (Oracle) - Properties.
- **Definition** panel - for details, see Job Queues (Oracle) - Definition.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object. For more information, see Previewing the DDL Generated to Create the New Object.

Job Queues (Oracle) - Properties

When creating or editing a job, this tab/panels lets you work with the following settings:

Option	Availability and Description
No Parse	Specifies the value of the NO_PARSE flag. If deselected, the associated procedure is parsed on submission. If selected, the associated procedure is parsed the first time the job is executed.
Submit as disabled	If selected, the DBMS_JOB.SUBMIT call generated to create the job, is followed by a DBMS_JOB.BROKEN call. Marking the job as broken, this disables the job and in order to execute the job, you must use the **Run Job** object action. For details, see Run Job (job queues).
Next Date	Use this control to open a calendar that lets you select a date for the job to run with a default time of midnight. You can then modify the time value.
Run Again	
Every	Only available if **Run Again** is selected and **Use Custom Expression** is not selected..
Time Unit	
Use Custom Expression and **Custom Expression**	Only available if **Run Again** is selected. **Custom Expression** becomes available when **Use Custom Expression** is selected.

Job Queues (Oracle) - Definition

When creating or editing a job, the tab/panel lets you provide the DBMS_JOB.SUBMIT call's WHAT parameter, specifying the name of the PL/SQL procedure to run.

Jobs wizard (Oracle)

This wizard builds and submits a call to DBMS_SCHEDULER.CREATE_JOB with optional DBMS_SCHEDULER.SET_JOB_ARGUMENT_VALUE, DBMS_SCHEDULER.SET_ATTRIBUTE, and DBMS_SCHEDULER.ENABLE calls. This lets you create a fully-specified job.

Note: This functionality is available as of Oracle 10g.

Note: Before working with jobs, consult Oracle documentation for general information on DBMS_SCHEDULER. For more information, see Accessing Third Party Documentation.

To create a job using a wizard

1 In DBArtisan, on a connected Oracle datasource, expand the **Instance** and **Scheduler** nodes, right-click the **Jobs** node and select **Create** from the context menu.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Jobs (Oracle) - Properties.
- **Action** panel - for details, see Jobs (Oracle) - Action.
- **Arguments** panel - for details, see Jobs (Oracle) - Arguments.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Finish** button to create the object.

Jobs (Oracle) - Properties

When creating or editing a job, this tab/panel lets you work with the following settings:

Group	Settings and Description
Creation	Lets you provide an **Owner** and **Name** for the job.
Attributes	Lets you provide **Enabled**, **Description** (COMMENTS), and **Job Class** (JOB_CLASS) attributes.
Schedule	Lets you select a **Schedule Type** of IMMEDIATE, ONCE, REPEATING, or specified SCHEDULE. Depending on your selection, the following settings may be available: **Repeat Interval** - corresponds to a REPEAT_INTERVAL attribute. Calendaring syntax, as described in Oracle DBMS_SCHEDULER documentation, can be typed free-form, and will be passed directly to the server. For information on third-party documentation access, see Accessing Third Party Documentation. **Start Date** and **End Date** - correspond to START_DATE and END_DATE, attributes. **Schedule Owner** and **Schedule Name** - correspond to SCHEDULE_NAME attribute components.
Properties	The **Auto Drop** setting corresponds to a CREATE_JOB AUTO_DROP attribute. Other settings result in one or more DBMS_SCHEDULER.SET_ATTRIBUTE calls being generated with RESTARTABLE, JOB_PRIORITY, MAX_RUNS, MAX_FAILURES, FAILED RUNS, STOP_ON_WINDOW_CLOSE, INSTANCE_STICKINESS, or JOB_WEIGHT attributes specified, as appropriate.

Jobs (Oracle) - Action

When creating or editing a job, this tab/panel lets you specify a JOB_ACTION attribute for the DBMS_SCHEDULER.CREATE_JOB call, specifying the job action for the job.

To specify a job action

1 From **Job Type**, select one of PLSQL BLOCK, STORED PROCEDURE, EXECUTEABLE, or CHAIN.

2 Based on your **Job Type** selection, provide additional details as follows:

- PLSQL BLOCK - Type or paste the SQL to be executed.
- STORED PROCEDURE - Provide the **Procedure Owner** and **Procedure Name**.
- EXECUTEABLE - Use the browse button to locate the executable.
- PROGRAM - Provide the **Program Owner** and **Program Name**.
- CHAIN - Provide the **Chain Owner** and **Chain Name**.

Jobs (Oracle) - Arguments

When creating or editing a job, if you selected an **Action** tab choice of PROGRAM, EXECUTABLE, or STORED PROCEDURE, this tab/panel lets you manage parameters to be passed.

Note: If you provide arguments with an **Action** tab choice of PROGRAM, the provided arguments override those specified in the program definition. For details, see Programs wizard (Oracle).

Libraries Wizard (Oracle)

Libraries are an object type that represent a call to an operating system shared library cache. After the call is made, libraries can be used by SQL or PL/SQL to link to external procedures or functions. Libraries are only to be used on operating systems that support shared libraries and dynamic linking. Libraries serve as pointers or aliases to physical operating system shared library files and do not exist as a physical object; rather they rely on the physical existence of the files in the external operating system library to which they refer. To access the function or procedures stored in the library, you need execute privileges at the operating system level where the shared library resides.

- To create a library in your own schema, you need CREATE ANY LIBRARY privileges. To use the functions or procedures stored in the library, you need object EXECUTE privileges on the library.

To Open the Library Wizard

1 On the Navigator/Explorer, find the datasource where you want to create a Library and expand the Schema node.

2 Right-click the **Libraries** node, and select **New**.

The table that follows describes the fields you may encounter using this wizard:

Required Information	Description
Who owns the library?	Self-explanatory.
What is the name of the library?	Self-explanatory.
What is the file specification?	Lets you type the file name and location. You must type the complete location (for example, D:\Embarcadero\ETLIB21D.DLL).

Materialized Views Wizard (Oracle)

A materialized view gives you indirect access to table data by storing a query's results in a separate schema object. Essentially, a materialized view is a database object that contains the results of a query.

The Materialized View Wizard lets you:

- Specify the materialized view owner and to name the materialized view.
- Specify the materialized view's refresh configuration.
- Place the materialized view on a tablespace and specify the query that should be used to populate the materialized view.
- Specify how Oracle should allocate data blocks to store the materialized view.
- Specify how Oracle should manage the growth of the materialized view.
- Specify if Oracle updates the materialized view, register an existing table, and specify how to populate a materialized view.
- Specify if the data for the materialized view is cached, if you want the updates logged, and to specify a number of threads for a parallel operation.
- Specify rollback segments, and enable query rewrites.

To Open the Materialized View Wizard

1 On the Navigator/Explorer, find the datasource where you want to create a Materialized View and expand the Schema node.

2 Right-click the **Materialized Views** node, and select **New**.

The table that follows describes the fields you may encounter as you complete the wizard:

Required Information	Description
Who owns the materialized view?	Self-explanatory
What is the name of the materialized view?	Self-explanatory.
How should the materialized view be refreshed?	Fast - Using the information logged on the materialized view logs or a partition maintenance operation, the refresh applies incremental changes. See Fast Refresh Requirements for more information. Complete - This refresh recalculates the materialized view's defining query. Force - Applies fast refresh when feasible, otherwise uses a complete refresh. Never - Materialized view will not be refreshed with the optional refresh mechanisms.
Choose a refresh mechanism	On Demand - This option requires that all refreshes be manually executed. On Commit - Select to refresh the materialized view whenever Oracle processes a transaction. Only select this option for materialized views on single table aggregates and materialized views containing joins. Automatically - Select to refresh the materialized view automatically. In the On this date: boxes select a time and date, and then select a refresh amount and a unit of time.
Where do you want to place the materialized view?	Select the tablespace where you want the materialized view placed.
What is the materialized view query?	Type the SQL query to be used to populate and to refresh the materialized view.
Select a refresh method	Primary Key - A primary key's values uniquely identify the rows in a table. Changes are propagated according to row changes as identified by the primary key value of the row. Only one primary key can be defined for each table.The primary key of the master table is the basis for this refresh option, which is the default option. ROWID - A globally unique identifier for a row in a database based on the physical row identifiers. A RowID is created at the time the row is inserted into a table, and destroyed when it is removed from a table. ROWID materialized views cannot contain distinct or aggregate functions or GROUP BY subqueries, joins and set operations.
How many transaction entries are allowed for each datablock in the materialized view?	A transaction is a logical unit of work that contains one or more SQL statements. Each transaction that updates a data block requires a transaction entry. Initial (1-255) - Ensures that a minimum number of concurrent transactions can update a data block, avoiding the overhead of allocating a transaction entry dynamically. Maximum (1-255) - Limits concurrency on a data block.
What is the percent of space reserved for future updates?	Percent Free (0 99) - This sets the percentage of a data block to be reserved for possible row updates that are included in the block. The value you set is the percent kept free.
What is the minimum percentage of used space that Oracle maintains for each datablock?	Percent Used (0-99) - Set the amount of space to be used for each datablock. NOTE: The sum of percent free and the percent used cannot exceed 100.

How large are the materialized views extents?	The unit of space allocated to an object whenever the object needs more space. An extent is a specific number of contiguous data blocks set aside for storing a specific type of information. Initial Extent (KB) - The initial space extent (in bytes) allocated to the object. Next Extent - The next extent (in bytes) that the object will attempt to allocate when more space for the object is required.
How many extents should be allocated to the materialized view?	Minimum Extents - The appropriate minimum extents value for the object. Maximum Extents - The appropriate maximum extents value for the object.
What is the growth rate for sizing additional materialized views?	Percent Increase - Magnifies how an object grows and can materially affect available free space in a tablespace. Select a value in the corresponding box.
Can the materialized view be updated?	Yes/No
Do you want to register a prebuilt table to the view?	Yes/No. This option is particularly useful for registering large materialized views in a data warehousing environment.
Should the materialized view be immediately filled?	Yes/No: Select Yes if you want the materialized view populated immediately or during the next refresh operation.
Should data for the materialized view be cached?	Yes/No: Select if you want Oracle to put data you access frequently at the most recently used end of the list in the buffer cache when a full table scan is performed. This option is useful for small lookup tables.
Do you want updates to be logged?	Yes/No.
Do you want to specify the number of threads used in a parallel operation?	Parallel processes means that multiple processes work simultaneously to run a single statement. This can cut the amount of time it takes to get a response. Specify the degree of parallelism if you so desire. If you leave the default at 1, the operation will not be "parallelized."
Would you like to specify rollback segments to be used for the materialized view refresh?	A rollback segment temporarily stores old data that has changed in a SQL statement transaction until it is committed. The "before" image of the database, as it were. Local Rollback Segment - Default indicates that Oracle will select the rollback segment to use on the local machine. Master Rollback Segment - Specify the remote rollback segment used at the remote master site for the individual materialized view.
Is the materialized view eligible for query rewrite?	Select to enable the materialized view for query rewrite. Only enable query rewrite if expressions in the statement are repeatable.
Do you want to partition this materialized view?	Yes/No Partitioning methods available are: Range: Data is mapped to partitions based on ranges of column values. This is the default. Composite: Based on the range method of partitioning, you can create subpartitions within each partition. Hash: Data is distributed evenly over a specified number of partitions. Data need not fit into a logical range. List: You control explicitly how rows map to partitions. List partitions allow you to group and organize unrelated sets of data.

Do you want to enable Row Movement?	Yes/No Enabling row movement allows you to specify whether Oracle can move a table row when you are compressing a table or performing an update on partitioned data.
Select the partitioning columns	Self-explanatory
Select the subpartitioning method	Self-explanatory
Select the subpartitioning columns	Self-explanatory
Hash Partitioning methods	None Partition Definition: Specify number of partitions and (optionally) tablespaces Specify individual partitions by name and (optionally) tablespaces
Create list/ordered list of partitions	Self-explanatory The **Add Partition** dialog may open.
Specify number of subpartions	Self-explanatory Click **Add**, **Insert**, or **Edit** to open a dialog that lets you work with subpartition properties.
Select the default tablespaces to contain the subpartitions (optional)	Self-explanatory

Fast Refresh Requirements

	When the Materialized View has:		
	Only Joins	Joins and Aggregates	Aggregate on a Single Table
Detail tables only	X	X	X
Single table only			X
Table Appears only once in the FROM list	X	X	X
No non-repeating expressions like SYSDATE and ROWNUM	X	X	X
No references to RAW or LONG RAW	X	X	X
No GROUP BY	X		
Rowids of all the detail tables must appear in the SELECT list of the query	X		
Expressions are allowed in the GROUP BY and SELECT clauses provided they are the same		X	X
Aggregates allowed but cannot be nested		X	X
AVG with COUNT		X	X
SUM with COUNT			X
	Only Joins	Joins and Aggregates	Aggregate on a Single Table
VARIANCE with COUNT and SUM		X	X
STDDEV with COUNT and SUM		X	X
WHERE clause includes join predicates which can be ANDed bit not ORed.	X	X	
No WHERE clause			X
No HAVING or CONNECT BY	X	X	X
No subqueries, inline views, or set functions like UNION or MINUS	X	X	X
COUNT(*) must be present			X
No MIN and MAX allowed			X
If outer joins, then unique constraints must exist on the join columns of the inner join table	X		

Materialized View logs must exist and contain all columns referenced in the materialized view and have been created with the LOG NEW VALUES clause			X
Materialized View Logs must exist with rowids of all the detail tables	X		
Non-aggregate expression in SELECT and GROUP BY must be straight columns			X
DML to detail table	X		X
Direct path data load	X	X	X
ON COMMIT	X		X
ON DEMAND	X	X	X

Materialized View Logs Wizard (Oracle)

Materialized view logs are tables that maintain a history of modifications to the master table, and they are used to refresh simple materialized views. When you create a materialized view log, Oracle automatically creates a log table to track data changes in the master table and a log trigger to maintain the data in the log table. A log can refresh the materialized view incrementally and is therefore often less time-consuming than a complete refresh.

The Materialized View Log Wizard lets you:

- Specify the materialized view log owner and master table.
- Select refresh types and select column filters.
- Specify how Oracle should allocate data blocks to store the materialized view log.
- Specify how Oracle should manage the growth of the materialized view.
- Specify if you want the data for the materialized view log cached, if you want updates logged, and to enable parallel query.
- Specify if you want the log to hold new values.

To Open the Materialized View Logs Wizard

1 On the Navigator/Explorer, find the datasource where you want to create a Materialized View Log and expand the Schema node.

2 Right-click the **Materialized View Log** node, and select **New**.

The table that follows describes the fields you may encounter as you complete the wizard:

Required Information	Description
Who owns the materialized view log's master table?	Self-explanatory
Which table will serve as the materialized view log's master table?	Self-explanatory
On which tablespace do you want to place the log?	Self-explanatory
Which refresh types would you like to use?	Primary Key - The log records changes to the master table based on the primary key of affected rows. A primary key's values uniquely identify the rows in a table. Changes are propagated according to row changes as identified by the primary key value of the row. Only one primary key can be defined for each table.The primary key of the master table is the basis for this refresh option, which is the default option. ROWID - The log records changes to the master table based on the RowID of the affected rows. A ROWID is a globally unique identifier for a row in a database based on the physical row identifiers. A ROWID is created at the time the row is inserted into a table, and destroyed when it is removed from a table. ROWID materialized views cannot contain distinct or aggregate functions or GROUP BY subqueries, joins and set operations.
Optional: Select any filter column(s) to be recorded in the materialized view log.	A filter column is a column whose values you want to be recorded in the materialized view log for any rows that are changed. You can specify only one primary key, one ROWID, and one filter column list per materialized view log.
How many transaction entries are allowed for each data block in the materialized view log?	A transaction is a logical unit of work that contains one or more SQL statements. Each transaction that updates a data block requires a transaction entry. Initial (1-255) - Ensures that a minimum number of concurrent transactions can update a data block, avoiding the overhead of allocating a transaction entry dynamically. Maximum (1-255) - Limits concurrency on a data block.
What is the percent of space reserved for future updates?	Percent Free (0 99) - This sets the percentage of a data block to be reserved for possible row updates that are included in the block. The value you set is the percent kept free.
What is the minimum percentage of used space that Oracle maintains for each data block?	Percent Used (0-99) - Set the amount of space to be used for each datablock. NOTE: The sum of percent free and the percent used cannot exceed 100.
How large are the materialized view log's extents?	Initial Extent (KB) -The default is the value specified for the tablespace where the materialized view log resides.
Should the data for a materialized view log be cached?	Yes/No - Select Yes if you want Oracle to put data you access frequently at the most recently used end of the least recently used list in the buffer cache when a full table scan is performed. This option is useful for small lookup tables. No indicates that your most frequently accessed data blocks are put at the least recently used end of the least recently used list of the buffer cache.

Do you want updates to be logged?	Yes/No
Do you want to enable parallel query for the log?	Degree. The integer is number of parallel threads used in the parallel operation. The Parallel server query option lets you process queries using many query server processes running against multiple CPUs. This option provides substantial performance gains such as reduction of the query completion time.
Should the log hold new values?	Yes/No - Yes indicates both old and new values should be saved in the materialized view log. No disables recording of new values in the log. This is the default.

Outlines Wizard (Oracle)

Outlines are a set of results for the execution plan generation of a particular SQL statement. When you create an outline, plan stability examines the optimization results using the same data used to generate the execution plan. That is, Oracle uses the input to the execution plan to generate an outline, and not the execution plan itself.

Note: To create an outline, you must have CREATE ANY OUTLINE system privileges.

To create a new outline using a wizard:

1 Open a creation wizard for an outline. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Outlines (Oracle) - Properties.
- **Definition** panel - Outlines (Oracle) - Definition.
- **DDL** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Outlines (Oracle) - Properties

When creating or editing an outline, this tab/panel lets you work with the following settings:

Setting	Description
Stored Outline Name	Provide a unique name for the outline.
Category	The category is a name you want to use to group stored outlines. (You can type over the word Default that appears automatically.)

Outlines (Oracle) - Definition

Type the SQL statement you want to store as an outline.

Note: The only SQL statements possible with stored outlines are SELECT, DELETE, UPDATE, INSERT…SELECT, and CREATE TABLE…AS SELECT.

Packages Wizard (Oracle)

A package is a collection of related program objects stored together in the database. A package specification or package header, declares variables, constants, etc., that are visible outside the package's immediate scope. The package body defines objects declared by the specification but are not visible to applications outside the package. Packages contain all the information needed to process SQL statements from a single source file. You can use packages to process and call batches of SQL.

To Open the Package Wizard

1 On the Navigator/Explorer, find the datasource where you want to create a package and expand the Schema node.

2 Right-click the **Packages** node, and select **New**.

You're asked to name an owner for the package and give it a name. When you click **Finish**, the Packages editor opens to the header tab where you indicate any package specifications and create the package body. For more information, see Package Bodies Editor (Oracle).

Primary Keys Wizard (Oracle)

Primary key (constraint)s are a set of table columns that can uniquely identify every row of a table. No fields that are a part of the primary key can have null values, and each table can have only one primary key.

To create a new primary key using a wizard:

1 Open a creation wizard for a primary key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Primary Keys (Oracle) - Properties.
- **Columns** panel - Primary Keys (Oracle) - Columns.
- **Storage** panel - Primary Keys (Oracle) - Storage.
- **Partition** panel - Primary Keys (Oracle) - Partition.
- **DDL** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Primary Keys (Oracle) - Properties

When creating or editing a primary key, this tab/panel lets you work with the following settings:

Setting	Description
Table Owner and **Table Name**	Choose the owner and name of the table in which the primary key is being created.
Name	Provide a name for the primary key being created.
No Sort	Enable this feature if the rows in the table already stored in ascending order. This increases the speed of the index creation process. Oracle does not sort the rows.
Logging	Enabling logs this operation to the redo file.
Reverse	Enabling this feature stores the bytes of the index block in reverse order and excludes the ROWID. The ROWID is a globally unique identifier for a row in a database. It is created at the time the row is inserted into a table, and destroyed when it is removed from a table.
Validate	Enabling this option indicates that existing data is checked against the constraint when the primary key is enabled. Leaving it disabled indicates that only new data is to be cheeked against the constraint.
Deferrable	Dictates whether constraint checking can be deferred until the end of a transaction.
Deferred	This option is enabled only if you enabled the **Deferrable** option. Select IMMEDIATE to have the constraint checked at the end of every DDL statement. Select DEFERRED to have the constraint checked only at the end of a transaction.
Enabled	Enables or disables the primary key.

Primary Keys (Oracle) - Columns

From the **Column** dropdown, select a column for the index and specify a **Sort** option. To add more columns, click the **New** button and then follow the steps in the last instruction. Use the Delete button to drop columns.

Primary Keys (Oracle) - Storage

When creating or editing a primary key, this tab/panel lets you work with the following settings:

Setting	Description
Data Block Storage group	Select the DEFAULT **Tablespace** only if you are creating a local partitioned index and want the partitions in the same tablespace as the partitions in the underlying table. (Each partition of a local index is associated with one partition of the table. Oracle can then keep the index partitions in synch with table partitions.) A transaction entry is needed for each INSERT, UPDATE, DELETE, etc. statement that accesses one or more rows in the block. Transaction entries in many operating systems require approximately 23 bytes. **Percent Free** identifies how much space you want to allocate for new rows or updates to existing rows. **Initial Transactions** ensures that a minimum number of concurrent transactions can update an index block, avoiding the overhead of allocating a transaction entry dynamically. **Maximum Transactions** limits concurrency on an index block.
Extents group	An extent is the unit of space allocated to an object whenever the object needs more space. **Initial Extent** - The initial space extent (in bytes) allocated to the object. **Next Extent** - The next extent (in bytes) that the object will attempt to allocate when more space for the object is required. **Percentage Increase** - Lets you type the percentage. NOTE: You should be careful when setting Percent Increase because it magnifies how an object grows and, therefore, can materially affect available free space in a tablespace. **Minimum Extents** - For a dictionary managed tablespace, this is the total number of extents to be allocated when the index is first created. For a locally managed tablespace, this is simply the initial amount of space allocated. **Maximum Extents** - For a dictionary managed tablespace, this is the total number of extents that can ever be allocated to the index. In a locally managed tablespace, the database will automatically manage the extents.
Freelists group	**Free lists** let you manage the allocation of data blocks when concurrent processes are issued against the index. You can potentially improve the performance of the index by identifying multiple free lists, which can reduce contention for free lists when concurrent inserts take place. The default and minimum value is 1. You should increase this number if multiple processes access the same data block. **Free List Groups** is the number of groups of free lists. NOTE: This option is only applicable for the parallel server option.
Buffer Pool	DEFAULT - Choose this if you want to use the default bufferpool. KEEP - Use this to retain the object in memory to avoid I/O conflicts. This type of bufferpool stores frequently referenced data blocks in a separate cache. RECYCLE - Select this option to save cache space by ridding data blocks from memory as soon as they are no longer in use.

Primary Keys (Oracle) - Partition

Clicking **Create Partition** opens a wizard that lets you create a partition.

Procedures Wizard (Oracle)

Procedures are a reusable block of PL/SQL, stored in the database, that applications can call. Procedures streamline code development, debugging, and maintenance by being reusable. Procedures enhance database security by letting you write procedures granting users execution privileges to tables rather than letting them access tables directly. Procedures do not return values, unlike functions.

Note: To create a procedure in your own schema, you need CREATE PROCEDURE privileges. To create a procedure in someone else's schema, you need CREATE ANY PROCEDURE privileges.

To create a new procedure using a wizard:

1 Open a creation wizard for a procedure. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Procedures (Oracle) - Properties.
- **Definition** panel - for details, see Procedures (Oracle) - Definition.

3 Finally, use the **Execute** button to create the object.

Procedures (Oracle) - Properties

When creating or editing a procedure, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the owner of the procedure.
Name	Provide a name for the procedure

Procedures (Oracle) - Definition

Complete the CREATE PROCEDURE outline provided by typing or pasting the body of the procedure.

Profiles Wizard (Oracle)

Profiles are a mechanism for allocating system and database resources to users. In essence, a profile is a set of limits with a name attached to them. If the profile is active, Oracle will limit use and resources to the limits defined by the profile.

The Profile Wizard lets you:

- Name the profile.
- Set composite limit.
- Set session limits for SGA shared pool.
- Set limits on total connection time per session and Idle time per session.
- Set limits on concurrent sessions per user, CPU time per session, and data blocks read per session.
- Set limits on CPU time per call, and number of data blocks read for a call to process an SQL statement.
- Set the number of failed login attempts, and the days an account locks.

Note: To create a profile, you need the CREATE PROFILE system privilege.

The Default option is subject to the limit specified for that particular resource. The default profile initially permits unlimited resources. Limits to the default profile can be made using an Alter statement.

The Unlimited option allows the user with that profile to take use unlimited amounts of that resource.

The Other options take on the resource limits that you indicate.

To Open the Profile Wizard

1 On the Navigator/Explorer, find the datasource where you want to create a procedure and expand the Security node.

2 Right-click the **Profile** node, and select **New**.

The table that follows describes the fields you may encounter as you complete the wizard.

Required Information	Description
What is the name of the profile?	Write a name that's 30 characters or less
What is the composite limit on resources per session?	You can set a single composite limit for all resource limits in a profile in addition to setting specific resource limits for a profile. Explicit and composite limits can peaceably coexist. The limit that is reached first stops the session's activity. Service units are a weighted sum of CPU per session, connect time, logical reads per session, and private SGA. Default/Unlimited/Other (service units)
What is the limit on the amount of private space a session can allocate in the shared pool of the SGA?	This limit only applies if you are using Shared Server architecture. Default/Unlimited/Other (KB)
What is the limit on the total connection time per session?	The total elapsed time limit for a session Default/Unlimited/Other (Minutes)
What is the limit on idle time per session?	Permitted periods of continuous inactive time during a session, expressed in minutes. Long-running queries and other operations are not subject to this limit. Default/Unlimited/Other (Minutes)
What is the limit on concurrent sessions per user?	Default/Unlimited/Other
What is the limit on CPU time per session?	Default/Unlimited/Other (Hundredths of a Second)
What is the limit on data blocks read per session?	Default/Unlimited/Other
What is the limit on CPU time per call?	The limit on a parse, execute, or fetch. Default/Unlimited/Other (Hundredths of a Second)
What is the limit on the number of data blocks read for a call to process a SQL statement?	This is the number of logical reads per call. Default/Unlimited/Other
How many failed login attempts will be allowed before an account is locked?	Default/Unlimited/Other
How long will an account be locked after the specified number of failed login attempts?	Default/Unlimited/Other (Days)
What is the lifetime of the password?	Default/Unlimited/Other (Days)

How many days must pass before a password can be reused?	Default/Unlimited/Other (Days)
How many password changes are required before the current password can be reused?	Default/Unlimited/Other
What is the grace period allowed for a password to be changed without expiring?	Default/Unlimited/Other (Days)
What is the name of the password complexity verification routine?	Default Null - Specifies no password verification is performed. Function Name

Programs wizard (Oracle)

This wizard builds and submits a call to DBMS_SCHEDULER.CREATE_PROGRAM with optional DBMS_SCHEDULER.DEFINE_PROGRAM_ARGUMENT calls, letting you create a program.

Note: This functionality is available as of Oracle 10g.

Note: Before working with jobs, consult Oracle documentation for general information on DBMS_SCHEDULER. For more information, see Accessing Third Party Documentation.

To create a program using a wizard

1 In DBArtisan, on a connected Oracle datasource, expand the **Instance** and **Scheduler** nodes, right-click the **Programs** node and select **Create** from the context menu.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Action** panel - for details, see Programs (Oracle) - Action.
- **Arguments** panel - for details, see Programs (Oracle) - Arguments.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Finish** button to create the object.

Programs (Oracle) - Action

When creating or editing a job, this tab/panel lets you work with the following settings:

Group	Settings and Description
Creation	Lets you provide an **Owner** and **Name** for the job as well as ENABLED and COMMENTS (**Description**) attribute values
Type of Program	The **Type of Program** setting lets you choose among EXECUTABLE, STORED PROCEDURE, or PLSQL Block. Depending on your choice, take one of the following actions: PLSQL BLOCK - Type or paste the SQL to be executed. STORED PROCEDURE - Provide the **Procedure Owner** and **Procedure Name**. EXECUTEABLE - Use the browse button to locate the executable.

Programs (Oracle) - Arguments

When creating or editing a program, if you selected an Action tab choice of EXECUTABLE or STORED PROCEDURE, this tab/panel lets you manage parameters to be passed.

Redo Log Groups Wizard (DBArtisan - Oracle)

A redo log makes it possible to replay SQL statements because a redo log stores all changes made to a database as they occur. If something unanticipated happens to a datafile, you can restore a backed-up datafile and use the redo log to bring the datafile back to the state it was in before the unanticipated activity. The redo log can consist of two parts: the online redo log and the archived redo log. In archive log mode, redo log files will not be overwritten unless they have been archived somewhere. Each log is a member of the group.

Note: Redo should not be confused with Undo, which is a transaction based process.

If you find that you have problems with redo log group availability and Oracle consistently has to wait for redo log groups to become free, you can create additional redo log groups. DBArtisan lets you create additional redo log groups to ensure that there is always a recycled redo log group available.

The Redo Log Group Wizard lets you:

- Assign a number to the new redo log group and determine the file size for the group members.
- Choose a thread for the redo log group for Oracle Parallel Server.
- Add redo log members with the Add Redo Log Member dialog.

To Open the Redo Log Wizard

1 On the Navigator/Explorer, find the datasource where you want to create a redo log and expand the **Storage** node.

2 Right-click the **Redo Log Groups** node, and select **New**.

The table that follows describes the fields you may encounter as you complete the wizard.

Required Information	Description
Do you want to use Oracle Managed Files (OMF) for the redo log group? You must be set up to use OMF to use this option	Lets you specify to use Oracle Managed Files to automatically create the redo log group. Oracle automatically creates and removes operating system files as you create and drop storage objects. **Note:** If you select OMF, you do not have to enter a redo log group number or file size.
What is the group number for the redo log group?	When numbering your redo log groups, it is better not to skip numbers (e.g., 5, 10, 15). Skipping numbers causes Oracle to consume extra space in the control files of the database.
What file size should the group's redo log members be? Note: All members of the group must be the same. With OMF, you can let Oracle assign a size or enter it manually.	size in KB or MB
To which thread should the redo log group be added? For this option to be available, you need to be using Oracle with the parallel server option in parallel mode.	OPTIONAL: If you are using Oracle Parallel Server, type the thread. A thread is an operating system component that allows the logic of multiuser applications to be performed as several separate, asynchronous execution paths. Threads help ensure that work is being performed for some user connections even when other connections are blocked (for example, when waiting for a disk read or write operation to complete).
Add redo log group members	The name of each group member should be a file name. For example, D:\ORACLE\ORADATA\FINANCEDB\REDO02.LOG is a valid group member name. If you do not specify a full path, DBArtisan creates files in either the default or current directory of the database server.

Roles Wizard (Oracle)

The Role Wizard constructs the necessary CREATE ROLE statement from the information that you have supplied. The Role Wizard lets you specify a name for the role and whether or not the role should be identified.

Note: To create a role, you need the CREATE ROLE system privilege.

To Open the Role Wizard

1 On the Navigator/Explorer, find the datasource where you want to create a role and expand the Security node.

2 Right-click the **Roles** node, and select **New**.

The table that follows describes the fields you may encounter as you complete the wizard.

Required Information	Description
What is the name of the role?	Self-explanatory.
How should the role be identified?	Not identified - Selecting this means the role you are creating will be enabled immediately. Identified: If this is your choice, you're indicating one of the following authorization methods will be followed: Globally - Select to indicate that Oracle permits access to the user by obtaining user name and password information from the security domain central authority. Externally - Select to indicate that Oracle should verify the database user name against an existing operating system user name. Password - Select to indicate that Oracle should identify the role with the password you provide. In the Password box, type the password for the user.

For information on activating and deactivating roles for the current login in the current session, see Role Activation.

Rollback Segments Wizard (Oracle)

Rollback segments manage all transactions in your Oracle databases. A transaction is a read, modify, and write cycle for an Oracle database. A rollback entry is made for all transactions unless a particular clause is specified. So, a rollback segment is a transaction log that consists of a pre-update image value and transaction status, among other things. The rollback segments maintain read consistency among concurrent users in a database and if the transaction fails for any reason, the old image is taken from the rollback segment. By maintaining a history of data changes, rollback segments can rollback uncommitted transactions so that data is rolled back to the prior state. SYS owns all rollback segments no matter who created them and are not accessible to users, just Oracle.

Oracle, it should be mentioned, strongly recommends that you use automatic undo management to simplify managing databases. Tuning rollback segments is a manual process that has largely been deprecated by Oracle, but it is supported for backward compatibility reasons.

The Rollback Segment Wizard lets you:

- Name the rollback segment and to place it online or off-line.
- Place the rollback segment on a tablespace.
- Specify the initial next and optimal extent size as well a the minimum and maximum number of extents that should be allocated to the rollback segment.

Note: This wizard is not available if auto-UNDO management is enabled.

Tip: Make sure enough rollback segments exist on a database to handle the imposed workload. One rule of thumb is to create one rollback segment for every four concurrent users.

To Open the Rollback Segment Wizard

1 On the Navigator/Explorer, find the datasource where you want to create a rollback segment and expand the Storage node.

2 Right-click the **Rollback Segments** node, and select **New**.

The table that follows describes the fields you may encounter as you complete the wizard.

Required Information	Description
What is the name of the rollback segment?	Self-explanatory.
Should this rollback segment be made public?	Yes - A public rollback segment can be brought online by any instance in a parallel server. Public rollback segments form a pool of rollback segments that can be used by any instance that needs one. No - This is the default. A private rollback segment can only be acquired by the instance specifying the segment in its initialization file.
Do you want to place the rollback segment to be online following its creation?	Online - To be useful, a rollback segment must be online. Offline - You may want to take rollback segments offline if you want to take a tablespace offline and it contains rollback segments that you want to keep from being used.
On which tablespace do you want to place this rollback segment?	Self-explanatory. Oracle suggests that you create one or more tablespaces specifically to hold all rollback segments. This way, the data contained in the rollback segments is held apart from other data types.
What extent sizes do you want to assign to this rollback segment?	Initial size (KB) Next size (KB) Optimal size (KB) Null/Default
What are the minimum and maximum number of extents to allocate to the rollback segment?	Minimum Maximum

Schedules Wizard (Oracle)

This wizard builds and submits a call to DBMS_SCHEDULER.CREATE_SCHEDULE call, letting you create a schedule.

Note: This functionality is available as of Oracle 10g.

Note: Before working with jobs, consult Oracle documentation for general information on DBMS_SCHEDULER. For more information, see Accessing Third Party Documentation.

To create a schedule using a wizard

1 In DBArtisan, on a connected Oracle datasource, expand the **Instance** and **Scheduler** nodes, right-click the **Schedules** node and select **Create** from the context menu.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Schedules (Oracle) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Finish** button to create the object.

Schedules (Oracle) - Properties

When creating or editing a job, this tab/panel lets you work with the following settings:

Group	Settings and Description
Creation	Lets you provide an **Owner** and **Name** for the schedule.
Attributes	**Description** - corresponds to a COMMENT attribute. **Repeat Interval** - corresponds to a REPEAT_INTERVAL attribute. Calendaring syntax, as described in Oracle DBMS_SCHEDULER documentation, can be typed free-form, and will be passed directly to the server. For information on third-party documentation access, see Accessing Third Party Documentation. **Start Date** and **End Date** - correspond to START_DATE and END_DATE, attributes.

Sequences Wizard (Oracle)

Sequences are programmable database objects that provide numbers in sequence for input to a table. A sequence can be used to automatically generate primary key values for tables.Once defined, a sequence can be made available to many users. When you create a sequence, you can define its initial value, increment interval, and maximum value.

The Sequence Wizard lets you:

- Specify the name and owner of the sequence.
- Set both the value of the sequence, and an interval and ranges for incrementing it.
- Cache the sequence, cycle the sequence when it reaches its minimum or maximum values, and guarantee that Oracle generates sequence numbers in the order of request.

Note: To create a sequence, it must belong to your schema or you need CREATE SEQUENCE privilege.

To Open the Sequence Wizard

1 On the Navigator/Explorer, find the datasource where you want to create a rollback segment and expand the Schema node.

2 Right-click the **Sequences** node, and select **New**.

The table that follows describes the fields you may encounter as you complete the wizard.

Required Information	Description
Who owns the sequence?	Self-explanatory.
What is the sequence name?	Self-explanatory.
What is the first sequence number to be generated?	Start with: Pick an integer.
What is the interval between sequence numbers?	Increment by: Positive numbers will generate ascending values, and a negative number will generate descending numbers
What is the sequence's minimum value?	The default is 1 for ascending sequences; This integer value can have 28 or fewer digits. None Minimum value - Identify how low the sequence can go.
What is the sequence's maximum value?	For descending values, the default is 1. Lets you specify the maximum value the sequence can generate. This integer value can have 28 or fewer digits. None Maximum value - Indicate the highest sequence value that will be allowed.
Should Oracle preallocate sequence numbers and cache them for faster access?	This is the number of sequence values you want to specify in the SGA buffers. This speeds access, but cached numbers are erased when the database is shut down. The default value is 20. Yes Number of values No
Should the sequence continue to generate values after reaching either its maximum or minimum value?	If you say no, the sequences will automatically recycle to the minimum value when you've hit the maximum for ascending sequences and vice versa for descending sequence values. Yes No
Should the sequence numbers be generated in the order of request?	This may be required when sequences are required for timestamping. But generally, because sequences are naturally ordered, this is only necessary for if you use Oracle RAC clusters for parallel mode. Yes No

Snapshots and Snapshot Logs Wizards (Oracle)

Oracle has replaced the snapshot functionality with materialized views. Refer to the Materialized Views Wizard (Oracle) and Materialized View Logs Wizard (Oracle).

Synonyms Wizard (Oracle)

Synonyms are alternate names for database objects to be used as a reference by users or applications. A synonym offers you security and convenience so that you can refer to an object without revealing who owns it or what database it belongs to. Synonyms Depending on the platform, you can define synonyms on tables, views, sequences, procedures, functions, packages, and materialized views. If an underlying object needs to be renamed or moved, it's easy enough to redefine the synonym without having to modify any applications based on the synonym.

Note: To create a private synonym, you need CREATE SYNONYM privileges. To create a public synonym, you need CREATE PUBLIC SYNONYM privileges.

To create a new synonym using a wizard:

1 Open a creation wizard for a synonym. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Synonyms (Oracle) - Properties.
- **Definition** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Synonyms (Oracle) - Properties

When creating or editing a synonym, this tab/panel lets you work with the following settings:

Setting	Description
Owner and **Name**	Choose the owner and provide a name for the synonym being created.
Referenced Object Owner	Select the owner of the object to which the synonym is to refer.
Referenced Object Type	Select the type of the object to which the synonym is to refer.
Referenced Object Name	Select the specific object to which the synonym is to refer.
Database Link	If the object resides on a remote database, select a database link.

Tables Wizard (Oracle)

Tables are the most basic data storage units for Oracle. As you might expect, data is stored in rows and columns. The Table Wizard constructs the necessary CREATE TABLE statement from the information that you supply. The Table Wizard varies slightly in content based on the version of Oracle on the target datasource. But in all cases, you name columns and determine column width or precision and scale depending on the column's data type. A row collects the column information that corresponds to a single record.

You can set rules for the columns to live by, and these are called integrity constraints. For example, if you select NOT NULL, that column will have to have a value in each row.

Also, before beginning, consider what kind of table you want to create as the wizard will ask you to choose:

Heap organized table	This is a basic table where data is stored as an unordered collection, i.e., heap.
Index-organized table	A B-tree index structure stores data, sorted by primary key. Nonkey column values are stored too.
Partitioned table	Data is broken down into smaller, more manageable pieces called partitions or subpartitions. Each partition can be managed individually and operate independent of the other partitions.
Clustered table	This is a table that, once created, is part of a cluster. A cluster is a group of tables that share some data blocks and columns and are often used together. To create a cluster, use the Clusters Wizard (Oracle).

Note: The table wizard panels differ depending on what options you select.

Note: To simplify the process of creating a table, the Table Wizard focuses on creating the basic table definition with a primary key constraint. After you create the basic table definition you can add unique and foreign keys to the table on the **Constraints** tab of the Tables Editor.

To create a new table using a wizard:

1 Open a creation wizard for a table. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Tables (Oracle) - Properties.
- **Columns** panel - for details, see Tables (Oracle) - Columns.
- **Indexes** panel (not available with a **Row Organization** of **EXTERNAL**) - for details, see Tables (Oracle) - Columns.
- **Constraints** panel (not available with a **Row Organization** of EXTERNAL) - for details, see Tables (Oracle) - Constraints.

- **Storage** panel (not available with a **Row Organization** of EXTERNAL) - for details, see Tables (Oracle) - Storage.
- **IOT Properties** panel (not available with a **Row Organization** of EXTERNAL) - for details, see Tables (Oracle) - IOT Properties.
- **Partition** (not available with a **Row Organization** of **EXTERNAL**) - for details, see Tables (Oracle) - Partition.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **Permissions** panel - for details, see Setting Permissions or Privileges for an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Tables (Oracle) - Properties

When creating or editing a table, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the owner of the table.
Name	Provide a name for the table
Cache	Enabling this feature keeps a block in memory by placing it at the most recently used end. This option is useful for small lookup tables.
Row Movement	Enabling this option permits the migration of a row to a new partition if its key is updated.
Parallel Degree	A value indicating the number of query server processes that should be used in the operation.
Parallel Instances	A value indicating how you want the parallel query partitioned between the Parallel Servers.
Physical group	Choose a Row Organization of INDEX, HEAP, or **EXTERNAL**. If you chose INDEX or HEAP, enable or disable **Logging**. If you chose EXTERNAL, provide an **External Type**, **Default Directory**, **Access Parameters**, **Reject Limit**, and **Location**.
Logging	Redo logs minimize the loss of data in the event that an uncontrolled shutdown happens.

Tables (Oracle) - Columns

When creating or editing a table, this tab/panel lets you manage the columns for the table.

To add a column to the table

1 Click **Add Column**, provide a **Name** he column in the **Property/Value** list, and press TAB or ENTER.

The column is added to the columns list on the left, with default attribute values.

2 Proceed to edit the column attributes.

To edit column attributes

3 Select the column in the columns list on the left. The controls in the Property/Values list are updated with values of the selected column.

4 Use the following table as a guide to providing additional property values, noting that availability of a property differs by data type and other property selections.

Property or group	Description
Virtual	Selecting this check box defines the column as an Oracle virtual column, with a value calculated from a column expression. Virtual columns do not use any disk space as there is o data to store, and INSERT/UPDATE operations are not supported. Consult Oracle documentation before working with virtual columns to familiarize yourself with topics such as column expressions and virtual column-specific details such as automatic type conversion. For more information, see Accessing Third Party Documentation. After specifying a column as virtual, use the **Default Value** box to provide the calculation.
Datatype	This group lets you select a valid **Type** for the column. Depending on your selection, additional properties such as **Scale**, **Size**, **Width**, and **Unused** may be available.
Allow Nulls	Select this check box to allow nulls in this column.
Encryption	Use controls in this group to have the column encrypted using transparent data encryption. Select the **Encrypted** check box to enable transparent data encryption for the column. Type a **Password** value used to build an IDENTIFIED BY clause. The column key will be derived from the value you provide. Select the **Salted** check box to add a SALT option, appending a random 'salt' string, to the clear text of the column before encrypting. Leaving the check box unselected adds a NO SALT option. From the **Encryption Algorithm** dropdown, select an algorithm (AES256, NONE, DES168, AES192, AES128, or DES156) that will be used to build a USING clause.
Default Value	If you selected the **Virtual** check box, type a valid Oracle column expression that will calculate the value of the column. Otherwise, either select a pseudocolumn (CURRENT_TIMESTAMP, USER, SYSDATE, or UID) from the dropdown or type an expression resulting in a value that matches the selected **Datatype**.
Comment	Lets you add a comment to the column.
LOB Storage	This group is available for bfile, blob, clob, and nclob types. Settings include **Segment Name**, **Configuration** properties (**Tablespace**, **Chunk**, **Percent Version**, **Enable Storage In Row**, **Cache**, and **Logging**), and **Storage** properties (**Initial Extent**, **Next Extent**, **Percent Increase**, **Minimum Extents**, **Maximum Extents**, **Free Lists**, and **Free List Groups**).

To delete a column

1 Select a column from the column list on the left.

2 Click **Delete** to remove the column from the table.

To change a column's ordering position

1 Select a column from the column list on the left.

2 Use the arrow buttons to move the column up or down.

Tables (Oracle) - Indexes

Note: This tab/panel is not available with a **Row Organization** of **EXTERNAL**.

Click **Add** to open the Index Wizard. For details, see Indexes Wizard (Oracle).

Tables (Oracle) - Constraints

Note: This tab/panel is not available with a **Row Organization** of **EXTERNAL**.

Selecting a constraint type and clicking **Add** opens the object wizard for that object type. For details see:

- Primary Keys Wizard (DB2 LUW)
- Unique Keys Wizard (DB2 LUW)
- Foreign Keys Wizard (DB2 LUW)

Tables (Oracle) - Storage

Note: This tab/panel is not available with a **Row Organization** of **EXTERNAL**.

When creating or editing a table, this tab/panel has the following settings:

Settings	Description
Data Block Storage group	Select the DEFAULT **Tablespace** only if you want the partitions in the same tablespace as the partitions in the underlying table. **Percent Free** identifies how much space you want to allocate for new rows or updates to existing rows. **Initial Transactions** ensures that a minimum number of concurrent transactions can update a primary key block, avoiding the overhead of allocating a transaction entry dynamically. **Maximum Transactions** limits concurrency on a primary key block.
Extents group	An extent is the unit of space allocated to an object whenever the object needs more space. **Initial Extent** - The initial space extent (in bytes) allocated to the object. **Next Extent** - The next extent (in bytes) that the object will attempt to allocate when more space for the object is required. **Percentage Increase** - Lets you type the percentage. NOTE: You should be careful when setting Percent Increase because it magnifies how an object grows and, therefore, can materially affect available free space in a tablespace. **Minimum Extents** - For a dictionary managed tablespace, this is the total number of extents to be allocated when the index is first created. For a locally managed tablespace, this is simply the initial amount of space allocated. **Maximum Extents** - For a dictionary managed tablespace, this is the total number of extents that can ever be allocated to the index. In a locally managed tablespace, the database will automatically manage the extents.
Freelists group	**Free lists** let you manage the allocation of data blocks when concurrent processes are issued against the primary key. You can potentially improve the performance of the primary key by identifying multiple free lists, which can reduce contention for free lists when concurrent inserts take place. The default and minimum value is 1. You should increase this number if multiple processes access the same data block. **Free List Groups** is the number of groups of free lists. NOTE: This option is only applicable for the parallel server option.
Buffer Pool	DEFAULT - Choose this if you want to use the default bufferpool. KEEP - Use this to retain the object in memory to avoid I/O conflicts. This type of bufferpool stores frequently referenced data blocks in a separate cache. RECYCLE - Select this option to save cache space by ridding data blocks from memory as soon as they are no longer in use.

Tables (Oracle) - IOT Properties

Note: This tab/panel is not available with a **Row Organization** of **EXTERNAL**.

Provide compression and space details for an index-organized table.

Tables (Oracle) - Partition

Note: This tab/panel is not available with a **Row Organization** of **EXTERNAL**.

Prior to working with partitions, you should be familiar with the material in Oracle Partitioning.

Click **Create Partition** to Partition a table.

Oracle Partitioning

Partitioning your tables lets you get around the problem of supporting large tables. Partitioning lets you break large tables into smaller pieces, which are called partitions. Partitions make the data in your table easier to manage and analyze. Your SQL statements can access the partitions rather than the entire table. Partitions are most useful in data warehouse applications, which store large amounts of data.

The table below describes the types of partitions in Oracle:

Partition Type	Description
Range	Use range partitioning to map rows to partitions based on ranges of column values. This type of partitioning is useful when dealing with data that has logical ranges into which it can be distributed; for example, months of the year. Performance is best when the data evenly distributes across the range. If partitioning by range causes partitions to vary dramatically in size because of unequal distribution, you may want to consider one of the other methods of partitioning.
Hash	Use hash partitioning if your data does not easily lend itself to range partitioning, but you would like to partition for performance and manageability reasons. Hash partitioning provides a method of evenly distributing data across a specified number of partitions. Rows are mapped into partitions based on a hash value of the partitioning key. Creating and using hash partitions gives you a highly tunable method of data placement, because you can influence availability and performance by spreading these evenly sized partitions across I/O devices (striping).
Composite	Hash partitions partition the table according to a hash function. Composite partitions use both range and hash types, first partitioning the data by a range of values, and then further dividing the partitions into subpartitions by way of a hash function. This option is not available for index-organized tables.
List	Use list partitioning when you require explicit control over how rows map to partitions. You can specify a list of discrete values for the partitioning column in the description for each partition. This is different from range partitioning, where a range of values is associated with a partition, and from hash partitioning, where the user has no control of the row to partition mapping.

Partition a table

The Add partition wizard lets you set up partitions for a table. Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

Step	Settings and tasks
Properties	Select a **Partition Type** and optionally, a Subpartition Type.
Columns	For each column, click the **New** button and select a name from the **Column** dropdown. Use the **Delete** button to drop a selected column.
Subpartition Columns (only available with a **Partition Type** of **RANGE** and **Subpartition Type** of **HASH** or **LIST**)	For each column, click the **New** button and select a name from the **Column** dropdown. Use the **Delete** button to drop a selected column.
Subpartitions (only available with a **Subpartition Type** of **HASH**)	Specify a **Default number of partitions**. For each partition, click the **New** button and then select a tablespace from the dropdown.
Range Definitions (only available with a **Partition Type** of **RANGE**)	Click the **New** button to Add a partition definition.
Partition Definition (only available with a **Partition Type** of **HASH**)	To specify a partition method other than **None**, take one of the following actions: (1) Select the **Number Of Partitions** radio box, specify the **Number Of Partitions**, and for each partition, click the **New** button and choose a tablespace from the dropdown, or (2) select the **By Partition Name** radio box and for each partition, click the **New** button provide a name and then choose a tablespace from the dropdown.
List Definitions (only available with a **Partition Type** of **List**)	Click **New** to Add a partition definition.

Add a partition definition

Use the following topics as a guide in completing the settings in this wizard:

Step		Settings and tasks
Partition Definition	**Name**	Provide a name.
	Tablespace	Select a tablespace from the dropdown
	Logging	Enable or disable logging.
Subpartitions	To specify a partition method other than **None**, select the **By Subpartition Name** radio box and for each partition, click the **New** button to open a dialog that lets you provide subpartition values. **NOTE:** When you split a range-list partition, you cannot specify the new partitions' subpartition information.	
Storage	Provide or select Data Block Storage, Extents, Freelists, and Buffer Pool values.	

Tablespaces Wizard (Oracle)

Tablespaces are logical storage structures that act as partitions for the database. Each tablespace consists of one or more datafiles which are the physical structures holding the data. You can create a tablespace to store table data and other objects related to table performance such as indexes or large object data. Tablespaces are used to manage large complex databases. Once you have created a tablespace, you can place objects on it.

The Tablespace Wizard lets you:

- Name the tablespace, and specify space management.
- Specify what types of objects are stored on the tablespace, and place the tablespace online or offline.
- Add the datafiles that comprise the tablespace and specify the parameters for the datafiles.
- Specify how Oracle should manage the growth of the tablespace.

Important Notes

- For auto-UNDO management to be in effect, set init.ora parameter to undo_management. When set to MANUAL (the default), it disables auto-UNDO management. When to set AUTO, auto-UNDO management is enabled.
- To determine if the undo_management parameter is set to AUTO, use the following query:

SELECT VALUE

FROM SYS.V_$PARAMETER

WHERE NAME = 'undo_management'

Note: This parameter cannot be set dynamically via the ALTER SYSTEM or ALTER SESSION.

One of the best ways to avoid fragmentation in a tablespace is to pre-allocate the space that your objects will use. If possible, plan for one to two years' growth for each object and allocate your space accordingly. Having initial empty objects will not affect table scan times as Oracle only scans up to the high-water mark (the last used block) in a table.

Of all your tablespaces, you want to avoid fragmentation problems in your SYSTEM tablespace the most as this is the major hotbed tablespace for Oracle activities. The easiest way to avoid this is to not allow any user (even the default DBA ID's SYS and SYSTEM) to have access to it. There are three ways to do this:

- Ensure no user has a DEFAULT or TEMPORARY tablespace assignment of SYSTEM.
- Ensure no user has a quota set for SYSTEM.
- Ensure no user has been granted the UNLIMITED TABLESPACE privilege.

To create a new tablespace using a wizard:

1 Open a creation wizard for a tablespace. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - see Tablespaces (Oracle) - Properties.
- **Datafiles** panel - see Tablespaces (Oracle) - Datafiles.
- **DDL** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Tablespaces (Oracle) - Properties

When creating or editing a tablespace, this tab/panel lets you work with the following settings:

Setting	Description
Name	Provide a name for the tablespace.
Big File	Selecting this check box results in a CREATE BIGFILE TABLESPACE statement being generated. This results in a large, single file tablespace. Leaving this check box unselected results in a tablespace containing multiple smaller datafiles.
Type	Leaving the default PERMANENT selection creates a permanent tablespace, intended to contain schema objects you want to keep on an ongoing basis. The objects are stored in datafiles. The other two options let you generate a CREATE TABLESPACE statement that specifies TEMPORARY or UNDO keywords. An Undo tablespace is a kind of permanent tablespace that holds undo data if your database is operating in automatic undo mode. Oracle recommends the automatic undo mode as the wiser choice than using rollback segments to undo. In a Temporary tablespace, shema objects will last only as long as your session continues.
Tablespace Group	This option is only available with a **Type** value of TEMPORARY. If there are currently entries in the SYS.DBA_TABLESPACE_GROUPS table, you can either select an existing group or type the name for a new tablespace group. If there are currently no entries in the SYS.DBA_TABLESPACE_GROUPS table, you can type the name of a new tablespace group to be created. When editing a tablespace, clearing the entry from the **Tablespace Group** box deletes the <tablespace_name, group_name> entry from the SYS.DBA_TABLESPACE_GROUPS table.
Retention Guarantee	This option is only available with a **Type** value of UNDO. Selecting this check box specifies that unexpired data should be preserved for all Undo segments of the tablespace, regardless of whether this may force failure of ongoing operations that require space in those segments. Leaving this check box unselected specifies default RETENTION NOGUARANTE behavior. Space currently being used by unexpired undo data in undo segments can be consumed as necessary by ongoing transactions.
Status	This option is only available with a **Type** value of PERMANENT. Use this control to specify an ONLINE/OFFLINE clause in the generated DDL. The ONLINE selection makes the tablespace available immediately after creation. The OFFLINE selection results in a table is unavailable immediately after creation.
Logging	This option is only available with a **Type** value of PERMANENT. Use this control to specify a LOGGING/NOLOGGING clause in the generated DDL. This sets the default logging attributes of all tables, indexes, materialized views, materialized view logs, and partitions within this tablespace.
Force Logging	This option is only available with a **Type** value of PERMANENT. Selecting this check box specifies that a FORCE LOGGING clause be included in the generated DDL. Changes to all objects in the tablespace (temporary segments excepted) are logged, overriding any NOLOGGING setting for individual objects.

Compression Type	This option is only available with a **Type** value of PERMANENT. This lets you select default compression type of ALL OPERATIONS (corresponds to COMPRESS FOR ALL OPERATIONS in the DDL), COMPRESS_DIRECT_LOAD (corresponds to COMPRESS FOR DIRECT_LOAD OPERATIONS in the DDL) or no compression.
Flashback	This option is only available with a **Type** value of PERMANENT. Use this control to specify a FLASHBACK ON/FLASHBACK OFF clause in the generated DDL. Select this check box to put the tablespace in FLASHBACK mode. In FLASHBACK mode, Oracle saves Flashback log data for the tablespace and the tablespace can participate in a FLASHBACK DATABASE operation. If you leave this check box unselected, Oracle does not save Flashback log data for the tablespace. Prior to FLASHBACK DATABASE operations, you must either take datafiles in the tablespace off line or drop them OR take the entire tablespace offline. The database does not drop existing Flashback logs.
Encrypted and **Algorithm**	This option is only available with a **Type** value of PERMANENT. Selecting the **Encryption** check box enables the **Algorithm** control. Together, they let you add an ENCRYPTION USING clause to the generated DDL. The **Algorithm** lets you select from AES 128, AES192, AES256, DES56, DES168, or NONE encryption algorithms.
Locally Managed, **Uniform Allocation**, **Size**, **Automatic Segment Space Management**, and **Minimum Extent Size**	These option are only available with a **Type** value of PERMANENT or UNDO. These controls let you work with the extent-related clauses that will be included in the generated DDL. Selecting the **Locally Managed** check box lets you generate an EXTENT MANAGEMENT LOCAL clause. Subsequently leaving the **Uniform Allocation** check box deselected adds an AUTOALLOCATE option. Alternatively, selecting the **Uniform Allocation** check box enables the **Size** check box, letting you add a UNIFORM SIZE option. Select the **Automatic Segment Space Management** check box to generate a SEGMENT SPACE MANAGEMENT AUTO or leave it deselected to generate a SEGMENT SPACE MANAGEMENT MANUAL clause. Leaving the **Locally Managed** check box deselected generates an EXTENT MANAGEMENT DICTIONARY clause. This also enables the **Minimum Extent Size** control, letting you select a MINIMUM EXTENT clause value.
Use Default Block Size and **Block Size**	Disabling the **Use Default Block Size** check box enables the **Block Size** control, generating a BLOCKSIZE clause to specify a nonstandard block size for the tablespace.

Tablespaces (Oracle) - Datafiles

When creating or editing a tablespace, this tab/panel lets you build DATAFILE or TEMPFILE clauses that will be submitted with the CREATE TABLESPACE statement creating or editing this tablespace. This lets you manage the files that make up the tablespace.

To add a datafile

1 Click **Add**, either select an existing file name from **File Name** dropdown or type the name of a new file, and then press TAB or ENTER.

The file is added to the datafiles list on the left, with default attribute values.

2 Proceed to edit the datafile attributes.

To edit datafile attributes

3 Select the file from the list on the left. The controls in the Property/Values list are updated with values of the selected file.

4 Use the following table as a guide to modifying property values.

Properties	Description
Size	Lets you specify the SIZE option for the DATAFILE/TEMPFILE clause, to specify the size of the file
Reuse Existing File	Lets you specify the REUSE option for the DATAFILE/TEMPFILE clause, to specify that an existing file can be reused.
Auto Extend and **Growth Amount**	Use these options to select an AUTOEXTEND option for the DATAFILE/TEMPFILE clause. Leaving the **Auto Extend** check box unselected disables the **Growth Amount** disabled and specifies an AUTOEXTEND OFF option in the generated DDL. Selecting the **Auto Extend** check box lets you specify an AUTOEXTEND ON NEXT clause. Selecting **Auto Extend** enables the **Growth Amount** control, letting you specify the size of the next increment of disk space automatically allocated when more extents are required.
Unlimited and **Value**	These options are only available with **Auto Extend** selected. They let you specify a MAXSIZE clause. Selecting the **Unlimited** check box lets you specify a MAXSIZE UNLIMITED clause, placing no limit on the size of the file. Leaving the **Unlimited** check box unselected enables the **Value** disabled. This specifies a MAXSIZE clause with a size provide by the **Value** control.

To delete a datafile

1 Select a datafile from the column list on the left.

2 Click **Delete** to remove the datafile from the tablespace.

Triggers Wizard (Oracle)

This wizard lets you build and submit a CREATE OR REPLACE TRIGGER statement, creating a trigger to be associated with a table, view, schema, or database. BEFORE, AFTER, and COMPOUND triggers are supported for a variety of DDL, DML, and Database manipulation events.

Important Notes

- To create triggers in your own schema, you need CREATE TRIGGER privileges. To create triggers in other schemas, you need CREATE ANY TRIGGER privileges.

To create a new trigger using a wizard:

1 Open a creation wizard for a trigger. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Triggers (Oracle) - Properties.
- **Events** panel - for details, see Triggers (Oracle) - Events.
- **Column Selection** panel - for details, see Triggers (Oracle) - Column Selection.
- **Action** panel - for details, see Triggers (Oracle) - Action.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Triggers (Oracle) - Properties

When creating or editing a trigger, this tab/panel lets you work with the following settings:

Setting	Availability and description	
Parent Type	Select the type (TABLE, DATABASE, SCHEMA, or VIEW) of element that this trigger is to be associated with.	
Parent Schema, and **Parent Name**	If available and appropriate, select the specific object for which the trigger is being created.	
Schema and **Name**	Select a schema and provide a name for the trigger being created.	
Enabled	Deselecting this check box adds a DISABLE clause to the CREATE OR REPLACE TRIGGER statement, disabling the trigger.	
Trigger Timing	**BEFORE**: These triggers serve as extensions to the constraint subsystem and are most often used to validate input data, generate values for newly inserted rows, and read from other tables for cross-reference purposes. Note: Before triggers must be created as a For Each Row. **AFTER**: Such a trigger is run after the integrity constraint validations; they can be used to modify operations in the database or be used for activities beyond the database, like supporting an alert notification. **COMPOUND** (11.1.0.7 ^): These triggers fire at more than one timing point.	
Trigger Type	Not available with a **Parent Type** of **Database** or **Schema** and not available with a **Trigger Timing** value of **COMPOUND**	Selecting **ROW** adds a FOR EACH ROW clause to the CREATE OR REPLACE TRIGGER statement, making this a row trigger that fires for each affected row. Selecting **STATEMENT** designates this as a statement trigger that only fires once.
Old Table Alias	Not available with a **Parent Type** of **Database** or **Schema**	Type the name of a temporary table of rows as they exist before they are updated or deleted.
New Table Alias		Type a name for a temporary table of rows as they exist after they are inserted or updated.
When Clause		Type a WHEN clause or open a WHEN CLAUSE wizard to further qualify the trigger behavior.

Triggers (Oracle) - Events

When creating or editing a trigger, this tab/panel lets you provide BEFORE, AFTER, or FOR clause options.

If you selected a **Parent Type** of **Table** or **View**, select one or more of the INSERT, UPDATE, or DELETE check boxes to specify the events that are to fire the trigger.

If you selected a **Parent Type** of **Database** or **Schema**, select one or more of the ALTER, CREATE, ANALYZE, ASSOCIATE_STATISTICS, AUDIT, COMMENT, DISASSOCIATE_STATISTICS, DROP, GRANT, NOAUDIT, RENAME, REVOKE, TRUNCATE, LOGON, SERVERERROR, or SUSPEND check boxes to specify the events that are to fire the trigger.

Triggers (Oracle) - Column Selection

If you chose UPDATE as the **Trigger Event**, this tab/panel lets you provide FOR UPDATE OF, BEFORE UPDATE OF, or AFTER UPDATE OF clause options. Select the check box beside each column that is to fire the trigger.

Triggers (Oracle) - Action

If you selected a **Trigger Timing** value of BEFORE or AFTER, provide the complete body of trigger. Keep the following in mind:

- If the SQL statement you have in mind for a trigger is longer than 60 lines of code, you would be better off creating a stored procedure.
- INSTEAD OF statements can only be used by view triggers.
- BEFORE and AFTER options cannot be used for view triggers.

If you selected a **Trigger Timing** value of COMPOUND, a compound trigger block template is generated. Customize the generated code provide a valid trigger body.

Object Types Wizard (Oracle)

Types define an abstract data type or object composed of a collection of similar types of data. For example, you can create an object type that defines a full address rather than the pieces of an address, such as city, state and postal code. An object type stores the pieces of an address in a single type, storing them in the same location, and allowing the full address to be accessed and manipulated as single unit rather than multiple units.

To Open the Object Type Wizard

1 On the Navigator/Explorer, find the datasource where you want to create a Type and expand the Schema node.

2 Right-click the **Type** node, and select **New**.

The single page wizard asks that you pick an owner for the type and name the type. The fun begins after you make your choices and click **Finish**. The Object Editor, where you can finalize the object type's creation, opens.

The Object Type Editor's tabs are:

Header: Here you flesh out the specification for the type you're creating including any methods (a subprogram), attributes, and any parameters or return types for a function.

Body: For every specification, the object type body defines the code for the method. If the object type header declares only attributes, not a method, the body is unnecessary.

You need to create the object type before you can see the following tabs. After you've added what you need to the Header and Body pages of the Editor, click **Create** on the toolbar.

Information: Read-only page that indicates the vital statistics for the object type.

Dependencies: A tree structure displays the objects that depend on the type you have created.

Privileges: Displays individual, group, and role permissions associated with the object type.

Unique Keys Wizard (Oracle)

A unique key constraint requires that every value in a column or set of columns be unique. Thus, no two rows of a table have duplicate values in the column or set of columns you identified. So, for example, you can use a unique key constraint to make sure you haven't duplicated a social security number in a list of employees.

To create a new unique key using a wizard:

1 Open a creation wizard for a unique key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Unique Keys (Oracle) - Properties.
- **Columns** panel - for details, see Unique Keys (Oracle) - Columns.
- **Storage** panel - for details, see Unique Keys (Oracle) - Storage.
- **Partition** panel - for details, see Unique Keys (Oracle) - Partition.
- **DDL** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Unique Keys (Oracle) - Properties

When creating or editing a unique key, this tab/panel lets you work with the following settings:

Setting	Description
Table Owner and **Table Name**	Choose the owner and name of the table in which the unique key is being created.
Name	Provide a name for the unique key being created.
No Sort	Enable this feature if the rows in the table already stored in ascending order. This increases the speed of the index creation process. Oracle does not sort the rows.
Logging	Enabling logs this operation to the redo file.
Reverse	Enabling this feature stores the bytes of the index block in reverse order and excludes the ROWID. The ROWID is a globally unique identifier for a row in a database. It is created at the time the row is inserted into a table, and destroyed when it is removed from a table.
Validate	Enabling this option indicates that existing data is checked against the constraint when the primary key is enabled. Leaving it disabled indicates that only new data is to be checked against the constraint.
Deferrable	Dictates whether constraint checking can be deferred until the end of a transaction.
Deferred	This option is enabled only if you enabled the **Deferrable** option. Select IMMEDIATE to have the constraint checked at the end of every DDL statement. Select DEFERRED to have the constraint checked only at the end of a transaction.
Enabled	Enables or disables the primary key.

Unique Keys (Oracle) - Columns

From the **Column** dropdown, select a column for the index and specify a **Sort** option. To add more columns, click the **New** button and then follow the steps in the last instruction. Use the Delete button to drop columns.

Unique Keys (Oracle) - Storage

When creating or editing a primary key, this tab/panel lets you work with the following settings:

Setting	Description
Data Block Storage group	Select the DEFAULT **Tablespace** only if you are creating a local partitioned index and want the partitions in the same tablespace as the partitions in the underlying table. (Each partition of a local index is associated with one partition of the table. Oracle can then keep the index partitions in synch with table partitions.) A transaction entry is needed for each INSERT, UPDATE, DELETE, etc. statement that accesses one or more rows in the block. Transaction entries in many operating systems require approximately 23 bytes. **Percent Free** identifies how much space you want to allocate for new rows or updates to existing rows. **Initial Transactions** ensures that a minimum number of concurrent transactions can update an index block, avoiding the overhead of allocating a transaction entry dynamically. **Maximum Transactions** limits concurrency on an index block.
Extents group	An extent is the unit of space allocated to an object whenever the object needs more space. **Initial Extent** - The initial space extent (in bytes) allocated to the object. **Next Extent** - The next extent (in bytes) that the object will attempt to allocate when more space for the object is required. **Percentage Increase** - Lets you type the percentage. NOTE: You should be careful when setting Percent Increase because it magnifies how an object grows and, therefore, can materially affect available free space in a tablespace. **Minimum Extents** - For a dictionary managed tablespace, this is the total number of extents to be allocated when the index is first created. For a locally managed tablespace, this is simply the initial amount of space allocated. **Maximum Extents** - For a dictionary managed tablespace, this is the total number of extents that can ever be allocated to the index. In a locally managed tablespace, the database will automatically manage the extents.
Freelists group	**Free lists** let you manage the allocation of data blocks when concurrent processes are issued against the index. You can potentially improve the performance of the index by identifying multiple free lists, which can reduce contention for free lists when concurrent inserts take place. The default and minimum value is 1. You should increase this number if multiple processes access the same data block. **Free List Groups** is the number of groups of free lists. NOTE: This option is only applicable for the parallel server option.
Buffer Pool	DEFAULT - Choose this if you want to use the default bufferpool. KEEP - Use this to retain the object in memory to avoid I/O conflicts. This type of bufferpool stores frequently referenced data blocks in a separate cache. RECYCLE - Select this option to save cache space by ridding data blocks from memory as soon as they are no longer in use.

Unique Keys (Oracle) - Partition

Clicking **Create Partition** opens a wizard that lets you create a partition.

Users Wizard (Oracle)

Whoever you add as a user will have access to the Oracle database. You also can set up the means by which the database recognizes the user.

Important Notes

- To create a user, you need the CREATE USER system privilege.

To create a new user using a wizard:

1 Open a creation wizard for a user. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Users (Oracle) - Properties.
- **Roles** panel - for details, see Users (Oracle) - Roles.
- **Quotas** panel - for details, see Users (Oracle) - Quotas.
- **DDL** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Users (Oracle) - Properties

When creating or editing a user, this tab/panel lets you work with the following settings:

Setting	Description
Name	Provide a name for the user.
Default Tablespace and **Temporary Tablespace**	Select the default tablespace for objects the user creates and the tablespace to be used for the user's temporary segments. If you do not specify a default tablespace when creating a user, the assigned tablespace will be that set using the Set Default function. For details, see Set Default.
Profile	The profile you choose affects the amount of database resources available to the user.
Identified By	REQUIRED_YES - A user identified by password is a local user. REQUIRED_EXTERNAL - This is a user who is validated by an external service like an operating system or third-party service. The login authentication is handled by that external entity. REQUIRED_GLOBAL - The user you're creating is a global user who is authenticated by the enterprise directory service.
Password	If you specified an **Identified By** value of REQUIRED_YES, provide a password for the user.
External Name	If you specified an **Identified By** value of REQUIRED_EXTERNAL, provide an external name for the user.
Account Locked	When enabled, the account cannot be altered by anyone but the creator. It also means that after a specified number of failed attempts to access the database, the database will remain closed to the user for a period of time.
Password Expired	Marks the password as expired, forcing the user to change the password before being allowed to connect to the database.

Users (Oracle) - Roles

For each role to be assigned to the user, select the check box beside that role.

Users (Oracle) - Quotas

To assign a tablespace quota for a user, select a tablespace, select the **Other** radio button, and provide the value in the **Quota** box.

Views Wizard (Oracle)

Views are SQL queries stored in the system catalog that customize the display of data contained in one or more tables. Views behave like tables because you can query views and perform data manipulation operations on them. However, views do not actually store any data. Instead, they depend on data contained in their base tables. You can use views to help enforce corporate security policies by creating a view that limits information a user can see.

Important Notes

- To create a view in your own schema, you need CREATE VIEW privileges. To create a view in someone else's schema, you need CREATE ANY VIEW privileges.

To create a new view using a wizard:

1 Open a creation wizard for a view. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Views (Oracle) - Properties.
- **Definition** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Views (Oracle) - Properties

When creating or editing a view, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the owner of the view. The owner of the view must have SELECT privileges for the tables in the CREATE view statement or DBADM authority on the database that contains the table.
Name	Provide a name for the view.

Views (Oracle) - Definition

This is where you build the query that will display the data you're interested in seeing. The template is CREATE VIEW <name> AS
SELECT: Identify the columns you want to show in the view
FROM: Identify the table(s) you want to draw the data from
WHERE: Write the equivalence you want to view.

You can use the Query Builder to help you write the appropriate SQL statement. For details, see Query Builder.

PostgreSQL Object Wizards

You create PostgreSQL objects using the following wizards:

- Check Constraints Wizard (PostgreSQL)
- Domains Wizard (PostgreSQL)
- Exclusion Constraints, Primary Keys, and Unique Keys Wizards (PostgreSQL)
- Foreign Keys Wizard (PostgreSQL)
- Functions Wizard (PostgreSQL)
- Indexes Wizard (PostgreSQL)
- Roles Wizard (PostgreSQL)
- Rules Wizard (PostgreSQL)
- Schemas Wizard (PostgreSQL)
- Tables Wizard (PostgreSQL)
- Tablespaces Wizard (PostgreSQL)
- Triggers Wizard (PostgreSQL)
- Views Wizard (PostgreSQL)

Check Constraints Wizard (PostgreSQL)

This wizard lets you build and submit an ALTER TABLE... ADD CONSTRAINT... CHECK... statement, adding a constraint to an existing table.

To create a new check constraint using a wizard:

1 Open a creation wizard for a check constraint. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Check Constraints (PostgreSQL) - Properties.
- **Definition** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Check Constraints (PostgreSQL) - Properties

When creating or editing a check constraint, this tab/panel lets you work with the following settings:

Setting	Description
Table Schema and **Table Name**	Identify the table for which the constraint is being created.
Name	Lets you type the name of the constraint.
No Inherit	If selected, a NO INHERIT parameter is added to the generated ALTER TABLE statement, letting you create a non-inherited check constraint.
Is Validated	If not selected, a NOT VALID parameter is added to the generated ALTER TABLE statement, letting you ensure that initial check to verify that all rows satisfy the constraint is not executed.
Check Condition	Type the check constraint condition. As a convenience, use the **Table Columns** button to paste in column names as part of the condition.

Domains Wizard (PostgreSQL)

This wizard lets you build and submit a CREATE DOMAIN... AS... statement, adding a new, datatype-based domain.

Note: Before working with this object type, consult PostgreSQL documentation for details on topics such as NULL and default values. For more information, see Accessing Third Party Documentation.

To create a new domain using a wizard:

1 Open a creation wizard for a domain. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Domains (PostgreSQL) - Properties.
- **Constraints** panel - Domains (PostgreSQL) - Constraints.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Domains (PostgreSQL) - Properties

When creating or editing a domain, this tab/panel lets you work with the following settings:

Setting	Description
Owner and **Schema**	Lets you select the schema and owner of the new domain.
Datatype	Lets you provide a name for the new domain.
Type	Lets you select the underlying datatype of the new domain.
Allow Nulls	If unselected, a NOT NULL parameter is included with the generated DDL, generally assuring that the domain cannot take a NULL value.
Default	Lets you enter a value or expression that defines the default value for columns of this domain type.
Collation Schema and **Collation Name**	Let you provide the value of a COLLATE parameter, specifying an optional collation for the domain.

Domains (PostgreSQL) - Constraints

When creating or editing domains, this tab/panel lets you manage foreign key, primary key, unique key, exclusion constraints, and check constraints for the domain:

Click **Add** to open the Check Constraints Wizard. Use the wizard to create a new constraint for the domain. See Check Constraints Wizard (PostgreSQL)

Select an existing constraint and click **Edit** to open a wizard that lets you add modify that constraint. See Check Constraints Editor (PostgreSQL)

Select a constraint and click **Drop** to drop a constraint from the table.

Exclusion Constraints, Primary Keys, and Unique Keys Wizards (PostgreSQL)

The wizards for creating these object types let you build and submit an ALTER TABLE... ADD CONSTRAINT... statements specific to the object type.

Note: Before working with these object types, consult PostgreSQL documentation. For more information, see Accessing Third Party Documentation.

To create a new exclusion constraint, primary key, or unique key using a wizard

1 Open a creation wizard for the target object type. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Exclusion Constraints, Primary Keys and Unique Keys (PostgreSQL) - Properties.
- **Constraints** panel - Exclusion Constraints, Primary Keys, and Unique Keys (PostgreSQL) - Columns.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Exclusion Constraints, Primary Keys and Unique Keys (PostgreSQL) - Properties

When creating or editing an exclusion constraint, primary key, or unique key, this tab/panel lets you work with the following settings:

Setting	Availability and Description
Table Schema and **Table Name**	Select the schema and name of the parent table.
Name	Provide a name for the object being created.
Index Name	Primary Keys and Unique Keys only. Lets you provide the name of an existing index that will be used as the base for the primary or unique key.
Deferrable	If selected, a DEFERRABLE parameter is included in the generated DDL, specifying that checking of constraints can be postponed until the end of the transaction.
Deferred	If IMMEDIATE is not selected, an INITIALLY DEFERRED parameter is added to the generated DDL, ensuring that the constraint is checked at the end of the transaction.
Predicate	Exclusion Constraints only. Lets you provide an expression used in a WHERE clause that defines the constraint on a subset of the table.
Access Name	Exclusion Constraints only. Lets you select an index method (**btree**, **gist**, **hash**, or **spgist**) for the USING option of the EXCLUDE clause.
Tablespace	Lets you optionally select the tablespace on which the new or altered object is to be created. If not specified, the default tablespace will be used.
Fill Factor	Lets you provide a FILLFACTOR value, specifying the percentage of tables that can be packed.
Buffering	Excusion constraints only. Only available with an **Access Name** of **gist**. Lets you select a BUFFERING value (**AUTO**, **ON**, or **OFF**).

Exclusion Constraints, Primary Keys, and Unique Keys (PostgreSQL) - Columns

When creating or editing an exclusion constraint, this tab/panel lets you set up the operators on the columns or expressions that define the exclusion constraint.

Foreign Keys Wizard (PostgreSQL)

This wizard lets you build and submit an ALTER TABLE... ADD CONSTRAINT .. statement, adding a new, foreign key to an existing table.

To create a new foreign key using a wizard:

1 Open a creation wizard for a foreign Key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Foreign Keys (PostgreSQL) - Properties.
- **Column Mapping** panel - Foreign Keys (PostgreSQL) - Column Mapping.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Foreign Keys (PostgreSQL) - Properties

When creating or editing a domain, this tab/panel lets you work with the following settings:

Setting	Description
Table Schema and **Table Name**	Let you select the table in which the foreign key is being defined.
Name	Lets you provide the ADD CONSTRAINT parameter value, naming the new foreign key.
Validate	If not selected, a NOT VALID parameter is included in the generated DDL, specifying that the initial check to verify that all rows satisfy the constraint is not executed.
Deferrable	If selected, a DEFERRABLE parameter is included in the generated DDL, specifying that checking of constraints can be postponed until the end of the transaction
Deferred	If IMMEDIATE is not selected, an INITIALLY DEFERRED parameter is added to the generated DDL, ensuring that the constraint is checked at the end of the transaction.
Delete Rule and **Update Rule**	Choices correspond to ON DELETE... and ON UPDATE parameter variations added to the generated DDL: **CACHE**, **SET DEFAULT** , **SET NULL** , and **RESTRICT**.
Match Full	If selected, a MATCH FULL parameter is added to the generated DDL, ensuring that no column of a multicolumn foreign key can be null unless all foreign key columns are null.

Foreign Keys (PostgreSQL) - Column Mapping

When creating or editing a foreign key, this tab/panel lets you specify the REFERENCES and FOREIGN KEY parameters of the generated ALTER TABLE... DDL.

1 Under **Main Table**, select the check boxes corresponding to the columns in the current table that are to use values in the referenced table.

2 Under **Referenced Table**, choose the **Owner** and then the **Table** name of the referenced table and then select the check boxes corresponding to the columns in the referenced table whose values are to be used in the main table.

Functions Wizard (PostgreSQL)

This wizard lets you build and submit a CREATE OR REPLACE FUNCTION statement, creating a new function or replacing an existing function.

Note: Before working with this object action, consult PostreSQL documentation for details on CREATE FUNCTION parameter values. For more information, see Accessing Third Party Documentation.

To create a new function using a wizard:

1 Open a creation wizard for a function. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Functions (PostgreSQL) - Properties.
- **Parameters** panel - Functions (PostgreSQL) - Parameters.
- **Body** panel - Functions (PostgreSQL) - Body.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Functions (PostgreSQL) - Properties

When creating or editing a domain, this tab/panel lets you work with the following settings:

Setting	Description
Schema and **Name**	Let you identify the owning schema and provide a name for the function.
Return Type	Lets you provide a RETURNS clause value, specifying a valid type (base, composite, domain type, or a reference to the type of a table column) fir the value returned by the function.
Language	Lets you provide a LANGUAGE clause value, specifying a valid language implementation.
Object File and **Link Symbol**	Let you specify the AS clause appropriate to dynamically loadable C language functions for which the function name in the C language source code is identical to the name of the SQL function. **Object File** is the name of the file containing the dynamically loadable object. **Link Symbol** is the function name in the C language source code.
Is Window	If selected, specifies that this C language function is a window function.
Leakproof	If selected, a LEAKPROOF clause is added to the generated DDL, indicating that the function provides no information about its arguments other than by its return value.
Execution Cost	Lets you provide a COST clause value, estimating the execution cost in CPU operator cost units.
Result Rows	Lets you provide a ROWS clause value, estimating the number of rows the function is expected to return.
Behavior	Lets you select an IMMUTABLE, STABLE, or VOLATILE clause to the generated DDL, indicating the functions behavior with respect to database lookups and ability to make changes to the database.
Strict	Lets you add a STRICT clause to the generated DDL, specifying whether non-NULL values can be returned if any provided input parameters are NULL.
Is Security Definer	Lets you add a SECURITY DEFINED clause to the generated DDL, specifying that the function is to execute with the same privileges as the user who created it.

Functions (PostgreSQL) - Parameters

When creating or editing a function, this tab/panel lets you build and maintain a list of input/output parameters for the function. For each input or output parameter for this function, use the New button to add a new parameter and then provide a name for the parameter.

To edit a parameter in the attributes list, first select the parameter in the list. Then, in the **Attributes** area, select a **Datatype**, select a **Parameter Mode** (INPUT, OUTPUT, INPUT_OUTPUT, or VARIADIC), and if you chose INPUT, you can optionally provide a **Default** value.

To delete a parameter from the list, select the parameter and click the Delete button.

To reposition a parameter in the list, select the parameter and use the arrow buttons to move the parameter up or down in the list.

Functions (PostgreSQL) - Body

When creating or editing a function, this tab/panel lets you build and maintain a valid AS clause value that defines the function. You can use an internal function name, the path to an object file, a SQL query, or text in a procedural language.

Indexes Wizard (PostgreSQL)

This wizard lets you build and submit a CREATE INDEX or CREATE UNIQUE INDEX statement, creating a new index on a table.

To create a new index using a wizard:

1 Open a creation wizard for an index. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Indexes (PostgreSQL) - Properties.
- **Columns** panel - Indexes (PostgreSQL) - Columns.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Indexes (PostgreSQL) - Properties

When creating or editing a domain, this tab/panel lets you work with the following settings:

Setting	Availability and Description
Parent Type	Lets you specify whether the index is being defined for a table or view.
Table Schema and **Table Name**	Select the schema and name of the parent table.
Name	Provide a name for the index being created.
Unique	If selected, CREATE UNIQUE INDEX DDL is generated, enforcing uniqueness. Otherwise simple CREATE INDEX DDL i generated.
Concurrently	If selected, s CONCURRENTLY parameter is added to the generated DDL, specifying that the index is built without taking any locks that would prevent concurrent inserts, updates, or deletes on the target table.
Clustered	If selected, an ALTER TABLE... CLUSTER ON... statement is generated following the DDL to generate the index, specifying this index as the default for subsequent cluster operations.
Tablespace	Lets you optionally select the tablespace on which the index is to be created. If not specified, the default tablespace will be used.
Index Method	Lets you specify a USING option value of **btree**, **hash**, **gist**, **gin**, or **spgist**.
Fill Factor	Lets you add a FILLFACTER= option value, specifying the percentage that the index method will try to fill index pages.
Fast Update	Lets you add a FASTUPDATE= ON option, enabling fast update.
Buffering	Available with an **Index Method** of **gist**. Determines whether the buffering build technique is set to ON, OFF, or AUTO.
IndexPredicate	Lets you provide a WHERE condition that will be used as a constraint expression for a partial index.

Indexes (PostgreSQL) - Columns

From the Column dropdown, select a column for the index and specify **Sort**, **Collation**, and **Operator Class** option. To add more columns, click the **New** button and then follow the steps in the last instruction. Use the Delete button to drop columns.

Roles Wizard (PostgreSQL)

This wizard lets you build and submit a simple CREATE ROLE statement, defining a new database role.

To create a new role using a wizard:

1 Open a creation wizard for a role. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Roles (PostgreSQL) - Properties.
- **Membership** panel - Roles (PostgreSQL) - Membership.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Roles (PostgreSQL) - Properties

When creating or editing a role, this tab/panel lets you work with the following settings:

Setting	Availability and Description
Name	Lets you provide the name of the new role.
Password and **Encrypted Password**	These items control how the password is created for the new role. The **Password** control lets you provide a password for the new role. If **Encrypted Password** is selected, an ENCRYPTED PASSWORD... parameter is generated. Otherwise, an UNENCRYPTED PASSWORD... parameter is generated.
Account Expires	Selecting this control enables a calendar widget that lets you select a value for a VALID UNTIL parameter in the generated DDL.
Connection Limit	Lets you provide the value for a CONNECTION LIMIT parameter, setting a limit on the number of concurrent connections the new role can make.
Login	This control determines whether a role is allowed to log in. If **Login** is selected, a LOGIN parameter is included in the DDL. Otherwise, a NOLOGIN parameter is included in the DDL.
Superuser	This control determines whether a role can override all access restrictions within the database. If **Superuser** is selected, a SUPERUSER parameter is included in the DDL. Otherwise, a NOSUPERUSER parameter is included in the DDL.
Create Database and **Create Roles**	These controls determine whether a role can create databases or roles, respectively. When selected, CREATEDB or CREATEROLE parameters are added to the generated DDL. Otherwise, NOCREATEDB or NOCREATEDB parameter are included in the DDL.
Update Catalog	Only available if **Superuser** is selected. By default, this option is selected, dictating that this role can update system catalogs directly. If deslected, an UPDATE pg_catalog.pg_authid SET rolcatupdate=false WHERE rolname=... statement is generated and appended to the CREATE ROLE statement, explicitly disabling catalog update for the new role.
Replication	This setting determine whether a role can initiate streaming replication or can control backup mode. If **Replication** is selected, a REPLICATION parameter is included in the DDL. Otherwise, a NOREPLICATION parameter is included in the DDL.

Roles (PostgreSQL) - Membership

When creating or editing a role, this tab/panel lets you append one or more GRANT statements to the generated CREATE ROLE statement, granting membership in the role to other roles. For each GRANT statement to be generated, select a role from the **Groups** list and click **Join** to assign membership in the selected role to the role being created.

Rules Wizard (PostgreSQL)

This wizard lets you build and submit a simple CREATE OR REPLACE RULE statement, defining a new database rule.

To create a new rule using a wizard:

1 Open a creation wizard for a rule. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Rules (PostgreSQL) - Properties.
- **Definition** panel - Rules (PostgreSQL) - Definition.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Rules (PostgreSQL) - Properties

When creating or editing a rule, this tab/panel lets you work with the following settings:

Setting	Availability and Description
Name	Lets you provide the name of the new role.
Parent Type	Select the type (TABLE or VIEW) of element to which this rule is to be applied.
Parent Schema and **Parent Name**	Select the schema and name of the parent object.
Connection Limit	Lets you provide the value for a CONNECTION LIMIT parameter, setting a limit on the number of concurrent connections the new role can make.
Name	Provide a name for the rule being created.
IsInstead	If unselected, a simple DO clause will be generated for the rule. If selected, a DO INSTEAD clause will be generated. The DO or DO INSTEAD clause command or commands are provided on the **Definition** tab/panel.
Event	Lets you select an AS ON event of SELECT, INSERT, UPDATE, or DELETE.

Rules (PostgreSQL) - Definition

When creating or editing a rule, this tab/panel lets you provide the DO or DO INSTEAD clause value, specifying the additional or alternative commands for this rule.

Schemas Wizard (PostgreSQL)

This wizard lets you build and submit a CREATE SCHEMA... AUTHORIZATION... statement, creating a new schema.

To create a new schema using a wizard:

1 Open a creation wizard for a schema. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Schemas (PostgreSQL) - Properties.
- **Permissions** panel - for details, see Setting Permissions or Privileges for an Object
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Schemas (PostgreSQL) - Properties

When creating or editing a check constraint, this tab/panel lets you work with the following settings:

Setting	Description
Name	Lets you type the name of the schema.
Owner	Lets you select the name of the user who will own the schema.

Tables Wizard (PostgreSQL)

This wizard lets you build and submit a CREATE TABLE statement, creating a new table.

To create a new table using a wizard:

1 Open a creation wizard for a table. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Tables (PostgreSQL) - Properties.
- **Ancestors** panel - for details, see Tables (PostgreSQL) - Ancestors.
- **Columns** panel - for details, see Tables (PostgreSQL) - Columns.
- **Constraints** panel - for details, see Tables (PostgreSQL) - Constraints.
- **Permissions** panel - for details, see Setting Permissions or Privileges for an Object
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Tables (PostgreSQL) - Properties

When creating or editing a table, this tab/panel lets you work with the following settings:

Setting	Description
Owner, **Schema**, and **Name**	Lets you provide ownership details and provide a name for the table.
Tablespace	Lets you select a CREATE TABLE... TABLESPACE parameter value, specifying the tablespace in which the new table will be created.
With OIDS	If selected, a WITH (OIDS=TRUE) parameter is added to the generated DDL, specifying that rows of the table will have object identifiers. Otherwise, a WITH (OIDS=FALSE) parameter is added to the generated DDL.
Persistence	Lets you change from the default form of CREATE TABLE statement by selecting either **TEMPORARY** (CREATE TEMPORARY TABLE) or **UNLOGGED** (CREATE UNLOGGED TABLE).
Of Type	Lets you select the value for an OF clause, designating this as a typed table. Any columns defined using the **Ancestors** or **Columns** tab are overwritten when a type is selected.
On Commit	Only available with a **Persistence** selection of **TEMPORARY**. Lets you change the ON COMMIT... parameter DDL generated between **DROP** (ON COMMIT DROP), **PRESERVE ROWS** (ON COMMIT PRESERVE ROWS), and **DELETE ROWS** (ON COMMIT DELETE ROWS).
Fill Factor	Lets you provide a FILLFACTOR parameter value, an integer expressing a percentage between 10 and 100, dictating the upper limit to which INSERTs can pack table pages.
Table Autovacuum... settings	Lets you enable or disable the autovacuum daemon as well as specify values for parameters used in autovacuum calculations (Threshold, Scale Factor, Analyze Threshold, Analyze Threshold, Cost Delay, Cost Limit, Freeze Min Age, Freeze Max Age, Freeze Table Age, Multixact Freeze Min Age, Multixact Freeze Max Age, and Multixact Freeze Table Age.
Toast Table Autovacuum... settings	Lets you enable or disable the autovacuum daemon for the table's secondary toast table, as well as specify values for parameters used in autovacuum calculations (Threshold, Scale Factor, Cost Delay, Cost Limit, Freeze Min Age, Freeze Max Age, Freeze Table Age, Multixact Freeze Min Age, Multixact Freeze Max Age, and Multixact Freeze Table Age.

Tables (PostgreSQL) - Ancestors

This tab/panel lets you add an INHERITS clause to the generated DDL, building a list of tables from which this table inherits columns. For each table to be added to the list, use the New button to enable a dropdown that lets you select a table. Use the Delete button to remove a selected table from the list.

Tables (PostgreSQL) - Columns

When creating or editing a table, this tab/panel lets you manage the columns for the table:

To add a column to the table:

1 Use the **Add Column** dropdown to add a new column (**Add Column**) to the bottom of the column list or to insert a new column (**Insert Column**) above the currently selected column in the column list.

2 Provide a Name for the column.

3 Use the following table as a guide to providing additional property values, noting that availability of a property differs by data type and other property selections:

Setting	Description
Type	Lets you select the datatype for the column.
Allow Nulls	If not selected, a NOT NULL argument is added to the column specification.
Default	Lets you provide a default value for the column.
Comment	Lets you append a COMMENT ON COLUMN statement to the generated table creation DDL.
Statistics Target	Lets you append an ALTER TABLE... ALTER COLUMN... SET STATISTICS statement to the generated table creation DDL, setting the per-column statistics-gathering target for ANALYZE operations.
Storage Mode	Lets you append an ALTER TABLE... ALTER COLUMN... SET STORAGE statement to the generated table creation DDL, with a storage specifier of EXTERNAL, EXTENDED, or MAIN.
NDistinct Values and **NDistinct Values Inherited**	Let you append ALTER TABLE... ALTER COLUMN... SET (n_distinct or n_distinct_inherited) statements to the generated table creation DDL. These override the number-of-distinct-values estimates made by ANALYZE operations for the table or its inherited children.

To edit a column:

- Select the column from the list and edit property values as per the descriptions above.

To drop a column:

- Select the column from the list and click the Delete button.

To reorder the column list:

- Move a selected column up or down the list using the Move Item Up or Move Item Down buttons.

Tables (PostgreSQL) - Constraints

When creating or editing tables, this tab/panel lets you manage foreign key, primary key, unique key, exclusion constraints, and check constraints for the table:

- Select a constraint type and click **Add** to open a wizard that lets you add a new constraint of that type to the table.

- Select an existing constraint and click **Edit** to open a wizard that lets you add modify that constraint. The wizards used to add and modify constraints offer functionality identical to the editors and wizards used when creating those object types. For details, see the following topics:
    -
        - Check Constraints Wizard (PostgreSQL) and Check Constraints Editor (PostgreSQL)
        - Foreign Keys Wizard (PostgreSQL) and Foreign Keys Editor (PostgreSQL)
        - Roles Wizard (PostgreSQL) and Roles Editor (PostgreSQL)
        - Exclusion Constraints, Primary Keys, and Unique Keys Wizards (PostgreSQL) and Exclusion Constraints, Primary Keys, or Unique Keys Editors (PostgreSQL)
- Select a constraint and click **Drop** to drop a constraint from the table.

Tablespaces Wizard (PostgreSQL)

This wizard lets you build and submit a simple CREATE TABLESPACE statement, registering a new, cluster-wide tablespace.

To create a new tablespace using a wizard:

1 Open a creation wizard for a tablespace. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Check Constraints (PostgreSQL) - Properties.
- **Definition** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Tablespaces (PostgreSQL) - Properties

When creating or editing a tablespace, this tab/panel lets you work with the following settings:

Setting	Description
Table Schema and **Table Name**	Identify the table for which the constraint is being created.
Name	Lets you type the name of the tablespace.
Owner	Lets you select the name of the user who will own the tablespace.
Location	Lets you provide a LOCATION parameter clause, identifying the directory used for the tablespace.
random_page_code and **seq_page_cost**	When either of these settings are assigned values, the generated CREATE TABLESPACE DDL is followed by an ALTER TABLESPACE statement that lets you override the RANDOM_PAGE_COST and SEQ_PAGE_COST configuration parameters.

Triggers Wizard (PostgreSQL)

This wizard lets you build and submit a CREATE TRIGGER or CREATE CONSTRAINT TRIGGER statement, creating a trigger to be associated with a table or view. Optionally, an ALTER TABLE statement can be generated to control the firing of the trigger.

To create a new trigger using a wizard:

1 Open a creation wizard for a trigger. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Triggers (PostgreSQL) - Properties.
- **Column Selection** panel - Triggers (PostgreSQL) - Column Selection.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Triggers (PostgreSQL) - Properties

When creating or editing a trigger, this tab/panel lets you work with the following settings:

Setting	Description
Parent Type	Select the type (TABLE or VIEW) of element that this trigger is to be associated with.
Parent Schema and **Parent Name**	Select the schema and name of the parent object.
Name	Provide a name for the trigger being created.
Constraint trigger, **Referenced Table Schema**, and **Referenced Table Schema**	Selecting the **Constraint Trigger** control specifies that a CREATE CONSTRAINT TRIGGER statement is generated instead of a CREATE TRIGGER statement. Optionally, you can use the **Referenced Table Schema**, and **Referenced Table Schema** controls to specify a FOR clause to indicate the referenced table.
Trigger Type	This control lets you specify a FOR clause, specifying whether the trigger is called once per row modified or once per operation. ROW specifies FOR EACH ROW while STATEMENT (not available if **Constraint Trigger** is selected) specifies FOR EACH STATEMENT.
Trigger Timing	Lets you select whether the trigger fires BEFORE or AFTER the operation.
Trigger Event List	Lets you build a list of events (INSERT, DELETE, UPDATE, and in the case of FOR EACH STATEMENT triggers, TRUNCATE) that cause the trigger to fire.
Deferrable and **Deferred**	These controls are only available if **Constraint Trigger** is selected. Selecting **Deferrable** generates a DEFERRABLE INITIALLY IMMEDIATE clause. Subsequently selecting **Deferred** changes the clause to DEFERRABLE INITIALLY DEFERRED.
Function Schema, **Function Name**, and **Function Arguments**	These controls let you specify details of the function to be called when the trigger fires.
Fire Mode	Optionally, lets you include an ALTER TABLE statement specifying an action of DISABLE TRIGGER, ENABLE ALWAYS TRIGGER. or ENABLE REPLICA TRIGGER. Selecting ORIGIN LOCAL results in no ALTER TABLE statement being appended to the main CREATE TRIGGER/CREATE CONSTRAINT TRIGGER statement.

Triggers (PostgreSQL) - Column Selection

If you chose UPDATE as the **Trigger Event List** option, this tab/panel lets you provide FOR UPDATE OF, BEFORE UPDATE OF, or AFTER UPDATE OF clause options. Select the check box beside each column that is to fire the trigger.

Views Wizard (PostgreSQL)

This wizard lets you build and submit a CREATE VIEW or CREATE MATERIALIZED VIEW statement.

To create a new view using a wizard:

1 Open a creation wizard for a view. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - Viewss (PostgreSQL) - Properties.
- **Definition** panel - Views (PostgreSQL) - Definition.
- **Comment** panel - for details, see Adding a Comment to an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Views (PostgreSQL) - Properties

When creating or editing a view, this tab/panel lets you work with the following settings:

Setting	Availability and Description
Schema	Lets you select the owner of the new view.
Name	Lets you provide the name of the new view.
Is Materialized	When selected, a CREATE MATERIALIZED VIEW statement is generated. When unselected, a CREATE VIEW statement is generated.
Security Barrier	This setting is only available with **Is Materialized** unselected. If selected, a WITH (security_barrier=true) clause is added to the generated DDL, specifying that the view is to provide row-level security.If unselected, a WITH (security_barrier=false) clause is added..
Tablespace	This setting is only available with **Is Materialized** selected. It lets you add a TABLESPACE clause to the generated DDL, specifying the tablespace where the materialized view is to be created.
Fill Factor	This setting is only available with **Is Materialized** selected. It lets you add a WITH (fillfacter= ...) clause to the generated DDL, specifying a an integer expressing a percentage between 10 and 100, dictating the upper limit to which INSERTs can pack pages.
With Data	This setting is only available with **Is Materialized** selected. If selected, a WITH DATA clause is appended to the generated DDL, specifying that the backing query is executed to provide new data. If unselected, a WITH NO DATA clause is appended to the generated DDL, specifying that no new data is generated.
View Autovacuum... settings	Lets you enable or disable the autovacuum daemon as well as specify values for parameters used in autovacuum calculations (Threshold, Scale Factor, Analyze Threshold, Analyze Threshold, Cost Delay, Cost Limit, Freeze Min Age, Freeze Max Age, Freeze Table Age, Multixact Freeze Min Age, Multixact Freeze Max Age, and Multixact Freeze Table Age.
Toast View Autovacuum... settings	Lets you enable or disable the autovacuum daemon for the table's secondary toast table, as well as specify values for parameters used in autovacuum calculations (Threshold, Scale Factor, Cost Delay, Cost Limit, Freeze Min Age, Freeze Max Age, Freeze Table Age, Multixact Freeze Min Age, Multixact Freeze Max Age, and Multixact Freeze Table Age.

Views (PostgreSQL) - Definition

Complete the CREATE VIEW statement by typing or pasting in the relevant query.

Sybase ASE Object Wizards

You create Sybase ASE objects using the following wizards:

- Aliases Wizard (Sybase ASE)
- Databases Device Wizard (DBArtisan - Sybase ASE)
- Databases Wizard (Sybase ASE)
- Data Caches Wizard (DBArtisan - Sybase ASE)
- Defaults Wizard (Sybase ASE)
- Dump Devices Wizard (DBArtisan - Sybase ASE)
- Extended Procedures Wizard (Sybase ASE)
- Foreign Keys Wizard (Sybase ASE)
- Functions Wizard (Sybase ASE)
- Groups Wizard (Sybase ASE)
- Indexes Wizard (Sybase ASE)
- Logins Wizard (Sybase ASE)
- Primary Keys Wizard (Sybase ASE)
- Procedures Wizard (Sybase ASE)
- Remote Servers Wizard (DBArtisan - Sybase ASE)
- Role Wizard (DBArtisan - Sybase ASE)
- Rules Wizard (Sybase ASE)
- Segment Wizard (Sybase ASE)
- Tables Wizard (Sybase ASE)
- Triggers Wizard (Sybase ASE)
- Unique Keys Wizard (Sybase ASE)
- User Datatypes Wizard (Sybase ASE)
- User Messages Wizard (Sybase ASE)
- Users Wizard (Sybase ASE)
- Views Wizard (Sybase ASE)

Aliases Wizard (Sybase ASE)

The Alias Wizard lets you map a login to an existing user in the database. You can set up aliases so that multiple users log in to the same account and therefore have the same privileges. You can also set up an alias based on individual log ins and give those users access to the same alias with the advantage that you can track their activity within the database.

To create a new alias using a wizard:

1 Open a creation wizard for an alias. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Aliases (Sybase ASE) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Aliases (Sybase ASE) - Properties

When creating or editing an alias, this tab/panel lets you work with the following settings:

Setting	Description
Name	Provide a name for the alias being created.
User	Select the user to which the alias refers. The user has to have a valid account on SQL Server but cannot be a user in the current database.

Databases Device Wizard (DBArtisan - Sybase ASE)

A database device is part or all of a disk device used for storing a database and its transaction log.

The Database Device Wizard lets you specify basic creation attributes and override many obscure, rarely-used parameters of the DISK INIT statement.

Note: Make sure that your Sybase ASE is configured for enough devices that it can accommodate the creation of another device. If not, increase that parameter from the **Server Configuration** dialog. For details, see Session Recording (DBArtisan).

To Open the Database Device Wizard

1 On the Navigator/Explorer, find the datasource where you want to add a database device and expand the **Storage** node.

2 Right-click the **Database Devices** node, and select **New**.

3 Use the following topics as a guide to understanding and setting options on the panels of this wizard:

- **Properties** panel - for details, see Database Devices (Sybase ASE) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

4 Finally, use the **Execute** button to create the object.

Database Devices (Sybase ASE) - Properties

When creating or editing a database device, this tab/panel lets you work with the following DISK INIT command, SP_DISKDEFAULT, or SP_DBEXTEND parameters:

Setting	Availability)	Description
Name		The abstract name you want to assign to the device. If for any reason the physical device is down, you can redefine the logical name to a new device.
In-memory	This feature is available as of Sybase ASE 15.5.	Adds a TYPE='inmemory' parameter, creating an in-memory device.
Device Number		DBArtisan automatically calculates the next open device number and defaults its value in the box. Numbering is sequential.
Size		In kilobytes, megabyte, gigabytes, or terabytes.
Cache Name	Available only if **In-memory** is selected	Lets you select from available in-memory type data caches, to provide a PHYSNAME= parameter value. For information on setting up an in-memory data cache, see Data Caches Wizard (DBArtisan - Sybase ASE).
Physical Name	Available only if **In-memory** is not selected	Specify the full path for the file, including the hard drive letter or machine name, directory and full file name with the *.DAT extension (for example, D:\SYB\DATA\TEST.DAT)
Controller Number		Lets you provide a CNTRLTYPE= parameter value. Normally, Sybase ASE uses 0
Virtual Address		Lets you provide a VSTART= parameter value, specifying a starting virtual address, or offset, to begin using the database device
Default Device		The DISK INIT used to create or edit the device is followed by an EXEC SP_DISKDEFAULT call. This control specifies whether it is called with a defaulton or defaultoff option.
Dsync		Lets you select a DSYNC= option, specifying whether database writes are made directly to the storage media, or are buffered using operating system files.
DirectIO		Lets you select a DIRECTIO= option, specifying whether you want to transfer data directly to disk, bypassing the operating system buffer cache.
Autogrowth properties	Available only if **In-memory** is not selected	Enabling **Auto Growth** results in the DISK INIT being followed by an EXEC SP_DBEXTEND 'SET', 'DEVICE'... call. This lets you set the **Growth rate** and **Maximum Size** for the device.

Databases Wizard (Sybase ASE)

The Database Wizard lets you create a database (a structured collection of data that can be updated or queried) without knowing the underlying commands. Databases can be simple, that is one file with many records and the same fields, or much more complicated with multiple files with different fields.

To create a new database using a wizard:

1 Open a creation wizard for a database. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Databases (Sybase ASE) - Properties.
- **Placement** panel - for details, see Databases (Sybase ASE) - Placement.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Databases (Sybase ASE) - Properties

When creating or editing a database, this tab/panel lets you work with the following settings:

Category	Setting (and availability)	Description
Creation Properties	**Database Name**	Provide a name for the database.
	Type	Generate variations on the CREATE DATABASE DDL generated to create the database, as follows: DEFAULT - generates a generic CREATE DATABASE statement. TEMP - generates a CREATE TEMPORARY DATABASE variation. ARCHIVE - generates a CREATE ARCHIVE DATABASE variation.
	In-memory (available with a **Type** of TEMP or DEFAULT) This feature is available as of Sybase ASE 15.5.	Generates CREATE INMEMORY DATABASE DDL, indicating that this is an in-memory database. For information setting up data caches and database devices for in-memory databases using DBArtisan, see Data Caches Wizard (DBArtisan - Sybase ASE) and Databases Device Wizard (DBArtisan - Sybase ASE).
	Scratch Database (available with a **Type** of ARCHIVE)	Lets you provide a WITH SCRATCH_DATABASE parameter value, specifying the name of an existing database that stores information about the archive database.

	For Proxy Update (available with a **Type** of DEFAULT and not available if **For Load** is selected) This feature is available as of Sybase ASE 12.5.	Adds a FOR PROXY_UPDATE parameter to the generated DDL, forcing resynchronization of proxy tables with the proxy database. For information on how to resynchronize an existing database, see Resynchronize.
	For Load (available with a **Type** of DEFAULT and not available if **For Proxy Update** is selected)	This option speeds loading by eliminating the step for pre-initializing panels. The for load option is appropriate when creating a database for recovering from media failure or moving a database from one machine to another.
	With Override (available with a **Type** of TEMP or DEFAULT)	Enable this option to specify whether logs and data are to be kept on the same logical device
	With Default Location (available with a **Type** of DEFAULT)	Lets you add a WITH DEFAULT_LOCATION clause value, specifying a storage location for new tables. If you also select **For Proxy Update**, one proxy table will be automatically created for each remote table or view.
	Durability (available with a **Type** of TEMP or DEFAULT) Relaxed Durability databases are available as of Sybase ASE 15.5.	Lets you select a WITH DURABILITY= parameter value of NO RECOVERY, FULL, or AT SHUTDOWN, specifying the durability level of the database. With **In-memory** selected, the only valid selection is NO RECOVERY.
Options	The GENERATE... DDL is followed by an sp_dboption call for each option selected in this group.	
	With a **Type** of ARCHIVE	**single user** - generates a sp_dboption call specifying the SINGLE USER option, restricting database access to one user at a time.
	With a **Type** of TEMP or DEFAULT	In addition to the **single user** option, the following sp_dboption database options are available: **abort tran on log full**, **allow nulls by default**, **auto identity**, **dbo use only**, **ddl in tran**, **identity in nonunique index**, **no chkpt on recovery**, **no free space acctg**, **read only**, **select into/bulkcopy/pllsort**, **single user**, **trunc log on chkpt**, and **unique auto_identity index**. For Sybase ASE 15.0 databases, **scratch database**, **async log service option**, and **delayed commit** options are available.

Databases (Sybase ASE) - Placement

In the **Fragment Properties** area provide or select values for the default fragment: **Device Name**, **Size** for the fragment, and a **Device Type** value of DATA ONLY, LOG ONLY, or DATA AND LOG. For databases with a **Properties** tab/panel **Type** selection of ARCHIVE, no LOG ONLY **Device Type** selection is available.

If necessary use the **New** button to add a new fragment and repeat the steps above to provide details for that fragment. Use the **Delete** button to drop a fragment.

Data Caches Wizard (DBArtisan - Sybase ASE)

This wizard lets you build and submit an sp_cacheconfig call, creating a new data cache. One or more sp_bindcache calls can also be built and submitted simultaneously, letting you bind databases, tables, indexes, text object, or image object to the new data cache. For each object bound, you can specify sp_cachestrategy parameters, letting you select caching strategies for those objects.

To create a new data cache using a wizard

1 Open an object creation wizard for a data cache. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Data Caches (Sybase ASE) - Properties.
- **Objects** panel - for details, see Data Caches (Sybase ASE) - Objects.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

On creating a data cache, all space is allocated to the logical page size memory pool. You can subsequently use the Data Cache Editor to subdivide the data cache into additional pools with larger I/O sizes. For details, see Data Cache Editor (DBArtisan - Sybase ASE).

Data Caches (Sybase ASE) - Properties

When creating or editing a data cache, this tab/panel lets you work with settings corresponding to sp_cacheconfig parameters:

Category	Setting	Description
Creation	**Name**	Lets you provide a name for the cache.
General	**Type**	Lets you specify a cache type parameter value of MIXED, INMEMORY, or LOGONLY. The In-memory databases feature is available as of Sybase ASE 15.5.
	Policy (not available with a **Type** of INMEMORY)	Lets you specify a replacement policy of STRICT or RELAXED.
	Status	Displays the current status of the cache: ACTIVE, PENDING ACTIVE, or PENDING DELETE. You can only change a mixed data cache if it does not have any non-log objects bound to it.
	Partitions	Lets you specify the number of partitions to be created for the cache. Selecting a legal value (1 - 64, in powers of 2) adds that sp_cacheconfig cache_partition parameter value to the DDL generated to create the data cache.

Size	**Configured Size**	Lets you specify the size of the cache after the server is restarted. Consult Sybase documentation for details on this property. See Accessing Third Party Documentation.
	Run Size	Displays the size that the cache is currently using. For the default cache, this is the amount of all cache space not explicitly configured for another cache

Data Caches (Sybase ASE) - Objects

When creating or editing a data cache, this tab/panel lets you manage the objects bound to the data cache. You use it to bind and unbind objects and to specify the caching strategy for each bound object.

The **Objects** list on the left displays objects currently bound to the data cache. Specific objects are grouped under the **Databases**, **Tables**, and **Index** subfolders.

The **Property/Value** list on the right lets you work with caching strategy parameters for a selected table or index. Consult Sybase documentation for information on the **Apply to text only** (applies to indexes only), **Prefetch Strategy**, and **MRU Replacement Strategy** caching strategies. See Accessing Third Party Documentation.

Maintain the **Objects** list as follows:

- **Adding a new database, table, or index to the bound objects list** - click the **Bind** button. This opens a Bind wizard. For detailed information, see Binding Databases, Indexes, or Tables to a Sybase ASE Data Cache. Use this process repeatedly to bind objects of different types.
- **Enabling or disabling caching strategy options for an object** - select an object from the **Objects** list and use the check box associated with a strategy to enable or disable that strategy.
- **Removing an object from the bound objects list** - select the object in the **Objects** list and click **Unbind**.

Binding Databases, Indexes, or Tables to a Sybase ASE Data Cache

When creating or editing a data cache, the Bind Wizard lets you select the specific objects that are to be bound to the cache.

To bind one or more databases, tables or indexes to a data cache

1 Open the Data Cache Wizard or Data Cache Editor and select the **Objects** tab. For more information, see **Data Caches Wizard (DBArtisan - Sybase ASE)** or Data Cache Editor (DBArtisan - Sybase ASE).

2 Click **Bind** to open the Bind Wizard.

3 From the **Objects** dropdown select **Entire Database**, **Tables**, or **Indexes**.

4 If you selected **Tables**, or **Indexes**, from the **Databases** dropdown select the database containing the specific table or index.

5 If you selected **Tables**, optionally select the **Bind text or image from table objects** check box to bind storage of text and images from the table to this cache.

6 Click **Objects** in the navigation frame to display the **Objects** panel.

7 Select the check box associated with each specific object that is to be bound to the cache.

8 Click **Finish**.

Defaults Wizard (Sybase ASE)

Here you create a default for table column or user-defined datatype in the event that no value is available when the data is inserted. The default value you specify will be inserted only in the current database. You can then bind the default to a specific column or user-datatype.

To create a new default using a wizard:

1 Open a creation wizard for a default. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Defaults (Sybase ASE) - Properties.
- **Dependencies** panel - for details, see Defaults (Sybase ASE) - Dependencies.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Defaults (Sybase ASE) - Properties

When creating or editing a default, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the schema that is to own the default.
Name	Provide a name for the default.
Value	Provide the value of the default.

Defaults (Sybase ASE) - Dependencies

From the **Type** dropdown, choose Column or Datatype, and if you chose **Column**, choose a Table from the **Table** dropdown. The list on the left is populated with candidate columns or datatypes. To move a candidate from the list on the left to the dependencies column on the right, select the candidate and click **Add**. Remove columns or datatypes from the dependencies list on the right by selecting the column or datatype and clicking **Remove**.

Dump Devices Wizard (DBArtisan - Sybase ASE)

Your system writes its memory image to a dump device so you can retrieve data if your system crashes. All load and dump operations require that you name a device before the operation will complete. When you create a new dump device, ASE adds a row to the sysdevices table in the master database. Sybase recommends that you back up the master database before you create a new device and again after you have created the device. Be sure you have enough disk space on the physical drive to perform these operations.

To create a new dump device using a wizard:

1 Open an object creation wizard for a dump device. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Dump Devices (Sybase ASE) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Dump Devices (Sybase ASE) - Properties

When creating or editing a dump device, this tab/panel lets you work with the following settings:

Setting	Description
Name	This is the device's logical name.
Physical Name	The device or file name that is recognized by the operating system. (For example, a:sqltable.dat).
Device Type	Select TAPE or DISK.

Extended Procedures Wizard (Sybase ASE)

Extended stored procedures provide a method for calling external procedural language functions from within the Adaptive Server. A procedural language is a language capable of calling a C language function or manipulating C language datatypes.

To create a new extended procedure using a wizard:

1 Open a creation wizard for an extended procedure. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Extended Procedures (Sybase ASE) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Extended Procedures (Sybase ASE) - Properties

When creating or editing an extended procedure, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the owner of the extended procedure.
Name	Provide the name of the procedure.
Library Name	Provide the name of the library containing the procedure.

Foreign Keys Wizard (Sybase ASE)

Foreign keys are used to relate information from one table to another. Foreign keys are unique values that refer to specific columns of other tables. Thus, a foreign key links two tables together. The Foreign Key Wizard makes it easy for you to create a relational link between two tables, thereby speeding queries and giving you faster access to data. The column in the initial table, the parent table, is called the primary key. The corresponding column in the (child) table that references the primary key, is the foreign key. Foreign keys can also refer to columns within the same table.

To create a new foreign key using a wizard:

1 Open a creation wizard for a foreign key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Foreign Keys (Sybase ASE) - Properties.
- **Column Mapping** panel - for details, see Foreign Keys (Sybase ASE) - Column Mapping.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Foreign Keys (Sybase ASE) - Properties

When creating or editing a foreign key, this tab/panel lets you work with the following settings:

Setting	Description
Table Owner	Select the owner of the referring, or child, table
Table Name	Select the name of the referring, or child, table.
Match Full	Specify a referential integrity option.

Foreign Keys (Sybase ASE) - Column Mapping

Under **Referenced Table**, choose the **Owner** and then the **Name** of the referenced, or parent, table.

Under the **Main Table**, select check boxes corresponding to the columns that are to reference columns in the referenced table. Then, under **Referenced Table**, select the corresponding column check boxes.

Functions Wizard (Sybase ASE)

The Functions Wizard creates and submits a CREATE FUNCTION command. This lets you create a user-defined function, a saved Transact-SQL routine that returns a specified value. The wizard collects basic identifying information and generates basic CREATE FUNCTION statement syntax, letting you provide the body of the function.

Note: This functionality is supported for Sybase ASE 12.5.

To create a new function using a wizard:

1 Open a creation wizard for a function. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties panel** - for details, see Functions (SQL Server) - Properties.
- **Definition panel** - for details, see Functions (SQL Server) - Definition.

3 Finally, use the **Execute** button to create the object.

Functions (Sybase ASE) - Properties

When creating or editing a function, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the owner of the function.
Name	Provide a name for the function.

Functions (Sybase ASE) - Definition

Complete the CREATE FUNCTION outline provided by typing or pasting the body of the function.

Groups Wizard (Sybase ASE)

A group is a collection of privileges that the DBA assigns to designated users.

To create a new group using a wizard:

1 Open a creation wizard for a group. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Groups (Sybase ASE) - Properties.
- **Object Permissions** and **System Permissions** panels - see Setting Permissions or Privileges for an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

After you click **Finish** and **Execute**, use the Group Editor to assign users to the new group. For details, see Groups Editor (Sybase ASE).

Groups (Sybase ASE) - Properties

When creating a group, this panel lets you provide a name for the group.

Indexes Wizard (Sybase ASE)

Like an index in a book, a table index helps you get at the data you want without having to read through the whole table. Indexes can exist on single column or on multiple columns. Indexes appear in the form of B-trees. And, as for books, you can have multiple indexes for a single table.

To create a new index using a wizard:

1 Open a creation wizard for an index. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Indexes (Sybase ASE) - Properties.
- **Columns** panel - for details, see Indexes (Sybase ASE) - Columns.
- **Partition** panel - for details, see Indexes (Sybase ASE) - Partition.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Indexes (Sybase ASE) - Properties

When creating or editing an index, this tab/panel lets you work with the following settings:

Setting	Description
Table Owner, **Table Name**, and **Name**	Select or provide identifying information.
Attributes	Provide or select values for the following Sybase options: **Index Type**, **Clustered**, **Ignore Duplicate Key**, **Ignore Duplicate Rows**, and **Maximum Rows Per Page** properties.
Storage	Provide or select values for the following Sybase options: **Reserve Page Gap**, **Segment Name**, **Fill Factor**, **Prefetch Strategy**, and **MRU Replacement Strategy**.

Indexes (Sybase ASE) - Columns

Select a column from the **Columns** dropdown and specify a **Sort** option.

Use the New button to add more columns to the index. Use the Delete button to remove selected columns from the index.

Indexes (Sybase ASE) - Partition

Click **Add** to open the Index Partition wizard (Sybase ASE).

Index Partition wizard (Sybase ASE)

The Sybase ASE Index Partition wizard can be opened from the following editors and wizards:

- **Indexes Wizard (Sybase ASE)**
- Indexes Editor (Sybase ASE)

Use the following topics as a guide to understanding and setting the options on this wizard:

Step	Settings and tasks
Properties	Select a **Locality**.
Partition Definitions	Click the **Define a new partition** button to open a dialog that lets you provide a name, select a segment, click the **New** button to add values, and then **Add** the partition definition.

When finished, click the **Finish** button.

Logins Wizard (Sybase ASE)

The Login wizard lets you build and submit an **sp_addlogin** or CREATE LOGIN call, adding a new login with a basic set of specified database, language, and password properties. An additional **sp_locklogin** call can be issued to lock the new login. The Login wizard also lets you:

- Grant roles to the login
- Create user accounts on databases for the role, optionally with an alias or by changing ownership

To create a new login using a wizard:

1 Open a creation wizard for a login. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Logins (Sybase ASE) - Properties.
- **Roles** panel - for details, see Logins (Sybase ASE) - Roles.
- **Users** panel - for details, see Logins (Sybase ASE) - Users.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Note: Login triggers, configured for individual logins, are also supported. For details, Logins Editor (Sybase ASE).

Logins (Sybase ASE) - Properties

When creating or editing a login, this tab/panel lets you work with the following settings:

Group	Setting	Description
Creation	**Name**	The user login name.
	Full Name	The full name of the user, an optional but useful identifier.
	Suid (version 15.7 ^)	Lets you optionally add a SUID parameter to the generated DDL, specifying a system user ID.
General	**Default Database**	The database where the user starts each session.
	Default language	The language in which prompts and messages are displayed.
	Profile (version 15.7 ^)	Lets you add a LOGIN PROFILE parameter: **ignore** eliminates profile binding while selecting a login profile (if available) binds the profile to this login.

	Locked or **Account Locked**	If selected, the **sp_addlogin** or CREATE LOGIN call is immediately followed by an **sp_locklogin** call, locking the new login and effectively disabling it.
	NOTE: After creating a login, this group also displays an assigned **Login trigger**. For details, see Logins Editor (Sybase ASE).	
Authentication	**Password**	The initial password for the new user.
	Password expiration	The duration, in days, before the password expires.
	Minimum password length	The minimum number of characters a password can contain.
	Maximum login attempts	The maximum number of login attempt failures before the login is suspended.
	Authentication mechanism	Lets you select ASE, LDAP, PAM, or ANY authentication.
	Exempt Inactive Lock (version 15.7 ^)	When selected, an EXEMPT INACTIVE LOCK TRUE parameter is added to the generated DDL, exempting the login from being locked due to inactivity.

Logins (Sybase ASE) - Roles

When creating or editing a login, this tab/panel lets you grant roles to the login by selecting the the check boxes assocaited with those roles.

For information on creating roles in DBArtisan, see Role Wizard (DBArtisan - Sybase ASE).

For information on activating and deactivating roles for the current login in the current session, see Role Activation.

Logins (Sybase ASE) - Users

When creating or editing a login, this tab/panel lets you maintain database user accounts for a login as follows:

- Add a user account for this login to a database by selecting a database from the **Databases where the Login does NOT have a User Account** and clicking the **Add** button. This opens the **Add User** dialog letting you provide additional details. For more information, see Adding a login account to a database.
- Remove a user account for this login from a database by selecting a database from the **Databases where the Login HAS a User Account** and clicking the **Remove** button.

Adding a login account to a database

When adding a user account for a login to a database, the **Add user** dialog lets you provide details on the method used (**sp_adduser**, **sp_addalias**, or **sp_changedbowner**). Use the following table as a guide to setting options in this dialog:

Setting or Group	Description
User Type	Lets you select the method by which the user account will be added to the database: USER (**sp_adduser**), ALIAS (**sp_addalias**) or DBO (**sp_changedbowner**).
Creation	**Login Name -** displays the user name from the Master database. If you selected a **User Type** of USER, the following settings are available: **User Name -** is a new name for the user in the target database. **Group Name -** lets you select a group in the target database to which the user will be added.
Alias	**LoginUserName -** If you selected a **User Type** of ALIAS, this control lets you select the database user name to alias the login name to.
dbo	**TransferAliasesAndPermissions -** If you selected a **User Type** of DBO, this control lets you transfer aliases and their permissions to the new database owner.

For more information, see Logins Editor (Sybase ASE).

Primary Keys Wizard (Sybase ASE)

Primary key constraints make sure that no duplicate values or NULLS are entered in the columns you specify. You can use primary key constraints to enforce uniqueness and referential integrity. A table can only have a single primary key constraint.

To create a new primary key using a wizard:

1 Open a creation wizard for a primary key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Primary Keys (Sybase ASE) - Properties.
- **Columns** panel - for details, see Primary Keys (Sybase ASE) - Columns.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Primary Keys (Sybase ASE) - Properties

Provide or select the following:

- The **Table Owner** and **Table Name** of the table for which to create the index.
- A **Name** for the Index.
- Whether the index is **Clustered**.
- The **Maximum Rows Per Page**.
- A **Reserve Page Gap**.
- A **Segment Name**.
- A **Fill Factor** value.

Primary Keys (Sybase ASE) - Columns

Use the **Column** dropdown to choose a column name and choose a **Sort** option for that column. Use the New button to add an additional column to the index or use the Drop button to delete selected columns.

Procedures Wizard (Sybase ASE)

Procedures are a reusable block of PL/SQL, stored in the database, that applications can call. Procedures streamline code development, debugging, and maintenance by being reusable. Procedures enhance database security by letting you write procedures granting users execution privileges to tables rather than letting them access tables directly.

To create a new procedure using a wizard:

1 Open a creation wizard for a procedure. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Procedures (Sybase ASE) - Properties.
- **Definition** panel - for details, see Procedures (Sybase ASE) - Definition.

3 Finally, use the **Execute** button to create the object.

Procedures (Sybase ASE) - Properties

When creating or editing a procedure, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the owner of the procedure.
Name	Provide the name of the procedure.
Procedure Number	Provide a procedure number if you want to group identically-named procedures.
Recompile	Optionally, enable this feature to prevent Sybase ASE from saving a plan for the procedure.

Procedures (Sybase ASE) - Definition

Complete the provided CREATE PROCEDURE statement by typing or pasting the body of the procedure.

Remote Servers Wizard (DBArtisan - Sybase ASE)

Remote Servers are servers on a network that you can access through your local server. Instead of logging directly into a remote server, you can log into your local server and execute a remote procedure call to it.

Note: Only stored procedures are allowed against remote servers. You can use a linked server to execute distributed queries and stored procedures.

Before beginning

- Before you can set up a remote server, you first configure both the local and remote servers to accept remote access. To configure the Microsoft SQL Servers for remote access, you need to log into each server. The table below describes the parameters you need to configure on each server:

Parameter	Description
Remote Access	1(enabled)
Remote Connections	Number of remote connections required
Remote Logins	Number of remote logins required
Remote Sites	Number of remote sites required

- After setting these server configuration parameters, shut down and restart the server so the new values can take effect. When you have configured both servers, proceed to the steps below.

To create a new remote server using a wizard:

1 Open an object creation wizard for a remote server. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Remote Servers (Sybase ASE) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Remote Servers (Sybase ASE) - Properties

When creating or editing a remote server, this tab/panel builds an sp_addserver stored procedure call and a set of sp_serveroption calls defining and configuring a remote server. It lets you work with the following settings:

Setting	Description
Name	Provide a name for the remote server.
Network Name	Optionally, provide the server's physical/network name?
Server Type	Select a server type of LOCAL or REMOTE.
Timeout Inactive Connections (only available with a **Server Type** of REMOTE)	Indicate whether SQL server should time out inactive physical connections with this remote server.
Network Password Encryption (only available with a **Server Type** of REMOTE)	Indicate whether encryption should be used to protect passwords sent to this remote server
Security Model (only available with a **Server Type** of REMOTE)	Security Model A: This is equivalent to the level of security in release 11.0.x and prior releases. Security Model B: Implements security using DCE, Kerberos, or NT LAN Manager.
Security Mechanism (only available with a **Server Type** of REMOTE)	Specify a security mechanism
Message Integrity (only available with a **Server Type** of REMOTE and a **Security Model** of SECURITY MODEL B)	If enabled, messages between the servers are checked for tampering.
Message Confidentiality (only available with a **Server Type** of REMOTE and a **Security Model** of SECURITY MODEL B)	If enabled, messages between the local and remote servers are encrypted.
Mutual Authentication (only available with a **Server Type** of REMOTE and a **Security Model** of SECURITY MODEL B)	If enabled, the local server authenticates the remote server by retrieving and verifying the remote server credentials using the security mechanism. The credentials of both servers are authenticated and verified.

Role Wizard (DBArtisan - Sybase ASE)

Roles are sets of user privileges you associate with access to objects within a database. Roles streamline the process of granting permissions. You can use roles to grant sets of permissions and privileges to users and groups. Roles can help you comply with Sarbanes Oxley regulations by limiting which users can have access to what privileges, for example a Human Resources Role versus an Accounting Role.

To create a new role using a wizard:

1 Open an object creation wizard for a role. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Roles (Sybase ASE) - Properties.
- **Object Permissions** and **System Permissions** panels - see Setting Permissions or Privileges for an Object.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

After creating a role, you can use the Roles Editor to assign the role to logons or other roles. For details, see Roles Editor (DBArtisan - Sybase ASE).

For information on activating and deactivating roles for the current login in the current session, see Role Activation.

Roles (Sybase ASE) - Properties

When creating or editing a role, this tab/panel lets you work with the following settings:

Settings	Description
Name	The name of the role?
Password	Enter the password, if any, that must be entered to activate the role (seven characters is the default).
Password Expiration	Enter the time, in days, after which users must change the password
Minimum Password Length	Enter the minimum password length, if any.
Maximum Failed Logins	Enter the minimum number of login attempts, if any, after which the role is locked out.

Rules Wizard (Sybase ASE)

Rules promote data integrity by allowing you to validate the values supplied to a table column. They are reusable objects that you can bind to table columns or user datatypes. Check constraints are similar to rules, and are in fact the preferred way of restricting data. A column or user-defined data type can have only one rule bound to it, but a column can have both a rule and one or more check constraints associated with it. Not that a rule cannot apply to data already existing in the database at the time you're creating the rule and can't be bound to a system-created data type. If you create a new rule when one already exists, the new rule will override the previous one.

To create a new rule using a wizard:

1 Open a creation wizard for a rule. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Rules (Sybase ASE) - Properties.
- **Dependencies** panel - for details, see Rules (Sybase ASE) - Dependencies.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Rules (Sybase ASE) - Properties

When creating or editing a rule, this tab/panel lets you work with the following settings:

Setting	Description
Owner and **Name**	Select an **Owner** and provide a **Name** for the rule.
Restriction	Type the condition. The rule restriction is the condition that defines the rule and can be any expression valid in a WHERE clause and can include such elements as arithmetic operators, relational operators, and predicates (for example, IN, LIKE, BETWEEN).
Type	Choose STANDARD_RULE, AND_ACCESS_RULE, or OR_ACCESS_RULE.

Rules (Sybase ASE) - Dependencies

From the **Type** dropdown, choose Column or Datatype, and if you chose **Column**, choose a Table from the **Table** dropdown. The list on the left is populated with candidate columns or datatypes. To move a candidate from the list on the left to the dependencies column on the right, select the candidate and click **Add**. Remove columns or datatypes from the dependencies list on the right by selecting the column or datatype and clicking **Remove**.

Segment Wizard (Sybase ASE)

This wizard lets you build and submit an `sp_addsegment` procedure call, optionally followed by `sp_dbextend 'set’, 'database'`... and `sp_dbextend 'set', 'threshold'`... calls. This defines a segment on a database device, and specifies the size by which to grow the database, its maximum size, and a free space level.

To create a new segment using a wizard:

1 Opening an Object Wizard for a segment.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Segments (Sybase ASE) - Properties.
- **Location** panel - for details, see Segments (Sybase ASE) - Location.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object. For more information, see Previewing the DDL Generated to Create the New Object.

The Segments Editor opens. You can extend the segment and see the objects stored in the segment. For more information, see Segments Editor (Sybase ASE).

Segments (Sybase ASE) - Properties

When creating or editing a segment, this tab/panel lets you work with the following settings:

Setting	Description
Name	Provide a **Name** for the segment.
Grow By	Provide an `sp_dbextend 'set', 'database'`... growth rate parameter value, specifying the rate at which the segment grows each time threshold procedures are run.
Threshold Free Space	Provide an `sp_dbextend 'set', 'threshold'`... free space parameter value, specifying when the threshold procedure is installed on the segment.
Max Size	Provide an `sp_dbextend 'set', 'database'`... maximum segment size parameter value.

Segments (Sybase ASE) - Location

When creating a segment, this tab/panel lets you select the database device for the segment. Select a device from the **Available Devices** list and click **Place** to include the device in the **Used Devices** list.

Similarly, when editing a segment, this tab/panel lets you extend a segment by adding additional devices. Select a device from the **Available Devices** list and click **Extend** to include the device in the **Used Devices** list.

Tables Wizard (Sybase ASE)

A table is a column-based arrangement of data in which the content of one column has a bearing on the other column(s). So, for example, a table might have a column for authors, another column for the books each author has written, and a third for the number of copies each title by a given author has sold. The data moves across the columns in rows.

Note: You must have CREATE TABLE permissions to generate a new table.

To create a new table using a wizard:

1 Open a creation wizard for a table. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Tables (Sybase ASE) - Properties.
- **Columns** panel - for details, see Tables (Sybase ASE) - Columns.
- **Indexes** panel - for details, see Tables (Sybase ASE) - Indexes.
- **Constraints** panel - for details, see Tables (Sybase ASE) - Constraints.
- **Partitions** panel - for details, see Tables (Sybase ASE) - Partition.
- **Permissions** panel - for details, see Tables (Sybase ASE) - Permissions.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Tables (Sybase ASE) - Properties

When creating or editing a table, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the owner of the table.
Name	Provide a name for the table
Segment Name	Specify the segment on which you want to place the table.
Maximum Rows Per Page	Specifying a number allows you to override the default. The default, 0, creates indexes with full pages and nonclustered indexes with full leaf pages. This number can be changed at any time.
Reserve Page Gap	
Identity Gap	Specify the size of the identity gap for the table.
MRU Replacement Strategy	When enabled, new pages are read into the least recent end of the page chain. When pages reach the most recent end of the chain, the pages are flushed.
Prefetch Strategy	Enabling this feature allows you to fetch as many as eight 2K data pages simultaneously instead of one at a time (the default).
Lock Scheme	Select a locking scheme of ALLPAGES, DATAPAGES, or DATAROWS.
Expected Row Size	If you specified a **Lock Scheme** of DATAROWS or DATAPAGES, provide an expected row size

Tables (Sybase ASE) - Columns

For each column in the table, click the **Add Column** button to create a column, provide a **Name** for the column and provide or select the remaining column attributes.

Use the **Delete** button to drop a selected column.

Use the arrow buttons to reposition the columns.

Note: Because the smalldatetime datatype stores dates and time with less precision than the datetime datatype, before outputting you use the CAST or CONVERT functions to convert any boxes with the smalldatetime datatype to either VARCHAR or datetime datatypes.

Tables (Sybase ASE) - Indexes

Click **Add** to open the Index Wizard. For details, see Indexes Wizard (Sybase ASE).

Tables (Sybase ASE) - Constraints

Selecting a constraint type and clicking **Add** opens the object wizard for that object type. For details see:

- Primary Keys Wizard (Sybase ASE)
- Unique Keys Wizard (Sybase ASE)

- Foreign Keys Wizard (Sybase ASE)
- Create Synonym

Tables (Sybase ASE) - Partition

Click **Add** to open the Table Partition Wizard. For details, see Table Partition wizard for Sybase ASE.

Tables (Sybase ASE) - Permissions

Set up the user permissions for this table.

Table Partition wizard for Sybase ASE

See the following topics for information on editors and wizards that let you open the Sybase ASE Table Partition Wzard:

- **Tables Wizard (Sybase ASE)**
- Tables Editor (Sybase ASE)

Use the following topics as a guide to understanding and setting the options on this wizard:

Step	Settings and tasks
Properties	Select a **Partition Type** of ROUNDROBIN, RANGE, HASH, or LIST
Columns (not available with a **Partition Type** of ROUNDROBIN)	For each column, click the **Insert a new column** button to create a column, provide a **Name** for the column and provide or select the remaining column attributes. Use the **Delete** button to drop a selected column. Use the arrow buttons to reposition the columns.
Partition Definition (only available with a **Partition Type** of HASH or ROUNDROBIN)	Take one of the following actions: (1) Select the **Number Of Partitions** radio box, specify the **Number Of Partitions**, and for each partition, click the **New** button and choose a tablespace from the dropdown, or (2) select the **By Partition Name** radio box and for each partition, click the **New** button provide a name and then choose a tablespace from the dropdown.
Range Definitions (only available with a **Partition Type** of RANGE)	Click the **Define a new partition** button to open a dialog that lets you provide a name, select a segment and **Add** the partition definition.
List Definitions (only available with a **Partition Type** of LIST)	Click the **Define a new partition** button to open a dialog that lets you provide a name, select a segment, click the **New** button to add values, and then **Add** the partition definition.

When finished, click the **Finish** button.

Triggers Wizard (Sybase ASE)

The Triggers Wizard lets you build and submit a CREATE TRIGGER command, selecting a parent object type of table or view, and selecting the triggering events for the trigger. Depending on whether you are creating a trigger for a view or a table, the Wizard generates customizable CREATE TRIGGER syntax as follows:

- **Tables** - The wizard generates a FOR INSERT, UPDATE, DELETE trigger code template (all supported Sybase ASE versions)
- **Views** - The wizard generates an INSTEAD OF trigger code template (Sybase ASE 15.0.2)

In both cases, you provide the SQL statements that are to make up the body of the FOR or INSTEAD OF trigger.

To create a new trigger using a wizard:

1 Open a creation wizard for a trigger. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties panel** - for details, see Triggers (Sybase ASE) - Properties.
- **Definition panel** - for details, see Triggers (Sybase ASE) - Definition.

3 Finally, use the **Execute** button to create the object.

Triggers (Sybase ASE) - Properties

When creating or editing a trigger, this tab/panel lets you work with the following settings:

Setting	Description
Parent Type, **Parent Owner**, and **Parent Name**	Provide or select details on the target table or view.
Owner and **Name**	Select the owner and provide a name for the trigger being created.
Fire On Insert, **Fire On Update**, and **Fire On Delete**	Enable the events that fire the trigger. An INSERT trigger must be associated with an INSERT statement. For example, if a data load operation doesn't include an INSERT statement, the trigger won't be invoked. An UPDATE trigger can be associated with specific columns of the base table and will only be activated if those columns are updated. A **DELETE** trigger fires automatically after items in the table are deleted.

Note: For triggers with a table parent type, the editor lets you enable and disable the trigger, after creation. For details, see Triggers Editor (Sybase ASE).

Triggers (Sybase ASE) - Definition

Complete the CREATE TRIGGER statement by typing or pasting in content.

  - See the Microsoft SQL Server Transact-SQL documentation for information on the syntax for Trigger bodies. For more information, see Accessing Third Party Documentation.

Unique Keys Wizard (Sybase ASE)

Unique keys can enforce logical keys that are not chosen as the primary key. In other words, you can use a unique key to ensure no duplicate values are entered in specific columns that are not a part of the primary key. Although you can only attach one primary key to a table, you can attach multiple unique keys. Also, you can use unique keys on columns that allow null values.

To create a new unique key using a wizard:

1 Open a creation wizard for a unique key. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Unique Keys (Sybase ASE) - Properties.
- **Columns** panel - for details, see Unique Keys (Sybase ASE) - Columns.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

Important Notes

- If you are creating a non-clustered index constraint, you should place it on a separate segment from the target table.

Unique Keys (Sybase ASE) - Properties

Provide or select the following:

- The **Table Owner** and **Table Name** of the table for which to create the index.
- A **Name** for the Index.
- Whether the index is **Clustered**.
- The **Maximum Rows Per Page**.
- A **Reserve Page Gap**.
- A **Segment Name**.
- A **Fill Factor** value.

Unique Keys (Sybase ASE) - Columns

Use the **Column** dropdown to choose a column name and choose a **Sort** option for that column. Use the New button to add an additional column to the index or use the Drop button to delete selected columns.

User Datatypes Wizard (Sybase ASE)

User datatypes promote domain consistency by streamlining the definition of commonly used table columns in a database. You can build a customized datatype from system datatypes and bind defaults and rules to it to enhance integrity. When you reference the user datatype in a column, the column assumes all of the properties of the user datatype.

To create a new user datatype using a wizard:

1 Open a creation wizard for a user datatype. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see User Datatypes (Sybase ASE) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

User Datatypes (Sybase ASE) - Properties

When creating or editing a user datatype, this tab/panel lets you work with the following settings:

Setting	Description
Datatype	Provide a name for the datatype.
Type	Select the base datatype.
Size	Provide the size of the datatype.
Allow Nulls	Null has no explicitly assigned value. Null is not equivalent to zero or blank. A value of null is not considered to be greater than, less than, or equivalent to any other value, including another value of null.
Identity	This allows you to specify whether or not you want to limit access to the new datatype based on the user’s privileges.
Default Binding	Defaults promote data integrity by supplying a default value to a column if the user does not explicitly provide one. They are reusable objects that you can bind to user datatypes.
Rule Binding	Rules promote data integrity by allowing you to validate the values supplied to a column. They are reusable objects that you can bind to user datatypes.

User Messages Wizard (Sybase ASE)

A user message let's you write the error message users will see when a user-defined event transpires. The User Message Wizard lets you multiple language versions of a user message, specifying the message numbers, specifying the text for each language version, and binding the message to one or more integrity constraints.

To create a new user message using a wizard:

1 Open a creation wizard for a user message. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see User Messages (Sybase ASE) - Properties.
- **Information** panel - for details, see User Messages (Sybase ASE) - Information.
- **Bindings** panel - for details, see User Messages (Sybase ASE) - Bindings.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

User Messages (Sybase ASE) - Properties

When creating or editing a user message, this tab/panel lets you provide the message number parameter for the sp_addmessage system procedure calls that will create each language-based version of this user message.

To specify the message number for this user message

In the **Message Number** box, type a unique message number (20000 or greater).

User Messages (Sybase ASE) - Information

For each language in which the user message is to be available, the User Message Wizard/Editor builds a sp_addmessage system procedure call. This tab/panel manages the language and message text parameters for each sp_addmessage call.

To add a language version of the user message

1 Click the **Add new text** for the user message button. The **Create User Message Text** dialog opens.

2 From the **Language** dropdown, select the language for the message.

3 In the **Message Text** box, type the text of the message.

4 Click **OK**.

To edit an existing language version of the message

1 Select the version from the **Language/Message** list.

2 Click the **Modify user message text** button. The **Edit User Message Text** dialog opens.

3 In the **Message Text** box, replace the text of the message.

4 Click **OK**.

To delete an existing language version of the message

1 Select the version from the **Language/Message** list.

2 Click the **Remove user message text** button and verify when prompted.

User Messages (Sybase ASE) - Bindings

When creating or editing a user message, this tab/panel manages one or more sp_bindmsg system procedure calls that will be issued with the sp_addmessage calls that create or edit a user message. This lets you bind the user message to, or unbind the user message from foreign keys or check constraints.

To bind a user message to an integrity constraint

1 From the **Constraint Type** dropdown, select the type of integrity constraint (**Check Constraints** or **Foreign Keys**) to which you are binding the user message.

2 Select one or more check constraints or foreign keys from the list on the left and click **Bind**.

To unbind a user message from a currently bound integrity constraint

1 Select one or more check constraints or foreign keys from the list on the right and click **Unbind**.

Users Wizard (Sybase ASE)

The User Wizard lets you create a user who will then have access to the database where you are registering him or her. You can also identify the appropriate user group and the system privileges you want to assign to the new user.

To create a new user using a wizard:

1 Open a creation wizard for a user. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Users (Sybase ASE) - Properties.
- **DDL View** panel - for details, see Previewing the DDL Generated to Create the New Object.

3 Finally, use the **Execute** button to create the object.

You can use the User Editor to assign System and Object privileges as well as create an alias.

Users (Sybase ASE) - Properties

When creating or editing a user, this tab/panel lets you work with the following settings:

Setting	Description
Login Name	Choose a login ID from the drop-down list.
Name	Provide the user name.
Group Name	Specify a group for the user.

Views Wizard (Sybase ASE)

Views are SQL queries stored in the system catalog that customize the display of data contained in one or more tables. Views behave like tables because you can query views and perform data manipulation operations on them. However, views do not actually store any data. Instead, they depend on data contained in their base tables. Views can only be

To create a new view using a wizard:

1 Open a creation wizard for a view. For details, see Opening an Object Wizard.

2 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Properties** panel - for details, see Views (Sybase ASE) - Properties.
- **Definition** panel - for details, see Views (Sybase ASE) - Definition.

3 Finally, use the **Execute** button to create the object.

You can use the User Editor to assign System and Object privileges as well as create an alias.

Views (Sybase ASE) - Properties

When creating or editing a view, this tab/panel lets you work with the following settings:

Setting	Description
Owner	Select the owner of the view.
Name	Provide a name for the view.
Check Type	When enabled, when a row is modified through a view, this option makes sure the data remains visible through the view after the modification is committed.

Views (Sybase ASE) - Definition

Complete the CREATE VIEW statement by typing or pasting in the relevant query.

Modifying objects using editors

An Object editor is a tabbed dialog box that stores information about existing server and database types and objects, and lets you modify those items. For an introduction to object editors, see Overview and common usage of object editors.

Note: Availability of editors for specific object type varies from DBMS to DBMS. Similarly, functionality offered for an editor for an object type common to two or more DBMS, will differ from DBMS to DBMS.

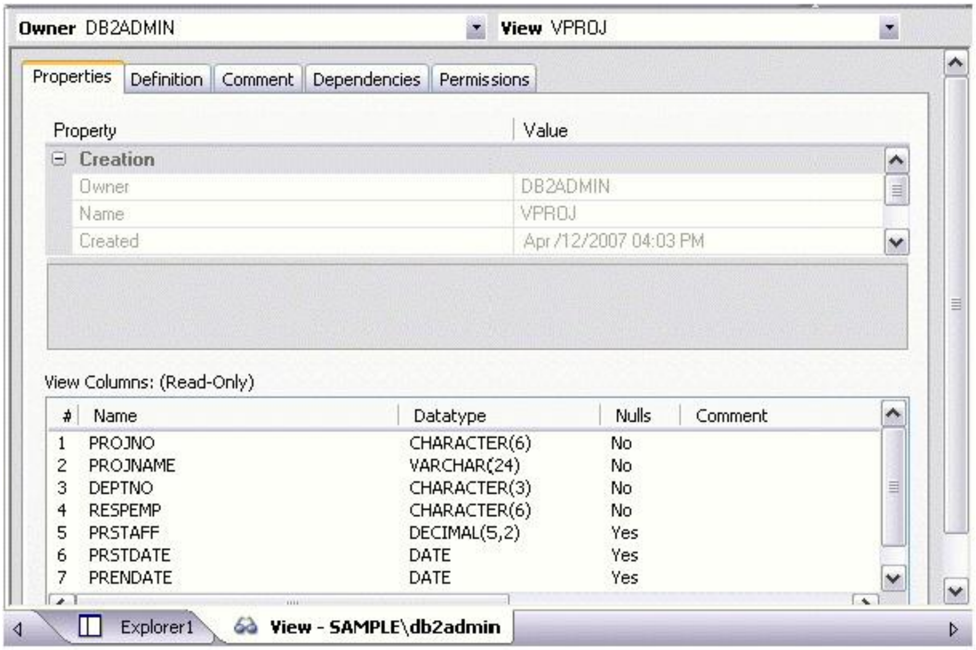
See the following topics for information the object editors available for each supported platform:

- IBM DB2 for Linux, Unix, and Windows Object Editors
- IBM DB2 for z/OS Object Editors
- Microsoft SQL Server Object Editors
- MySQL Object Editors
- Oracle Object Editors
- PostgreSQL Object Editors
- Sybase ASE Object Editors

Overview and common usage of object editors

An Object editor lets you view and modify settings and properties of existing object types and servers on a datasource. It also lets you add new resources and provides access to related, datasource management facilities. For example, the Tables editor lets you add or insert, edit, or drop columns, work with permissions to work with that table, access information on physical storage and the distribution of data and indexes across table spaces, and so on.

Each tab on the Object editor lets you perform a logical task or collection of logical tasks for that object type. The Object editor toolbar has a set of commands common to all object types and also includes a **Command** menu with commands specific to the object type you are currently working with. For example:



Owner DB2ADMIN **View** VPROJ

Property	Value
Creation	
Owner	DB2ADMIN
Name	VPROJ
Created	Apr /12/2007 04:03 PM

View Columns: (Read-Only)

#	Name	Datatype	Nulls	Comment
1	PROJNO	CHARACTER(6)	No	
2	PROJNAME	VARCHAR(24)	No	
3	DEPTNO	CHARACTER(3)	No	
4	RESPEMP	CHARACTER(6)	No	
5	PRSTAFF	DECIMAL(5,2)	Yes	
6	PRSTDATE	DATE	Yes	
7	PRENDATE	DATE	Yes	

In order to work with object editors, you must be familiar with the following tasks:

- Opening an Object Editor on a server or database object.
- Viewing and Modifying Object Properties using functionality common to all object editors as well as object-specific functionality available to specific object editors.
- Previewing and Submitting Object Editor Changes to effect your changes to the datasource.

Opening an Object Editor

Object editors are accessed from the Datasource Navigator/Explorer pane of the main window.

To open an Object editor on a specific database object in DBArtisan

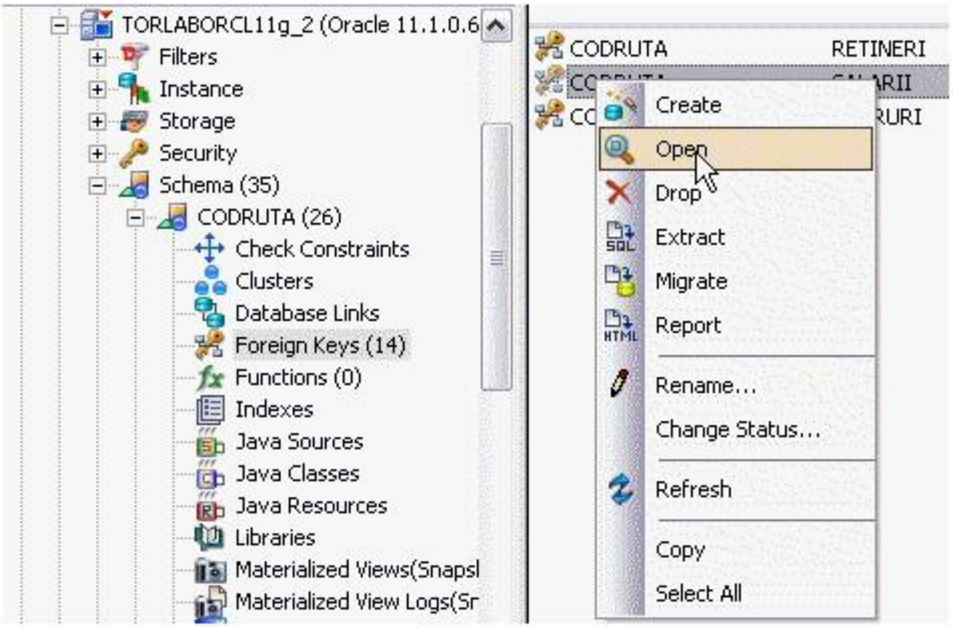
1 Connect to the datasource with the database object to be edited. For more information, see Datasource and Server Management.

2 On the Datasource Navigator/Explorer, expand the target datasource.

3 Continue to expand folders under the datasource until the type of object you want to view or modify is visible.

4 On the Datasource Navigator/Explorer, click on the specific type of object that you want to view or modify.

All objects of that type on the datasource are displayed.



5 In the Navigator/Explorer, right-click the specific object you want to view or modify, and then select **Open**.

An editor for that object type opens on the object you selected.

For information on editing objects, see Viewing and Modifying Object Properties.

Viewing and Modifying Object Properties

There are two categories of tasks you can perform using an Object editor:

- Work with Tasks and Settings of Specific Object Types
- Work with Tasks and Settings Common to Object Editors

Work with Tasks and Settings of Specific Object Types

Many of the tasks you perform using an Object editor are specific to the object type you are working with. For example, the Triggers editor lets you work directly with SQL code for the trigger while the Tables editor lets you work with columns of the table. Object-specific tasks are documented in the topics for specific Object editors provided later in this chapter.

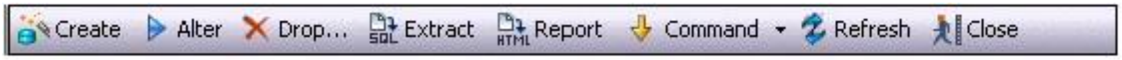
Work with Tasks and Settings Common to Object Editors

The following tasks can be performed against many of the supported Object editors:

- Using the object editor toolbar
- Opening Another Object in the Same Object Editor Explorer
- Viewing the SQL/DDL for an Object
- Working with Privileges and Permissions
- Working with Object Dependencies
- Adding a Comment to an object

Using the object editor toolbar

The Object editor toolbar appears above the tabs of the Object editor.



It provides the following functions:

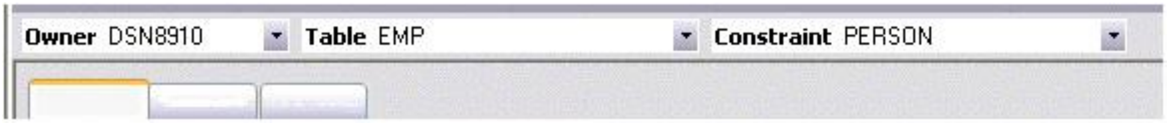
- **Create** - Launches an object creation wizard. For more information, see Creating objects.
- **Alter** - Enabled when a property of the current database has been modified. Opens a dialog that lets you preview the SQL code that will effect the modification and then submit the change. For more information, see Preview.
- **Drop** -Lets you drop one or more database objects and remove their definition from the system catalog. For more information, see Drop.
- **Extract** - Lets you extract data from one database to another database and extract the statements required to create objects into an Interactive SQL window. For more information, see Extract.

- **Migrate (DBArtisan)** - Lets you move object schema between DBMS or servers by copying the DDL between the source and target databases. For details, see Migrate (DBArtisan).
- **Report** - Lets you generate detailed or summary reports on database objects. For details, see Report.
- **Command Menu** - Provides menu commands specific to the object being viewed or modified. For a listing of commands available, see the topic for the specific Object editor, later in this chapter.
- **Refresh** - Refreshes or clears Object editor contents.
- **Close** - Closes the Object editor and if appropriate, prompts you to verify any changes.

Commands on the Object editor toolbar are disabled if they do not apply to the object type being viewed or modified.

Opening Another Object in the Same Object Editor Explorer

When an object editor is open, the area between the object editor toolbar and the tabs has one or more dropdown lists. For example:



Owner DSN8910
Table EMP
Constraint PERSON

These lists allow you to qualify any object of the same type as the object currently displayed in the Object editor, and display information for that object in the current Object editor. The number of, and specific dropdown lists, differ according to the type of Object editor but there are always sufficient controls to uniquely identify another object of the same type.

To display the information for another object of the same type

- Use the dropdown lists to specify a different object.

Viewing the SQL/DDL for an Object

The object editors for most object types feature a **DDL** tab that lets you view an object's underlying SQL code.

To view the underlying DDL/SQL for a database object

1 Open an editor on an object type for which DDL code can be displayed. For details, see Opening an Object Editor.

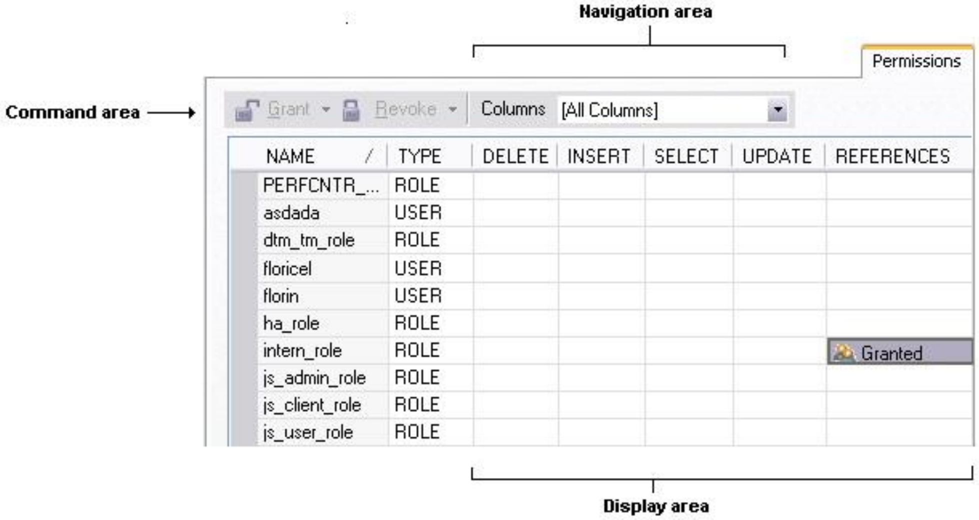
See the topics for specific Object editors later in this chapter for information on whether that object type supports display of underlying SQL code.

2 Click the **DDL** tab.

Working with Privileges and Permissions

When you open an Object editor on an object with associated privileges, the **Permissions** (or **Object Permissions** or **System Permissions**) tab for that editor displays the relevant privileges and lets you make changes accordingly:

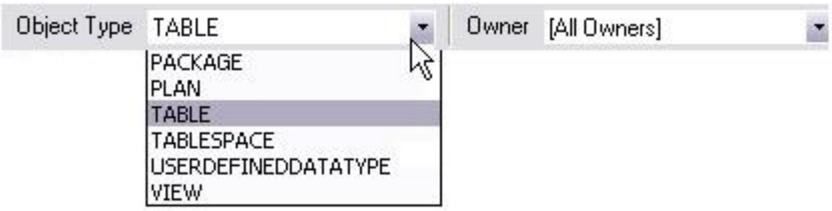
For database objects such as tables or procedures, to which permissions or privileges are granted, a **Permissions** tab lets you manage permissions or privileges to that object from all grantees, such as users or groups, on that datasource. The tab has a navigation area, a command area, and a display area.



Columns [All Columns]

NAME	TYPE	DELETE	INSERT	SELECT	UPDATE	REFERENCES
PERFCNTR_...	ROLE					
asdada	USER					
dtm_tm_role	ROLE					
floricel	USER					
florin	USER					
ha_role	ROLE					
intern_role	ROLE					Granted
js_admin_role	ROLE					
js_client_role	ROLE					
js_user_role	ROLE					

The Navigation area lets you change the content of the Display area to view more specific privilege details. For example, when viewing privileges for a database object such as a table, the dropdown lists in the navigation area, if present, let you drill down to populate the display area with privileges for a lower level component, such as a particular column. Similarly, when viewing privileges for a grantee such as a user, the **Object Permissions** tab's Navigation area lets you populate the display area with that user's permissions on specific object types, such as tables or procedures.



Object Type TABLE

Owner [All Owners]

The Display area shows privilege details for an object or grantee. When viewing privileges for a grantee such as a user or role, the display area shows privileges for that recipient on objects currently qualified by the current selection in the Navigation area. When viewing privileges for an object such as a table, the display area shows the privileges for all grantees. Each cell in the Display area corresponds to a specific permissions and a cell representing a granted permission shows a distinctive icon.

The Command area lets you initiate granting or revocation of permissions.

To view or modify privileges using an Object editor

1 Open an editor on a database object with associated privileges or permissions. For details, see Opening an Object Editor.

2 Click the **Permissions** (or **Privileges** or **Object Privileges** or **System Privileges**) tab.

3 Use the dropdown lists in the navigation area, if present, to populate the Display area with details for more specific or different permissions.

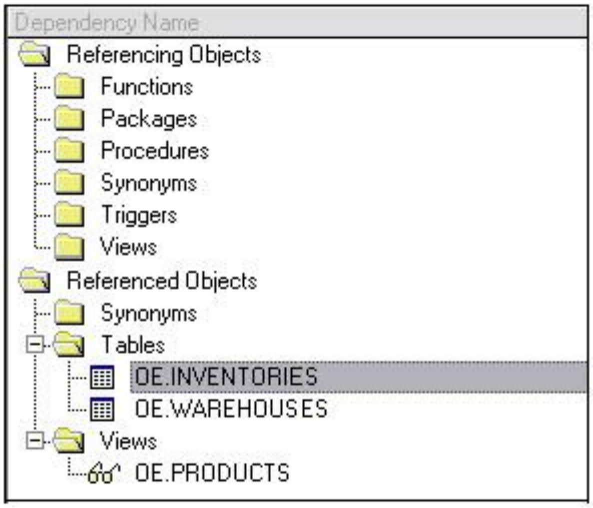
4 In the Display area, select a cell corresponding to a specific permission.

5 Use the Command area controls to grant or revoke permissions.

6 Submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Working with Object Dependencies

For objects such as views and procedures, whose definition references other objects, such as tables, you can view all dependencies. The Object editors for referencing or referenced objects have a **Dependencies** tab that shows all dependencies for an object.



In addition to letting you view dependencies, Object editors let you drop a referencing or referenced object, or open an Object editor on that object.

To manage dependences for an object

1 Open an editor on a database object with referencing or referenced objects. For details, see Opening an Object Editor.

See the object editor descriptions in subsequent topics to verify that an Object editor supports display of dependencies.

2 Click the **Dependencies** tab.

The **Dependency Name** area lists all referenced or referencing objects. Objects are grouped according to their object type.

3 Optionally, select a referenced or referencing object in the right pane and either:

- Use the **Edit** button to open an Object editor on that object
- Use the **Drop** button to initiate dropping that object.

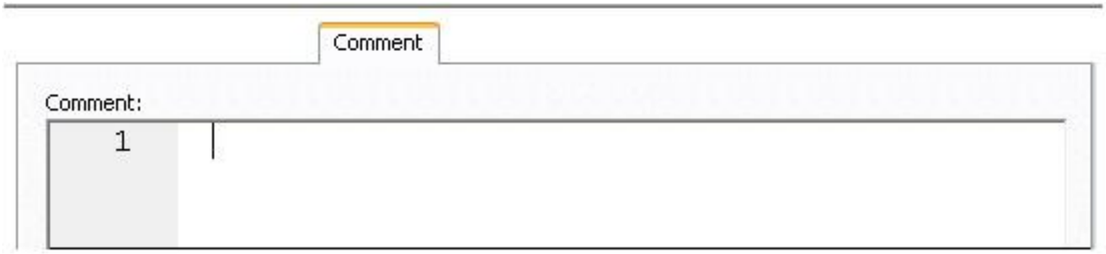
Adding a Comment to an object

The object editors for certain object types feature a **Comment** tab that lets you add an explanatory note to specific object definitions. Comments are stored in the REMARKS column of the object's system-catalog.

To add a comment to an object

1 Open an editor on an object type that permits comments. For details, see Opening an Object Editor.

2 Click the **Comment** tab.



3 In the **Comment** area, type an explanatory note of up to 254 characters long.

Note: Right-clicking in the **Comment** area opens a context menu that lets you perform edit text operations such as search, selection, copying and pasting, as well as coding-specific operations such as enabling/disabling line numbers and indenting lines.

4 Submit your changes. For details, see Previewing and Submitting Object Editor Changes.

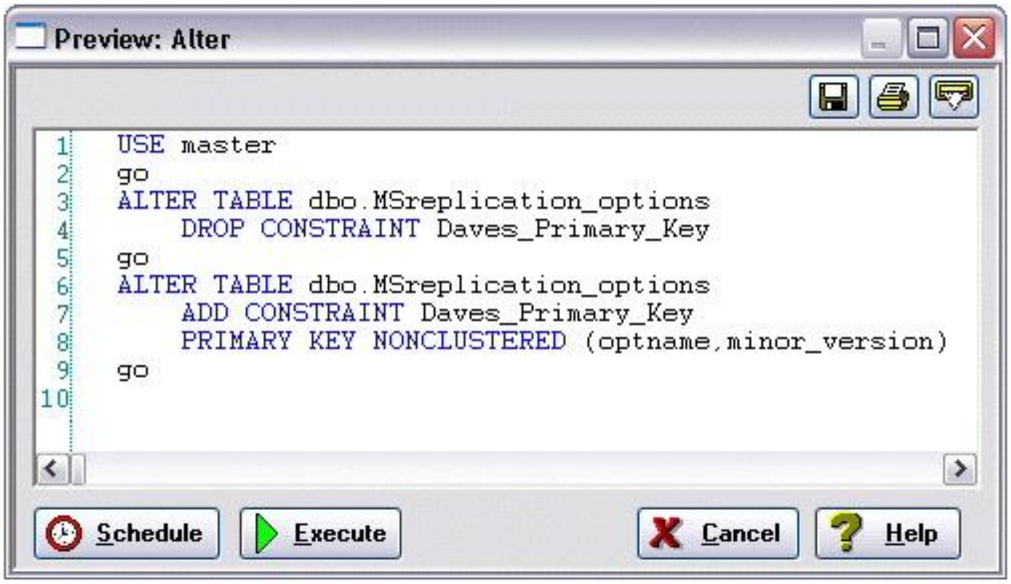
Previewing and Submitting Object Editor Changes

After you use an Object editor to modify the settings or properties of a database object, you can preview the SQL that will be executed to effect those changes on the datasource. You can then submit the SQL for execution on the server.

To preview and submit object editor changes to a database object

1 Click the **Alter** button on the Object editor toolbar. For details, see Overview and common usage of object editors.

A **Preview: Alter** dialog opens.



```
USE master
go
ALTER TABLE dbo.MSreplication_options
    DROP CONSTRAINT Daves_Primary_Key
go
ALTER TABLE dbo.MSreplication_options
    ADD CONSTRAINT Daves_Primary_Key
    PRIMARY KEY NONCLUSTERED (optname,minor_version)
go
```

2 Use one of the following options to submit and execute your changes:

- **Execute** button - executes the changes immediately.
- **Schedule** button - opens a dialog that lets you schedule the change. For more information, see Scheduling.
- **Batch** button - (IBM DB2 for z/OS only) lets you specify the job cards used to execute the changes as a batch job. For information on how to check status, see Database Monitor (DBArtisan).

IBM DB2 for Linux, Unix, and Windows Object Editors

An Object Editor is offered for all supported IBM DB2 or Linux, Unix, and Windows objects. To see an Editor for a specific object, click the corresponding link below:

- Aliases Editor (IBM DB2 LUW)
- Bufferpool Editor (DBArtisan - IBM DB2 LUW)
- Check Constraints Editor (IBM DB2 LUW)
- Databases Editor (IBM DB2 LUW)
- Event Monitor Editor (DBArtisan - IBM DB2 LUW)
- Foreign Keys Editor (IBM DB2 LUW)
- Functions Editor (IBM DB2 LUW)
- Groups Editor (DBArtisan - IBM DB2 LUW)
- Indexes Editor (IBM DB2 LUW)
- Materialized Query Tables Editor (IBM DB2 LUW)
- Packages Editor (IBM DB2 LUW)
- Nodegroups Editor (DBArtisan - IBM DB2 LUW)
- Primary Keys Editor (IBM DB2 LUW)
- Procedures Editor (IBM DB2 LUW)
- Sequences Editor (IBM DB2 LUW)
- Structured Types Editor (IBM DB2 LUW)
- Tables Editor (IBM DB2 LUW)
- Tablespaces Editor (IBM DB2 LUW)
- Triggers Editor (IBM DB2 LUW)
- Unique Keys Editor (IBM DB2 LUW)
- User Datatypes Editor (IBM DB2 LUW)
- Users Editor (IBM DB2 LUW)
- Views Editor (IBM DB2 LUW)

For access to object editors for different supported DBMS platforms and an introduction to editor usage, see Modifying objects using editors.

Aliases Editor (IBM DB2 LUW)

The Aliases Editor lets you view basic alias properties, change the comment associated with an alias, and view the DDL that will be issued to alter the alias definition.

To edit an alias

1 Open an editor on the alias. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	In addition to the owning **Schema**, this tab lets you view the **Target Owner**, **Target Type**, and **Target Name**, of the referenced object.
Comment	For details on using this tab, see Adding a Comment to an object.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Bufferpool Editor (DBArtisan - IBM DB2 LUW)

The Bufferpool editor lets you manage the basic properties, nodegroups, and partitions size of a bufferpool.

To edit a bufferpool

1 Open an editor on the bufferpool. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	**Deferred** - controls IMMEDIATE or DEFERRED update of the bufferpool. **Page size** - displays the current page size. **Automatic Size** and **Number of Pages** - Lets you specify the size of the bufferpool. You can use the **Automatic Size** option to enable resizing based on self-tuning or you can specify a fixed **Number of Pages**. **No. of Block Pages** and **Block Size** - Let you specify NUMBLOCKPAGES and BLOCKSIZE values.
Nodegroups	Displays all defined nodegroups and lets you associate nodegroups with this bufferpool by selecting the relevant check boxes.
Partitions	Displays all partitions for the bufferpool. Optionally you can click in the **Size** field of an entry to activate editing and provide a new size.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Check Constraints Editor (IBM DB2 LUW)

The Check Constraints Editor lets you modify and enable/disable check constraints.

To edit a check constraint:

1 Open an editor on the check constraint. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Definition	Use the **Enabled** control to enable/disable the check constraint. You can also edit the condition in the **Check Condition** box. The **Table Column** button opens a dialog that lets you select and paste column names into the check condition expression.
Comment	For details on using this tab, see Adding a Comment to an object.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Databases Editor (IBM DB2 LUW)

The Databases Editor lets you manage database tablespace placement details and modify configuration parameters for the database.

To edit a database

1 Open an editor on the database. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Placement	Select a tablespace and click the **Edit** button to open an object editor on that tablespace. For more information, see Tablespaces Editor (IBM DB2 LUW).
Options	Lets you modify the DB2 Database Manager configuration file parameters for the database. Select a parameter and click the **Edit** button to open a dialog that lets you change the current value. **NOTE:** To set database options for all future databases, set the database options on the model database.
DDL	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Event Monitor Editor (DBArtisan - IBM DB2 LUW)

The event monitor editor lets you manage the event types associated with an event monitor, modify the condition for each event type, and specify where and how records are written.

To edit an event monitor

1 Open an editor on the event monitor. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you work with the following settings: **Auto Start** - lets you specify whether the event monitor is activated each time the database starts. **Target Type** - lets you specify where the output is written: FILE, PIPE, or TABLE. If you specify a **Target Type** of TABLE, you also have options to provide a **Buffer SIze** and specify whether the event monitor is to be **Blocked**. If you specify a **Target Type** of PIPE, you must also specify the **Target** named pipe identifier. If you specify a **Target Type** of FILE, you also have options to provide a **Buffer SIze** and whether the event monitor is to be **Blocked**. You also specify a **Write Mode** of APPEND or REPLACE, provide a **Max Files** value and a **Max File Size** value.
Definition	Lets you modify the **Event Types** used to construct the CREATE EVENT MONITOR statement FOR clause. The base set of event types is BUFFERPOOLS, CONNECTIONS, DATABASE, DEADLOCKS, DEADLOCK WITH DETAILS, STATEMENTS, TABLES, TABLESPACES, and TRANSACTIONS. For IBM DB2 for Linux, Unix, and Windows version 9.0, DEADLOCK WITH DETAILS HISTORY and DEADLOCKS WITH DETAILS HISTORY VALUES event types are available. For IBM DB2 for Linux, Unix, and Windows version 9.5, ACTIVITIES, THRESHOLD VIOLATIONS, and STATISTICS event types are available. NOTE: If your choices are all from the DEADLOCKS... types, you can also specify **Deadlock Monitoring** of **Global** or **Local**. If you select Local, specify the **Database Partitions** the event monitor is to run against.
Conditions	Available only for CONNECTION, STATEMENT, and TRANSACTION **Event Type**s, this tab lets you modify a condition used to determine which CONNECTION, STATEMENT, or TRANSACTION the event monitor will collect data for.
DDL	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Foreign Keys Editor (IBM DB2 LUW)

The Foreign Keys Editor lets you manage column mapping for a foreign key, modify update and delete rule actions, and specify a NOT ENFORCED value.

To edit a foreign key

1 Open an editor on the foreign key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Column Mapping	The existing column mapping for the foreign key is represented by selected columns in the **Main Table** and **Referenced Table** lists. Additional candidates in the **Referenced Table** list are indicated by enabled column check boxes. If necessary, use the **Table** dropdown in the **Referenced Table** group to choose a new table for this foreign key. Select or deselect columns in the **Main Table** list and **Referenced Table** list to form the referential constraint between the two tables.
Properties	Lets you modify the **Delete Rule** (NO ACTION, RESTRICT, CASCADE, SET NULL) and **Update Rule** (NO ACTION, RESTRICT) actions. This tab also lets you deselect the **Enforced** check box to specify a NOT ENFORCED option.
Comment	For details on using this tab, see Adding a Comment to an object.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Functions Editor (IBM DB2 LUW)

The Functions Editor lets you manage the body, inputs and outputs, and properties for a function.

To edit a function

1 Open an editor on the function. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Displays the **Language**, **Return Type**, **External Name**, **SQL Access Level**, **Origin**, **Threadsafe**, **Fenced**, **Scratchpad**, **Scratchpad Length**, **Allow Parallel**, **Final Call**, **Parameter Style**, **Inherit Special Registers**, **DBInfo**, **Deterministic**, **External Action**, **Called on Null Input**, and **Parameter CCSID** properties.
Parameters	Lets you manage function parameters. On opening, this tab shows the existing parameters. Optionally you can: Select a parameter from the list, modify the **Type** and if appropriate, the **Precision**, **Size**, and **Scale** of the parameter. Click the **New** button, provide a name for the new parameter and modify its attributes. Select a parameter and click the **Drop** button to delete the parameter. Select a parameter and use the arrow buttons to reorder the parameter list.
Return Scalar (only available for functions created with a **Function Type** of EXTERNAL SCALAR, SOURCED or TEMPLATE and for SQL **Function Type** functions with an explicit **Return Type** of SCALAR)	Lets you change the **Type** of a return scalar and depending on the type specified, modify the **Precision**, **Scale**, and **Size** of the return scalar.
Return Columns (only available for functions created with a **Function Type** of OLEDB and EXTERNAL TABLE and for SQL **Function Type** functions with an explicit **Return Type** of TABLE or ROW)	Lets you manage return columns for a function. On opening, this tab shows the existing columns. Optionally you can: Select a column from the list, modify the **Type** and if appropriate, the **Precision**, **Size**, and **Scale** of the parameter. You can also specify that the return column is to be treated **As Locator**. Click the **New** button, provide a name for the new column and modify its attributes. Select a parameter and click the **Drop** button to delete the parameter.

	Select a parameter and use the arrow buttons to reorder the parameter list.
Body	Lets you edit the body of the function.
Comment	For details on using this tab, see Adding a Comment to an object.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Permissions	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Groups Editor (DBArtisan - IBM DB2 LUW)

The group editor lets you manage privileges for a group.

To edit a group

1 Open an editor on the group. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Object Permissions and **System Permissions**	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Indexes Editor (IBM DB2 LUW)

The Index Editor lets you manage columns and properties, and view space details for an index.

Note: IBM DB2 for Windows, Unix, and Linux, lets you segregate *Include columns*; columns that are to be part of the index but not part of the unique key.

To edit an index

1 Open an editor on the index. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Columns	Lets you manage columns that make up the index. On opening, this tab shows the existing columns. Optionally you can:	
		Change the **Sort** order of a column.
		Click the **New** button and select a column name from the dropdown, to add a column to the index.
		Select a column and click the **Drop** button to delete the column from the index.
		Select a column and use the arrow buttons to reorder the columns in the index.
Properties	Lets you work with properties in the following categories:	
	Attributes	Lets you view the **Defined By** property. Lets you set the **Index Type** (UNIQUE or NONUNIQUE), specify the index as **Clustered**, **Allow Reverse Scans**, and **Compression**.
	Storage	Lets you set **Percent Free** and **Minimum Percent Used** settings.
Include Columns	Lets you Include columns that are to be part of the index but not part of the unique key. On opening, the tab shows a listing of the Include columns currently defined as part of this index. Optionally, take one of the following actions:	
		Click the **New** button to add a new column to the index.
		Select an existing column and click the **Delete** button to delete that column.
Space	Lets you view the following property groups:	
	Attributes	**Make Unique**, **System Required**, **Total Keys**, **Page Fetch Pairs**, and **Distinct Keys**
	Statistics	**Index Level**, **Cluster Ratio**, **Cluster Factor**,**Leaf Pages**, **Sequential Pages**, and **Density**
	Cards	**First Key,First 2 Keys, First3 Keys, First 4 Keys,** and **Full Keys**

Comment	For details on using this tab, see Adding a Comment to an object.
Permissions	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Materialized Query Tables Editor (IBM DB2 LUW)

The Materialized Query Tables Editor lets you manage the columns, base query and options for a materialized query table, and work with storage and performance settings.

To edit a materialized query table

1 Open an editor on the materialized query table. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Columns	Displays the currently defined columns for the materialized Query Table. Optionally, you can **Add**, **Insert**, or **Edit** columns for the materialized query table.	
Definition	**Base Query**	Lets you view the query that this materialized query table is based on.
	Materialized Query Table Options	Lets you specify an immediate or deferred **Refresh**, and explicitly set a Materialized Query Table as **Enabled** or **Disabled** or go with the **Defaul**t. This tab also displays whether the Materialized Query Table is maintained by the system or by a user.
Performance	**Tablespace Placement**	Displays **Table Tablespace**, **Index Tablespace**, and **Long Tablespace** settings for the materialized query table.
	Log Options	Lets you select logging options: **Logged initiallyData Capture -** Lets you select none or change to **Include Longvar Columns**. NOTE: If you select the **Data Capture** option, the table name / column name cannot be longer than 18 bytes.
	Options	Let you set the **Percent Free**, **Locksize**, **Append**, and **Volatile** settings for the materialized query table.
	Partition Columns	Use the **Add** and **Delete** buttons to manage the partition columns for the materialized query table.
Space	**Page Information**	Lets you view **Row Count** and **Num. of Overflow Rows** values.
	Row Information	Lets you view **Num. of Pages with Rows**, **Num. of Pages**, **Percent Free** values.
Comment	For details on using this tab, see Adding a Comment to an object.	
Permissions	For details on using this tab, see Working with Privileges and Permissions.	
Dependencies	For details on using this tab, see Working with Object Dependencies.	
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Nodegroups Editor (DBArtisan - IBM DB2 LUW)

The nodegroup editor lets you add partitions to, and delete partitions from the nodegroup.

To edit a nodegroup

1 Open an editor on the nodegroup. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Partitions	Use the **Join Nodegroup** and **Leave Nodegroups** buttons to move partitions between the **Partitions Not in Nodegroup** and **Partitions in Nodegroup** lists.
Comment	For details on using this tab, see Adding a Comment to an object.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Packages Editor (IBM DB2 LUW)

The Packages Editor lets you view contents and settings for a package.

To edit a package

1 Open an editor on the package. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Definition	Lets you view the following settings: **Binder**, **Definer**, **Default Schema**, **Degree**, **Function Path**, **Language Level**, **SQL Math Warn**, **SQL Warn**, **Buffered Inset**, **Status**, **Code Page**, **Total Sections**, **Multi-node Bound**, **Intra-partition**, **Query Optimization**, **Cursor Blocking**, **Isolation Level**, **Date Time Format**, **Last Bind Time**, **Explicit Bind Time**, **Explain Level**, **Explain Snapshot**, and **Explain Mode**.
Statements	Shows the contents of a package statement. Optionally, you can select a statement and click **Explain** to copy the statement to an ISQL Editor window. For more information, see Using the SQL Editor.
Comment	For details on using this tab, see Adding a Comment to an object.
Dependencies	For details on using this tab, see Working with Object Dependencies.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Primary Keys Editor (IBM DB2 LUW)

The Primary Keys Editor lets you manage columns and properties for a primary key, and view space usage/allocation details.

To edit a primary key

1 Open an editor on the primary key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Columns	Displays the columns that make up the primary key. Optionally, you can:	
		Select a column and use the Delete key to remove the column from the primary key.
		Select a column and use the arrow keys to reorder the columns in the primary key.
Properties	Displays the owning **Table Schema** and **Table Name**, the **Name** of the primary key, a **Defined By** property, and **Percent Free** and **Minimum Percent Used** properties.	
Space	Lets you view the following property groups:	
	Attributes	**Make Unique**, **System Required**, **Total Keys**, **Page Fetch Pairs**, and **Distinct Keys**
	Statistics	**Index Level**, **Cluster Ratio**, **Cluster Factor**,**Leaf Pages**, **Sequential Pages**, and **Density**
	Cards	**First Key,First 2 Keys, First3 Keys, First 4 Keys,** and **Full Keys**
Comment	For details on using this tab, see Adding a Comment to an object.	
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Procedures Editor (IBM DB2 LUW)

The Procedures Editor lets you manage the body and parameters for a procedure.

To edit a procedure

1 Open an editor on the procedure. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you modify the following settings: **SQL Access Level** (MODIFIES SQL DATA, CONTAINS SQL, READS SQL DATA, NO SQL), number of **Result Sets**, **External Action**, New Save Point, Inherit Special Registers, **Deterministic**, and **Parameter CCSID**. Lets you view the following settings: **Language** (C, COBOL, JAVA, OLE, SQL CLR), **External Name**, Threadsafe, **Fenced**, **Parameter Type** (GENERAL, DB2, GENERAL WITH NULLS, SQL), **Program Type** (MAIN, SUB and not enabled for a **Parameter Style** of SQL), and **DBInfo**.
Parameters	Lets you manage procedure parameters. On opening, this tab shows the existing parameters. Optionally you can: Select a parameter from the list, modify the **Type**, and if appropriate, the **Precision**, **Size**, and **Scale** of the parameter. You can also set the **Parameter Mode** to INPUT, OUTPUT, or INPUT_OUTPUT. Click the **New** button, provide a name for the new parameter and modify its attributes. Select a parameter and click the **Drop** button to delete the parameter. Select a parameter and use the arrow buttons to reorder the parameter list. **NOTE:** You cannot use host variables in the CALL statement for the name of the procedure.
Comment	For details on using this tab, see Adding a Comment to an object.
Body	Lets you modify the body of the procedure.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Permissions	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Sequences Editor (IBM DB2 LUW)

The Sequences Editor lets you manage cycle numbers, increments, the datatype, and other options for a sequence.

To edit a sequence

1 Open an editor on the sequence. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Definition	Lets you work with settings in the following categories:	
	Parameters	**Increment By -** Lets you specify the interval between sequence numbers. This integer value can be any positive or negative integer, but it cannot be 0. This value can have 28 or fewer digits. The absolute of this value must be less than the difference of MAXVALUE and MINVALUE. If this value is negative, then the sequence descends. If the increment is positive, then the sequence ascends. If you omit this clause, the interval defaults to 1. **Minimum Value -** Lets you specify the minimum value of the sequence. This integer value can have 28 or fewer digits. **Maximum Value -** Lets you specify the maximum value the sequence can generate. This integer value can have 28 or fewer digits.
	Next Sequence Numbers	Lets you work with sequence cycle numbers.
	Sequence Datatype	Lets you specify the datatype and width for the sequence.
	Options	Lets you specify **Cache Size**,**Cycle When Reach Max/Min**, and **Generate Numbers in Order** (useful when you are using the sequence number as a timestamp) values.
Dependencies	For details on using this tab, see Working with Object Dependencies.	
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Structured Types Editor (IBM DB2 LUW)

The Structured Types Editor lets you manage the attributes, methods and body for a structured type.

To edit a structured type

1 Open an editor on the structured type. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Displays the initially-defined values for the following properties: **Instantiable**, **Final Type**, **With Function Access**, **Without Comparisons**, **Inline Length**, **Supertype Schema**, **Supertype Name**, **Cast (Source as Ref) With**, **Cast (Ref as Source) With**, **Reference Using**, **Size**, **Precision**, and **Scale**. For details on these properties, see Structured Types Wizard (DB2 LUW).
Attributes	Displays the currently defined attributes for the structured type. Optionally, take one of the following actions: Select an attribute and change the **Datatype** in the Property/Value list. Depending on the datatype you choose you can also provide additional datatype options. For details on option availability, see Structured Types Wizard (DB2 LUW). Add an attribute by clicking the **New** button and typing a name for the attribute. Select an attribute and click **Drop** to delete the attribute.
Methods	This tab lets you initiate creation of method specifications and prepopulate associated method bodies. On opening, this tab lists all method specifications associated with the structured type and for each method, includes name and language details. Optionally, take one of the following actions: Click **Add** to begin the process of adding a new method specification. Similarly, select a method and click **Edit** to modify the method specification. For more information, see Adding or editing structured type methods. Select a method specification and click **Drop** to delete that method specification from the structured type.

	The **Create Body** and associated controls are available for selected methods for which no method body has yet been defined. They lets you prepopulate the method body definition: **External name** - lets you provide a 'string" or SQL identifier for a method specified with a **Language** of C, JAVA, or OLE. **Transform group** - lets you specify the transform group used when invoking the method. This setting is available for methods specified with a **Language** of C, JAVA, or OLE. **As identifier** - This setting is available for methods specified with a **Language** of C. When this check box is checked, the method body is created with the external name as provided. Otherwise, the external named provided will appear within quotes in the CREATE METHOD statement on the **Body** tab. **Inherit isolation level with lock request** - specifies whether the INHERIT clause is specified as INHERIT ISOLATION LEVEL WITHOUT LOCK REQUEST or INHERIT ISOLATION LEVEL WITH LOCK REQUEST. This setting is available for methods specified with a Language of SQL. After specifying options, click **Create Body** to work with the CREATE METHOD statement you generated on the editor's **Body** tab.
Body	Lets you view and modify the CREATE METHOD statement generated with your choices on the **Methods** tab.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Adding or editing structured type methods

The **Add Method** and **Edit Method** wizards let you work with the methods of a structured type. They let you specify basic properties, parameter and return value details for methods.

To add or edit a structured type method:

1 Open an editor on a structured type. For details, see Opening an Object Editor.

2 Navigate to the **Methods** panel, and then open a wizard using one of the following techniques:

- Click **Add** to create a new method
- Select an existing method and click **Edit**.

3 Use the following table as a guide to understanding and modifying the settings on the tabs of this wizard:

Tab	Settings and tasks
Properties	When adding a method this pane lets you specify a **Name**, **Specific Name**, **Language** (OLE, SQL, JAVA, C), and **SQL Access Level** (CONTAINS SQL, READS SQL DATA, NO SQL). When editing a method, these properties are for display only.
Advanced (available after clicking the **Advanced** button on the **Properties** pane)	Lets you select or enable the following ADD METHOD options: **Deterministic**, **External Action**, **Called on NULL Input**, **DBINFO**, **Fenced**, **Allow Parallel**, **Scratchpad**, **Scratchpad Length**, **Parameter Style** (SQL, DB2GENERAL), and **Final Call**.
Parameters	Lets you work with input parameters: Add a parameter by clicking the **New** button and typing a name for the parameter. Select a parameter and change the **Datatype** in the Property/Value list. Depending on the datatype you choose you can also provide **Precision**, **Size**, and **Scale** options. Delete a selected parameter by clicking the **Delete** button. Order the parameters using the arrow buttons.
Return	Lets you provide details of the return value of the method: **Return Datatype** Lets you specify the base **Type** of the return value. Depending on the datatype you choose you can also provide **Precision**, **Size**, and **Scale** options. **Cast Datatype** This group lets you optionally use the CAST FROM form of the ADD METHOD RETURN clause. It lets you have the method return a different datatype, cast from the datatype specified in the **Return Datatype** group. This feature must be **Enabled** and provide the same type options as the **Return Datatype** group. An **As Locator** option specifies that the method return a LOB locator instead of the actual value. The option is only available for LOB and LOB-based datatypes. **As Locator** applies to a LOB type in the **Cast Datatype** group, if specified. Otherwise it applies to a LOB type in the **Return Datatype** group

4 When ready click **Finish**.

Tables Editor (IBM DB2 LUW)

The Tables Editor lets you manage basic properties, columns, dimension columns, distribution key columns, partitions, tablespaces, and constraints for a table.

Tip: Before modifying a table, familiarize yourself with the material in Altering Tables for IBM DB2 LUW for Linux, Unix, and Windows.

To edit a table

1 Open an editor on the table. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Columns	Displays the currently defined columns in the table. Optionally, you can: Select a column, and in the **Property**/**Value** list modify property values for that column. Click **Add Column**, provide a name for the new column, and set property values for the column. Select a column and click **Delete** to remove the column from the table.
Properties	Displays the **Name,Created**, **Last RunStats**, **Invalidate Time**, and **Defined By** properties. Lets you set **Percent Free**, **Lock Size**, **Append Data**, **Volatile**, **Compress**, **Row Compression**, **Security Policy**, **RestrictDrop**, **LogIndexBuild**, **CCSID**, **Do not initially log**, and **Data Capture** properties.
Status	Lets you view the following statistics: **Total Number of Rows**, **Number of Overflow Rows**, **Number of Pages With Rows**, **Number of Page**s, **Table Status**, **Row Organization**, **RowTypeSchema**, **RowTypeName**, **AccessMode**, **ActiveBlocks**, **AvgCompressedRowSize**, **AvgRowCompressionRatio**, **CodePage**, **PercentOfPagesSaved**, **LastRegeneratedTime**, and **ProtectionGranularity**.
Partition	This tab provides details on the partition columns and data partitions for the table. Optionally you can: Use the New or Delete buttons to create or drop a partition column or data partition. Use the Edit button to edit a data partition. Use the **Commands** menu to attach or detach a data partition.

Tablespaces	**Data Tablespace**, **Long Tablespace**, and **Index Tablespace**	[JavaScript:popup.TextPopup(Poptext1,Popfont,9,9,-1,-1) Lets you view the **Database partition Group**], **Managed By**, **Page Size**, and **Extent Size** properties. Lets you choose the **Name** of a tablespace.
Dimensions	Lets you group columns to form a dimension:	Click the **New** button to add a new column to the dimension for the table. Select a column and click the **Edit** button to modify the dimension column properties. Select a column and click the **Delete** button to drop a column from the dimension.
Distribution Key Columns	Lets you group one or more columns to form a distribution key:	Click the **New** button to add a new column to the dimension for the table. Select a column and click the **Edit** button to modify the dimension column properties. Select a column and click the **Delete** button to drop a column from the dimension.
Indexes	Lets you manage indexes for a table:	Click **Add** to open a dialog that lets you add a new index to the table. Select an index and click **Edit** to open a dialog that lets you edit index properties. Select an index and click **Drop** to open a dialog that lets you remove the index from the table.
Constraints	Lets you manage primary key, unique key, foreign key, and check constraints for a table. Constraints are grouped by type, under folders:	Select a constraint type folder and click **Add** to open a dialog that lets you add a constraint of that type. Select a constraint and click **Edit** to open a dialog that lets you modify the constraint details. Select a constraint and click **Drop** to remove the constraint.

Comment	For details on using this tab, see Adding a Comment to an object.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Permissions	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Altering Tables for IBM DB2 LUW for Linux, Unix, and Windows

The ALTER TABLE command of Transact-SQL is limited to adding NULL columns to the end of a table and to adding or dropping constraints. Unfortunately, this scenario does not address many requirements of administrators and developers who need to add, delete or modify columns more broadly:

- Add columns anywhere in a table, not just the end
- Add columns that do not permit a NULL value.
- Change the NULL/NOT NULL status of table columns
- Change column datatypes to other compatible types
- Change the length of datatypes
- Delete a column

Due to the limitations of the ALTER TABLE command, the only way to make broader modifications is to write SQL scripts that step through all desired changes. To perform an enhanced table alter, SQL script that completes the following steps is constructed:

1 Renames the existing table so that the original and its data remain intact

2 Builds a CREATE TABLE statement with the new table definition, including declared defaults, primary key and check constraints

3 Builds an INSERT statement to copy data from the original, renamed table to the new one

4 Builds foreign keys on the new table

5 Reapplies any privileges granted on the table

6 Rebuilds all dependencies on the new table, including indexes, triggers, procedures, packages, functions and views. When rebuilding procedures, functions, packages and views, any permissions on them are also rebuilt.

Tablespaces Editor (IBM DB2 LUW)

The Tablespaces Editor lets you manage containers, basic properties, performance, space usage, and associated objects for a tablespace.

To edit a tablespace

1 Open an editor on the tablespace. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Properties	Lets you work with settings in the following categories:	
	Tablespace	Lets you view the **Type** (REGULAR, LARGE, TEMPORARY, or USER TEMPORARY), **Use Automatic Storage**, **Managed By**, and **Database Partition Group** properties. Lets you modify the **Buffer Pool** and **Drop Recovery** properties.
	Performance	Lets you view the **Page SIze** and **Extent Size** properties. Lets you specify or select the **Prefetch Automatic**, **Overhead**, **Transfer Rate**, and **File System Caching** properties.
	Automatic Storage	Lets you work with the **AutoResize** (enables following size settings), **Initial Size**, **Increase Size**, **Max Size Unlimited**, and **Max Size** (enabled if **Max Size Unlimited** is not set) attributes. The **Initial Size** and **Max Size** values can be provided as kB, mB, or gB values.
Container	This option is only available for database managed tablespaces. This tab lets you add or delete containers of a tablespace.	
		Select an existing container and in the Property/Value list, use the **Size** controls to RESIZE, EXTEND, or REDUCE the container.
		Add a container by clicking **New**, and then in the Property/Value list, provide **Name**, number of **DatabasePartitions**, **Type** (DEVICE, FILE) and original **Size**.
		Click **Delete** to remove a container.
Performance	Lets you manage settings in the following categories:	
	Page Setting	Lets you view **Page SIze** and **Extent Size** settings and modify the **Prefetch Size** setting.
	I/O Setting	Let you modify the **Overhead** and **Transfer Rate** settings.
	Dropped Table Settings	Lets you view the **Recovery Status**.
	Defaults	Lets you view the **Nodegroup** and specify a **Bufferpool**.

Comment	For details on using this tab, see Adding a Comment to an object.
Space	Lets you view the table usage and the distribution of space for a tablespace. Specific statistics include **Free Pages**, **Used Pages**, **Reserved Pages**, and **Total Pages**.
Objects	Lets you manage database objects associated with the tablespace. Objects are organized in a tree structure with folders containing the objects. Optionally, take one of the following actions: Select an object and click **Edit** to open an object editor on the selected object. Select an object and click **Drop** to initiate dropping the selected object.
Privileges	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Triggers Editor (IBM DB2 LUW)

The Triggers Editor lets you view properties for a trigger.

To edit a trigger

1 Open an editor on the trigger. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view the **Trigger Timing**, **Trigger Events**, **Trigger Type**, **Object Status**, **Define**r, and **Function Path** properties.
Comment	For details on using this tab, see Adding a Comment to an object.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Definition	Lets you view and modify the CREATE TRIGGER DDL that will implement your changes.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Unique Keys Editor (IBM DB2 LUW)

The Unique Keys Editor lets you manage columns and basic properties for a unique key, as well as view space details.

To edit a unique key

1 Open an editor on the unique key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Columns	[JavaScript:popup.TextPopup(Poptext1,Popfont,9,9,-1,-1) Lets you manage columns that make up the unique key. On opening, this tab shows the existing columns. For each column, the datatype (and if applicable the precision in brackets) and whether the table definition permits nulls in the target table column.] Optionally you can:	
		Click the **New** button and select a column name from the dropdown, to add a column to the index.
		Select a column and click the **Drop** button to delete the column from the index.
Properties	Displays the owning **Table Schema** and **Table Name**, the Name of the primary key, a **Defined By** property, and **Percent Free** and **Minimum Percent Used** properties.	
Space	Lets you view settings in the following categories:	
	Attributes	Lets you view **Make Unique**, **System Required**, **Total Keys**, **Distinct Keys,** and **Page Fetch Pairs** settings.
	Statistics	Lets you view **Index Level**, **Cluster Ration**, **Cluster Factor**,**Leaf Pages**, **Sequential Pages**, and **Density** settings.
	Cards	Lets you view **First Key**, **First 2 Keys**, **First 3 Keys**, **First 4 Keys** and **Full Keys** settings.
Comment	For details on using this tab, see Adding a Comment to an object.	
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

User Datatypes Editor (IBM DB2 LUW)

The User Datatypes Editor lets you manage the basic properties of a user datatype.

To edit a user datatype

1 Open an editor on the user datatype. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Displays the user **Datatype** name. Lets you select a new **Owner** or **Type**, and depending on the type specified, offers additional type-specific properties such as **Size**. Also lets you modify the **Allow Bit Data** property, if appropriate.
Comment	For details on using this tab, see Adding a Comment to an object.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Users Editor (IBM DB2 LUW)

The Users editor lets you manage object dependencies and permissions for a user.

To edit a user

1 Open an editor on the user. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Displays the user **Name**.
Objects	Lets you manage database objects associated with the user. Objects are organized in a tree structure with folders containing the objects. Optionally, take one of the following actions: Select an object and click **Edit** to open an object editor on the selected object. Select an object and click **Drop** to initiate dropping the selected object.
Object Permissions and **System Permissions**	For details on using these tabs, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Views Editor (IBM DB2 LUW)

The Views Editor lets you manage the columns as well as view and modify properties for a view.

To edit a view

1 Open an editor on the view. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Displays all columns for the view. Details for each column include the **Column Name**, the **Datatype** (and if applicable, with the precision in parentheses), and whether or not **Nulls** are allowed for that column. You can also set the **EnableQueryOptimization** property for the view.
Definition	Lets you view and modify the CREATE VIEW DDL that will implement any changes you make in this editor. To modify a view, the existiing view is automatically dropped and then recreated.
Comment	For details on using this tab, see Adding a Comment to an object.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Permissions	For details on using this tab, see Working with Privileges and Permissions.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

IBM DB2 for z/OS Object Editors

An Object Editor for all supported IBM DB2 for z/OS objects:

- Aliases Editor (IBM DB2 Z/OS)
- Buffer Pools Editor (DBArtisan - IBM DB2 Z/OS)
- Check Constraints Editor (IBM DB2 Z/OS)
- Databases Editor (IBM DB2 Z/OS)
- DBRMs Editor (DBArtisan - IBM DB2 Z/OS)
- Foreign Keys Editor (IBM DB2 Z/OS)
- Functions Editor (IBM DB2 Z/OS)
- Indexes Editor (IBM DB2 Z/OS)
- Packages Editor (IBM DB2 Z/OS)
- Plans Editor (IBM DB2 Z/OS)
- Primary Keys Editor (IBM DB2 Z/OS)
- Procedures Editor (IBM DB2 Z/OS)
- Stogroups Editor (IBM DB2 Z/OS)
- Synonyms Editor (IBM DB2 Z/OS)
- Tables Editor (IBM DB2 Z/OS)
- Tablespaces Editor (IBM DB2 Z/OS)
- Triggers Editor (IBM DB2 Z/OS)
- Unique Keys Editor (IBM DB2 Z/OS)
- User Datatypes Editor (IBM DB2 Z/OS)
- Users Editor (IBM DB2 Z/OS)
- Views Editor (IBM DB2 Z/OS)

For access to object editors for different supported DBMS platforms and an introduction to editor usage, see Modifying objects using editors.

Aliases Editor (IBM DB2 Z/OS)

The Aliases Editor lets you view details of an alias definition.

To edit an alias

1 Open an editor on the alias. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	In addition to the owning **Schema** and **Name** of the alias, this tab lets you view the **Target Owner**, **Target Type**, and **Target Name**, of the referenced object.
Comment	For details on using this tab, see Adding a Comment to an object.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Buffer Pools Editor (DBArtisan - IBM DB2 Z/OS)

The buffer pool editor lets you manage basic properties of a buffer pool and view usage details.

To edit a buffer pool

1 Open an editor on the buffer pool. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Definition	Lets you work with settings in the following areas: **Storage** Lets you set the **Size** (VPSIZE), **Type** (VPTYPE), **Hiperpool Size** (HPSIZE), **Castout**, **Page Stealing Algorithm** (PGSTEAL) properties. **Threshold** Lets you set the **Sequential Steal** (VPSEQT), **Parallel Sequential** (VPPSEQT), **Assisting Parallel Sequential** (VPXSEQT), **Hiperpool Sequential Steal Threshold** (HPSEQT), **Deferred Write Threshold** (DWQT), and **Vertical Deferred Write Threshold** (DDWQT) properties
Usage Details	Displays space utilization details for each associated database.
Privileges	For details on using this tab, see Working with Privileges and Permissions.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Check Constraints Editor (IBM DB2 Z/OS)

The Check Constraints Editor lets you view definition details and edit a check condition expression.

To edit a check constraint

1 Open an editor on the check constraint. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Definition	Lets you view basic identification information on the check constraint: **Table Schema** and **Table Name**, the **Name** of the constraint and the date it was **Created**. You can also edit the condition in the **Check Condition** box. The **Table Columns** button acts a time saver in editing the condition. It opens a dialog that lets you select and paste column names into the check condition expression.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Databases Editor (IBM DB2 Z/OS)

The Databases Editor lets you manage basic properties, permissions, and object dependencies for a database.

To edit a database

1 Open an editor on the database. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	The **Database Creation** group lets you view the **Name** and **Group Member** properties and set the **Type** property. The **Database Attributes** group lets you view the **Encoding Scheme**, **Create Date**, and **Last Altered** properties and lets you set the **Tablespace Buffer Pool**, **Index Buffer Pool**, **Storage Group**, and **CCSID** properties.
Permissions	For details on using this tab, see Working with Privileges and Permissions.
Dependencies	For details on using this tab, see Working with Object Dependencies.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

DBRMs Editor (DBArtisan - IBM DB2 Z/OS)

The DBRM Editor lets you view into the metadata of each DBRM, view and cross-reference all associated Plans and Packages as well as view and Explain all Package SQL Statements.

To edit a DBRM

1 Open an editor on the DBRM. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Description	Displays the main contents of the DBRM header information, including the **Precompile Timestamp**, **Language**, and **Consistency Token**, as well as **String Delimiter**, **Katakana Charset**, **Mixed Character Set**, and **Decimal Point** specifiers.
Plans	Lists details of the plans containing the DBRM. Optionally, you can: Select a plan and click **Edit** to open the Plans editor on the plan. For more information, see Plans Editor (IBM DB2 Z/OS). Select a plan and click **Rebind** to initiate rebinding the plan. For more information, see Rebind Plans. Select a plan and click **Free** to initiate deleting the plan. For more information, see Drop.
Packages	Displays a cross-reference to the collections, packages, and plans where the DBRM was bound. Optionally, you can: Select a package and click **Edit** to open the Packages editor on the package. For more information, see Packages Editor (IBM DB2 Z/OS). Select a package and click **Rebind** to initiate rebinding the package. For more information, see Rebind Packages. Select a package and click **Free** to initiate deleting the package. For more information, see Free (Packages).
Statements	Displays the information about the SQL statements in the DBRM. The information is for programmers reference only. Optionally, click **Explain** to copy the selected statement(s) to an ISQL window. The Explain output displays in the report pane. For more information, see Using the SQL Editor.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Foreign Keys Editor (IBM DB2 Z/OS)

The Foreign Keys Editor lets you manage column mapping and specify a delete rule for a foreign key.

To edit a foreign key

1 Open an editor on the foreign key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Column Mapping	The existing column mapping for the foreign key is represented by selected columns in the **Main Table** and **Referenced Table** lists. Additional candidates in the **Referenced Table** list are indicated by enabled column check boxes. If necessary, use the **Table** dropdown in the **Referenced Table** group to choose a new table for this foreign key. Select or deselect columns in the **Main Table** list and **Referenced Table** list to form the referential constraint between the two tables.
Properties	Lets you specify a **Delete Rule** (CASCADE, NO ACTION, RESTRICT, SET NULL).
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Functions Editor (IBM DB2 Z/OS)

The Functions Editor lets you view and modify properties for a function, manage its inputs and outputs, and modify the code in the body of the function.

To edit a function

1 Open an editor on the function. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Properties	Lets you work with properties in the following categories:	
	Identification	Lets you view the **Create Timestamp**, **Schema**, **Routine ID**, and **Origin** properties. Lets you set the **External Name** and **Collection ID** properties.
	Run Time	Lets you view the **Result Sets** property. Lets you set the **WLM Environment**, **WLM For Nested**, **ASU Time**, and **Run Options** properties.
	Structure	Lets you view the **Language**, **Parameter Style**, **Number of LOB Columns**, **Number of Parameters**, and **Allow Parallel** properties. Lets you set the **Program Type**, **Security Type**, **SQL Access Level**, **Inherit Special Registers**, **Fenced**, **DBINFO**, **Deterministic**, **Called On Null Input**, **External Action**, **Stay Resident**, **Final Call**, **Scratchpad**, and **Scratchpad Length** properties.
	Run Estimates	Lets you view the **Initial I/Os**, **I/Os Per Invocation**, **Initial Instructions**, and **Instructions Per Invocation** properties.
	Java Structure	Lets you view the **Java Class**, **Jar ID**, **Package ID**, **Method ID**, **Jar Schema**, and **Java Signature** properties.
Parameters, **Return Columns**, and **Return Scalar**	On opening, these tabs displays the current function parameters, return columns or return scalar. Optionally, take one of the following actions:	
		Select a parameter, return column, or return scalar in the **Attributes** area, modify values, as permissible. Attributes differ by the tab you chose but typically include items such as **Type**, **Precision**, **Scale**, **As Locator**, and **Size**.
		Click the **New** button to provide datatype and size details for a new parameter, return column, or return scalar.
		Select a parameter, return column, or return scalar and click **Drop** to delete that parameter.

Body	Lets you modify the code in the text area of the tab.
Comment	For details on using this tab, see Adding a Comment to an object.
Permissions	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Indexes Editor (IBM DB2 Z/OS)

The Indexes Editor lets you manage basic, storage, and space properties for an index, as well as work with its columns and partitions.

To edit an index

1 Open an editor on the index. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Columns	Lets you manage the columns or key-expressions that make up the index. On opening, this tab shows the existing columns/key-expressions. **NOTE:** Key-expression functionality is only enabled if the **Index on Expression** setting is enabled on the **Properties** tab. Optionally you can: Change the **Sort** order of a column or key-expression. If **Index on Expression** is disabled, click the **New** button and select a column name from the dropdown to add a column to the index. If **Index on Expression** is enabled, click the **New** button and type a valid key-expression in the **Expression** field, to add an expression to the index. Select a column or key-expression and click the **Drop** button to delete the column from the index.
Properties	Lets you work with the **Buffer Pool**, **Piece Size**, **Close**, **Copy**, **Compress**, **Index on Expression**, and **Padded** properties. For more information on these properties, see Indexes (DB2 z/OS) - Properties.
Storage	Lets you select a dataset management scheme and provide associated attribute values. For details, see Indexes (DB2 z/OS) - Storage.
Partitions	Lets you work with partitions for the index. For each data partition, the listing shows the storage group, VCAT Catalog, primary and secondary space allocations, and if appropriate, the free space percentage, GBP Cache, limit key value, and erase on delete settings. Optionally, you can select a partition and click the **Edit** button top open an editor that lets you edit that partition.
Space	Lets you view the following property groups: **Attributes**: **Make Unique**, **System Required**, **Total Keys**, **Page Fetch Pairs**, and **Distinct Keys** **Statistics**: **Index Level**, **Cluster Ratio**, **Cluster Factor**,**Leaf Pages**, **Sequential Pages**, **Density**, **DASD Storage**, and **Data Blocks/Key**. **Cards**: **First Key** and **Full Keys**

Comment	For details on using this tab, see Adding a Comment to an object.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Packages Editor (IBM DB2 Z/OS)

The Packages Editor lets you manage properties and bind parameters for a package, as well as view environment information, package contents and associated plans.

To edit a package

1 Open an editor on the package. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you work with the following properties: **Average Size**, **Bind Time**, **Consistency Token**, **Dec31**, **Decimal Point**, **Group Member**, **Katakana Charset**, **Language**, **Mixed Character Set**, **Operative**, **Package Size**, **Package Source**, **PDS Name**, **Precompile Timestamp**, **Release Bound**, **String Delimiter**, **SysEntries**, **Timestamp**, and **Valid**.
Bind Parameters	Lets you set the **Schema Path**, **Creator**, **Qualifier**, **Explain**, **Validate**, **Degree**, **Dynamic Rules**, **Lock Isolation**, **Encoding**, **Page Writes**, **Resource Release**, **CurrentData**, **DB Protocol**, **Reoptvar**, **Keep Dynamic**, **Defer Prepare**, **SQL Error**, and **Optimization Hint** properties.
Plan/Packlists	Displays the plans contained in the package if the package was bound individually, or as part of a complete collection ID. Optionally, you can: Select a plan and click **Edit** to open the Plans editor on the plan. For more information, see Plans Editor (IBM DB2 Z/OS). Select a plan and click **Rebind** to initiate rebinding the plan. For more information, see Rebind Plans. Select a plan and click **Free** to initiate deleting the plan. For more information, see Drop.
Statements	Shows the contents of any package statement on the datasource. Optionally, select a statement and click **Explain** to copy the statement to an ISQL Editor window. NOTE: For more information, see Using the SQL Editor.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Privileges	For details on using this tab, see Working with Privileges and Permissions.
Environments	Displays run-time environments information for a package. Use the arrow buttons to move environments between the **Enabled Environments** and **Disabled Environments** lists. To modify a connection, select the environment and click **Edit Connections**. For more information, see Connection Editor.
Command	Displays the command that originally built the package.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Connection Editor

The Connection Editor lets you modify plan and package connections. It is opened from the Packages editor. For more information, see **Packages Editor (IBM DB2 Z/OS)**.

The table below describes the options and functionality of the Connection Editor:

Option	Description
Connections	Displays the connections for the plan or package.
Add	Click to add the connection.

Plans Editor (IBM DB2 Z/OS)

The Plans Editor lets you manage plan properties, view DBRM information, manage packages and contents, and manage run-time environments for a plan.

To edit a plan

1 Open an editor on the plan. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Definition	Lets you work with the following attribute groups: **Properties** Lets you view the **Timestamp**, **Group Member**, **Plan Size**, **Average Size**, **Valid, Operative**, **Pkg. List Entries**, **SysEntries**, and **Release Bound** properties. Lets you set the **Owner**, **Qualifier**, and **Current Server** properties. **Bind Parameters** Lets you set the **Schema Path**, **Explain**, **Validate**, **Degree**, **Dynamic Rules**, **Lock Isolation**, **Resource Acquire**, **Resource Release**, **Disconnect**, **Current Data**, **DB Protocol**, **Reoptvar**, **SQL Rules**, **Keep Dynamic**, **Defer Prepare**, **Encoding**, **Page Writes**, **Flag**, **Optimization Hint**, and **Cache Size** properties.
DBRMs	Lists details for each DBRM associated with a plan. Optionally, select a DBRM from the list and click **Edit** to open the DBRM editor on that DBRM.
Packages	Displays details for each package associated with a plan. Optionally, you can: Select a package and click **Edit** to open the Package editor on that package. For details, see Packages Editor (IBM DB2 Z/OS). Select a package and click **Rebind** to open the Rebind Packages dialog. Select a package and click **Free** to open the Free (Packages) dialog.
DBRM/Packages	Displays the entire contents of the plan, DBRMs and packages, in a single display. The first column contains either a 'D' for DBRM or 'P' for packages. Optionally, take one of the following actions: Select a package list entry and click **Edit** to open the Package editor on that package. For details, see Packages Editor (Oracle). Select a package and click **Rebind** to open the Rebind Packages dialog. Select a package and click **Free** to open the Free (Packages) dialog.

Dependencies	For details on using this tab, see Working with Object Dependencies.
Privileges	For details on using this tab, see Working with Privileges and Permissions.
Environments	Displays run-time environments information for a plan. Optionally, you can: Use the arrow buttons to move environments between the **Enabled Environments** and **Disabled Environments** lists. Select an environment and click **Edit Connections** to open the Connection editor.
Command	Displays the command that originally built the plan.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Primary Keys Editor (IBM DB2 Z/OS)

The Primary Keys Editor lets you manage primary key columns as well as work with storage, space, and partitions for a primary key.

Tip: The refresh button lets you refresh or clear the editor's contents, and log SQL.

To edit a primary key

1 Open an editor on the primary key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Column	Lets you manage columns that make up the primary key. On opening, this tab shows the existing columns. For each column, the listing shows the datatype (and if applicable the precision in brackets) and whether the table definition permits nulls in the target table column. Optionally you can: Change the **Sort** order of a column. Click the **New** button and select a column name from the dropdown, to add a column to the primary key. Select a column and click the **Drop** button to delete the column from the primary key. Select a column and use the arrow buttons to reorder the columns in the primary key.
Properties	Lets you set **Buffer Pool**, **Close**, **Copy** and **Piece Size** properties.
Storage	Lets you view **Storage Group** and **VCAT Catalog** properties. Lets you set **Primary Space Allocation**, **Secondary Space Allocation**, **Erase**,**Free Page, Percent Free,** and **GBP Cache** properties.
Partitions	Lets you work with partitions for the primary key.
Space	Lets you view values in the following property groups: **Attributes** **Make Unique**, **System Required**, **Total Keys**, **Page Fetch Pairs**, and **Distinct Keys**. **Statistics** **Index Level**, **Cluster Ratio**, **Cluster Factor**,**Leaf Pages**, **Sequential Pages**, and **Density**. **Cards** **First Key** and **Full Keys**.

Comment	For details on using this tab, see Adding a Comment to an object.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Procedures Editor (IBM DB2 Z/OS)

The Procedures Editor lets you manage the properties and input/output parameters of a procedure.

To edit a procedure

1 Open an editor on the procedure. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Properties	Lets you work with properties in the following categories:	
	Identification	Lets you view the **Schema** and **Routine ID** properties. Lets you set the **External Name** and **Collection ID** properties.
	Run Time	Lets you set the **Result Sets**, **WLM Environment**, **WLM For Nested**, **ASU Time**, and **Run Options** properties.
	Structure	Lets you view the **Language**, **Number of LOB Columns**, **Number of Parameters**, and **DBINFO** properties. Lets you set the **Program Type**, **Security Type**, **SQL Access Level**, **Parameter Style**, **Inherit Special Registers**, **Fenced**, **Commit on Return,Deterministic**, and **Stay Resident** properties.
	Run Estimates	Lets you view the **Initial I/Os**, **I/Os Per Invocation**, **Initial Instructions**, and **Instructions Per Invocation** properties.
	Java Structure	Lets you view the **Java Class**, **Jar ID**, **Package ID**, **Method ID**, **Jar Schema**, and **Java Signature** properties.
Parameters	Displays a listing of the existing input/output parameters for the procedure. For the selected parameter, the **Datatype** list shows details for that parameter, including the **Type** and **Parameter Mode** (INPUT, OUTPUT, INPUT_OUTPUT). Depending on the type other parameters such as **Size**, **Precision**, or **Scale** may be available for viewing or modification. Optionally you can:	
		Select a parameter and in the **Datatype** list, modify details for that parameter.
		Click the **Add** button to add a new parameter, provide a name for the new parameter, and edit the parameter values in the **Datatype** list.
		Click the **Delete** button to drop a selected parameter.
		Use the arrow buttons to change the order location of a selected parameter.

Comment	For details on using this tab, see Adding a Comment to an object.
Body	Lets you modify the SQL code for procedures on the current datasource
Permissions	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Stogroups Editor (IBM DB2 Z/OS)

The Stogroups Editor lets you view and modify volumes, manage function privileges, and view DDL for a stogroup.

To edit a stogroup

1 Open an editor on the stogroup. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Volume Devices	[JavaScript:popup.TextPopup(Poptext1,Popfont,9,9,-1,-1) Shows details on volumes associated with the stogroup. Optionally you can:] Click **Add** to open a dialog that lets you add volumes. Select a volume and click **Remove** to delete the volume.
Privileges	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Synonyms Editor (IBM DB2 Z/OS)

The Synonyms Editor lets you view base object information and manage database object dependencies for a synonym.

To edit a synonym

1 Open an editor on the synonym. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view the owning **Schema**, the **Name** of the synonym, and the synonym's **Referenced Object Owner**, **Referenced Object Type**, and the **Referenced Object Name**.
Dependencies	For details on using this tab, see Working with Object Dependencies.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Tables Editor (IBM DB2 Z/OS)

The Tables Editor lets you: manage columns, basic properties, partitions, indexes, and constraints for a table.

Note: Before editing tables, refer to the material in Altering Tables for IBM DB2 z/OS.

To edit a table

1 Open an editor on the table. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Columns	Displays the currently defined columns in the table. For each column, the Property/Value list displays the **Name** and **Type** for the column. In addition, depending on the **Type** selected, the list also displays **Size**, **Scale**, **Identity Column**, **Allow Nulls**, **Default Value**, **Comment**, and **For Data** property values, as appropriate. The **Statistics** group displays a **Number of Distinct Values in the Column** value. Optionally, you can: Select a column, and in the **Property**/**Value** list modify property values for that column. Click **Add Column**, provide a name for the new column, and set property values for the column. Select a column and click **Delete** to remove the column from the table.
Properties	Lets you work with settings in the following categories: **Table** Lets you view **EditProcedure**, **Table Type**, **Table Status**, **Check Flag**, **DBID**, and **OBID** properties. Lets you set **Volatile**, **Audit**, **RestrictDrop**, **Label**, and **ValidProc** properties. **Tablespace Placement** Lets you select a **Tablespace**. **Log Options** Lets you select a **Data Capture** option of DATA CAPTURE NONE or DATA CAPTURE CHANGES. **Statistics** Lets you view the following statistics: **Last Runstats**, **Total number of rows**, **Average Row Length**, **Number of Pages**, **Percent Compressed Rows**, **Max Record Length**, and **DASD storage**.
Partitions	Displays existing partition columns and data partitions. Optionally you can add, edit, or delete partition columns and data partitions.
Indexes	Displays the list of indexes for the table. Optionally, take one of the following actions: Click **Add** to open a dialog that lets you add a new index to the table. Select an index and click **Edit** to open a dialog that lets you edit index properties.

	Select an index and click **Drop** to open a dialog that lets you remove the index from the table.
Constraints	Displays constraints in a tree structure. The tree contains folders which contain all constraints associated with the target table. The objects are organized in folders based on the type of constraint. Optionally take one of the following actions: Select a constraint type folder and click **Add** to open a dialog that lets you add a constraint of that type. Select a constraint and click **Edit** to open a dialog that lets you modify the constraint details. Select a constraint and click **Drop** to remove the constraint.
Comment	For details on using this tab, see Adding a Comment to an object.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Permissions	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Altering Tables for IBM DB2 z/OS

The ALTER TABLE command of Transact-SQL is limited to adding NULL columns to the end of a table and to adding or dropping constraints. Unfortunately, this scenario does not address many requirements of administrators and developers who need to add, delete or modify columns more broadly:

- Add columns anywhere in a table, not just the end
- Add columns that do not permit a NULL value.
- Change the NULL/NOT NULL status of table columns
- Change column datatypes to other compatible types
- Change the length of datatypes
- Delete a column

Due to the limitations of the ALTER TABLE command, the only way to make broader modifications is to write SQL scripts that step through all desired changes. To perform an enhanced table alter, an SQL script that completes the following steps is constructed:

1 Renames the existing table so that the original and its data remain intact

2 Builds a CREATE TABLE statement with the new table definition, including declared defaults, primary key and check constraints

3 Builds an INSERT statement to copy data from the original, renamed table to the new one

4 Builds foreign keys on the new table

5 Reapplies any privileges granted on the table

6 Rebuilds all dependencies on the new table, including indexes, triggers, procedures, packages, functions and views. When rebuilding procedures, functions, packages and views, any permissions on them are also rebuilt.

Tablespaces Editor (IBM DB2 Z/OS)

The Tablespaces Editor lets you work with the basic properties and partitions for a tablespace as well as view space details, status, and objects stored on the tablespace.

To edit a tablespace

1 Open an editor on the tablespace. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Properties	Lets you work with properties in the following categories:	
	Bufferpool	**Buffer Pool**
	Partitions and size	**Number of partitions**, Partition size (DSSIZE), Segment size, and Max rows per page
	Other parameters	**GBPCACHE**, **Compress**, **Track modified pages**, **Encoding scheme**, **Log**, **Define**, **Member Cluster**, **Close rule**, **Lock Size**, and **Maximum locks**
	For more information on these properties, see Tablespaces (DB2 z/OS) - Properties.	
Partitions	Displays a list of partitions for the tablespace. Details for each partition include the storage group, VCAT, primary and secondary space allocations, the free space portion of each page and free page frequency, the group buffer cache scheme, and whether modifications are tracked. Optionally, you can:	
		Select a partition and click **Edit** to open a dialog that lets you modify properties for that partition.
		Select a partition and click **Clone** to open a dialog that lets you apply the attributes of the selected partition to another partition.
Status	Lets you display CLAIMERS, LOCKS, LPL, USE, or WEPR status details	
Space	Lets you view space usage and allocation details for the tablespace.	
Objects	Displays the objects stored on the tablespace. Objects are organized in a tree structure with folders containing the objects. Optionally, you can:	
		Select an object and click **Edit** to open an object editor on that object.
		Select an object and click **Drop** to initiate dropping that object.
Permissions	For details on using this tab, see Working with Privileges and Permissions.	
DDL	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Triggers Editor (IBM DB2 Z/OS)

The Triggers Editor lets you modify the CREATE TRIGGER statement and manage properties for a trigger.

To edit a trigger

1 Open an editor on the trigger. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view the **Trigger Timing**, **Trigger Events**, **Trigger Type**, **Object Status**, **Definer**, and **Function Path** properties.
Comment	For details on using this tab, see Adding a Comment to an object.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Definition	Lets you modify the CREATE TRIGGER body for a trigger.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Unique Keys Editor (IBM DB2 Z/OS)

The Unique Keys Editor lets you manage columns, basic properties, and partitions for a unique key, as well as view storage details.

To edit a unique key

1 Open an editor on the unique key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Column	Lets you manage columns that make up the primary key. On opening, this tab shows the existing columns. For each column, the listing shows the datatype (and if applicable the precision in brackets) and whether the table definition permits nulls in the target table column. Optionally you can:	
		Change the **Sort** order of a column.
		Click the **New** button and select a column name from the dropdown, to add a column to the primary key.
		Select a column and click the **Drop** button to delete the column from the primary key.
		Select a column and use the arrow buttons to reorder the columns in the primary key.
Properties	Lets you set the **Buffer Pool**, **Piece Size**, **Close**, and **Copy** properties.	
Storage	Lets you view the **Storage Group** and **VCAT Catalog** properties. Lets you set the **Primary Space Allocation**, **Secondary Space Allocation**, **Erase**, **Free Page**, **Percent Free**, and **GBP Cache** properties.	
Partition	Lets you work with partitions for the index.	
Space	Lets you view values in the following property groups:	
	Attributes	**Make Unique**, **System Required**, **Total Keys**, **Page Fetch Pairs**, and **Distinct Keys**
	Statistics	**Index Level**, **Cluster Ratio**, **Cluster Factor**,**Leaf Pages**, **Sequential Pages**, and **Density**
	Cards	**First Key** and **Full Keys**
Comment	For details on using this tab, see Adding a Comment to an object.	
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

User Datatypes Editor (IBM DB2 Z/OS)

The User Datatypes Editor lets you manage basic properties of a user datatype.

To edit a user datatype

1 Open an editor on the user datatype. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Properties	Lets you work with properties in the following categories:	
	Base Datatype	Lets you set **Type** and **Size** settings.
	Character Options	Lets you set the **For Data** and **CCSID** properties.
Comment	For details on using this tab, see Adding a Comment to an object.	
Permissions	For details on using this tab, see Working with Privileges and Permissions.	
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Users Editor (IBM DB2 Z/OS)

The Users Editor lets you manage permissions for a user and the objects owned by that user.

To edit a user

1 Open an editor on the user. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Displays the user **Name**.
Objects	Lets you manage database objects associated with the user. Objects are organized in a tree structure with folders containing the objects. Optionally, take one of the following actions: Select an object and click **Edit** to open an object editor on the selected object. Select an object and click **Drop** to initiate dropping the selected object.
Object Permissions and **System Permissions**	For details on using these tabs, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Views Editor (IBM DB2 Z/OS)

The Views Editor lets you view columns for a view and work with the dependencies and permissions for the view.

To edit a view

1 Open an editor on the view. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Displays all columns for the view. Details for each column include the **Column Name**, the **Datatype** (and if applicable, with the precision in parentheses), and whether or not **Nulls** are allowed for that column.
Definition	Lets you view and modify the CREATE VIEW DDL that will implement any changes you make in this editor.
Comment	For details on using this tab, see Adding a Comment to an object.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Permissions	For details on using this tab, see Working with Privileges and Permissions.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Microsoft SQL Server Object Editors

An Object Editor is provided for all supported Microsoft SQL Server objects. To see an Editor for a specific object, click the corresponding link below:

- Alerts Editor (DBArtisan - SQL Server)
- Asymmetric Keys Editor (SQL Server)
- Backup Devices Editor (DBArtisan - SQL Server)
- Certificates Editor (SQL Server)
- Check Constraints Editor (SQL Server)
- Databases Editor (SQL Server)
- Database Triggers Editor (SQL Server)
- Defaults Editor (SQL Server)
- Extended Procedures Editor (SQL Server)
- Foreign Keys Editor (SQL Server)
- Full-text Catalogs Editor (SQL Server)
- Full-text Indexes Editor (SQL Server)
- Functions Editor (SQL Server)
- Indexes Editor (SQL Server)
- Jobs Editor (DBArtisan - SQL Server)
- Linked Servers Editor (DBArtisan - SQL Server)
- Logins Editor (SQL Server)
- Operators Editor (DBArtisan - SQL Server)
- Partition Functions Editor (SQL Server)
- Partition Schemes Editor (SQL Server)
- Primary Keys Editor (SQL Server)
- Procedures Editor (SQL Server)
- Remote Servers Editor (DBArtisan - SQL Server)
- Roles Editor (SQL Server)
- Rules Editor (SQL Server)
- Schemas Editor (SQL Server)

  - Sequences Editor (SQL Server)
  - Server Roles Editor (DBArtisan - SQL Server)
  - Server Triggers Editor (DBArtisan - SQL Server)
  - Symmetric Key Editor (SQL Server)
  - Synonyms Editor (SQL Server)
  - Tables Editor (SQL Server)
  - Triggers Editor (SQL Server)
  - Unique Keys Editor (SQL Server)
  - Users Editor (SQL Server)
  - User Datatypes Editor (SQL Server)
  - User Messages Editor (SQL Server)
  - Views Editor (SQL Server)

For access to object editors for different supported DBMS platforms and an introduction to editor usage, see Modifying objects using editors.

Alerts Editor (DBArtisan - SQL Server)

The Alerts Editor lets you modify the definition of a SQL Agent alert.

To edit an alert

1 Open an editor on the alert. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

<table>
<tr><th>Tab</th><th>Settings and tasks</th></tr>
<tr><td><b>Properties</b></td><td>This tab/panels lets you view the <b>ID</b>, <b>Name</b>, and <b>Error Description</b> settings. It lets you modify the <b>Enabled</b>, <b>Type</b>, <b>Error #</b>, <b>Severity</b>, <b>Database</b>, and <b>Event Description</b> settings. For details on these settings, see Alerts (SQL Server) - Properties. The <b>History</b> and <b>Occurrence</b> groups display the following properties:
<table>
<tr><td><b>Last Occurrence Date</b></td><td>The date and time that this alert was last raised.</td></tr>
<tr><td><b>Last Response Date</b></td><td>The date and time that this alert was responded to by the SQLServerAgent service.</td></tr>
<tr><td><b>Occurrence Count</b></td><td>The number of times this alert occurred since last being reset.</td></tr>
<tr><td><b>Last Reset Date</b></td><td>The date and time that the <b>Occurrence Date</b> was last reset.</td></tr>
<tr><td><b>Reset Count</b></td><td>Use this button to set the <b>Occurrence Date</b> to zero.</td></tr>
</table></td></tr>
<tr><td><b>Response</b></td><td>This tab lets you provide job execution details, set content of email, pager, and net send notifications, and select operators notified, for this alert. For details on the specific properties, see Alerts (SQL Server) - Response.</td></tr>
<tr><td><b>DDL View</b></td><td>For details on using this tab, see Viewing the SQL/DDL for an Object.</td></tr>
</table>

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Asymmetric Keys Editor (SQL Server)

The Asymmetric Keys Editor lets you view and modify asymmetric key properties.

To edit an asymmetric key

1 Open an editor on the asymmetric key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	This tab lets you view all properties assigned when the asymmetric key was created. For more information, see Asymmetric Keys (SQL Server) - Properties. In addition, this tab lets you modify the following properties: **Authorization Owner** - the AUTHORIZATION argument. **Change Encryption Mechanism** - activates the **Encryption Type** control, letting you change the encryption method used for this certificate. **Encryption Type** - lets you select among encryption using the MASTER KEY, a USER_PASSWORD, or NONE. **Encryption Password** and **Decryption Password** - available if you select an **Encryption Type** of USER_PASSWORD, they lets you provide the specific encryption/decryption passwords.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Backup Devices Editor (DBArtisan - SQL Server)

The Backup Device editor lets you view key properties for a backup device.

To edit a backup device

1 Open an editor on the backup device. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Definition	Displays the following property values: **Name -** the logical name. **Device Type -** DISK or TAPE. **Physical Name -** A file name (sqltable.dat for example) or full path (a:\sqltable.dat for example) identifying the device.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Certificates Editor (SQL Server)

The Certificates Editor lets you view and modify check certificate properties.

To edit a certificate

1 Open an editor on the certificate. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	This tab lets you view all properties assigned when the certificate was created. For more information, see Certificates (SQL Server) - Properties. In addition, this tab lets you modify the following properties: **Authorization Owner -** the AUTHORIZATION argument. **Active for Begin Dialog -** the ON/OFF values for a ACTIVE FOR BEGIN_DIALOG = option. **Change Encryption Mechanism** - activates the **Encryption Type** control, letting you change the encryption method used for this certificate. **Encryption Type -** lets you select among encryption using the MASTER KEY, a USER_PASSWORD, or NONE. **Encryption Password -** available if you select an **Encryption Type** of USER_PASSWORD, it lets you provide the specific encryption password.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Check Constraints Editor (SQL Server)

The Check Constraints Editor lets you view and modify check constraints properties and edit the check constraint expression.

To edit a check constraint

1 Open an editor on the check constraint. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Definition	Use the **Enabled** check box to enable or disable the check constraint. Use the **Not For Replication** check box to enable or disable the check constraint for replication. You can also edit the condition in the **Check Condition** box. The **Table Column** button opens a dialog that lets you select and paste column names into the check condition expression.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Databases Editor (SQL Server)

The Databases Editor lets you manage basic properties, log and data files for a database.

To edit a database

1 Open an editor on the database. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Options	Lets you work with options in the following categories:	
	Creation Properties	Lets you choose values for the **Compatible Level** and **Owner** settings. **NOTE:** The Compatible level option sets certain database behaviors to be compatible with the specified earlier version of Microsoft® SQL Server. The compatibility level affects the behaviors in the specified database, not the entire server.To set database options for all future databases, set the database options on the model database. **NOTE:** When changing the database owner (dbo) select the check box to transfer the existing aliases of users who could act as the old dbo (including their permissions) to the new dbo.
	Properties	Lets you set the following properties: **ANSI null default**, **ANSI nulls**, **ANSI padding**, **ANSI warnings**, **arithabort**, **auto create statistics**, **auto update statistics**, **auto close**, **auto shrink**, **concat null yields nul**, **cursor close on commit**, **db chaining**, **dbo use only**, **default to local cursor**, **merge publish**, **numeric roundabout**, **offline**, **published**, **quoted identifier**, **read only**, **recursive triggers**, **select into/bulkcopy/pllsort**, **single user**, **subscribed**, **torn page detection**, and **trunc log on chkpt**.
Placement	Displays the currently defined data files for the database. Optionally, you can:	
		Select a file from the list on the left and in the **Device File Properties** group, modify the **Size**, **FileGrowth Rate**, **Max Size**, and **Unlimited Max Size** settings.
		Click the **New** button, provide a **Device File Name** for the new file, and use the other settings in the **Device File Properties** group to provide additional details for the file.
		Select a file from the list on the left and click **Delete** to remove the file.
Transaction Log	Displays the currently defined transaction logs for the database. Optionally, you can:	
		Select a transaction log from the list on the left and in the **Log Device Properties** group, modify the **Size**, **FileGrowth Rate**, **Max Size**, and **Unlimited Max Size** settings.
		Click the **New** button, provide a **Device File Name** for the new transaction log, and use the other settings in the **Log Device Properties** group to provide additional details for the transaction log.

	Select a transaction log from the list on the left and click **Delete** to remove the transaction log.
Space	Lets you view pie charts showing the data space usage and the transaction log (if available) space usage for the database.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Database Triggers Editor (SQL Server)

The Database Triggers Editor lets you enable or disable a database trigger and modify the body of the generated CREATE TRIGGER statement.

To edit a database trigger:

1 Open an editor on the database trigger. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Select the **Enabled** check box to enable the database trigger or deselect it to disable the trigger. In addition to the name and creation date details, this tab also lets you view the **Trigger Timing** and **Encrypted** property settings you chose when creating the trigger. For details on these properties, see Database Triggers (SQL Server) - Properties.
Definition	Lets you modify the CREATE TRIGGER statement.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Working with Object Dependencies.
Trigger Events	Lets you view the DDL events you selected to fire this trigger.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Defaults Editor (SQL Server)

The Defaults Editor lets you change the owner and value of a default.

To edit a default

1 Open an editor on the default. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you modify the **Value** of the Default and the owning **Schema**.
Dependencies	For details on using this tab, see Working with Object Dependencies.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Extended Procedures Editor (SQL Server)

The Procedures Editor lets you modify the library name for an extended procedure. It also lets you work with dependencies and permissions.

Note: Extended Procedures are only available on the master database.

To edit an extended procedure

1 Open an editor on the extended procedure. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you provide a dynamic-link library (DLL) **Library Name** for the extended procedure.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Permissions	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Foreign Keys Editor (SQL Server)

The Foreign Keys Editor lets you manage the columns and basic properties of a foreign key.

To edit a foreign key

1 Open an editor on the foreign key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Column Mapping	The existing column mapping for the foreign key is represented by selected columns in the **Main Table** and **Referenced Table** lists. Additional candidates in the **Referenced Table** list are indicated by enabled column check boxes. If necessary, use the **Table** dropdown in the **Referenced Table** group to choose a new table for this foreign key. Select or deselect columns in the **Main Table** list and **Referenced Table** list to form the referential constraint between the two tables.
Properties	Lets you specify **Enabled**, **With NOCHECK**, and **Not For Replication** properties. This tab also lets you select a **Delete rule** and an **Update rule** (NONE, SET NULL, SET DEFAULT, CASCADE).
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Full-text Catalogs Editor (SQL Server)

The Full-text Catalogs Editor lets you work with the basic definition and dependencies of a full-text catalog.

To edit a full-text catalog:

1 Open an editor on the full-text catalog. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view the **Parent Type**, **Parent Schema**, **Parent Name**, **Key Index**, and **Full-text Catalog Name** properties for the partition scheme. It also lets you modify the **Filegroup Name** (SQL Server 2008 ^), **Change Tracking**, and **Stoplist** (SQL Server 2008 ^) properties. For more information on these properties, see Full-text Catalogs (SQL Server) - Properties.
Dependencies	For details on using this tab, see Working with Object Dependencies.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Full-text Indexes Editor (SQL Server)

The Full-text Indexes Editor lets you work with the basic definition and dependencies of a full-text index.

To edit a full-text index:

1 Open an editor on the full-text index. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view the **Name**, property and modify the **Accent Sensitive**, **Default**, and **Authorization Owner** properties. For more information on these properties, see Full-text Indexes (SQL Server) - Properties.
Dependencies	For details on using this tab, see Working with Object Dependencies.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Functions Editor (SQL Server)

The Functions Editor lets you view and modify function definitions and dependencies.

To edit a function

1 Open an editor on the function. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you modify the owning **Schema** of the function.
Definition	Lets you view and modify the dynamic-link library (DLL) or view the data definition language (DDL).
Dependencies	For details on using this tab, see Working with Object Dependencies.
Permissions	For details on using this tab, see Working with Privileges and Permissions.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Indexes Editor (SQL Server)

The Indexes Editor lets you manage columns and basic properties of an index and view index statistics.

To edit an index

1 Open an editor on the index. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Columns	Lets you manage columns that make up the index. On opening, this tab shows the existing columns. For each column, the datatype (and if applicable the precision in brackets) and whether the table definition permits nulls in the target table column. Optionally you can: Change the **Sort** order of a column. Click the **New** button and select a column name from the dropdown, to add a column to the index. Select a column and click the **Drop** button to delete the column from the index. Select a column and use the arrow buttons to reorder the columns in the index.
Properties	Lets you work with settings in the following categories: **Creation** In addition to displaying identifying information, you can modify the **Build Online** and **Max degree of parallelism** properties. For details on these properties, see Indexes Wizard (SQL Server). **Attributes** Lets you set **Index Type**, **Clustered**, **Ignore Duplicate Key** (for **Index Type** of UNIQUE), **Statistics Recompute**, **Allow Row Locks**, and **Allow Page Locks** properties. For details on these properties, see Indexes Wizard (SQL Server). **NOTE:** You cannot reorganize an index (primary key, or unique key) that has an **Allow Page Locks** property set to FALSE. For information on reorganizing indexes, see Reorganize (SQL Server indexes, primary keys, and unique keys). **Storage** Lets you set **Partitioned**, **Partition Scheme**, **Parttition Column, Fill Factor**, **File Group**, **Pad Index**, and **Sort in Tempdb** settings.
Statistics	Lets you view statistics in the following categories: **Page Statistics** **Data Pages**, **Reserved Pages**, **Used Pages**, and **Total Pages Modified**. **Row Statistics** **Maximum Row Size**, **Minimum Row Size**, **Max Size of Non-Leaf Index Row**, and **Total Rows Modified**.

DDL View For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Jobs Editor (DBArtisan - SQL Server)

The Jobs Editor lets you modify the definition of a SQL Agent job.

To edit a job

1 Open an editor on the job. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	This tab/panels lets you specify an **Enabled**/disabled status and select a **Category**, as well as specify notify and delete levels for the job. For details on these settings, see Jobs (SQL Server) - Properties.
Steps	This tab/panels lets you work with the steps for a job, including subsystem specifics, success/failure/ retry details, and logging and file specifications. For details on these settings, see Jobs (SQL Server) - Steps.
Schedules	This tab/panels lets you create new schedules for this job, add schedules that were defined in other job definitions, delete schedules currently associated with the current job definition, and display all jobs using a selected schedule. For detailed instructions, see Jobs (SQL Server) - Schedules.
Alerts	On opening, this tab provides a list of all alerts that specify this job as the target job to be executed in response to those alerts. You can create new alert definitions and edit existing definitions from the list. When you execute the job that creates or edits this job, DDL to create all alerts on this tab is submitted. For detailed instructions, see Jobs (SQL Server) - Alerts.
Target	Lets you specify the server or servers where the job executes. For details, see Jobs (SQL Server) - Target.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Linked Servers Editor (DBArtisan - SQL Server)

The Linked Servers Editor lets you manage basic properties and logins for a linked server.

To edit a linked server

1 Open an editor on the linked server. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Properties	This panel lets you work with settings in the following categories:	
	General	**Is SQL Server** - Setting this check box indicates a linked server defined through Microsoft SQL Server. Any tables retrieved from the server are from the default database defined for the login on the linked server. Setting this check box disables the remaining OLE DB Provider settings. **Product Name** - The product name of the OLE DB datasource to which you are linking. **Datasource** - The name of the OLE DB provider managing access to the linked server. **OLE DB Provider Name** - Select the name of the OLE DB provider managing access to the specified linked server. **Provider Location** - The OLE DB location property corresponding to the linked server. **Provider Connection String** - The OLD DB provider-specific connection string that identifies a unique datasource. **Catalog** - The name of the catalog to be used when making a connection to the OLE DB provider.
	Server Options	**Data Access Server** - Indicates whether the target linked server is enabled for distributed query access. **RPC Out Server** - Lets you select the server with RPC encryption. **RPC** - Enables RPC encryption from the server. **Collation Compatible Server** - Indicates Change DIstributed Query execution against linked servers. If you select this option, Microsoft SQL Server assumes that all characters in the linked server are compatible with the local server, with regard to character set and collation sequence (or sort order). This option is for Microsoft SQL Server 8.0 or later. **Lazy Schema Validation** - This option lets you optimize performance by ensuring the query processor does not request metadata for any of the linked tables until data is actually needed from the remote member table. This option is for Microsoft SQL Server 8.0 or later. **Use Remote Collation** and **Collation Name** - Let you indicate that the linked server should use remote collation and select the name of the collation. This option is for Microsoft SQL Server 8.0 or later. **Query Timeout** - Type the query timeout value. This option is for Microsoft SQL Server 8.0 or later. **Connection Timeout** - Type the connection timeout value. This option is for Microsoft SQL Server 8.0 or later.

Logins	Includes a table listing every user who has a login on the target linked server and their local login names. Optionally, you can: Create a new login by clicking **Add linked login** button and providing the following information in the dialog that opens: **SQL Server Authentication -** Use SQL Server authenticated login credentials to login to linked server. If you choose this option, you must provide a **Local Login Name**. **No Remote Login -** Indicates that no remote login or password are required login to the linked server. **All Users -** All users will use the same login when connecting to the linked server. If you choose this option, you must provide a **Linked Login Name** and a **Password**. **Local Login Name -** Lets you select a local login name. **Linked Login Name -** Lets you type in a linked login name. **Password -** Lets you set a password. Select a login, click the **Modify** button, and change the login names and password. Select a login and click the **Drop** button to delete that login.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Logins Editor (SQL Server)

The Logins Editor lets you manage basic properties for a login, associated users and roles, and accounting.

To edit a login:

1 Open an editor on the login. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Properties	Lets you view or modify the following properties set when creating the login: **Name**, **Default Database**, **Default Language**, **Password**, **Check Policy**, **Check Expiration**, **Must Change**, **Certificate**, and **Asymmetric Key**. For detailed information on these settings, see Logins Wizard (SQL Server). In addition, when editing an existing login, you have access to the following settings:	
	Currently Logged In	Displays whether the user is currently logged in.
	Enabled	Enabled, an ALTER LOGIN... ENABLE is submitted following the CREATE LOGIN statement issued by the editor. Disabled, an ALTER LOGIN... DISABLE is submitted
Server Roles	Lets you add the login as a member of one or more fixed server roles. For more information, see Logins (SQL Server) - Server Roles.	
Users	Lets you add and remove user accounts for this login to specified databases. For more detailed information, see Logins (SQL Server) - Users.	
DDL	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Operators Editor (DBArtisan - SQL Server)

The Operators Editor lets you modify the definition of a SQL Agent operator.

To edit an operator

1 Open an editor on the operator. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	This tab/panel lets you change the operator's enabled/disabled status, modify contact information, and change the pager duty schedule. For details, see Jobs (SQL Server) - Properties.
Notifications	This tab lets you modify the set of SQL Agent alerts for which this operator will be notified. For details, see Operators (SQL Server) - Notifications.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Partition Functions Editor (SQL Server)

The Partition Functions Editor lets you view the basic definition and work with dependencies of a partition function.

To edit a partition function:

1 Open an editor on the partition function. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view the **Name**, **Input Parameter Type**, **Range**, and **Function Values** properties for the partition function. For more information on these properties, see Partition Functions (SQL Server) - Properties.
Dependencies	For details on using this tab, see Working with Object Dependencies.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Partition Schemes Editor (SQL Server)

The Partition Schemes Editor lets you view the basic definition and work with dependencies of a partition scheme.

To edit a partition scheme:

1 Open an editor on the partition scheme. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view the **Name**, **Partition Function**, and **Filegroup Name** properties for the partition scheme. For more information on these properties, see Partition Schemes (SQL Server) - Properties.
Dependencies	For details on using this tab, see Working with Object Dependencies.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Primary Keys Editor (SQL Server)

The Primary Keys Editor lets you manage the columns and basic properties for a primary key and lets you view page and row statistics.

To edit a primary key

1 Open an editor on the primary key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Columns	Lets you manage columns that make up the primary key. On opening, this tab shows the existing columns. For each column, the listing shows the datatype (and if applicable the precision in brackets) and whether the table definition permits nulls in the target table column. Optionally you can: Change the **Sort** order of a column. Click the **New** button and select a column name from the dropdown, to add a column to the primary key. Select a column and click the **Drop** button to delete the column from the primary key. Select a column and use the arrow buttons to reorder the columns in the primary key.
Properties	Lets you work with settings in the following categories: **Creation** In addition to displaying identifying information, you can modify the **Build Online** and **Max degree of parallelism** properties. For details on these properties, see Primary Keys Wizard (SQL Server). **Attributes** **Clustered** - Indicates whether the target primary key is clustered. **Ignore Duplicate Key** - Indicates whether the target primary key ignores duplicate key values. If you select his option, the transaction that generated the duplicate key values can continue. **Statistics Recompute** - Indicates that index statistics are automatically recomputed as the index is updated. Microsoft does not recommend this. **Storage** **File Group** - Lets you select a file group. **Fill Factor** - Lets you specify the fill factor that specifies how full each index page can be. If no fill factor is specified, Microsoft SQL Server uses the database's default fill factor. **Pad Index** - If you specified a Fill factor of more than 0 percent, and you selected the option to create a unique index, you can specify to use the same percentage you specified in Fill Factor as the space to leave open on each interior node. By default, Microsoft SQL Server sets a 2 row index size.
Statistics	Lets you view statistics in the following categories: **Page Statistics** **Data Pages**, **Reserved Pages**, **Used Pages**, and **Total Pages Modified**.

	Row Statistics	**Maximum Row Size**, **Minimum Row Size**, **Max Size of Non-Leaf Index Row**, and **Total Rows Modified**.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Procedures Editor (SQL Server)

The Procedures Editor lets you manage properties, the definition, dependencies, and permissions for a procedure.

To edit a procedure

1 Open an editor on the procedure. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you change the owning **Schema**.
Definition	Lets you view the SQL code for the procedure.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Permissions	For details on using this tab, see Working with Privileges and Permissions.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Remote Servers Editor (DBArtisan - SQL Server)

The Remote Server Editor lets you manage the connection and security properties for a remote server, and manage the associated logins.

To edit a remote server

1 Open an editor on the remote server. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Properties	Lets you work with the following settings:	
	Server Type	Displays the server type, LOCAL or REMOTE. The local server routes all commands to the remote server. To communicate with a remote server, you log into a local server and submit your remote procedure calls for remote servers. The remote server receives remote procedure calls from the local server and processes them accordingly.
	Publication Server	Specifies database publishing. When replication is installed, this option permits the tables of a database to be published for replication. When selected, this option enables publication. When not selected, this option disables publishing, drops all publications, and unmarks all transactions that were marked for replication in the transaction log.
	Distribution Server	Enables the remote server to be a distribution server.
	Subscription Server	Specifies database subscriptions. When selected, the database can be subscribed for publication. When not selected, the database cannot be subscribed for publication.
	Publisher/Subscriber	Enables the remote server to be a publisher/subscriber server.
	Collation Compatible Server	Indicates Change DIstributed Query execution against remote servers. If you select this option, Microsoft SQL Server assumes that all characters in the remote server are compatible with the local server, with regard to character set and collation sequence (or sort order).
	Data Access Server	Indicates whether the target remote server is enabled for distributed query access.
	RPC Out Server	Enables the remote server to execute stored procedures on the local server using a remote procedure call (RPC).

Logins	Includes a table listing the remote name, local name, and trusted status of logins for a remote server.Optionally you can: Click **Add** to open a dialog that lets you add a new login. The dialog displays or lets you provide: **Remote Server Name -** Displays the remote server name. **Remote Login Name** - Lets you specify a remote login name. **All remote users** - Select this box if you want to use this name for all remote logins. **Local Login Name** - Lets you select a local user name. **Is Trusted** - Lets you specify this remote login as a trusted login. Select a login, click **Modify**, and use the table above as a guide to modifying login options. Select a login, and click **Drop** to delete the login.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Roles Editor (SQL Server)

The Roles Editor lets you manage the users for a role.

To edit a role

1 Open an editor on the role. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Displays the **Name**, and either a **Role Type** of STANDARD and an **Authorization Owner** or a **Role Type** of APPLICATION and a **Password**.
Users	Let you manage users for the role. A user becomes associated with an application role after running the target application. Use the **Join Rule** and **Leave Role** buttons to move users between the **Users In Role** and **Users Not In Role** lists.
Object Privileges and **System Privileges**	For details on using these tabs, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Rules Editor (SQL Server)

The Rules Editor lets you view or modify basic properties of a rule.

To edit a rule

1 Open an editor on the rule. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you specify whether the rule is **Enabled** and view the **Fire On Insert**, **Fire On Update**, **Fire On Delete**, and **Encrypted** settings.
Definition	Lets you view and modify the SQL code that will implement any changes you make using this editor.
Dependencies	For details on using this tab, see Working with Object Dependencies.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Schemas Editor (SQL Server)

The Schemas Editor lets you change the owner of a group and manage the objects contained in the schema.

To edit a user

1 Open an editor on the user. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you modify the **Owner** of the schema.
Permissions	For details on using this tab, see Working with Privileges and Permissions.
Objects	Displays the specific objects owned by the schema, grouped under object type folders. Optionally, you can: Select an object and click **Edit** to open an object editor on that object. Select an object and click **Drop** to initiate dropping the object.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Sequences Editor (SQL Server)

This editor lets you view creation properties for a sequence as well as modify the scope, starting value and increment value.

To edit a sequence:

1 Open an editor on the sequence. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view the **Schema**, **Name**, **Create Date**, **Data Type Schema**, **Base Data Type**, **Data Type Name**, and **Data Type Precision** properties for the synonym. Lets you modify the **Start With**, **Increment By**, **Minimum Value**, **Maximum Value**, **Cycle**, **Cache**, and **Cache Size** properties. For more information on these properties, see Sequences (SQL Server) - Properties. This tab also includes the following, display-only properties: **Current Value -** The value most recently returned by the NEXT VALUE FOR function or the last value resulting from an `sp_sequence_get_range` procedure execution. **Is Exhausted -** Indicates that the sequence object has reached the **Maximum Value** and **Cycle** is not set. The NEXT VALUE FOR function will return an error until the sequence is restarted using ALTER SEQUENCE.
Dependencies	For details on using this tab, see Working with Object Dependencies.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Server Roles Editor (DBArtisan - SQL Server)

The Server Roles Editor lets you generate a set of sp_addsrvrolemember calls. This lets you change the login members of SQL Server's fixed server roles: bulkadmin, dbcreator, diskadmin, processadmin, securityadmin, serveradmin, setupadmin, and sysadmin.

To edit the login members of a server role:

1 Open an editor on the server role. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Logins	Select all check boxes associated with logins that are to be members of this server role.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Server Triggers Editor (DBArtisan - SQL Server)

The Server Triggers Editor lets you enable or disable a logon-based or other event-based server trigger and modify the body of the generated CREATE TRIGGER statement.

To edit a server trigger:

1 Open an editor on the server trigger. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Select the **Enabled** check box to enable the server trigger or deselect it to disable the trigger. In addition to the name and creation date details, this tab also lets you view the **Trigger Timing**, **Fire On Logon**, and **Encrypted** property settings you chose when creating the trigger. For details on these properties, see Server Triggers (SQL Server) - Properties.
Definition	Lets you modify the CREATE TRIGGER statement.
Trigger Events	Lets you view the DDL events you selected to fire this trigger.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Symmetric Key Editor (SQL Server)

The Symmetric Key Editor lets you view creation details for a key and modify its encryption mechanisms.

To edit a symmetric key:

1 Open an editor on the symmetric key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view the **Authorization Owner**, **Name**, **Create Date**, **Update Date**, **Provider Name**, **Key Algorithm**, and **Key Length** properties. For more information, see Symmetric Keys (SQL Server) - Properties.
Encryption Mechanisms	Lets you manage the encryption mechanisms for this key. For detailed instructions, see Symmetric Keys (SQL Server) - Properties.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Synonyms Editor (SQL Server)

The Synonyms Editor lets you view the basic definition and work with dependencies of a partition scheme.

To edit a synonym:

1 Open an editor on the synonym. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view the **Schema**, **Name**, **Server**, **Database**, **Referenced Object Type**, **Referenced Object Owner**, and **Referenced Object Name** properties for the synonym. For more information on these properties, see Synonyms (SQL Server) - Properties. This tab also includes a **CreateDate** property showing the data and time the synonym was created.
Dependencies	For details on using this tab, see Working with Object Dependencies.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Tables Editor (SQL Server)

The Tables Editor lets you manage columns, basic properties, indexes, and constraints for a table and view space usage details.

To edit a table

1 Open an editor on the table. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Columns	Displays the currently defined columns in the table. Optionally, you can:	
		Select a column, and in the **Property**/**Value** list modify property values for that column.
		Click **Add Column**, provide a name for the new column, and set property values for the column.
		Select a column and click **Delete** to remove the column from the table.
Properties	Lets you work with settings in the following categories:	
	Physical Storage	Lets you view the **Partitioned**, **Text Image Filegroup**, and **FileStream On** (.version 2008^) properties and modify the **Filegroup** property. For details on these properties, see Tables (SQL Server) - Properties.
	Full-Text Index	Lets you view whether Full-Text Indexing is installed and active. The full-text index feature provides support for sophisticated word searches in character string data. A full-text index stores information about significant words and their location within a given column. This information is used to quickly complete full-text queries that search for rows with particular words or combinations of words. This feature is available for Microsoft SQL Server 8.0 or later.
	Text In Row	Lets you enable Text In Row Data and specify a limit.
	Filetable (Version 2012^)	Lets you view the **As Filetable**, **Filetable Collate Filename**, **FiletablePK Constraint Name**, F**iletable StreamID Unique Constraint Name**, and **Filetable Fullpath Unique Constraint Name** properties and lets you modify the **Filetable Directory Name** property. For details on these properties, see Tables (SQL Server) - Properties. This group also lets you modify an **Enabled** property. That control specifies whether system-defined constraints are enabled.
Indexes	Displays any indexes currently defined for the table. Optionally, you can:	
		Click **Add** to open an Index editor, letting you create a new index for this table. For more information, see Indexes Wizard (SQL Server).

	Select an index and click **Edit** to open an wizard that lets you modify that index. For more information, see Indexes Editor (SQL Server). Select an index and click **Drop** to open a dialog that lets you confirm that you want to drop the index. When dropping an index with the Clustered option enabled, the confirmation dialog includes an **Online** option that lets you specify an online drop (ONLINE-ON clause).
Constraints	Lets you manage constraints for the table. Constraints are grouped by type, under folders. Optionally take one of the following actions: Select a constraint type folder and click **Add** to open a dialog that lets you add a constraint of that type. Select a constraint and click **Edit** to open a dialog that lets you modify the constraint details. Select a constraint and click **Drop** to open a dialog that lets you confirm that you want to drop the index. When dropping a clustered unique key or clustered primary key, the confirmation dialog includes an **Online** option that lets you specify an online drop (ONLINE-ON clause).
Space	Lets you view pie charts showing the space usage for the table. Optionally you can double-click a slice in the pie chart for detailed statistics.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Permissions	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Triggers Editor (SQL Server)

The Triggers Editor lets you enable and disable a trigger, view and modify the CREATE TRIGGER statement that will implement changes, and manage dependencies.

To edit a trigger

1 Open an editor on the trigger. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you set the **Enabled** property. It also lets you view the **Trigger Timing** (AFTER, INSTEAD OF), **Fire On Insert**, **Fire On Update**, **Fire On Delete**, and **Encrypted** properties.
Definition	Lets you modify the CREATE TRIGGER body for a trigger. To modify a trigger, edit the text of the trigger body in the Trigger Text area. The existing trigger is automatically dropped and then recreated.
Dependencies	For details on using this tab, see Working with Object Dependencies.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Unique Keys Editor (SQL Server)

The Unique Keys Editor lets you manage columns and properties for a unique key and view associated statistics.

To edit a unique key

1 Open an editor on the unique key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Columns	Lets you manage columns that make up the unique key. On opening, this tab shows the existing columns. For each column, the listing displays the datatype (and if applicable the precision in brackets) and whether the table definition permits nulls in the target table column. Optionally you can:	
		Change the **Sort** order of a column. Click the **New** button and select a column name from the dropdown, to add a column to the index. Select a column and click the **Drop** button to delete the column from the index. Select a column and use the arrow buttons to reorder the columns in the index.
Properties	Lets you work with settings in the following categories:	
	Creation	In addition to displaying identifying information, you can modify the **Build Online** and **Max degree of parallelism** properties. For details on these properties, see Unique Keys Wizard (SQL Server).
	Attributes	**Clustered** - Indicates whether the target index is clustered. **Ignore Duplicate Key** - Indicates whether the target primary key ignores duplicate key values. If you select his option, the transaction that generated the duplicate key values can continue. **Statistics Recompute** - Indicates that index statistics are automatically recomputed as the index is updated. Microsoft does not recommend this.
	Storage	**File Group** - Lets you specify the filegroup on which to place the index. This is for Microsoft SQL Server 7.0 or later. **Fill Factor -** Lets you specify a percentage that indicates how full Microsoft SQL Server should make the leaf level of each index page during index creation. When an index page fills up, Microsoft SQL Server must take time to split the index page to make room for new rows, which is quite expensive. For update-intensive tables, a properly chosen Fill factor value yields better update performance than an improper Fill factor value. **Pad Index -** If you specified a Fill factor of more than 0 percent, and you selected the option to create a unique index, you can specify to use the same percentage you specified in Fill Factor as the space to leave open on each interior node. By default, Microsoft SQL Server sets a 2 row index size.

Statistics	Displays statistics in the following categories: **Page Statistics**: **Data Pages**, **Pages Reserved**, **Used Pages**, and **Total Pages Modified**. **Row Statistics**: **Maximum Row Size**, **Minimum Row Size**, **Max Size of Non-Leaf Index Row**, and **Total Rows Modified Since Last**.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Users Editor (SQL Server)

The Users Editor lets you manage properties, assign roles, and manage object bindings for a user.

To edit a user

1 Open an editor on the user. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view the **Without Login**, **Login Name**, **Name**, **User Type**, **Certificate**, **Asymmetric Key**, and **Default Schema** properties.
Roles	Lets you assign roles to a user by selecting the check boxes associated with the roles to be assigned.
Objects	Lets you manage database objects (Defaults, Indexes, Procedures, Rules, Tables, Triggers, User Datatypes, Views) associated with the user. Objects are organized in a tree structure with folders containing the objects. Optionally, take one of the following actions: Select an object and click **Edit** to open an object editor on the selected object. For information on each editor, use the links in Microsoft SQL Server Object Editors. Select an object and click **Drop** to initiate dropping the selected object.
Object Permissions and **System Permissions**	For details on using these tabs, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

User Datatypes Editor (SQL Server)

The User Datatypes Editor lets you modify the base datatype, rule and default bindings, and referencing or referenced objects for a user datatype.

To edit a user datatype

1 Open an editor on the user datatype. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Properties	In addition to letting you change the **Schema**, this tab lets you work with settings in the following categories:	
	Base Datatype	Lets you modify the **Type** and **Allow Nulls** properties. Depending on the **Type** selected, **Precision**, **Scale**, or **Size** properties are available as appropriate.
	Bindings	**Default Binding -** Lets you select a default. For information on creating defaults, see Defaults Wizard (SQL Server). **Rule Binding -** Lets you select a rule. For information on creating rules, see Rules Wizard (SQL Server).
Usage	Lets you manage database objects referencing or referenced by a user datatype. Objects are grouped within object type folders. Optionally, you can:	
		Select an object and click **Edit** to open an object editor on the selected object.
		Select an object and click **Drop** to initiate dropping the selected object.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

User Messages Editor (SQL Server)

Note: User Messages are only on the master database.

The User Messages Editor lets you view or modify basic properties of a user message. It also lets you add, edit, or delete individual language versions of the message.

To edit a user message

1 Open an editor on the user message. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

<table>
<tr><th>Tab</th><th>Settings and tasks</th></tr>
<tr><td><b>Properties</b></td><td>Lets you work with the following properties:
<table>
<tr><td><b>Message Number</b></td><td>Displays the message number chosen when the user message was created.</td></tr>
<tr><td><b>Severity</b></td><td>Lets you modify the SQL Server error severity level (001-025).</td></tr>
<tr><td><b>Write to NT Event Log</b></td><td>Lets you specify that the message is always written to the Windows NT Event Log.</td></tr>
</table></td></tr>
<tr><td><b>Information</b></td><td>Displays the currently-defined language versions for this message number. Optionally you can:
Create a new language version for this message number. Click the <b>Add new text for the user message</b> button, and in the dialog box that opens, select a <b>Language</b> and provide the <b>Message Text</b>.
Select a language version of the message and click the <b>Modify user message text</b> button. This opens a dialog that lets you modify the <b>Language</b> or <b>Message Text</b>.
Select a language version of the message and click the <b>Remove user message text</b> button.
<b>Note:</b> You cannot create two versions of a message for the same language. NOTE: FOr a given message number there should always be a least one, us_english version of the message.</td></tr>
<tr><td><b>DDL View</b></td><td>For details on using this tab, see Viewing the SQL/DDL for an Object.</td></tr>
</table>

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Views Editor (SQL Server)

The Views Editor lets you view columns and change the schema of a view.

To edit a view

1 Open an editor on the view. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Displays a list of the columns that make up the view. Details include the column **Name**, its **Datatype**, and whether the definition allows **Null**s. This tab also lets you change the **Schema** owning the view.
Definition	Lets you view the SQL for the View.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Permissions	For details on using this tab, see Working with Privileges and Permissions.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

MySQL editors

An Object Editor is offered for all supported MySQL objects. To see an Editor for a specific object, click the corresponding link below:

- Database editor (MySQL)
- Foreign Keys editor (MySQL)
- Functions editor (MySQL)
- Indexes, Primary Keys, and Unique Keys editors (MySQL)
- Tables editor (MySQL)
- Users editor (MySQL)

For access to object editors for different supported DBMS platforms and an introduction to editor usage, see Modifying objects using editors.

Database editor (MySQL)

The Databases Editor lets you manage basic properties and privileges for a database.

To edit a database:

1 Open an editor on the database. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you modify the **Default Character Set** and **Collation** properties. For details on these properties, see Databases - Properties.
Privileges	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Foreign Keys editor (MySQL)

The Foreign Keys Editor lets you manage column mapping, delete, and update actions for a foreign key.

To edit a foreign key:

1 Open an editor on the foreign key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Column Mapping	The existing column mapping for the foreign key is represented by selected columns in the **Main Table** and **Referenced Table** lists. Additional candidates in the **Referenced Table** list are indicated by enabled column check boxes. If necessary, use the **Table** dropdown in the **Referenced Table** group to choose a new table for this foreign key. Select or deselect columns in the **Main Table** list and **Referenced Table** list to form the referential constraint between the two tables.
Properties	Lets you modify the **On Delete** and **On Update** (NO ACTION, RESTRICT, CASCADE, SET NULL) reference options.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Functions editor (MySQL)

The MySQL Functions editor lets you view properties of a basic function declaration.

To edit a function:

1 Open an editor on the function. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view the **Function Name**, **Return Type**, and **Shared Object Library** properties. For details on these properties, see Functions - Properties. Under **Specify Columns** in Index, you can change the selection of columns that make up the index, primary key, or unique key.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Indexes, Primary Keys, and Unique Keys editors (MySQL)

These editors lets you manage columns and view properties of an index, primary key, or unique key.

To edit an index:

1 Open an editor on the index. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view the **Table Name**, **Index Name**, **Constraint Type**, and **Index Type** properties. For details on these properties, see Indexes, Primary Keys, or Unique Keys - Properties. Under **Specify Columns** in Index, you can change the selection of columns that make up the index, primary key, or unique key.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Tables editor (MySQL)

The MySQL Tables editor lets you modify basic properties and view usage statistics for a MySQL table.

To edit a table:

1 Open an editor on the index. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Columns	Displays the currently defined columns in the table. Optionally, you can: Select a column, and in the **Property/Value** list modify property values for that column. Click **Add Column**, provide a name for the new column, and set property values for the column. Select a column and click **Delete** to remove the column from the table.
Properties	Lets you work with the following settings: **Table Name**, **Storage Type**, **Row Format**, **Default Character Set**, **Default Collation**, **Auto Increment**, **Comment**, **Min Rows**, **Max Rows**, **Average Row Length**, **Pack Keys**, **Checksum**, and **Delay Key Write**. For details on these properties, see Tables - Properties.
Indexes	Lets you add, delete, or modify table indexes. For details on available options, see Tables - Indexes.
Foreign Keys	For INNODB tables, this tab lets you add, delete, or modify table foreign keys. For details on available options, see Tables - Foreign Keys.
MERGE tables	For MRG_MyISAM tables, this tab lets you work with the collection of identical MyISAM tables that are to be used as a single table. For more information, see Tables - MERGE Tables.
Privileges	For details on using this tab, see Working with Privileges and Permissions.
Space	Lets you view graphical and numerical table usage statistics. Values include **Data Length**, **Index Length**, **Data Free**, **Max Data Length**, **Rows**, and **Average Row Length**.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Users editor (MySQL)

The MySQL Functions editor lets you work with basic properties, host details, and privileges for a user.

To edit a user:

1 Open an editor on the user. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
User Information	Lets you view the **User Name** property and lets you modify the **Full Name**, **Description**, **Email**, and **Contact Information** properties. For details on these properties, see Users wizard (MySQL).
User Hosts	Add a new host to the list by clicking the **New** button. Use the associated dropdown to select a valid host-name value (such as localhost or %), overwriting the text as necessary to provide a specific quoted identifier. Provide a password in the **Password** and **Confirm** fields and click **Apply**. Remove a selected host from the list by clicking the **Delete** button.
Object Privileges and **System Privileges**	For details on using these tabs, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Oracle Object Editors

An Object Editor is available for all supported Oracle objects. To see an Editor for a specific object, click the corresponding link below.

Note: If an objects has dependent objects, such as tables, triggers, procedures and views, you can view and access their dependent objects in the editor.

- Check Constraints Editor (Oracle)
- Clusters Editor (Oracle)
- Database Links Editor (Oracle)
- Directories Editor (Oracle)
- Disk Groups Editor (DBArtisan - Oracle)
- Foreign Keys Editor (Oracle)
- Functions Editor (Oracle)
- Groups Editor (DBArtisan - Oracle)
- Indexes Editor (Oracle)
- Job Queue Editor (Oracle)
- Jobs Editor (Oracle)
- Libraries Editor (Oracle)
- Materialized Views Editor (Oracle)
- Materialized View Logs Editor (Oracle)
- Outlines Editor (Oracle)
- Package Bodies Editor (Oracle)
- Packages Editor (Oracle)
- Primary Keys Editor (Oracle)
- Procedures Editor (Oracle)
- Programs Editor (Oracle)
- Profiles Editor (Oracle)
- Redo Log Groups Editor (Oracle)
- Roles Editor (Oracle)
- Rollback Segments Editor (Oracle)

  - Schedules Editor (Oracle)
  - Sequences Editor (Oracle)
  - Synonyms Editor (Oracle)
  - Tables Editor (Oracle)
  - Tablespaces Editor (Oracle)
  - Triggers Editor (Oracle)
  - Type Bodies Editor (Oracle)
  - Types Editor (Oracle)
  - Unique Keys Editor (Oracle)
  - Users Editor (Oracle)
  - Views Editor (Oracle)

For access to object editors for different supported DBMS platforms and an introduction to editor usage, see Modifying objects using editors.

Check Constraints Editor (Oracle)

The Check Constraints Editor lets you modify and enable/disable check constraints.

To edit a check constraint

1 Open an editor on the check constraint. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Definition	Use the **Enabled** control to enable/disable the check constraint. You can also edit the condition in the **Check Condition** box. The **Table Columns** button acts a time saver in editing the condition. It opens a dialog that lets you select and paste column names into the check condition expression.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Clusters Editor (Oracle)

The Clusters Editor lets you manage cluster column and table details, view and modify storage and space parameters, and manage performance settings for a cluster.

To edit a cluster

1 Open an editor on the cluster. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Columns	Use the controls in the **General Properties** group to specify a hash or index **Cluster Type** and to specify a **Key Size** for the cluster. If you specify a hash cluster type, use the controls in the **Hash Specifications** group to specify the number of hash keys and to specify use of either the default hash function or a specified expression used as the hash function.	
Storage	Lets you manage storage for a cluster:	
	Data Block Storage	Each transaction that updates a data block requires a transaction entry. **Percent Free** - minimum percentage of free space in a block **Percent Used** - minimum percentage of used space in a block. **Initial transactions** - The initial parameter ensures that a minimum number of concurrent transactions can update a data block, avoiding the overhead of allocating a transaction entry dynamically. **Maximum transactions** - The maximum parameter limits concurrency on a data block.
	Extents	The unit of space allocated to an object whenever the object needs more space. **Initial Extent** - The initial space extent (in bytes) allocated to the object. **Next Extent** - The next extent (in bytes) that the object will attempt to allocate when more space for the object is required. **Minimum Extents** - The appropriate minimum extents value for the object. **Maximum Extents** - The appropriate maximum extents value for the object. **Percent Increase** - Magnifies how an object grows and, can materially affect available free space in a tablespace. Select a value in the corresponding box.
	Optionally, you can modify the **Percent Free**, **Percent Used**, or **Max Transactions** values.	
Performance	Lets you modify the following performance settings:	
	Parallel Query Option group	Lets you specify **Degrees** and **Instances** settings for processing queries using many query server processes running against multiple CPUs, which provides substantial performance gains such as reduction of the query completion time.
	Cache group	The **Cache** setting keeps the blocks in memory by placing it at the most recently used end. This option is useful for small lookup tables.
Space	Lets you view the following usage and the space distribution details:	
	Space Utilization group	Displays the percent of space reserved for future updates.

	FreeLists group	Displays the allocation of data blocks when concurrent processes are issued against the cluster. Identifying multiple freelists can reduce contention for freelists when concurrent inserts take place and potentially improve the performance of the cluster.
	Extents group	The unit of space allocated to an object whenever the object needs more space.
Tables	Lets you view details for each cluster column. Details for each column include the key column in the cluster, the clustered table name, and the key column in the table.	
DDL	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Database Links Editor (Oracle)

The Database Links Editor lets you view connection string information for a database link.

To edit a database link

1 Open an editor on the database link. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Definition	Displays connection string information for a database link. Details include the **Public** setting, the **User** name and **Password** associated with the database link, and a **Connect String**.
DDL	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Directories Editor (Oracle)

The Directories Editor lets you change the path for a directory and modify associated privileges.

To edit a directory object

1 Open an editor on the directory. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Definition	Lets you view or modify the SQL code for the directory.
Privileges	For details on using this tab, see Working with Privileges and Permissions.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Disk Groups Editor (DBArtisan - Oracle)

This editor lets you work with basic properties, individual disks, and attribute templates for an existing disk group. It also lets you work at the individual file and alias level as well as view current operations executing against the disk group.

Note: This functionality is available for Oracle 10 g.

Note: When opening the editor, if this is the first time you have initiated a disk group action since connecting, you are prompted to choose between a connection to the ASM instance or to continue using the default datasource connection. For details on the implications, see Manage ASM Connection (DBArtisan).

Note: If you do not have a connection to the ASM instance, you van view but cannot modify property values in the Disk Groups Editor.

To edit a disk group:

1 On a connected Oracle datasource, expand the **Instance** and **ASM** nodes.

2 In the right-hand pane, right-click a disk group and select **Edit** from the context menu.

3 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Later versions of Oracle offer different properties for this object type: 10G This tab lets you view the **Name**, **Redundancy Level**, and **Group State** values you chose on creating the disk group. For details on these properties, see Disk Groups (Oracle) - Properties. A **Group Number** for the disk group is also displayed. The **General** group displays properties corresponding to disk group attributes or information available from V$ASM views: **Block Size**, **Total MB**, **Free MB**, **Required Mirror Free MB**, **Usable File MB**, **Offline Disks**, and **Is Unbalanced**. The **Attributes** group displays **Allocation Unit Size**, **Compatibility**, **Database Compatibility**, and **Sector Size** properties. 11GR1 As of Oracle 11GR1, the **Compatibility** and **Database Compatibility**, properties can be modified. For details on these properties, see Disk Groups (Oracle) - Properties. A **Disk Repair Time** property can also be modified. 11GR2 The **File Access Control Enabled** and **Access Control Unmask** properties let you add ACCESS_CONTROL.ENABLED (whether Oracle ASM File Access Control is enabled for the disk group) or ACCESS_CONTROL.UMASK (a three-digit number that masks out permissions) attributes. Consult Oracle documentation for details on these attributes.

Disks	The **Path** list on this tab lets you view the disks currently assigned to this disk group. With a Path list disk selected, the **Property**/**Value** list displays attributes for that disk. The **Creation** category shows disk creation and mount date details as well as the setting you chose when adding this disk. For details on these properties, see Disk Groups (Oracle) - Disks. The Property/Value list differs by version 10G The **General**, **Status**, **Size**, and **Statistics** categories display information returned from the LSATTR, LSDSK, and LSDG commands (V$ASM_ATTRIBUTE, V$ASM_DISKGROUP_STAT, V$ASM_DISKGROUP, V$ASM_DISK_STAT, and V$ASM_DISK views) as follows: **General - Compound Index**, **Incarnation**, **Redundancy**, **Library**, **Label**, **UdID**, and **Product**. **Status - Mount Status**, **Header Status**, and **Mode Status**. **Size - Free MB**. **Statistics - Repair Timer**, **Reads**, **Writes**, **Read Errors**, **Write Errors**, **Read Time**, **Write Time**, **Bytes Read**, and **Bytes Written**. 11GR1 The key differences for the 11gR1 release are the **Mode Status**, **Online Wait**, and **Offline Timeout** controls. **Mode Status -** Lets you bring a single disk ONLINE or OFFLINE. **Online Wait -** Enabled when bring a disk ONLINE, lets you select WAIT or NOWAIT. **Offline Timeout -** Enabled when taking a disk offline, lets you specify the amount of time after which the disk will be dropped after the status changes to OFFLINE. 11GR2 A **Type** property with REGULAR and QUORUM values is available.

Volumes (available as of Oracle 11GR2)	Lets you manage (add, edit, delete) volumes associated with the disk group.
Templates	This tab lets you maintain the list of named attributes applied to disks in this disk group. Note that System Templates are automatically created for the disk group when the group is created. For details on template attributes and information on adding, editing, and deleting templates, see Disk Groups (Oracle) - Templates.
Files (disabled unless you have a direct connection to the ASM instance)	The **Directories** list is a representation of the ASM virtual folder. On creation, the list is populated with the system directory hierarchy dictated by the ASM fully qualified file name convention. Use the associated buttons to create new directories, delete user-created directories, or modify the directory name or base directory of a user-created directory. The **Files and Aliases** list contains files generated automatically as part of database operations (including tablespace creation) and user-created aliases. Selecting a file or alias from the list displays its attributes in the **Property**/**Value** list on the right. Use the associated buttons to create an alias to a selected file, delete files or aliases, or modify the name or base directory of a user-created alias.
Operations (disabled unless you have a direct connection to the ASM instance)	Displays any commands currently running on the ASM instance.
Users and **Groups** (available as of Oracle 11GR2)	These tabs apply only to Oracle 11GR2 datasources installed on Linux/Unix computers. Permissions/ownership are also dependent on the **File Access Control Enabled** and **Access Control Unmask** values on the **Properties** tab. These tabs lets you manage the File Access Control users and user groups.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

4 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Foreign Keys Editor (Oracle)

The Foreign Keys Editor lets you enable a foreign key, manage its column mappings, and specify a delete rule.

To edit a foreign key

1 Open an editor on the foreign key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Column Mapping	The existing column mapping for the foreign key is represented by selected columns in the **Main Table** and **Referenced Table** lists. Additional candidates in the **Referenced Table** list are indicated by enabled column check boxes. If necessary, use the **Table** dropdown in the **Referenced Table** group to choose a new table for this foreign key. Select or deselect columns in the **Main Table** list and **Referenced Table** list to form the referential constraint between the two tables.
Properties	Lets you set the foreign key as **Enabled** and specify a **Delete Rule** of CASCADE or NO ACTION.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Functions Editor (Oracle)

The Functions Editor lets you modify basic properties and modify the CREATE/REPLACE DDL for a function.

To edit a function

1 Open an editor on the function. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Properties	Lets you view properties in the following categories:	
	Properties	Lets you view **Status** and **Last Modified** details for the function
	Size Information	Lets you view **Source SIze**, **Parsed Size**, **Code Size**, and **Error Size** for the function
Definition	Lets you modify the CREATE OR REPLACE FUNCTION DDL for the function.	
Dependencies	For details on using this tab, see Working with Object Dependencies.	
Permissions	For details on using this tab, see Working with Privileges and Permissions.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Groups Editor (DBArtisan - Oracle)

The Groups Editor lets you modify privileges for a group.

To edit a group

1 Open an editor on the group. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Object Privileges and **System Privileges**	For details on using these tabs, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Indexes Editor (Oracle)

The Indexes Editor lets you manage columns, basic properties, storage, space, and partitioning for an index.

To edit an index

1 Open an editor on the index. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Columns	Displays a listing of the columns making up the index. Optionally, you can:	
		Use the **Sort** column check box to specify a sort option.
		Click the **New** button to add a new column to the index.
		Select an existing column and click the **Delete** button to delete that column from the index.
Properties	Lets you work with settings in the following categories:	
	Attributes	Lets you view **IsValid** and **Function-Based** properties. Lets you set**Index Type** (UNIQUE, NONUNIQUE, BITMAP), **No Sort**, **Logging**, **Reverse**, and **Invisible** properties.
	Parallel Query Option	Lets you view the **No Parallel Execution** property. Lets you set the **Parallel Degree** and **Parallel Instances** values.
Storage	Lets you work with settings in the following categories:	
	Data Block Storage	Lets you choose the **Tablespace**, and specify **Percent Free**, **Initial Transactions**, and **Max Transactions values**.
	Extents	Lets you specify **Initial Extent**, **Next Extent**, **Minimum Extents**, **Maximum Extents**, and **Percent Increase** values.
	Freelists	Lets you specify **Freelists** and **Freelist Groups** values.
	Buffer Pool	Lets you specify a buffer pool.
Space	Lets you view settings in the following categories:	
	Space Utilization	Lets you view the **Size** and number of **Blocks** settings.
	Statistics	Lets you view **Index Level**, **Distinct Keys**, **Cluster Factor**, **Leaf Blocks**, **Leaf Blks/Key** and **Data Blks/Key** properties.

<table>
<tr><td><b>Partition</b></td><td>If the index is not currently partitioned, you can click the <b>Convert To Partitioned</b> button to partition the index. For more information, see Partitioning Oracle indexes, primary keys, and unique keys. If the index is currently partitioned, this tab displays the following partition details:
<table>
<tr><td><b>Properties</b></td><td>Lets you view the <b>Locality</b> (Global/Local), <b>Alignment</b> (Prefixed/Non-Prefixed), <b>Partition Type</b> (RANGE or HASH), and <b>Subpartition type</b> properties.
Click the <b>Edit Partition</b> button to edit partition details. or more information, see Partitioning Oracle indexes, primary keys, and unique keys.
Click the <b>Drop Partition</b> button to revert to an unpartitioned index.
Use the <b>Partition Commands</b> menu items to initiate object actions. For details, see the following topics: Allocate ExtentAnalyzeCoalesceDeallocate Unused Space<b>Mark Unusable -</b> opens a dialog that lets you select one or more partitions/subpartitions to be marked as unusable. <b>Rebuild -</b> opens a dialog that lets you select one or more unusable subpartitions to be rebuilt. <b>Split -</b> opens a dialog that lets you divide a single partition into two partitions. You can split partitions if a single partition is causing maintenance problems because it is too large. <b>NOTES:</b> If you are preparing to drop or rebuild an index, mark local indexes as unusable. If you want to make unusable indexes valid or to recover space and improve performance, rebuild the unusable indexes. You cannot split a local index partition defined on a hash or composite table. Make sure that you specify an upper bound for the column that is lower than the upper bound for that column in the original partition.</td></tr>
<tr><td><b>Columns</b></td><td>Displays partitioning columns.</td></tr>
<tr><td><b>Partition Definitions</b></td><td>Displays the list of partition definitions. For each partition definition, listing shows the partition definition <b>Value</b>, the associated <b>Tablespace</b>, and whether the index is <b>Usable</b> or has been marked as Unusable by Oracle. Use the <b>Partition Commands</b> menu <b>Mark Unusable</b> or <b>Rebuild</b> commands to change the current <b>Usable</b> value.</td></tr>
</table>
</td></tr>
<tr><td><b>DDL View</b></td><td>For details on using this tab, see Viewing the SQL/DDL for an Object.</td></tr>
</table>

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Job Queue Editor (Oracle)

The Job Queue Editor lets you change the sql procedure specified for the job as well as manage the job's schedule and status.

To edit a job queue

1 Open an editor on the job queue. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you modify the **Submit as disabled**, **Next Date**, **Run Again**, **Every**, **Time Unit**, **Use Custom Expression**, and **Custom Expression** properties. For details on these properties, see Job Queues (Oracle) - Properties. This tab also displays the following properties: **Job** - the unique identifier assigned when the job was created. **User** - the user who submitted the job. **Schema User** - the user schema for which to parse the job **Priv User** - the user whose privileges the job is to run under.
Definition	This tab lets you modify the DBMS_JOB.SUBMIT call's WHAT parameter, specifying the name of the PL/SQL procedure to run.
DDL	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Jobs Editor (Oracle)

The Jobs Editor lets you modify details of a job.

To edit a job

1 Open an editor on the job. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	For details on using these settings, see Jobs (Oracle) - Properties.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Libraries Editor (Oracle)

The Libraries Editor lets you view and modify library definitions and manage dependencies and privileges for the library.

To edit a library

1 Open an editor on the library. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Definition	The **Definition** tab of the Libraries Editor lets you modify the file name and path for a library and view the current status and whether the library is dynamic.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Privileges	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Materialized Views Editor (Oracle)

The Materialized Views Editor lets you view and modify materialized view information and partitions.

To edit a materialized view

1 Open an editor on the materialized view. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Information	Lets you work with properties in the following categories:	
	Master	Lets you work with the **Table Name** and **Master View** properties.
	Last Refresh	Lets you work with the **Last Date** and **Errors** properties.
	Refresh Configuration	Lets you work with the **Type**, **Refresh Method**, and **Mode** properties.
	Rollback Usage	Lets you work with the Local and Master properties.
	Options	Lets you work with the **Updatable** and**Enable Query Rewrite** properties.
Storage	Lets you work with properties in the following categories:	
	Placement	Lets you work with **Tablespace** and **Cluster** values.
	Data Block Storage	Lets you work with **Percent Free**, **Percent Used**, **Initial transactions**, and **Maximum transactions** values.
	Extents	Lets you work with **Initial Extent**, **Next Extent, Minimum Extents**, **Maximum Extents**, and **Percent Increase** values.
Performance	Lets you work with settings in the following categories:	
	Parallel Query Option	The Parallel server query option lets you process queries using many query server processes running against multiple CPUs. This option provides substantial performance gains such as reduction of the query completion time. **Degrees -** Lets you type a value indicating the number of query server processes that should be used in the operation. **Instances -** Lets you type a value indicating how you want the parallel query partitioned between the Parallel Servers.
	Logging	Select **Logging** to create a log for all Materialized View updates.
	Cache	Select **Cache** if you want Oracle to put data you access frequently at the most recently used end of the list in the buffer cache when a full table scan is performed. This option is useful for small lookup tables.

Query	Displays the associated query.	
Partitions	**Partitioning Method**	Displays the partitioning method, including Range-Hash Composite or Range-List Composite. Hash partitions partition the table according to a hash function. Composite partitions use both range and hash types, first partitioning the data by a range of values, and then further dividing the partitions into subpartitions by way of a hash function. List partitioning lets you control how rows map to partitions. You can specify a list of discrete values for the partitioning column in the description for each partition.
	Row Movement	If its key is updated, migrates the row to a new partition.
	Partitioning Columns	Displays partitioning columns.
	Subpartitioning Columns	Displays subpartitioning columns.
	Partitions	Click **Add** or **Edit** to open the **Partition** dialog. Click **Drop** to drop a partition.
	Subpartition Template	If the partitioning type is Range-Hash Composite, displays a list of subpartitions in the subpartition template. Click **Add**, **Insert**, or **Edit** to open the **Subpartition** dialog. Click **Drop** to drop a subpartition.
Dependencies	For details on using this tab, see Working with Object Dependencies.	
Privileges	For details on using this tab, see Working with Privileges and Permissions.	
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Materialized View Logs Editor (Oracle)

The Materialized View Logs Editor lets you view replication log table details, and manage space, storage, and performance details for a materialized view log.

To edit a materialized view log

1 Open an editor on the materialized view log. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Information	Displays information on the log table used for replication.	
Storage	Lets you work with settings in the following categories:	
	Data Block Storage	Lets you view the associated **Tablespace** and **Initial Transactions** value, and modify the **Percent Free**, **Percent Used**, and**Max Transactions** values.
	Extents	Lets you view **Initial Extent**, **Next Extent**, **Minimum Extents**, **Maximum Extents**, and **Percent Increase** values.
	Column Filtering	Lets you select the filter columns to be recorded in the materialized view log. You can specify only one primary key, one ROWID and one filter column list per materialized view log. The ROWID is a globally unique identifier for a row in a database. It is created at the time the row is inserted into a table, and destroyed when it is removed from a table.
Performance	Lets you work with settings in the following categories:	
	Parallel Query Option	The Parallel server query option lets you process queries using many query server processes running against multiple CPUs. This option provides substantial performance gains such as reduction of the query completion time. **Degrees -** Lets you type a value indicating the number of query server processes that should be used in the operation. **Instances -** Lets you type a value indicating how you want the parallel query partitioned between the Parallel Servers.
	Logging	Select **Logging** to create a log for all Materialized View updates.
	Cache	Select **Cache** if you want Oracle to put data you access frequently at the most recently used end of the list in the buffer cache when a full table scan is performed. This option is useful for small lookup tables.
Dependencies	For details on using this tab, see Working with Object Dependencies.	
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Outlines Editor (Oracle)

The Outlines Editor lets you view information on an outline, and modify its category and associated SQL statement.

Note: The only SQL statements possible with stored outlines are SELECT, DELETE, UPDATE, INSERT…SELECT, and CREATE TABLE…AS SELECT.

To edit an outline

1 Open an editor on the outline. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	In addition to displaying basic creation and identification properties, this tab lets you select a new **Category** for the outline.
Definition	Lets you view and modify the SQL Statement associated with the outline.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Package Bodies Editor (Oracle)

While listed as separate objects in the Datasource Navigator/Explorer, package bodies are created and edited on the **Body** tab of the Packages Editor. For details, see Packages Editor (Oracle).

Packages Editor (Oracle)

The Packages Editor lets you view and modify header and body specifications of a package.

To edit package

1 Open an editor on the package. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

<table>
<tr><th>Tab</th><th>Settings and tasks</th></tr>
<tr><td><b>Header</b></td><td>Lets you modify the package header specifications.</td></tr>
<tr><td><b>Body</b></td><td>Lets you modify the package body specifications.</td></tr>
<tr><td><b>Information</b></td><td>Lets you work with status and size properties in the following categories:<table><tr><td><b>Header</b> and <b>Body</b></td><td>Lets you view <b>Status</b>, <b>Created</b>, and <b>Last ModifiedSource SIze</b>, <b>Parsed Size</b>, <b>Code Size</b>, and <b>Error Size</b> details for the header and body of the package.</td></tr></table></td></tr>
<tr><td><b>Dependencies</b></td><td>For details on using this tab, see Working with Object Dependencies.</td></tr>
<tr><td><b>Permissions</b></td><td>For details on using this tab, see Working with Privileges and Permissions.</td></tr>
</table>

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Primary Keys Editor (Oracle)

The Primary Keys Editor lets you manage columns, basic properties, storage and space, and partitions for a primary key.

To edit primary key

1 Open an editor on the primary key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab		Settings and tasks
Columns	Displays a listing of the columns making up the primary key. Optionally, you can:	
		Click the **New** button to add a new column to the index.
		Select an existing column and click the **Delete** button to delete that column from the index.
Properties	Lets you work with settings in the following categories:	
	Enforcing Index	Lets you view **User Defined**, **Index Owner**, and **Index Name** properties.
	Attributes	Lets you work with **No Sort** (only available if **Enabled** is set), **Logging** (YES, NO, or NONE), **Reverse** (disabled if **No Sort** is enabled), **Validate**, **Deferrable**, **Deferred** (IMMEDIATE or DEFERRED and only enabled if **Deferrable** is enabled), **Enabled**, **Cascade** (disabled if **Enabled** is set), **Rely** and **Update Date** properties.
Storage	Lets you work with settings in the following categories:	
	Data Block Storage	Lets you specify **Tablespace**, **Percent Free**,**Initial Transactions**, and **Max Transactions** values. NOTE: You should never place primary keys on the SYSTEM tablespace.
	Extents	Displays **Initial Extent**, **Next Extent**, **Percent Increase**, **Minimum Extents** and **Maximum Extents** values.
	Freelists	Lets you specify **Freelists** and **Freelist Groups** values.
	Buffer Pool	Lets you specify a buffer pool.
Space	Lets you work with settings in the following categories:	
	Space Utilization	Lets you view **Size** and **Blocks** properties.
	Statistics	Lets you view **Index Level**, **Distinct Keys**, **Cluster Factor**, **Leaf Blocks**, **Leaf Blks/Key**, and **Data Blks/Key** properties.

Partition If the primary key is currently partitioned, this tab displays the following partition details:

Properties	Lets you view the **Locality** (Global/Local), **Alignment** (Prefixed/Non-Prefixed), **Partition Type** (RANGE or HASH), and **Subpartition type** properties. Click the **Edit Partition** button to edit partition details. or more information, see Partitioning Oracle indexes, primary keys, and unique keys. Click the **Drop Partition** button to revert to an unpartitioned primary key. Use the **Partition Commands** menu items to initiate object actions. For details, see the following topics: Allocate ExtentAnalyzeCoalesceDeallocate Unused Space**Mark Unusable -** opens a dialog that lets you select one or more partitions to be marked as unusable. **Rebuild -** opens a dialog that lets you select a partition to be rebuilt. **Split** - opens a dialog that lets you divide a single partition into two partitions. You can split partitions if a single partition is causing maintenance problems because it is too large.
Columns	Displays partitioning columns.
Partition Definitions	Displays details for each partition.

If the primary key is not currently partitioned, you can click the **Convert To Partitioned** button to partition the primary key. For more information, see Partitioning Oracle indexes, primary keys, and unique keys.

DDL View For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Procedures Editor (Oracle)

The Procedures Editor lets you view and modify the SQL code and properties of a procedure.

To edit a procedure

1 Open an editor on the procedure. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Properties	Lets you work with properties in the following categories:	
	Properties	Lets you view **Status**, and **Last Modified** properties.
	Size Information	Lets you view **Source Size**, **Parsed Size**, **Code Size**, and **Error Size**
Definition	Lets you modify the SQL code for a procedure.	
Dependencies	For details on using this tab, see Working with Object Dependencies.	
Permissions	For details on using this tab, see Working with Privileges and Permissions.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Profiles Editor (Oracle)

The Profiles Editor lets you manage limits and manage user assignments for the profile.

To edit a profile

1 Open an editor on the profile. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Resources	Lets you work with settings in the following categories:	
	General Limits	Lets you specify **Composite Limit** and **Private SGA** settings.
	Session Limits	Lets you specify the limit on the amount of private space a session can allocate in the shared pool of the SGA. Specific settings are **Sessions Per User**, **CPU Per Session**, **Logical Reads**.
	Time Limits	Lets you specify the limit on total connection time per session. Specific settings are **Connect Time** and **Idle Time**.
	Call Limits	Lets you specify the CPU time limit for a call (a parse, execute, or fetch), expressed in hundredths of seconds. Specific settings are **CPU Per Call** and **Logical Reads**.
	Login Limits	Lets you specify the number of **Failed Login Attempts** on the user account before the account is locked and the **Account Lock Time**.
	Password Limits	Lets you specify the**Lifetime**, **ReuseTime**, **Reuse Max**, **Grace Period** and a **Verify Function** for passwords.
Users	Use the **Assign** button to open a dialog that lets you assign a user to this profile or select a user from the list, and click the **Unassign** button to open a dialog prompting you to confirm that the user is to be unassigned.	
DDL	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Programs Editor (Oracle)

The Programs Editor lets you modify details of a program.

To edit a program

1 Open an editor on the program. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Action	For details on using these settings, see Programs (Oracle) - Action.
Arguments	For details on using these settings, see Programs (Oracle) - Arguments.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Redo Log Groups Editor (Oracle)

The Redo Log Groups Editor lets you manage the members in a redo log group.

To edit a redo log group

1 Open an editor on the redo log group. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Redo Log Members	Lets you add new members to the redo log group, edit existing members, and delete members from the redo log group.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Roles Editor (Oracle)

The Roles Editor lets you manage authentication and grant/revoke profiles for users and logins.

To edit a role

1 Open an editor on the role. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Authentication	Lets you manage role identity. When creating a role, you must establish certain rules governing its use. You can specify whether or not a role must be identified when used. If you require role identification, you can authenticate the user externally through the operating system, or with a specific password. If you specified that the role requires identification, provide a **Password** and specify whether the role is to be authenticated **Globally** or **Externally**.
User/Roles	Displays permissions for this role to logins or users. Click **Grant** to open a dialog that lets you grant this role to a login or another role. Select a role or login and click **Revoke** to revoke the role or login.
Object Privileges and **System Privileges**	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

For information on activating and deactivating roles for the current login in the current session, see Role Activation.

Rollback Segments Editor (Oracle)

The Rollback Segments Editor lets you view rollback segment status, manage rollback segment storage, and view activity levels.

To edit a rollback segment

1 Open an editor on the rollback segment. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Status	Lets you enable and disable a rollback segment and displays status details for the rollback segment. The tab displays whether the rollback segment is online or offline and provides the associated **Tablespace**, **Size**, and **No. of Extents**.	
Storage	Lets you work with settings in the following categories:	
	Extents	The unit of space allocated to an object whenever the object needs more space. **Initial Extent** - The initial space extent (in bytes) allocated to the object. **Next Extent** - The next extent (in bytes) that the object will attempt to allocate when more space for the object is required. **Optimal Size** - optimal extent size **Minimum Extents** - The appropriate minimum extents value for the object. **Maximum Extents** - The appropriate maximum extents value for the object.
	Extent Detail	Displays extent details.
Activity	Lets you work with settings in the following categories:	
	Activity Levels	Displays **Active Transactions**, **Writes**, **Gets** and **Waits** values.
	Dynamic Sizing	Displays **High Watermark**, **Extends**, **Shrink**s, and **Wraps** values.
DDL	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Schedules Editor (Oracle)

The Schedules Editor lets you modify details of a schedule.

To edit a schedule

1 Open an editor on the schedule. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Action	For details on using these settings, see Schedules (Oracle) - Properties.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Sequences Editor (Oracle)

The Sequences Editor lets you manage parameters for a sequence, manage database objects dependent on the sequence, and manage privileges for the sequence.

To edit a sequence

1 Open an editor on the sequence. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Definition	Lets you work with settings in the following categories:	
	Parameters	Lets you specify increment, minimum value and maximum value settings.
	Current/Next Sequence Numbers	Lets you work with sequence cycle numbers.
	Options	Lets you specify **Cache Size**,**Cycle When Reach Max/Min**, and **Generate Numbers in Order** (useful when you are using the sequence number as a timestamp) values.
Dependencies	For details on using this tab, see Working with Object Dependencies.	
Privileges	For details on using this tab, see Working with Privileges and Permissions.	
DDL	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Synonyms Editor (Oracle)

The Synonyms Editor lets you view base object information and manage database objects dependent on a synonym.

To edit a synonym

1 Open an editor on the synonym. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Displays the type, owner, name, and other details of the object referenced by the synonym.
Dependencies	For details on using this tab, see Working with Object Dependencies.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Tables Editor (Oracle)

The Tables Editor lets you manage columns, constraints, storage and space, and partitions for a table.

To edit a table

1 Open an editor on the table. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Columns	Displays the currently defined columns in the table. For any selected column, the Property/Value list provides additional detail on that column. Available properties depend on the datatype you choose as well as on the property values you select: **Virtual** - indicates whether this column is defined as an Oracle virtual column. If selected, the value is provided by the calculation in the **Default Value** box. **Datatype** properties- Lets you select a **type** and depending on your selection, additional properties such as **Scale**, **Width**, and **Unused** may be available. **Allow Nulls** - Select this check box to allow nulls in this column. **Encryption** properties include **Password**, **Salted**, and **EncryptionAlgorithm**. **Default Value** - If **Virtual** is selected, this box lets you specify a valid Oracle column expression. Otherwise, it can contain a pseudocolumn (CURRENT_TIMESTAMP, USER, SYSDATE, or UID) or an expression. lets you choose among CURRENT_TIMESTAMP, USER, SYSDATE, and UID. **Comment** lets you add a comment to the column. **LOB Storage** settings are available for bfile, blob, clob, and nclob types. They include **Segment Name**, **Configuration** properties (**Tablespace**, **Chunk**, **Percent Version**, **Enable Storage In Row**, **Cache**, and **Logging**), and **Storage** properties (**Initial Extent**, **Next Extent**, **Percent Increase**, **Minimum Extents**, **Maximum Extents**, **Free Lists**, and Free List Groups). Optionally, you can:	
		Select a column, and in the **Property**/**Value** list modify property values for that column.
		Click **Add Column**, provide a name for the new column, and set property values for the column.
		Select a column and click **Delete** to remove the column from the table.
Properties	Lets you work with settings in the following categories:	
	Table	Lets you work with the following properties, corresponding to CREATE TABLE clauses: **Cache**, **Row Movement**, **Parallel Degree**, and **Parallel Instances**.
	Physical	Lets you view the **Row Organization** property. Lets you set the **Logging** and **Table Compression** properties.
Indexes	Lets you manage indexes for a table. On opening, the list of current indexes for the table is displayed. Optionally, take one of the following actions:	
		Click **Add** to open a dialog that lets you add a new index to the table.

		Select an index and click **Edit** to open a dialog that lets you edit index properties.
		Select an index and click **Drop** to open a dialog that lets you remove the index from the table.
Constraints	Lets you manage constraints for the table. Constraints are grouped by type, under folders. Optionally take one of the following actions:	
		Select a constraint type folder and click **Add** to open a dialog that lets you add a constraint of that type.
		Select a constraint and click **Edit** to open a dialog that lets you modify the constraint details.
		Select a constraint and click **Drop** to remove the constraint.
Storage	Lets you work with settings in the following categories:	
	Data Block Storage	Lets you choose the **Tablespace Name**, and specify **Percent Free**, **Initial Transactions**, and **Max Transactions** values.
	Extents	Lets you view **Initial Extent**, **Next Extent**, **Minimum Extents**, **Maximum Extents**, and **Percent Increase** values.
	Freelists	Lets you specify **Freelists** and **Freelist Groups** values.
	Bufferpool	Lets you specify a **Buffer Pool**.
IOT Properties	Lets you work with settings in the following categories:	
	Ungrouped	Lets you specify an overflow segment.
	Percent Threshold	Lets you specify the percentage of space reserved for an index-organized table.
	Key Compression	Lets you enable or disable compression and provide a compression value.
Space	Lets you work with settings in the following categories:	
	Space Utilization	Lets you view **Size** and number of **Blocks**.
	Row Information	Lets you view the **Number of Rows**, **Average Row Length**, and **Chain Rows** properties.
	Extents	Let you view the **Number of Extents** and **Maximum Extents** values.
LOB columns	Lets you work with settings in the following categories:	
	LOB Column Segment and **LOB Column Index**	Lets you view the **Segment Name**, **No. of Extents**, **Max Extents**, **Size**, and **Blocks** properties.

	Segment Extents and **Index Extents**	Lets you view the **Extent ID**, **File ID**, **Block ID** and number of **Blocks** properties for segment and index extents.
Partition	Lets you work with table partitions	
Comment	For details on using this tab, see Adding a Comment to an object.	
Dependencies	For details on using this tab, see Working with Object Dependencies.	
Permissions	For details on using this tab, see Working with Privileges and Permissions.	
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Tablespaces Editor (Oracle)

The Tablespaces Editor lets you manage datafiles, space, storage, quotas, and objects for a tablespace.

To edit a tablespace

1 Open an editor on the tablespace. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Datafiles	The tab lists details for each datafile on the tablespace. Optionally you can add or delete files or edit file attributes. For detailed information on using this tab, see Tablespaces (Oracle) - Datafiles.
Properties	This tab lets you view the **Name**, **Big File**, **Type**, **Encrypted**, **Locally Managed**, **Minimum Extent Size**, **Uniform Allocation**, **Use Default Block Size**, **Block Size**, **Automatic Segment Space Management**, **Initial Extent**, **Next Extent**, **Minimum Extents**, **Maximum Extents**, and **Percent Increase** properties. This tab lets you modify the **Status**, **Logging**, **Force Logging**, and **Compression Type** properties. For more information on these properties, see Tablespaces (Oracle) - Properties.
Extent Details	Provides details (**Name**, **Type**, **File ID**, **Extent ID**, **Block ID**, **Blocks**, and **Bytes**) for each extent in the tablespace.
Space	Displays free blocks versus used blocks statistics for the tablespace, including a pie chart representation.
Map	This tab presents a graphical, segment map of the tablespace. The map is color-coded to show usage of each segment (FREE, SELECTED, OTHER, TEMPORARY, ROLLBACK, CACHE, TABLE, INDEX, TABLE PARTITION, NESTED TABLE, INDEX PARTITION, INDEX SUBPARTITION, TEXT, CLUSTER, LOB, LOBSEGMENT, LOBINDEX, or LOB PARTITION). Optionally you can: Click **Legend** to view the color legend of the map. Hover the mouse over a segment to display its **Segment ID**, **Block ID**, **Size** (kb), **Size** (in blocks), **Tablespace File ID**, and **Segment Type** details. Click **Display** to open a window that aids in navigating the graphical segment map. The **Overview** window displays a smaller version of the segment map. The selected area can be dragged up and down the window, displaying the selected area in the larger, main map. The selected area can also be resized, to display a larger or smaller area in the main map. Select an INDEX or TABLE segment on the map and click **Reorganize** to initiate a **Rebuild Indexes** or **Reorganize** (respectively) action. For details, see Rebuild Index and Reorganize /Reorg. The **Object Demographics** area provides a tabular representation of the tablespace map. It provides details on a segment-by-segment basis. For background information, see About Oracle Tablespace Storage.

Objects	Displays the objects currently stored on the tablespace, grouped under object type folders. Optionally you can: Specify a logging option using the **Log Changes When Scheme/Data is Modified?** radio set. Select an object under one of the object folders and click **Edit** to open an object editor on that object.
Quotas	Oracle limits the amount of space that can be allocated for storage of a user's objects within the specified tablespace to the amount of the quota. Users with privileges to create certain types of objects can create those objects in the specified tablespace. The **Quotas** tab of the Tablespace editor lets you manage user space quotas for tablespaces on the current datasource. Optionally, you can: Click **Add** or select a user and click **Edit** to assign a user unlimited or a specific space usage quota on the tablespace. For details, see Adding or Editing User Tablespace Quotas. Select an existing user and click **Drop** to delete the quota for that user
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Adding or Editing User Tablespace Quotas

When you assign a quota:

- Users with privileges to create certain types of objects can create those objects in the specified tablespace.
- Oracle limits the amount of space that can be allocated for storage of a user's objects within the specified tablespace to the amount of the quota.

Note: This functionality is available for Oracle only.

User tablespace quotas are added and modified from the Tablespaces editor.

The table below describes the options and functionality on the **Add User Quota...** or **Edit User Quota...** dialogs:

Option	Description
User selection list (Add only)	Lets you select one or more users to assign a quota.
Quota	Lets you set a quota for the selected user or users. You can select an unlimited, or a specified size. Unlimited - Lets you place an unlimited quota on the tablespace. Other - Lets you place a specified quota in KB or MB on the tablespace.

About Oracle Tablespace Storage

The **Storage** tab of the Tablespace Editor lets you view storage details for tablespaces on the current datasource.

Tip: Always create tablespaces for user data and never place user tables and indexes in the SYSTEM tablespace. Placing user objects in the SYSTEM tablespace can degrade performance and introduce space-related headaches to the database.

Oracle or later supports locally managed tablespaces, which can all but eliminate the problem of tablespace fragmentation. It totally does away with the storage parameters of MINEXTENTS, MAXEXENTS, PCTINCREASE, and NEXT. With locally managed tablespaces you either specify the initial extent size and let Oracle automatically size all other extents, or specify a uniform extent size for everything.

Tip: One of the best ways to avoid fragmentation in a tablespace is to pre-allocate the space that your objects will use. If possible, plan for one to two years' growth for each object and allocate your space accordingly. Having initial empty objects will not affect table scan times as Oracle only scans up to the high-water mark (the last used block) in a table.

Of all your tablespaces, you want to avoid fragmentation problems in your SYSTEM tablespace the most as this is the major hotbed tablespace for Oracle activities. The easiest way to avoid this is to not allow any user (even the default DBA ID's SYS and SYSTEM) to have access to it. There are three ways to do this:

- Ensure no user has a DEFAULT or TEMPORARY tablespace assignment of SYSTEM.
- Ensure no user has a quota set for SYSTEM.

Ensure no user has been granted the UNLIMITED TABLESPACE privilege.

Triggers Editor (Oracle)

The Triggers Editor lets you modify actions, events, and other details for a trigger.

To edit a trigger

1 Open an editor on the trigger. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	In addition to displaying basic identification and creation properties, this tab lets you work with settings in the following categories: **Attributes** – Lets you specify whether the trigger is **Enabled**, select the **Trigger Timing** (BEFORE, AFTER, COMPOUND), and the **Trigger Type** (ROW, STATEMENT). **Correlation Names** – Lets you provide an **Old Table Alias** and a **New Table Alias** as well as construct a **When Clause**. **Status** – Lets you view **Object Status**, **Create Date**, **Last Modified**, and **Base Object Type** properties. **Size Information** – Lets you view **Source Size**, **Parsed Size**, **Code Size**, and **Error Size** properties. For detailed information on these properties, see Triggers (Oracle) - Properties.
Events	Lets you select the DDL, DML, or Database manipulation events that fire this trigger. For detailed information,see Triggers (Oracle) - Events.
Column Selection	For triggers firing on UPDATE events, lets you work with the associated columns. For detailed information,see Triggers (Oracle) - Column Selection.
Action	Lets you modify the trigger action PL/SQL block for any trigger on the datasource. You modify the body of the trigger in the Trigger Action (PL/SQL Block) area. For detailed information,see Triggers (Oracle) - Action.
Dependencies	For details on using this tab, see Working with Object Dependencies.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Type Bodies Editor (Oracle)

The Type Bodies Editor contains the code for the methods that implement an object type. To create or replace a type body in one of your schema, you must have the CREATE TYPE or CREATE ANY TYPE system privilege. To replace a type in another user's schema, you must have the DROP ANY TYPE system privilege.

While listed as separate objects in the Datasource Navigator/Explorer, type bodies are created and edited on the **Body** tab of the Types Editor. For details, see Types Editor (Oracle).

Types Editor (Oracle)

The Types Editor lets you manage header text and body text for a type.

To edit a type

1 Open an editor on the type. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Header	Lets you manage the type header text.
Body	Lets you create and modify type body text for the type. For information on creating type bodies, see Object Types Wizard (Oracle).
Information	Displays the header and body information for a type.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Privileges	For details on using this tab, see Working with Privileges and Permissions.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Unique Keys Editor (Oracle)

The Unique Keys Editor lets you manage columns, basic properties, storage and space, and partitions for a unique key.

To edit a unique key:

1 Open an editor on the unique key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Columns	Displays a listing of the columns making up the index. Details include the **Column** name, the **Datatype**, and whether the definition allows **Nulls**. Optionally, you can:	
		Click the **New** button to add a new column to the index.
		Select an existing column and click the **Delete** button to delete that column from the index.
Properties	In addition to displaying basic identification information, this tablets you work with settings in the following categories:	
	Enforcing Index	Lets you view **User-Defined**, **Index Owner**, and **Index Name** settings.
	Attributes	Lets you set **No Sort**, **Logging**,**Reverse**, **Validate**, **Deferrable**, **Deferred**, **Enabled**, **Cascade**, **Rely**, and **Update Date** properties.
Storage	Lets you work with settings in the following categories:	
	Data Block Storage	Lets you choose the **Tablespace Name**, and specify **Percent Free**, **Initial Transactions**, and **Max Transactions values**.
	Extents	Lets you specify **Initial Extent**, **Next Extent**, **Minimum Extents**, **Maximum Extents**, and **Percent Increase** values.
	Freelists	Free lists let you manage the allocation of data blocks when concurrent processes are issued against the cluster. Identifying multiple free lists can reduce contention for free lists when concurrent inserts take place and potentially improve the performance of the cluster. This tab lets you specify **Freelists** and **Freelist Groups** values.
	Buffer Pool	Lets you select a **Buffer Pool**.
Space	Lets you work with settings in the following categories:	
	Space Utilization	Displays the **Size** and number of **Blocks** properties.
	Statistics	Lets you view **Index Level**, **Distinct Keys**, **Cluster Factor**, **Leaf Blocks**, **Leaf Blks/Key** and **Data Blks/Key** settings.

Extents	The unit of space allocated to an object whenever the object needs more space. This tab lets you view **Number of Extents** and **Maximum Extents** properties.

Partition	If the unique key is currently partitioned, this tab displays the following partition details:

Properties	Lets you view the **Locality** (Global/Local), **Alignment** (Prefixed/Non-Prefixed), **Partition Type** (RANGE or HASH), and **Subpartition type** properties. Click the **Edit Partition** button to edit partition details. For more information, see Partitioning Oracle indexes, primary keys, and unique keys. Click the **Drop Partition** button to revert to an unpartitioned unique key. Use the **Partition Commands** menu items to initiate object actions. For details, see the following topics: Allocate ExtentAnalyzeCoalesceDeallocate Unused Space**Mark UnusableRebuildSplit**
Columns	Displays partitioning columns.
Partition Definitions	Displays details for each partition.

If the unique key is not currently partitioned, you can click the **Convert To Partitioned** button to partition the unique key. For more information, see Partitioning Oracle indexes, primary keys, and unique keys.

DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Users Editor (Oracle)

The Users Editor lets you manage basic properties, roles, tablespace quotas, and associated objects for a user.

To edit a user

1 Open an editor on the user. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you work with settings in the following categories: **Defaults** Lets you select a **Default Tablespace** and Temporary **Tablespace** as well as select the **Profile** used by this user. For more on creating profiles, see Profiles Wizard (Oracle). **Account Identified By -** Lets you select among REQUIRED YES, REQUIRED EXTERNAL, and REQUIRED_GLOBAL. **Password -** (available with **Identified By** value of REQUIRED_YES only) lets you specify a password. **External Name -** (available with **Identified By** value of REQUIRED_GLOBAL only) lets you specify the name of the user in the enterprise directory service. **Account Locked -** lets you lock this user account. **Password Expired -** (available with **Identified By** value of REQUIRED_YES only) lets you mark the password as expired, forcing the user to change their password before being allowed to connect.
Role	Lets you select the roles that are to be assigned to this user.
Quota	Lets you set tablespace quotas for the user. Set an unlimited quota by selecting a tablespace and selecting the **Unlimited** radio button. Set a specific quota by selecting a tablespace, selecting the **Other** radio button, and providing a **Quota** value in kilobytes or megabytes.
Objects	Lets you manage database objects associated with a user. Objects are grouped within object type folders. Optionally, you can: Select an object and click **Edit** to open an object editor on the selected object. Select an object and click **Drop** to initiate dropping the selected object.
Object Privileges and **System Privileges**	For details on using these tabs, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Views Editor (Oracle)

The Views Editor lets you work with view columns and manage the dependencies and permissions for the view.

To edit a view

1 Open an editor on the view. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Displays a listing of the columns making up the view.
Definition	Lets you view and modify the DDL that will implement any changes made with the editor.
Comment	For details on using this tab, see Adding a Comment to an object.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Permissions	For details on using this tab, see Working with Privileges and Permissions.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Partitioning Oracle indexes, primary keys, and unique keys

The **Add Partition** wizard lets you partition an Oracle index, primary key, or unique key.

To partition an index, primary key, or unique key

1 Open an editor on the index, primary key, or unique key. For details, see Opening an Object Editor.

2 On the **Partition** tab, click **Convert to Partitioned**.

Note: If the object is already partitioned, the **Convert to Partitioned** button is not present.

3 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	In addition to displaying basic identification and ownership properties, this pane lets you work with the following settings: **Locality** - GLOBAL or LOCAL. **Alignment** - displays whether this partitioned object is prefixed or non-prefixed. **Partition Type** - RANGE or HASH. **Subpartition Type** - displays the subpartition type.
Columns	For each column to be added, click **New** to add a column, select a **Column** name, and specify a **Sort** order.
Range Definitions (only available for a **Partition Type** of RANGE)	Lets you create an ordered list of partitions. For each partition, click **New** to open the **Add Partition Definition** wizard. On the **Partition Definition** pane, provide a **Name**, select a **Tablespace**, and enable or disable **Logging**. On the **Storage** pane, provide **Data Block Storage**, **Extents**, and **Freelists** property values. Click **Add**. The **Range Definitions** pane also lets you edit and remove partitions.
Hash Definitions (only available for a **Partition Type** of HASH)	Lets you specify a Partition method: **NoneNumber of Partitions** - Specify a **Number of Partitions**, and then for each partition, click the **New** button and select a tablespace. **By Partition Name** - For each partition, click **New**, specify a **Name** and select a **Tablespace**.

4 When ready, click **Finish**.

PostgreSQL Object Editors

You modify PostgreSQL objects using the following wizards:

- Check Constraints Editor (PostgreSQL)
- Domains Editor (PostgreSQL)
- Exclusion Constraints, Primary Keys, or Unique Keys Editors (PostgreSQL)
- Foreign Keys Editor (PostgreSQL)
- Functions Editor (PostgreSQL)
- Indexes Editor (PostgreSQL)
- Roles Editor (PostgreSQL)
- Rules Editor (PostgreSQL)
- Schemas Editor (PostgreSQL)
- Tables Editor (PostgreSQL)
- Tablespaces Editor (PostgreSQL)
- Triggers Editor (PostgreSQL)
- Views Editor (PostgreSQL)

Check Constraints Editor (PostgreSQL)

The Check Constraints Editor lets you view or modify check constraints properties.

To edit a check constraint

1 Open an editor on the check constraint. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view or modify properties defined when creating the object. For details on these properties, see Check Constraints Wizard (PostgreSQL).
Comment	For details on using this tab, see Adding a Comment to an object.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Domains Editor (PostgreSQL)

The Domains Editor lets you view or modify domain properties.

To edit a domain

1 Open an editor on the domain. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view or modify properties defined when creating the object. For details on these properties, see Domains Wizard (PostgreSQL).
Constraints	Lets you add, drop, or edit constraints on this object. For details on these properties, see Domains Wizard (PostgreSQL).
Dependencies	For details on using this tab, see Working with Object Dependencies.
Comment	For details on using this tab, see Adding a Comment to an object.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Exclusion Constraints, Primary Keys, or Unique Keys Editors (PostgreSQL)

The Exclusion Constraints Editor, Primary Keys Editor, and Unique Keys Editor offer the same steps in modifying these PostgreSQL objects.

To edit an exclusion constraint, primary key, or unique key

1 Open an editor on the target object. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view or modify properties defined when creating the object. For details on these properties, see Exclusion Constraints, Primary Keys, and Unique Keys Wizards (PostgreSQL).
Columns	Lets you work with the columns designated for this object. For details, see Exclusion Constraints, Primary Keys, and Unique Keys Wizards (PostgreSQL).
Dependencies	For details on using this tab, see Working with Object Dependencies.
Comment	For details on using this tab, see Adding a Comment to an object.
Statistics	Displays the following pg_stat_all_indexes or pg_statio_all_table view statistics: **Scans**, **Keys Read**, **Rows Fetched**, **Blocks Read**, **Buffer Hits**, and **Index Size**.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Foreign Keys Editor (PostgreSQL)

The Foreign Keys Editor lets you view or modify column mapping and properties for a foreign key.

To edit a foreign key

1 Open an editor on the foreign key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Column Mapping	Lets you work with the columns of the main table and mapped referenced table columns for this foreign key. For details, see Foreign Keys Wizard (PostgreSQL).
Properties	Lets you view or modify properties defined when creating the object. For details on these properties, see Foreign Keys Wizard (PostgreSQL).
Comment	For details on using this tab, see Adding a Comment to an object.
Dependencies	For details on using this tab, see Working with Object Dependencies.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Functions Editor (PostgreSQL)

The Functions Editor lets you view or modify the function body, parameters, and properties for a function.

To edit a function

1 Open an editor on the function. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you modify the settings defined when the function was created or last edited. For details on these properties, see Functions Wizard (PostgreSQL).
Parameters	Lets you modify the function inputs and outputs defined when the function was created or last edited. For details on these settings on this tab, see Functions Wizard (PostgreSQL).
Body	Lets you modify the function definition specified when the function was created or last edited. For details on working with this tab, see Functions Wizard (PostgreSQL).
Dependencies	For details on using this tab, see Working with Object Dependencies.
Comment	For details on using this tab, see Adding a Comment to an object.
Permissions	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Indexes Editor (PostgreSQL)

The Indexes Editor lets you view or modify the columns and properties for an index.

To edit an index

1 Open an editor on the index. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view or modify properties defined when creating the object. For details on these properties, see Indexes Wizard (PostgreSQL).
Columns	Lets you modify the columns that make up the index. For details, see Indexes Wizard (PostgreSQL).
Comment	For details on using this tab, see Adding a Comment to an object.
Dependencies	For details on using this tab, see Working with Object Dependencies.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Roles Editor (PostgreSQL)

The Roles Editor lets you manage properties and users for a role.

To edit a role

1 Open an editor on the role. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you modify the properties set for the role when it was created or last edited. For details on these properties, see Roles Wizard (PostgreSQL).
Membership	Lets you modify the set of users GRANTed to the role. For details, see Roles Wizard (PostgreSQL).
Comment	For details on using this tab, see Adding a Comment to an object.
Dependencies	For details on using this tab, see Working with Object Dependencies.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Rules Editor (PostgreSQL)

The Rules Editor lets you view or modify rules properties.

To edit a rule

1 Open an editor on the rules. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view or modify properties defined when creating the object. For details on these properties, see Rules Wizard (PostgreSQL).
Dependencies	For details on using this tab, see Working with Object Dependencies.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Schemas Editor (PostgreSQL)

The Schemas Editor lets you change the owner of a schema, manage its permissions, and view referencing and referenced dependencies.

To edit a schema

1 Open an editor on the schema. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you modify the **Owner** of the schema.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Comment	For details on using this tab, see Adding a Comment to an object.
Permissions	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Tables Editor (PostgreSQL)

The Tables Editor lets you manage settings for an existing table.

To edit a table

1 Open an editor on the table. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	In addition to displaying the identification properties for this table, this tab lets you view the **Persistence** setting. You can modify the **Tablespace**, **With OIDS**, **Fill Factor**, **Table Autovacuum**, and **Toast Table Autovacuum** settings. For details on these properties, see Tables Wizard (PostgreSQL).
Ancestors	This tab/panel lets you modify a table's INHERITS clause.. For each table to be added to the list, use the New button to enable a dropdown that lets you select a table. Use the Delete button to remove a selected table from the list.
Columns	Displays the currently defined columns in the table. For any selected column, the **Property**/**Value** list provides additional detail on that column. Available properties depend on the datatype you choose as well as on the property values you select. For details on working with this tab, see Tables Wizard (PostgreSQL). You can add a column by clicking **Add Column**, providing a name for the new column, and set its property values. Similarly, select a column and click **Delete** to remove the column from the table. Lastly, change a columns position by selecting the column and using the arrow buttons.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Constraints	Lets you manage constraints for the table. Constraints are grouped by type, under folders. Optionally take one of the following actions: To add a constraints, select a constraint type folder and click **Add** to open a dialog that lets you add a constraint of that type. To edit a constraint, select a constraint and click **Edit** to open a dialog that lets you modify the constraint details. To delete a constraint, select a constraint and click **Drop** to remove the constraint.
Comment	For details on using this tab, see Adding a Comment to an object.
Permissions	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Tablespaces Editor (PostgreSQL)

The Tablespaces Editor lets you manage properties and permissions as well as view dependencies for a tablespace.

To edit a tablespace

1 Open an editor on the tablespace. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view or modify settings specified when the tablespace was created or most recently edited. For details on these properties, see Tablespaces Wizard (PostgreSQL).
Dependencies	For details on using this tab, see Working with Object Dependencies.
Comment	For details on using this tab, see Adding a Comment to an object.
Permissions	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Triggers Editor (PostgreSQL)

The Triggers Editor lets you manage properties and targeted columns for a trigger.

To edit a trigger

1 Open an editor on the trigger. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view or modify settings specified when the trigger was created or most recently edited. For details on these properties, see Triggers Wizard (PostgreSQL).
Column Selection	Lets you view or modify columns targeted when the trigger was created or most recently edited. For details on these settings on this tab, see Triggers Wizard (PostgreSQL).
Dependencies	For details on using this tab, see Working with Object Dependencies.
Comment	For details on using this tab, see Adding a Comment to an object.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Views Editor (PostgreSQL)

The Views Editor lets you manage settings for an existing view.

To edit a view

1 Open an editor on the view. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you modify the settings defined when the function was created or last edited. For details on these properties, see Views Wizard (PostgreSQL).
Definition	Lets you modify the DDL originally generated to create the view.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Comment	For details on using this tab, see Adding a Comment to an object.
Permissions	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Sybase ASE Object Editors

An Object Editor is offered for all supported Sybase ASE objects. To see an Editor for a specific object, click the corresponding link below:

Note: If an objects has dependent objects, such as tables, triggers, procedures and views, you can view and access their dependent objects in the editor.

- Aliases Editor (Sybase ASE)
- Check Constraints Editor (Sybase ASE)
- Database Devices Editor (DBArtisan - Sybase ASE)
- Databases Editor (Sybase ASE)
- Data Cache Editor (DBArtisan - Sybase ASE)
- Defaults Editor (Sybase ASE)
- Dump Devices Editor (DBArtisan - Sybase ASE)
- Extended Procedures Editor (Sybase ASE)
- Foreign Keys Editor (Sybase ASE)
- Functions Editor (Sybase ASE)
- Groups Editor (Sybase ASE)
- Indexes Editor (Sybase ASE)
- Logins Editor (Sybase ASE)
- Primary Keys Editor (Sybase ASE)
- Procedures Editor (Sybase ASE)
- Remote Servers Editor (DBArtisan - Sybase ASE)
- Roles Editor (DBArtisan - Sybase ASE)
- Rules Editor (Sybase ASE)
- Segments Editor (Sybase ASE)
- Tables Editor (Sybase ASE)
- Triggers Editor (Sybase ASE)
- Unique Keys Editor (Sybase ASE)
- User Datatypes Editor (Sybase ASE)
- User Messages Editor (Sybase ASE)

  - Users Editor (Sybase ASE)
  - Views Editor (Sybase ASE)

For access to object editors for different supported DBMS platforms and an introduction to editor usage, see Modifying objects using editors.

Aliases Editor (Sybase ASE)

The Aliases Editor lets you view and modify an alias definition.

To edit an alias

1 Open an editor on the alias. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you change the user referenced by the alias.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Check Constraints Editor (Sybase ASE)

The Check Constraints Editor lets you manage a check constraint definition and modify the condition expression.

To edit a check constraint

1 Open an editor on the check constraint. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Definition	Lets you edit the condition in the **Check Condition** box. The **Table Columns** button acts a time saver in editing the condition. It opens a dialog that lets you select and paste column names into the check condition expression. NOTE: If the **Check Condition** box shows a **Source text for this compiled object is hidden** message, source text for this compiled object has been hidden. For more information, see Hide Text.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Database Devices Editor (DBArtisan - Sybase ASE)

The Database Devices Editor lets you view or modify basic properties of a database device and view space usage details.

To edit a database device

1 Open an editor on the database device. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Properties	Lets you work with settings in the following categories:	
	Attributes and **Autogrowth**	For information on working with properties in these categories, see Database Devices (Sybase ASE) - Properties.
	Page range	The **First Virtual Page** and **Last Virtual Page** numbers for the target database device.
	Mirror	These controls are only enabled if mirroring is enabled on the server. Mirrored - enables/disables mirroring for this device. Mirror Enabled - enables/disables the existing mirror. Mirror Name - name of the existing mirror.
Space	Details include the **Device Size**, **SpaceUsed**, **FreeSpace**, and the size and usage for all **Database fragments**.	
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Databases Editor (Sybase ASE)

The Databases Editor lets you manage properties, placement, and binds for a database as well as view associated space statistics.

To edit a database

1 Open an editor on the database. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	For information on working with these settings, see Databases (Sybase ASE) - Properties.
Placement	Displays device fragments. Optionally you can: Select a fragment to view the **Device Name**, **Size**, and **Device Type** (DATA AND LOG or DATA ONLY) for the fragment. Click the **New** button to open a dialog that lets you provide the data device and log device details for a new database fragment. NOTE: We strongly recommend that you place the transaction log on a separate device from all other database fragments.
Space	Lets you view pie charts showing the data space usage and the transaction log (if available) space usage for the database.
Bindings	This tab is only available for databases created with a **Type** of TEMP. It displays all applications or logins bound to the database. Optionally, you can: Bind another application or login to the database by selecting the APPLICATION or LOGIN folder as appropriate, clicking **Add**, and in the dialog that opens select a login or provide an application name. Remove bindings by selecting the specific application or login and clicking **Drop**.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Data Cache Editor (DBArtisan - Sybase ASE)

This editor lets you modify a data cache's basic properties, object binding, and pool configuration.

Note: Consult Microsoft SQL Server documentation for detailed information on cache creation and configuration. Before creating or editing a data cache, be familiar with caching strategies, pool configuration, object binding, and server restart requirements. For assistance, see Accessing Third Party Documentation.

To edit a data cache:

1 Open an editor on the data cache. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

<table>
<tr><th>Tab</th><th>Settings and tasks</th></tr>
<tr><td><b>Properties</b></td><td>Lets you view the <b>Name</b>, <b>Status</b>, <b>Partitions</b>, and <b>Run Size</b> properties and modify the <b>Configured Size</b> property. For caches with a <b>Type</b> of MIXED or LOGONLY, you can also modify the <b>Type</b>, and <b>Policy</b> properties. For more information on these properties, see Data Caches (Sybase ASE) - Properties.</td></tr>
<tr><td>Objects</td><td>Lets you bind and unbind databases, tables and indexes for this data cache as well as modify the caching strategy for bound objects. For details on these tasks, see Data Caches (Sybase ASE) - Objects.</td></tr>
<tr><td><b>Pools</b></td><td>Generates the sp_poolconfig calls required to add or modify buffer pools for this cache. It lets you add or delete pools, change the configured size of a pool, or modify the wash size or asynchronous prefetch percentage for a pool. The list on the left displays the pools currently configured for the cache. The <b>Memory Pool Attributes</b> list on the right displays the following attributes for a selected pool: <b>Configured Size</b> - total size of the memory pool <b>Wash Size</b> - the cache location at which dirty pages are written. <b>Run Size</b> - (display only) <b>Async Prefetch</b> - the percentage of buffers in the pool that can be used to hold buffers that have been read into cache using asynchronous prefetch, but that have not yet been used. <b>Affected Pool</b> - the size of I/O performed in the memory pool where the memory is to be deallocated. Optionally, you can:
<table>
<tr><td><b>Modify pool attributes</b></td><td>Select a pool from the list and modify values in the <b>Memory Pool Attributes</b> list</td></tr>
<tr><td><b>Add a new pool</b></td><td>Click <b>Add Pool</b> and select an <b>I/O Size</b> (in kilobytes). Modify pool attributes as required.</td></tr>
<tr><td><b>Delete a pool</b></td><td>Select a pool from the list and click <b>Delete Pool</b>.</td></tr>
</table>
</td></tr>
<tr><td><b>DDL View</b></td><td>For details on using this tab, see Viewing the SQL/DDL for an Object.</td></tr>
</table>

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Defaults Editor (Sybase ASE)

The Defaults Editor lets you modify the value and manage bindings for a default.

To edit a default

1 Open an editor on the default. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you change the **Value** of the default. NOTE: If the **Value** box shows a **Source text for this compiled object is hidden** message, source text for this compiled object has been hidden. For more information, see Hide Text.
Dependencies	For details on using this tab, see Working with Object Dependencies.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Dump Devices Editor (DBArtisan - Sybase ASE)

The Dump Devices Editor lets you view basic properties of a dump device.

To edit a dump device

1 Open an editor on the dump device. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Definition	Displays the **Name**, **Physical Name**, and **Device Type** for the dump device. For more information, see Dump Devices (Sybase ASE) - Properties. If the **Device Type** is TAPE, this tab also shows a **Tape capacity** setting.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Extended Procedures Editor (Sybase ASE)

The Procedures Editor lets you view and modify an extended procedure definition.

To edit an extended procedure

1 Open an editor on the extended procedure. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you specify the **Library Name** for the extended procedure. NOTE: If the **Library Name** box shows a **Source text for this compiled object is hidden** message, source text for this compiled object has been hidden. For more information, see Hide Text.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Permissions	For details on using this tab, see Working with Privileges and Permissions.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Foreign Keys Editor (Sybase ASE)

The Foreign Keys Editor lets you modify the column mapping and specify properties of a foreign key.

To edit a foreign key

1 Open an editor on the foreign key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Column Mappings	The existing column mapping for the foreign key is represented by selected columns in the **Main Table** and **Referenced Table** lists. Additional candidates in the **Referenced Table** list are indicated by enabled column check boxes. If necessary, use the **Table** dropdown in the **Referenced Table** group to choose a new table for this foreign key. Select or deselect columns in the **Main Table** list and **Referenced Table** list to form the referential constraint between the two tables.
Properties	Lets you specify a **Match Full** value of TRUE or FALSE.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Functions Editor (Sybase ASE)

The Functions Editor lets you view and modify the body of a function, as well as work with permissions and dependencies.

To edit a function:

1 Open an editor on the function. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Definition	Lets you view and modify the CREATE FUNCTION statement, including the function body, for the function. NOTE: If the **Definition** tab for the function shows a **Source text for this compiled object is hidden** message, source text for this compiled object has been hidden. For more information, see Hide Text.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Permissions	For details on using this tab, see Working with Privileges and Permissions.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Groups Editor (Sybase ASE)

The Groups Editor lets you manage users in a group and the group's associated privileges.

Tip: The refresh button lets you refresh or clear the editor's contents, and log SQL.

To edit a group

1 Open an editor on the group. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Users	Lets you add users to, or remove users from the group. To move a user from one of the **Users Not In Group** or **Users In Group** lists to the other list, select the user in the list and click the **Join Group** or **Leave Group** button.
Object Privileges	For details on using this tab, see Working with Privileges and Permissions.
System Privileges	For details on using this tab, see Working with Privileges and Permissions.
DDL	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Indexes Editor (Sybase ASE)

The Indexes Editor lets you manage columns, properties, and partitions, and view statistics for an index.

To edit an index

1 Open an editor on the index. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Columns	Lets you manage columns that make up the index. On opening, this tab shows the existing columns. For each column, the datatype (and if applicable the precision in brackets) and whether the table definition permits nulls in the target table column. Optionally you can:	
		Change the **Sort** order of a column.
		Click the **New** button and select a column name from the dropdown, to add a column to the index.
		Select a column and click the **Drop** button to delete the column from the index.
		Select a column and use the arrow buttons to reorder the columns in the index.
Properties	Lets you work with settings in the following categories:	
	Attributes	Has the **Index Type** (UNIQUE or NONUNIQUE), **Clustered**, **Ignore Duplicate Key** (available with **Index Type** of UNIQUE), **Ignore Duplicate Rows** (available with **Clustered** enabled), **Allow Duplicate Rows** (available with **Clustered** enabled), and **Maximum Rows Per Page** properties.
	Storage	Lets you set **Reserve Page Gap**, **Segment Name**, **Fill Factor**, **Prefetch Strategy**, **MRU Replacement Strategy** settings
Partition	Lets you work with settings in the following categories:	
	Properties	Lets you view the **Locality** (Global/Local), **Alignment** (Prefixed/Non-Prefixed), **Partition Type** (including Range-Hash Composite or Range-List Composite), and **Subpartition type** properties.
	Columns	Displays partitioning columns.
	Partition Definitions	Displays details for each partition.
Statistics	Lets you view statistics in the following categories:	
	Page Statistics	**Data Pages**, **Reserved Pages**, **Used Pages**, **OAM Page Ratio**, and **Index Page Ratio**.
	Row Statistics	**Maximum Row Size**, **Minimum Row Size**, **Max Size of Non-Leaf Index Row**, and **Maximum Rows Per Page**.

DDL View For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Logins Editor (Sybase ASE)

The Logins Editor lets you manage basic properties for a login, associated users and roles, and accounting.

To edit a login:

1 Open an editor on the login. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

<table>
<tr><th>Tab</th><th>Settings and tasks</th></tr>
<tr><td><b>Properties</b></td><td>Lets you work with the following properties set when creating the login: <b>Name</b>, <b>FullName</b>, <b>Default database</b>, <b>Default language</b>, <b>Locked</b> (or <b>Account Locked</b>), <b>Password</b>, <b>Password expiration</b>, <b>Minimum password length</b>, <b>Maximum Login attempts</b>, and <b>Authentication Mechanism</b>. Against Sybase 15.7 and higher, servers, <b>Profile</b> (with an additional <b>default</b> option available) and <b>Exempt Inactive Lock</b> parameters are available. For detailed information on these settings, see Logins Wizard (Sybase ASE). In addition, when editing an existing login, you have access to the following settings:
<table>
<tr><td><b>Is currently logged in</b></td><td>Displays whether the user is currently logged in.</td></tr>
<tr><td><b>Login trigger</b></td><td>Displays the login trigger currently configured for this login. This is the stored procedure automatically executed when the user logs on. For information on configuring login triggers for a login, see Add/Modify Login Trigger and Drop Login Trigger.</td></tr>
</table></td></tr>
<tr><td><b>Users</b></td><td>Lets you maintain database user accounts for a login. For more detailed information, see Logins (Sybase ASE) - Users.</td></tr>
<tr><td><b>Roles</b></td><td>Lets you maintain the roles granted to this login. For more detailed information, see Logins (Sybase ASE) - Users. For information on activating and deactivating roles for the current login in the current session, see Role Activation.</td></tr>
<tr><td><b>Accounting</b></td><td>The <b>Accounting</b> tab of the Logins Editor lets you manage chargeback accounting statistics for every login on the current server. Chargeback accounting statistics are CPU and I/O usage statistics that Sybase ASE accumulates for every login. To start a new accounting period, the system administrator must clear all previous login statistics. Optionally, you can click the <b>Clear Statistics</b> button to start a new accounting interval. The <b>Clear Statistics</b> dialog lets you clear statistics immediately or use scheduling options.</td></tr>
<tr><td><b>DDL</b></td><td>For details on using this tab, see Viewing the SQL/DDL for an Object.</td></tr>
</table>

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Primary Keys Editor (Sybase ASE)

The Primary Keys Editor lets you manage the columns that make up the primary key and its basic properties, as well as view statistics for the primary key.

To edit a primary key

1 Open an editor on the primary key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Columns	Lets you manage columns that make up the primary key. On opening, this tab shows the existing columns. For each column, the datatype (and if applicable the precision in brackets) and whether the table definition permits nulls in the target table column. Optionally you can:	
		Change the **Sort** order of a column.
		Click the **New** button and select a column name from the dropdown, to add a column to the primary key.
		Select a column and click the **Drop** button to delete the column from the primary key.
		Select a column and use the arrow buttons to reorder the columns in the primary key.
Properties	Lets you work with settings in the following categories:	
	Attributes	Lets you set the **Clustered** and **Maximum Rows Per Page** properties.
	Storage	Lets you set **Reserve Page Gap**, **Segment** (DEFAULT, LOGSEGMENT or SYSTEM), and **Fill Factor** properties.
Statistics	Lets you view statistics in the following categories:	
	Page Statistics	**Data Pages**, **Reserved Pages**, **Used Pages**, **OAM Page Ratio**, and **Index Page Ratio**.
	Row Statistics	**Maximum Row Size**, **Minimum Row Size**, **Max Size of Non-Leaf Index Row**, and **Maximum Rows Per Page**.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Procedures Editor (Sybase ASE)

The Procedures Editor lets you view and modify procedure definitions and modify associated privileges and dependencies.

To edit a procedure

1 Open an editor on the procedure. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Definition	Lets you modify the SQL code for a procedure. NOTE: If the **Definition** tab for the procedure shows a **Source text for this compiled object is hidden** message, source text for this compiled object has been hidden. For more information, see Hide Text.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Permissions	For details on using this tab, see Working with Privileges and Permissions.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Remote Servers Editor (DBArtisan - Sybase ASE)

The Remote Server Editor lets you manage the connection and security properties for a remote server, and manage the associated logins.

Note: Before using this editor, consult Sybase documentation for information on configuring remote server logins. For more information, see Accessing Third Party Documentation.

To edit a remote server

1 Open an editor on the remote server. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you work with the **Network Name**, **Server Type**, **Timeout Inactive Connections**, **Network Password Encryption**, **Security Model**, **Security Mechanism**, **Message Integrity**, **Message Confidentiality**, and **Mutual Authentication** settings. For details on these settings, see Remote Servers (Sybase ASE) - Properties.
Logins	Lets you build pairs of sp_addremotelogin and sp_remoteoption system procedure calls that authorize a new remote server user and configures remote login options for that user: Click the Add Remote Login button to open a dialog that displays or lets you provide: **Remote Server Name -** Displays the remote server name. **Remote Login Name -** Lets you specify a remote login name. Select the **All remote users** box if you want to use this name for all remote logins. **Local Login Name -** Lets you select a local user name. **Trusted Remote Login -** Lets you specify this remote login as a trusted login. Select a login, click **Modify**, and use the table above as a guide to modifying login options. Select a login, and click **Drop** to delete the login.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Roles Editor (DBArtisan - Sybase ASE)

The Roles Editor lets you manage authentication, logins, and privileges for a role.

Note: User-defined roles are available in Sybase ASE 11.5 or later.

To edit a role

1 Open an editor on the role. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Displays the role **Name**, and lets you wok with the **Password**, **Password Expiration**, **Minimum Password Length**, and **Maximum Failed Logins** settings. For detailed information on these settings, see Roles (Sybase ASE) - Properties. This tab also lets you work with the following settings: **Apply Server Wide** – Displays whether the **maximum failed logins** server configuration parameter has been set. If **Apply Server Wide** is unselected, the **Maximum Failed Logins** setting applies only to the current role. For information on setting the **maximum failed logins** configuration parameter, see Session Recording (DBArtisan). **Role Locked** – Lets you manually lock the role.
Logins/Roles	Displays the logins or roles that have been granted to this role in the **Objects that have this Role granted** list. Other roles and logins available to be granted to this role are shown in the Objects that don't have this Role granted list. Optionally, you can: Use the **Logins/Roles** radio boxes to toggle the two lists between logins and roles display. Grant a login or another role to this role by selecting the login/role from the Objects that don't have this Role granted list and clicking **Grant**. Revoke a login or role from this role by selecting the login/role from the **Objects that have this Role granted** list and clicking **Revoke**.
Object Privileges and **System Privileges**	For details on using these tabs, see Working with Privileges and Permissions.
DDL	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

For information on activating and deactivating roles for the current login in the current session, see Role Activation.

Rules Editor (Sybase ASE)

The Rules Editor lets you modify basic properties of a rule.

To edit a rule

1 Open an editor on the rule. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you specify a [JavaScript:popup.TextPopup(Poptext1,Popfont,9,9,-1,-1) **Restriction**] and select a **Type** (STANDARD_RULE, AND_ACCESS_RULE, OR_ACCESS_RULE). NOTE: If the **Restriction** box shows a **Source text for this compiled object is hidden** message, source text for this compiled object has been hidden. For more information, see Hide Text.
Dependencies	For details on using this tab, see Working with Object Dependencies.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Segments Editor (Sybase ASE)

The Segments Editor lets you manage segment location, associated objects, segment space, and segment thresholds.

To edit a segment

1 Open an editor on the segment. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	This tab lets you view the **Name** and **Max Size** settings you specified when creating the segment. It also lets you change the **Grow By** and **Threshold Free Space** settings. For information on these settings, see Segments (Sybase ASE) - Properties. This tab also displays a **Hysteresis** value, the global variable `@@thresh_hysteresis`, that controls threshold sensitivity to variations in free space.
Location	Lets you **Drop** or **Extend** selected segments.
Objects	Lets you manage database objects associated with the segment. Objects are organized in a tree structure. You can open an object editor on a table, index, or constraint by double-clicking that object. For information on specific editors, see Sybase ASE Object Editors.
Space	Displays a summary of segment space distribution, broken down into data, indexes, unused, and other. Also provides reserved, data, indexes, and unused space values for each object on the segment.
Threshold	Displays a list of the free pages thresholds for the segment. For a selected threshold, the **Property**/**Value** list displays the associated procedure name. Optionally you can: Click the **Add** button to specify the number of free pages and procedure name for a new segment threshold. Select an existing threshold and modify the threshold’s procedure name value. Select an existing threshold and click the **Delete** button to delete that threshold
DDL	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Tables Editor (Sybase ASE)

The Tables Editor lets you manage columns, basic properties, constraints, storage and space, and partitions of a table.

To edit a table

1 Open an editor on the table. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Columns	Displays the currently defined columns in the table. For any selected column, the Property/Value list provides additional detail on that column. Available properties depend on the datatype you choose as well as on the property values you select: **Computed** and **Computed Expression** - Let you define a column as a computed column and provide the computed column expression. **Type** - Lets you select a datatype (depending on the type, additional properties such as **Size**, **Width**, and **Scale** may be available). **Identity Column** - Select this check box to define the column as an identity column. **Allow Nulls** - Select this check box to allow nulls in this column. **Default Collation** - available for text/character datatypes, lets you specify a default collation. **Default Value** - Lets you type a constant value or select a function returning a constant value to serve as the default for the column **Default Binding** and **Rule Binding** - Let you bind a rule or default to a column. Optionally, you can:
	Select a column, and in the **Property**/**Value** list modify property values for that column.
	Click **Add Column**, provide a name for the new column, and set property values for the column.
	Select a column and click **Delete** to remove the column from the table.
Properties	Lets you work with properties in the following categories:
	Physical Storage – Lets you set the **Segment Name** (DEFAULT, SYSTEM), **Maximum Rows Per Page**, **Reserve Page Gap**, and **Identity Gap** properties.
	Cache Strategy – Lets you set **MRU Replacement Strategy** and**Prefetch Strategy** properties.
	Row Locking Strategy – Lets you view the **Expected Row Size** property and set the**Lock Scheme** (ALLPAGES, DATAPAGES, ROWPAGES) property.
Indexes	Lets you manage indexes for a table. On opening, the list of current indexes for the table is displayed. Optionally, take one of the following actions:
	Click **Add** to open a dialog that lets you add a new index to the table.
	Select an index and click **Edit** to open a dialog that lets you edit index properties.

	Select an index and click **Drop** to open a dialog that lets you remove the index from the table.
Constraints	Lets you manage constraints for the table. Constraints are grouped by type, under folders. Optionally take one of the following actions: Select a constraint type folder and click **Add** to open a dialog that lets you add a constraint of that type. Select a constraint and click **Edit** to open a dialog that lets you modify the constraint details. Select a constraint and click **Drop** to remove the constraint.
Space	Lets you view the table usage and the distribution of table space for a table. Optionally, double-click a slice in the pie chart for detailed statistics.
Partitions	Displays partition details for the table including the **Partition Type** (ROUNDROBIN, HASH, LIST, RANGE), partitioning **Columns** (name and datatype) and **Partition Definitions** (Name, values, segments). You can also add or modify partition definitions. For more information, see Partitioning Sybase ASE Tables.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Permissions	For details on using this tab, see Working with Privileges and Permissions.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Partitioning Sybase ASE Tables

The **Create Partition** and **Edit Partition** wizards let you work with table partitions.

To create or edit table partitions

1 Open an editor on a table. For details, see Opening an Object Editor.

2 On the **Partition** tab, click the **Create Partition** or **Edit Partition** button.

3 Use the following table as a guide to understanding and modifying the settings on the tabs of this wizard:

Tab	Settings and tasks
Properties	In addition to displaying basic identification and ownership properties, this pane lets you select a **Partition Type** (ROUNDROBIN, HASH, LIST, RANGE).
Columns (not available for a **Partition Type** of ROUNDROBIN)	For each column to be added, click **New** to add a column and select a **Column** name.
Range Definitions (only available for a **Partition Type** of RANGE)	Lets you create an ordered list of partitions. For each partition, click **New** to open the **Add Partition Definition** wizard. Provide a **Name**, select a **Segment**, (DEFAULT, SYSTEM, LOGSEGMENT) and click **Add**. The **Range Definitions** pane also lets you edit and remove partitions.
Partitions Definitions (only available for a **Partition Type** of HASH)	Lets you specify a Partition method: **NoneNumber of Partitions -** Specify a **Number of Partitions**, and then for each partition, click the **New** button and select a segment. **By Partition Name -** For each partition, click **New**, specify a **Name** and select a **Segment**.
List Definitions (only available for a **Partition Type** of LIST)	Lets you create a list of partition definitions. For each partition, click **New** to open the **Add Partition Definition** wizard. Provide a **Name**, select a **Segment**, (DEFAULT, SYSTEM, LOGSEGMENT) and click **Add**. The **List Definitions** pane also lets you edit and remove partitions.

4 When ready, click **Finish**.

Triggers Editor (Sybase ASE)

The Triggers Editor lets you modify the trigger body of a trigger and change its enabled/disabled status.

To edit a trigger:

1 Open an editor on the trigger. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view the basic identifying properties of the trigger and its parent view or table: **Parent Type**, **Parent Owner**, **Parent Name**, **Owner**, **Name**, **Create Date**, **Fire On Insert**, **Fire On Update**, and **Fire On Delete**. For details on these properties, see Triggers (Sybase ASE) - Properties. This panel also lets you specify whether the trigger is **Enabled**.
Definition	Lets you modify the CREATE TRIGGER body for a trigger. To modify a trigger, the trigger must be dropped then recreated. NOTE: If the **Definition** tab for the trigger shows a **Source text for this compiled object is hidden** message, source text for this compiled object has been hidden. For more information, see Hide Text.
Dependencies	For details on using this tab, see Working with Object Dependencies.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes

Unique Keys Editor (Sybase ASE)

The Unique Keys Editor lets you manage the columns and basic properties for a unique key as well as view page and row statistics.

To edit a unique key

1 Open an editor on the unique key. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Columns	Lets you manage columns that make up the unique key. On opening, this tab shows the existing columns. For each column, the datatype (and if applicable the precision in brackets) and whether the table definition permits nulls in the target table column. Optionally you can: Change the **Sort** order of a column. Click the **New** button and select a column name from the dropdown, to add a column to the index. Select a column and click the **Drop** button to delete the column from the index. Select a column and use the arrow buttons to reorder the columns in the index.
Properties	Lets you work with properties in the following categories: **Attributes** Lets you set **Clustered** and **Maximum Rows Per Page** properties. **Storage** Lets you set the **Reserve Page Gap**, and **Segment** (DEFAULT, LOGSEGMENT, SYSTEM) properties.
Statistics	Lets you view statistics in the following categories: **Page Statistics** **Data Pages**, **Reserved Pages**, **Used Pages**, **OAM Page Ratio**, and **Index Page Ratio**. **Row Statistics** **Maximum Row Size**, **Minimum Row Size**, **Max Size of Non-Leaf Index Row**, and **Maximum Rows Per Page**.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

User Datatypes Editor (Sybase ASE)

The User Datatypes Editor lets you view and modify user datatype definitions and view objects associated with a user datatype.

To edit a user datatype

1 Open an editor on the user datatype. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks	
Properties	Lets you work with properties in the following categories:	
	Base datatype	Lets you set **Type**, **Size**, **Allow Nulls**, and **Identity** settings.
	Bindings	Lets you set **Default Binding** and **Rule Binding** properties.
Usage	Displays all objects referring to or referenced by a user datatype.	
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.	

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

User Messages Editor (Sybase ASE)

The User Messages Editor lets you manage user messages and their respective object bindings.

Note: On Sybase ASE, user messages are only available under the **Master** database.

To edit a user message

1 Open an editor on the user message. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Information	Displays the message text for each language version of the message and offers the options to add a new language version, drop a language version, or modify a language version. For detailed instructions, see User Messages (Sybase ASE) - Information.
Bindings	Displays any constraints to which the user message is bound. This tab also lets you create new bindings and delete current bindings. For details, see User Messages (Sybase ASE) - Bindings.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Users Editor (Sybase ASE)

The Users Editor lets you view and modify user definitions and manage associated logins and objects.

To edit a user

1 Open an editor on the user. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Lets you view the user **Name** and **Login Name** for the user. Lets you specify a **Group Name** for the user.
Logins	Lets you assign logins to a user by selecting associated check boxes.
Objects	Lets you manage database objects associated with the user. Objects are organized in a tree structure with folders containing the objects. Optionally, take one of the following actions: Select an object and click **Edit** to open an object editor on the selected object. Select an object and click **Drop** to initiate dropping the selected object.
Object Permissions and **System Permissions**	For details on using these tab, see Working with Privileges and Permissions.
Comment	For details on using this tab, see Adding a Comment to an object.
DDL View	For details on using this tab, see Viewing the SQL/DDL for an Object.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Views Editor (Sybase ASE)

The Views Editor lets you view the columns for the view and work directly with the SQL to create the view.

To edit a view

1 Open an editor on the view. For details, see Opening an Object Editor.

2 Use the following table as a guide to understanding and modifying the settings on the tabs of this editor:

Tab	Settings and tasks
Properties	Displays all columns for the view. Details for each column include the **Column Name**, the **Datatype** (and if applicable, with the precision in parentheses), and whether or not **Nulls** are allowed for that column.
Definition	Displays the SQL (CREATE VIEW) for the view. NOTE: If the **Definition** tab for the view shows a **Source text for this compiled object is hidden** message, source text for this compiled object has been hidden. For more information, see Hide Text.
Dependencies	For details on using this tab, see Working with Object Dependencies.
Permissions	For details on using this tab, see Working with Privileges and Permissions.

3 When finished, you can submit your changes. For details, see Previewing and Submitting Object Editor Changes.

Object actions

In addition to letting you create and perform basic editing on various object types, there are a number of actions can be applied to objects. While object action support differs by DBMS platform, they are most commonly implementations of:

- ALTER *OBJECT TYPE* clauses/options not typically available from object editors, such as REBUILD INDEX and RENAME
- Other SQL commands such as TRUNCATE, REORG, and LOCK
- System procedures and other available utilities

Most object actions are implemented as multi-panel wizards that guide you through the action. For an introduction to invoking object actions and working through the panels, see Overview of object actions/operations execution.

For a listing of available actions and access to detailed instructions, see Available object actions by DBMS.

Overview of object actions/operations execution

Most commonly, initiating an object operation opens a wizard with panels that guide you through initiating the action. For details, see the following topics

- Initiating an object operation
- Using object operation wizards

Initiating an object operation

Object operations are most commonly available as right-click context menu selections on the Database Navigator/Explorer, available with one or more objects selected.

Note: To verify availability of an object action for a specific object type and DBMS platform, see the topic covering that object type in Supported Objects.

To Initiate an Object Operation in DBArtisan

1 Connect to the datasource that contains the object you want to perform the operation against. For more information, see Connecting to Datasources.

2 On the Datasource Navigator/Explorer, expand nodes until the target object type is visible. If working in the Datasource Navigator, expand the object type node or if working on the Datasource Explorer, click the target object type. See the following topics for more information:

- Datasource Navigator and Datasource Explorer. - for general information on working with the Navigator and Explorer.
- Supported Objects, specific object type topic - for information on navigating Datasource Navigator/Explorer nodes to the target object type.

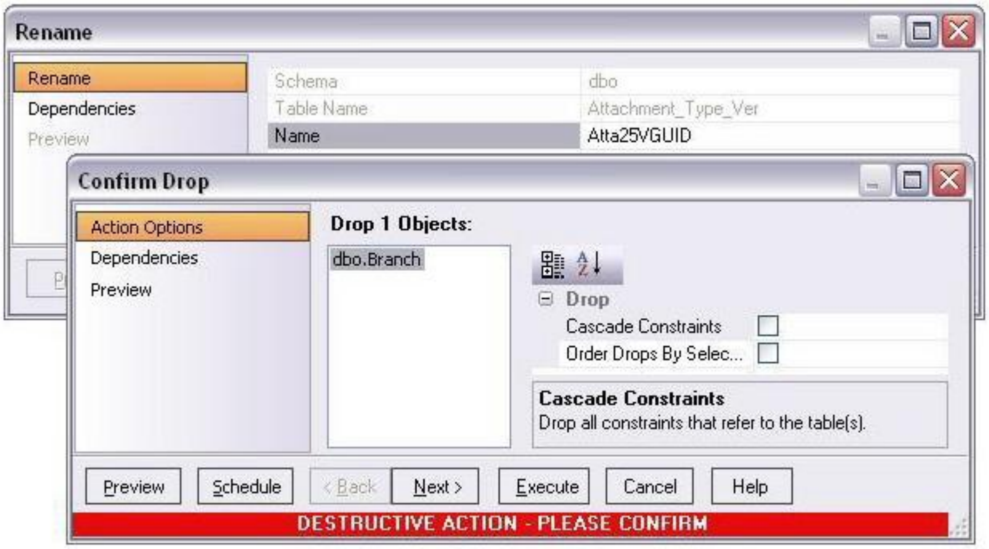
3 Take one of the following actions:

- To initiate an operation against a single object, right-click the object, and select the operation from the context menu. Alternatively, select the object and then, from the Datasource Navigator/Explorer Toolbar's **Command** menu, select the action.
- To initiate an operation against a multiple objects, first use CTRL-clicking to select multiple objects. Then, either right-click one of the selected objects and select the operation from the context menu, or select an operation from the Datasource Navigator/Explorer toolbar's **Command** menu.

For information on using the wizard panels to provide required information and execute an operation, see Using object operation wizards.

Using object operation wizards

A typical object operation wizard is a multi-panel window that walks you through the steps necessary to execute the operation. The wizard typically opens on a first panel named either **Action Options** or using a name specific to that object action. For example, the first panels for the **Rename** and **Drop** operations:



Schema: dbo
Table Name: Attachment_Type_Ver
Name: Atta25VGUID

Drop 1 Objects:
dbo.Branch

Drop
Cascade Constraints ☐
Order Drops By Selec... ☐

Cascade Constraints
Drop all constraints that refer to the table(s).

DESTRUCTIVE ACTION - PLEASE CONFIRM

The wizard's **Next** and **Back** buttons let you navigate through the wizard panels. While the remaining panels of the wizard differ by action type, the following are the most common:

- Dependencies
- Preview

After providing information and inspecting settings on the panels, there are two execution options:

- Use the **Execute** (or **Finish**) button to execute the operation immediately
- Use the **Schedule** button to schedule the operation to execute the operation at a later time or to be executed on a regular basis. For more information, see Scheduling.

Dependencies

For those object action wizards that include a **Dependencies** panel, it lists the objects potentially impacted by this object action. Within the main categories of **Referencing Objects** and **Referenced Objects**, objects are grouped within object type folders.

Caution: A **Dependencies** panel, if present, is provided for information only. No attempt is made to resolve the current action in referenced or referring objects.

Preview

The Preview panel lets you inspect the SQL generated by wizards or editors when you create, edit, or perform other actions against a database object. Depending on the action being taken, this panel may contain the following controls:

Button	Description
💾	Lets you save the SQL.
🖨	Lets you print the SQL.
✉	Lets you send the SQL via e-mail.
SQL	Opens an SQL Editor on the SQL contained in the Preview dialog. For more information, see Using the SQL Editor.
Schedule	Opens a dialog that lets you select job scheduling options. Once you have selected your options, a third-party job scheduler opens. For more information, see Scheduling.
Execute	Executes the SQL.
Cancel	Closes the Preview dialog without executing the SQL and returns you to the wizard, editor, or dialog box from which you invoked the Preview dialog. Since you cannot edit the SQL in the Preview dialog, this lets you make changes before once again previewing and then executing the SQL.

Available object actions by DBMS

The table below lists the available object actions by DBMS platform.

	DB2 LUW	DB2 z/OS	IITB/FBD *	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
Add or Modify Check Constraint	✓	✓	✓		✓		✓	✓	
Add/Modify Login Trigger								✓	
Add Private Key							✓		
Allocate Extent					✓		✓		
Analyze					✓		✓		
Analyze Tables				✓					
Attach Database							✓		
Backup (DBArtisan)							✓	✓	
Backup Certificate							✓		
Bind Package		✓							
Bind To Temporary Database								✓	
Build		✓							
Build Query	✓	✓	✓	✓	✓		✓	✓	
Cancel Thread (DBArtisan)		✓							
Change Access Status					✓				
Change Category					✓				
Change Job's Owner (DBArtisan)							✓		
Change Password					✓		✓		
Change Status			✓		✓		✓	✓	
Check Data (DBArtisan)		✓							
Check Disk Group (DBArtisan)					✓				
Check Index (DBArtisan)		✓							
Check Tables				✓					
Checkpoint							✓	✓	
Checksum Tables				✓					

Coalesce					✓			
Compile					✓			
Convert Tables				✓				
Copy Object Names	✓	✓			✓	✓	✓	
Copy Schema	✓							
Copy to Copy (DBArtisan)		✓						
Create Alias	✓	✓						
Create Clone		✓						
Create Insert Statements	✓	✓	✓	✓	✓	✓	✓	
Create Like	✓	✓	✓	✓	✓	✓	✓	
Create Synonym	✓	✓			✓			
Create View	✓	✓	✓		✓	✓	✓	
DBCC						✓	✓	
Deallocate Unused Space					✓			
Delete Statistics							✓	
Describe	✓							
Detach Database						✓		
Disable Index			✓			✓		
Disable Job					✓			
Disable Keys				✓				
Disable/Enable Triggers						✓	✓	
Disconnect ASM Connection (DBArtisan)					✓			
Drop	✓	✓	✓	✓	✓	✓	✓	✓
Drop Automatic Storage Path(s)	✓							
Drop By Category					✓			
Drop Clone		✓						

Drop Java					✓			
Drop Login Trigger							✓	
Drop Materialized Query Table	✓							
Drop Unused					✓			
Drop Unused					✓			
Backup Certificate						✓		
Enable Job (Job Queue)					✓			
Enable Keys				✓				
Enable Recycle Bin					✓			
Enable/Disable (Oracle Jobs)					✓			
Enable/Disable Job (DBArtisan - SQL Agent Job)						✓		
Estimate Size							✓	
Exchange Data With Clone		✓						
Execute			✓		✓		✓	
Extract	✓	✓	✓	✓	✓	✓	✓	✓
Extract Data as XML					✓	✓		
Flashback Table					✓			
Flashback Recycle Bin Entry					✓			
Flush Cache	✓							
Flush Tables				✓				
Free (Packages)		✓						
Free Plan		✓						
Hide Text							✓	
Import Data From File	✓	✓	✓	✓	✓	✓	✓	
Kill (DBArtisan)	✓			✓	✓	✓	✓	
Load Java					✓			

Lock	✓	✓					
Lower High Water Mark	✓						
Manage ASM Connection (DBArtisan)				✓			
Merge Copy (DBArtisan)		✓					
Migrate (DBArtisan)	✓	✓		✓	✓	✓	
Mirror (DBArtisan)						✓	
Modify Recovery (DBArtisan)		✓					
Modify Statistics (DBArtisan)	✓				✓	✓	
Mount Disk Group (DBArtisan)			✓				
Move Log						✓	
Move Table	✓						
Next Used Filegroup					✓		
Object Properties							✓
Open (DBArtisan - Logs and Error logs)					✓		
Open Control Files (DBArtisan)				✓			
Optimize Tables			✓				
Place					✓	✓	
Population status					✓		
Purge Recycle Bin				✓			
Purge Recycle Bin Entry				✓			
Quiesce	✓						
Quiesce Tablespaces (DBArtisan)		✓					
Reassign By Category				✓			
Rebalance	✓						

Rebalance Disk Group (DBArtisan)					√			
Rebind Packages	√	√						
Rebind Plans	√	√						
Rebuild (Full-text Catalogs)							√	
Rebuild Index		√	√		√		√	
Rebuild Outlines					√			
Rebuild Table				√				
Recompile							√	√
Recover (DBArtisan)		√						
Recover Abort (DBArtisan)		√						
Recover Commit (DBArtisan)		√						
Redistribute (DBArtisan)	√							
Refit (DBArtisan)								√
Refresh Materialized View						√		
Refresh Table	√							
Reinitialize (DBArtisan)								√
Rename	√	√	√	√	√		√	√
Reorganize /Reorg	√	√			√		√	√
Repair Tables				√				
Report	√	√	√	√	√		√	√
Resize (DBArtisan)								√
Restart (Object Action)	√							
Restart Sequence(s)							√	
Resynchronize		√						
Run Job (Oracle Jobs)					√			
Run Job (job queues)	√							

Role Activation					✓			✓
Rotate Log (DBArtisan)						✓		
Run Job (job queues)					✓			
Runstats Index (DBArtisan)		✓						
Runstats Tablespace (DBArtisan)		✓						
Schema (Object Action)	✓	✓	✓	✓	✓		✓	✓
Select * From	✓	✓	✓	✓	✓		✓	✓
Set Default					✓			
Set Integrity	✓	✓						
Set Online/Offline								✓
Set UNDO					✓			
Shrink					✓		✓	
Start Database		✓						
Start/Stop Job (DBArtisan - SQL Agent Job)							✓	
Stop Database		✓						
Stop Job					✓			
Switch Logfile (DBArtisan)					✓			
Switch Online	✓							
Take Checkpoint (DBArtisan)					✓			
Transfer Ownership	✓							
Truncate	✓	✓		✓	✓		✓	✓
Unbind From Temporary Database								✓
Undrop Disk (DBArtisan)					✓			
Unmirror (DBArtisan)								✓
Update Statistics	✓						✓	✓

View Contents (DBArtisan)	✓
View Job History (DBArtisan)	✓

Add or Modify Check Constraint

This action opens a Check Constraints Editor or Check Constraints Wizard.

Platform Availability

- DB2 LUW, DB2 z/OS, MySQL, ORCL, SQL SVR, SYB ASE

To Open the Check Constraint Editor or Check Constraint Wizard

1 On the Navigator/Explorer, right-click the **Check Constraints** node and select **Create**.

2 Use the following table as a guide to creating or editing the constraint. Note that settings available varies by DBMS type.

Step	Settings and tasks	
Definition	**Table Owner** or **Table Schema**	Select the owner of the table for which the check constraint is to be created.
	Table or **Table Name**	Lets you select where you want to place the check constraint.
	Name	Lets you type the name of the constraint, which can be up to 30 characters long and must be unique across the entire database.
	Enabled or **Enforced**	Enables or disables the check constraint.
	Not for Replication	Enabling this feature ensures that the constraint is not enforced if the table is replicated for another database.
	Check Condition	Type the check constraint condition. As a convenience, use the **Table Columns** button to paste in column names as part of the condition.
Comment	Lets you type a comment.	
DDL View	Preview the DDL generated by your choices. Finally, use the **Execute** button to create the check constraint.	

Add/Modify Login Trigger

This action lets you build and submit an **sp_modifylogin** or ALTER LOGIN call that specifies a login script option. The target login trigger can be any stored procedure located on the user's default database, and the procedure will run each time the user successfully logs in. The action can be used to:

- Add a login trigger to a login that currently has no login trigger configured
- Change the specified stored procedure for a login that already has a login trigger configured

Platform Availability

- SYB ASE

To Specify the Stored Procedure Used as a Login Trigger for a Login:

1 If using DBArtisan, on the Datasource Navigator/Explorer, expand the data source nodes until the **Logins** node is visible and then select the **Logins** node. DBArtisan displays the logins.

2 Right-click a login and select **Login Trigger Actions > Add/Modify Login Trigger** from the context menu. The **Add/Modify Login Trigger** dialog opens.

3 Use the following table as a guide to understanding and modifying settings in the wizard:

Step	Settings and tasks
Action options	From the **Login Procedure** dropdown, choose the default database stored procedure that is to be configured as the login trigger for this login.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation. For details, see Dependencies.
Preview	Preview the DDL generated for the operation and when ready, click **Execute**.

After executing this action, the specified stored procedure name is available in the Login editor. For details, see Logins Editor (Sybase ASE).

Related Information

- Drop Login Trigger

Add Private Key

The **Add Private Key** object action builds and submits an ALTER CERTIFICATE... WITH PRIVATE KEY statement, optionally including a DECRYPTION BY PASSWORD argument. This lets you change the private key for a certificate, add a private key to a certificate that currently does not have one, and optionally, specify a decryption password.

Note: This functionality is available for Microsoft SQL Server only.

To change a certificate's private key

1 Initiate an **Add Private Key** action against a certificate. For more information see Initiating an object operation.

The **Add Private Key** dialog box opens.

2 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks	
Action options	Lets you work with the following settings:	
	Private Key File	The full path and filename (local or UNC) of the file to which the private key is to be written.
	Decryption Password	The password required to decrypt the key.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation. For details, see Dependencies.	
Preview	Displays the DDL that will execute the object action. For details, see Preview.	

3 Finally, use the **Execute** button to create the object.

For related information, see Backup Certificate.

Allocate Extent

The Allocate Extent dialog lets you explicitly allocate extents for clusters, tables, and indexes in Oracle. Though Oracle dynamically allocates extents when additional space is required, explicit allocation of additional extents can be useful if you know that an object grows.

Explicit allocation of additional extents can be particularly helpful when using Oracle Parallel Server. When using Oracle Parallel Server and allocating additional extents, you can allocate an extent explicitly to a specific instance in order to minimize contention for free space among multiple instances.

Platform Availability

- ORCL

Important Notes

For composite-partitioned tables, you can allocate extents to subpartitions as well as partitions.

To Allocate an Extent for a Cluster, Table, or Index

1 Initiate an **Allocate Extent** action against a table, cluster, index, primary key, or unique key. For more information see Initiating an object operation.

2 Use the following table as a guide to setting the options on this dialog:

Option	Description
Extent Size	Specify an extent size in KBs or MBs.
Datafile	Lets you select the new datafile. You can choose a specific datafile from which to take space for the added extent. If you choose (Default), Oracle takes the space from any accessible datafile in the tablespace containing the table, index, or cluster.
Instance	Lets you specify a freelist from which to draw the extent. If you are using Oracle Parallel Server, you can assign the new extent to a free list group associated with a specific instance. The number you enter in the Instance text box should be the number of the freelist group that you wish to use, rather than the number of the specific instance. If you are using Oracle Parallel Server and you omit this parameter, Oracle allocates the extent, but the extent is drawn from the master freelist by default. Only use this parameter for Oracle Parallel Server. NOTE: The number you enter in the Instance field should be the number of the free list group that you wish to use, rather than the number of the specific instance.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Analyze

This action lets you work with statistics for tables, indexes (primary keys and unique keys), and clusters.

Platform Availability

- ORCL

This action lets you build and submit the following ANALYZE statement variations or DBMS_STATS package procedure calls:

Options	ANALYZE statement variation or DBMS_STATS call generated
Compute Statistics	ANALYZE INDEX/CLUSTER/TABLE... COMPUTE STATISTICS
Gather Statistics	DBMS_STATS.GATHER_INDEX_STATS DBMS_STATS.GATHER_TABLE_STATS,
Delete Statistics	ANALYZE INDEX/CLUSTER/TABLE... DELETE STATISTICS DBMS_STATS.DELETE_INDEX_STATS DBMS_STATS.DELETE_TABLE_STATS
Estimate Statistics	ANALYZE INDEX/TABLE/CLUSTER... ESTIMATE STATISTICS
Validate Structure	ANALYZE INDEX/TABLE/CLUSTER... VALIDATE STRUCTURE
Import Statistics	DBMS_STATS.IMPORT_INDEX_STATS DBMS_STATS.IMPORT_TABLE_STATS
List Chained Row Into Table	ANALYZE TABLE/CLUSTER... LIST CHAINED ROWS INTO

Note: Before using this object action, consult Oracle documentation for information on the ANALYZE statement and relevant DBMS_STATS package procedures. Be familiar with restrictions, required permissions, partition options, parameter values, and requirements. For more information, see Accessing Third Party Documentation.

To obtain or modify statistics for a database object:

1 Initiate an **Analyze** action against one or more clusters, indexes, primary keys, unique keys, or tables. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Analyze** wizard:

Step	Settings and tasks
Action Options	Take the following steps: If you are not working against a cluster, either select the **Use dbms_stats_package** check box to generate one of the DBMS_STATS procedure calls or deselect the **Use dbms_stats_package** check box to generate an ANALYZE statement variation. From the **Type** dropdown, select one of the options from the table above.

	Additional options are offered depending on the **Type** value you choose. They correspond to ANALYZE statement clauses or DBMS_STATS procedure parameters. See Oracle documentation for details.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Analyze Tables

This action lets you build and submit an ANALYZE TABLE statement, analyzing and storing the key distribution for a table.

Platform Availability

- MySQL

To analyze a table:

1 Initiate an **Analyze Tables** action against one or more tables. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in this wizard:

Step	Settings and tasks
Action Options	Lets you review the tables you selected.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Attach Database

The **Attach Database** object action lets you specify a set of previously detached data and transaction files to be used to attach a database to a Microsoft SQL Server instance. This makes the database available in exactly the same state it was in when it was detached

Platform Availability

- SQL SVR

To Attach a Database

1 Initiate an **Attach Database** action against a database. For more information see Initiating an object operation.

The **Attach Database File(s)** dialog opens.

2 In the **Database Name to be Attached** box, type the database name.

3 In the grid, do one of the following:

- Select the target database file(s).
- To add database file(s), click **Add** and then enter the name of the MDF (master data file) of the database to attach.

The appropriate *.ldf file is automatically added.

4 To drop database file(s), click **Drop** and then select the target file(s).

5 Click **Execute**. For information on the scheduling option, see Scheduling.

Related Information

- Detach Database

Backup (DBArtisan)

Right-clicking a SQL Server or Sybase ASE database and selecting **Backup** opens a tool that lets you backup that database. For details, see Database Backup (DBArtisan).

Backup Certificate

The **Backup Certificate** object action builds and submits a BACKUP CERTIFICATE... TO FILE statement with optionally, a WITH PRIVATE KEY clause. This lets you back up a certificate to file.

Platform Availability

- SQL SVR

To write a certificate to disk

1 Initiate a **Backup Certificate** action against a certificate. For more information see Initiating an object operation.

The **Attach Database File(s)** dialog box opens.

2 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks	
Action options	Lets you work with the following settings:	
	File	The full path and filename (local or UNC) of the file to which the certificate is to be written.
	Private Key File	The full path and filename (local or UNC) of the file to which the private key is to be written.
	Encryption Password	The password used to encrypt the private key before writing the key to the backup file.
	Decryption Password	The password used to decrypt the private key before backing up the key.
Dependencies	Lets you review the objects potentially impacted by this action. For more information, see Dependencies.	
Preview	Preview the DDL generated for the operation and when ready, use the **Schedule** or **Execute** button to perform this action.	

3 Finally, use the **Execute** button to create the object.

For related information, see Add Private Key.

Bind Package

The Bind Package Wizard lets you set package parameters, add environments, and set package properties.

Platform Availability

- DB2 z/OS

To Package Parameters, Add Environments, and Set Package Properties

1 Expand an owning schema, right-click **Packages**, and select **Bind** from the context menu.

The **Bind Package** wizard opens.

2 Use the following table as a guide to understanding and setting options on this wizard:

Step		Settings and tasks
1	**Location**	Lets you select the name of the location to connect to.
	Collection	Lets you select the location of the DBMS where the package binds and where the description of the package resides.
	Would you like to Copy an Existing Package from Another Collection	
	New Package (available if you select **No** for the **Would you like**... control)	Lets you select a Package Name and PDS Name.
	Copy Package (available if you select **Yes** for the **Would you like**... control)	Lets you qualify with the following: **Collection** - the collection to copy from. **Options** - COMMAND or COMPOSITE. **Package** - Lets you select a package. **Version** - Lets you select a version of the package.
2	Lets you select a package **Owner**, a **Qualifier** (the package creator), an **Action** of ADD or REPLACE, and the **Version** of the package.	
3	**Isolation**	Determines how far to isolate an application from the effects of other running applications.
	Keep Dynamic	Specifies that DB2 keeps dynamic SQL statements after commit points. The application does not need to prepare an SQL statement after every commit point. DB2 keeps the dynamic SQL statement until the application process ends, a rollback operation occurs or the application executes an explicit PREPARE statement with the same statement identifier. If the prepared statement cache is active, DB2 keeps a copy of the prepared statement in the cache. If the prepared statement cache is not active, DB2 keeps only the SQL statement string past a commit point. DB2 then implicitly prepares the SQL statement if the application executes an OPEN, EXECUTE, or DESCRIBE operation for that statement.

	Current Data	Determines whether to require data currency for read-only and ambiguous cursors when the isolation level of cursor stability is in effect. It also determines whether block fetching can be used for distributed, ambiguous cursors.
	Degree	Determines whether to attempt to run a query using parallel processing to maximize performance. Lets you select an option.
	DB Protocol	Specifies which protocol to use when connecting to a remote site that is identified by a three-part name statement.
	Dynamic Rules	Determines what values apply at run time for the following dynamic SQL attributes: The authorization ID that is used to check authorization The qualifier that is used for unqualified objects The source for application programming options that DB2 uses to parse and semantically verify dynamic SQL statements Whether dynamic SQL statements can include GRANT, REVOKE, ALTER, CREATE, DROP, and RENAME statements
	Release	**Commit** - Releases resources at each commit point. **Deallocate** - Releases resources only when the program terminates.
	Validate	Determines whether to recheck, at run time, errors found during bind. The option has no effect if all objects and needed privileges exist. Bind - If not all objects or needed privileges exist at bind time, the wizard displays an error messages, and does not bind the package. Run - If not all objects or privileges exist at bind time, the process issues warning messages, but the bind succeeds. DB2 checks existence and authorization again at run time for SQL statements that failed those checks during bind. The checks use the authorization ID of the package owner.
4	**Explain**	Obtains information about how SQL statements in the package are to execute, and then inserts that information into the table owner.PLAN_TABLE, where owner is the authorization ID of the owner of the plan or package. This option does not obtain information for statements that access remote objects.
	Optimization Hint	Query optimization hints are used for static SQL.
	Reoptvars	Re-determines the access path at run time.
	Encoding	Lets you select type of language for the package.
	Defer Prepare	Optionally, prepares dynamic SQL statements that refer to remote objects. Lets you choose from DEFER(PREPARE) and NODEFER(PREPARE).
	Degree	1 or ANY.
	Page Writes	NO, PH1, or YES.
	Flag	Lets you select flag for messages to display: all (default), completion, error or warning, and informational.

5. Use the arrow buttons to move system connections between the **Disabled Environments** and **Enabled Environments** lists.

3 When ready, click **Finish**.

Bind To Temporary Database

This action builds and submits a **sp_tempdb bind** system procedure call. It lets you bind a login to a temporary database.

Platform Availability

- SYB ASE

To Bind a Login to a Temporary Database

1 Initiate a **Bind To Temporary Database** action against one or more logins. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Bind To Temporary Database** wizard:

Step	Settings and tasks
Action Options	Select the target temporary database from the **Bind To Temporary Database** dropdown.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Build

This action lets you build a procedure, specifying a **Build utility function** (ALTER_REBUILD, REBUILD, REBIND, or DESTROY), a **Build owner**, **Precomplie options**, **Compile options**, **Prelink options**, **Link options**, and **Bind options**.

Platform Availability

- DB2 z/OS

To Build a Procedure

1 Initiate a **Build** action against a procedure. For more information see Initiating an object operation.

2 Type or select the build options in the **Procedure Build Options** dialog:

3 Click **Build**.

Build Query

Right-clicking a table and selecting **Build Query** opens a tool that lets you create, structure, and manipulate queries. For details, see Query Builder.

Cancel Thread (DBArtisan)

This action issues a -CANCEL THREAD command against an active or inactive thread, cancelling processing for those threads.

Platform Availability

- DB2 z/OS

To Cancel Processing for a Thread

1 In the Navigator/Explorer, expand the **Performance** node and then select the **Active Threads** or **Inactive Threads** node.

2 Right click an active thread or inactive thread and select **Cancel Thread** from the context menu.

3 When prompted to verify, click **Yes**.

Change Access Status

This action builds and submits a basic ALTER TABLE... READ ONLY/READ WRITE statement. This lets you toggle the table between READ ONLY and READ/WRITE modes.

Note: In DBArtisan, the Datasource Explorer listing for an Oracle table has a **Read-Only** field that displays the current mode for a table.

Platform Availability

- ORCL 11g

To Change Between READ ONLY and READ WRITE Modes for a Table

1 Initiate a **Change Access Status** action against one or more tables. For more information, see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks
Action options	Select the **Read-only** check box to generate a ALTER TABLE... READ ONLY statement, putting the table into read-only mode. Deselect the **Read-only** check box to generate a ALTER TABLE... READ WRITE statement, putting the table into read-write mode.
Dependencies	Lets you review the objects potentially impacted by this action. For more information, see Dependencies.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Change Category

You can change the category of a target Stored Outline.

Platform Availability

- ORCL

To Change the Category of a Stored Outline

1 Initiate a **Change Category** action against a stored outline. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Change Category** wizard:

Step	Settings and tasks
Action options	From the Category dropdown, choose a new, target category.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation. For more information, see Dependencies.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Related Information

- Drop By Category
- Reassign By Category

Change Job's Owner (DBArtisan)

This action builds and submits an **sp_update_job** call with an **@owner_login_name** argument. It lets you change the owner of a SQL Agent job.

Platform Availability

- SQL SVR

To Change the Owner of a Job

1 Initiate a Change job's owner action against a job. For more information see Initiating an object operation.

DBArtisan opens the Change job's owner dialog.

2 Use the following table as a guide to understanding and modifying settings in the Change job's **owner** wizard:

Step	Settings and tasks
Action Options	From the **New Owner** dropdown, select the new owner for the job.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Change Owner

This action lets you generate and submit an **ALTER***objecttype objectname***OWNER TO***newobjectname* statement, for one or more selected objects of the following types:

Domains	Functions	Schema	Tables
Tablespaces	Types	Views	

This lets you specify a new owner for the object.

Platform Availability

- PSTGRS *

To change the owner of an object

1 Initiate a **Change Owner** action against one or more objects of an object type shown in the table above. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Change Category** wizard:

Step	Settings and tasks
Action options	From the **New Owner** dropdown, choose a new owner for the object.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Related Information

- Change Schema

Change Password

The **Change Password** dialog lets you change user or login passwords, which you should do on a regular basis to guard against security leaks.

Platform Availability

- MySQL, ORCL, SQL SVR, SYB ASE

To Change the Password for a User or Login

1 Take one of the following actions:

- In DBArtisan, for Sybase ASE or SQL Server, select the Logins node.
- In DBArtisan, for MySQL and Oracle, select the Users node.

2 Right-click the user or login, and then select **Change Password** from the context menu.

The **Change Password** dialog opens.

3 For Oracle and SQL Server datasources, in the **Old Password** box, type the old password. Providing the old password is optional on SQL Server datasources.

4 In the **New Password** box, type the new password.

5 In the **Confirm Password** box, type the new password.

6 Click **Execute**. For information on the scheduling option, see Scheduling.

Change Schema

This action lets you generate and submit an **ALTER***objecttype objectname***SET SCHEMA***newobjectname* statement, for one or more selected objects of the following types:

Domains Functions Tables Types Views

This lets you specify a new schema for the object.

Platform Availability

- PSTGRS *

To change the schema for an object

1 Initiate a **Change Schema** action against one or more objects of an object type shown in the table above. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Change Category** wizard:

Step	Settings and tasks
Action options	From the **New Owner** dropdown, choose a new owner for the object.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Related Information

- Change Owner

Change Status

You can change the status of a number of relevant object types. For details, see the following topics:

- Change Status (Oracle, SQL Server, and Sybase ASE triggers)
- Change Status (SQL Server DDL triggers)
- Change Status (Full-text Indexes)
- Change Status (check constraints)
- Change Status (tablespaces)

Change Status (Oracle, SQL Server, and Sybase ASE triggers)

The **Change Status** object action lets you build an issue an ALTER TABLE statement that specifies a DISABLE (DISABLE TRIGGER) or ENABLE (ENABLE TRIGGER) option. This lets you enable or disable any triggers.

Loading a database from a previous dump causes any triggers defined in the database to fire. To speed the time required to load a database you should disable triggers.

Note: Disabling triggers can lead to problems with maintaining referential integrity and business rules

To enable or disable a trigger:

1 Initiate a **Change Status** action against one or more triggers. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Change Status** wizard:

Step	Settings and tasks
Action Options	Select the **Enabled** check box to enable the trigger. Deselect it to disable the trigger.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Change Status (SQL Server DDL triggers)

In DBArtisan, the **Change Status** object action lets you build and issue a DISABLE TRIGGER... ON DATABASE/ON ALL SERVER or ENABLE TRIGGER... ON DATABASE/ON ALL SERVER statement. This lets you enable or disable one or more database triggers or server triggers.

Note: The DBArtisan Datasource Explorer listing for a server trigger or database trigger indicates its current status.

To enable or disable a trigger:

1 Initiate a **Change Status** action against one or more triggers. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Change Status** wizard:

Step	Settings and tasks
Action Options	Select the **Enabled** check box to enable the trigger. Deselect it to disable the trigger.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Change Status (Full-text Indexes)

The action builds and submits an ALTER FULLTEXT INDEX statement with an ENABLE or DISABLE argument, letting you enable or disable the full-text index. Full-text searching is unavailable on tables or views with a disabled full-text index.

Platform Availability

- SQL SVR

To Enable or Disable a Full-text Index

1 Initiate a **Change Status** action against one or more full-text indexes. For more information, see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks
Action Options	Select the **Enabled** check box to enable the full-text index. Deselect the **Enabled** check box to disable the full-text index. Note that on opening the dialog, the **Enabled** check box state reflects whether the index is currently enabled or disabled.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Change Status (check constraints)

You can change the enabled/disabled status of check constraints, foreign key constraints, primary key constraints, and unique key constraints.

For detailed instructions, see:

- Setting constraint status for Microsoft SQL Server objects
- Setting Constraint Status for Oracle Objects

Setting constraint status for Microsoft SQL Server objects

The **Set Constraint Status** dialog lets you specify the ability of a group of constraints to be replicated, and (for Microsoft SQL Server version 7 or later) enable or disable check constraints, foreign key constraints, primary key constraints, and unique key constraints.

Completing the Set Constraint(s) Status Dialog Box

To complete this dialog, do the following:

1 Initiate a **Change Status** action against one or more check constraints. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying the settings on the **Change Status** wizard:

Step	Settings and tasks	
Action Options	**Enabled**	Deselect to temporarily override listed check constraints. Useful when you need to execute special processes that would ordinarily incur constraint-related errors.
	Not for Replication	When you duplicate the table schema and data of a source database containing constraints marked "Not for Replication", these objects are not carried over to the duplicate of the schema.
Dependencies	View the dependencies on the tablespace. For more information, see Dependencies.	
Preview	Preview the DDL generated from your choices. For more information, see Preview.	

3 Use the **Execute** or **Schedule** button to complete the operation.

Setting Constraint Status for Oracle Objects

The **Set Constraint(s) Status** dialog lets you change the status of check constraints, foreign key constraints, primary key constraints, and unique key constraints. You can enable or disable selected constraints and, in the case of primary key and unique key constraints, lets you enable with or without validation and disable with or without the changes cascading.

When enabled, the rule defined by the constraint is enforced on the data values in the columns on which the constraint is placed. When disabled, the constraint rule is not enforced but the constraint continues to be stored in the data dictionary.

Temporarily disabling constraints can improve performance when you are loading large amounts of data or when you are making massive changes to a table. Disabling constraints also can be useful if you are importing or exporting one table at a time.

Note: Primary keys for index-organized tables cannot be disabled.

Note: You cannot drop a unique or primary key constraint that is part of a referential integrity constraint without also dropping the foreign key. To drop the referenced key and the foreign key together, select the **Cascade** check box in the **Set Constraint(s) Status** dialog

The table below describes the options and functionality on the **Set Constraint(s) Status** dialog.

Note: The options differ by object.

Option	Description
Enable	Enabling the constraint and not selecting the Validate check box automatically uses Oracle ENABLE NOVALIDATE clause which enables a constraint so that it does not validate the existing data. A table using constraints in enable novalidate mode can contain invalid data but you cannot add new invalid data to that table.The enable novalidate mode is useful as an intermediate state or when you do not want the constraint to check for possible exceptions (e.g., after a data warehouse load).
Validate	Enabling the constraint and selecting the Validate check box causes Oracle to validate all existing data in the key columns of the table with the constraint. If an exception exists, Oracle returns an error and the constraint remains disabled.
Cascade	Selecting the Cascade check box when disabling a primary key or foreign key constraint instructs Oracle to simultaneously disable any constraints that depend on the primary or unique key. Selecting the Delete Cascade check box instructs Oracle to delete data in the child table (on which the foreign key is defined) if the referenced data is the parent table is deleted.

Completing the Set Constraint(s) Status Dialog Box

To complete this dialog box, do the following:

1 Initiate a **Change Status** action against one or more check constraints. For more information see Initiating an object operation.

2 Use the table above to select dialog box options.

3 Use the **Execute** or **Schedule** button to complete the operation.

Change Status (tablespaces)

You can change the online, offline, or read only status of a tablespace to control access to its segments. In addition, when setting a tablespace offline, you can choose between NORMAL, TEMPORARY, or IMMEDIATE modes of taking the tablespace offline.

Platform Availability

- ORCL

To Change the Status of a Tablespace

1 Initiate a **Change Status** action against one or more tablespaces. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Change Status** wizard:

Step	Settings and tasks
Action options	From the **Change Status** dropdown, select ONLINE, OFFLINE, or READ ONLY. If you select OFFLINE, from the **OfflineMode** dropdown, select NORMAL, TEMPORARY, or IMMEDIATE.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation. For more information, see Dependencies.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

Check Data (DBArtisan)

The Check Data Wizard checks tablespaces for violations of referential and check constraints. The wizard also checks for consistency between a base tablespace and a LOB tablespace.

Upon completion, the wizard reports information about violations that are detected and the status of tablespaces that violate any constraints is set to CHECK-PENDING. Also, the CHECK-PENDING status of all tablespaces that do not violate any constraints is reset. Optionally, the utility deletes rows that violate constraints.

Platform Availability

- DB2 z/OS

Important Notes

- Run the Check Data Wizard after a conditional restart or a point-in-time recovery on all tablespaces where parent and dependent tables might not be synchronized or where base tables and auxiliary tables might not be synchronized.
- Run the Check Data Wizard against a base tablespace only, not a LOB table space.

To Check Tablespace Data

1 Initiate a **Check Data** action against a tablespace. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Check Data Utility** wizard:

Step		Settings and tasks
1	**Select tablespaces:**	If the Wizard is opened from the Tablespaces Editor, the target tablespaces are initially displayed. To check additional tablespaces, click the **Add** button.
	Add	Opens theObject Selector Dialog Box.
	Delete	Deletes selected tablespace.
2	**Do you want the utility to be restartable?**	Lets you specify to execute Wizard in restartable mode. You can restart it from the Datasource Navigator/Explorer.
	Would you like to specify the scope?	Lets you specify scope options: Pending (default) - DBArtisan keeps track of the data rows that must be checked to ensure referential integrity. Auxonly - Select if you only want to check tables with LOB columns. All - Select whenever the scope information is in doubt. The scope information is recorded in the DB2 catalog. The scope information can become in doubt whenever you start the target table space with ACCESS(FORCE), or the catalog is recovered to a point in time. RefOnly - Select if you want to check all dependent tables in the specified tablespaces except tables with LOB columns.

	Would you like to specify the auxiliary error?	Lets you specify the auxiliary error options: Report (default) - DBArtisan sets the base tablespace to the auxiliary CHECK-pending (ACHKP) status. If CHECK DATA encounters only invalid LOB columns and no other LOB column errors, the base table space is set to the auxiliary warning status. Invalidate - DBArtisan sets the base table LOB column to an invalid status, and sets the base table space to the auxiliary warning (AUXW) status. You can use SQL to update a LOB column in the AUXW status, however, any other attempt to access the column will result in a -904 SQL return code.
	Exceptions	Lets you specify the number or exceptions, from 0 to 2147483647.
	Error ddn	Lets you specify the name of a error data definition (DD) statement that corresponds to a data control block that contains the same name. Lets you select a previously saved dataset template or click **Edit** to open the Dataset Template Editor.
	Work ddn	Lets you specify the name of a work data definition (DD) statement that corresponds to a data control block that contains the same name. Lets you select a previously saved dataset template or click **Edit** to open the Dataset Template Editor.
3	**Exceptions**	Displays the Source and Destination tables for each exception. You can select one or more rows from the grid.
	Add	Opens the **Exceptions** dialog. For details, see Exceptions Dialog Box.
	Delete	Deletes the selected exception.
4	**Do you want to specify the device type for temporary data sets by DFSORT?**	Lets you specify the device type for temporary data sets to be dynamically allocated by DFSORT.
	Do you want to specify the sort number?	Lets you type the number of datasets to be dynamically allocated by the sort program. The range of the text field is 0 to 2147483647.

3 When ready, click **Finish**.

Exceptions Dialog Box

The table below describes the options and functionality on the **Exceptions** dialog:

Option	Description
Base Table Owner	Lets you select the base table owner.
Base Table	Lets you select the base table.
Exception Table Owner	Lets you select the exception table owner.
Exception Table	Lets you select the exception table.

Check Disk Group (DBArtisan)

This action builds and submits an ALTER DISKGROUP CHECK... statement with FILE/DISK/ FAILGROUP/ALL and REPAIR options, letting you verify the internal consistency of disk group metadata.

Note: Before this object action is available you must have a direct connection to the ASM instance. For details, see Manage ASM Connection (DBArtisan).

Platform Availability

- ORCL 10g

To Verify Internal Consistency of Metadata for a Mounted Disk Group

1 On a connected Oracle datasource, expand the **Instance** and **ASM** nodes.

2 Right-click the **Disk Groups** node and select **Create** from the context menu.

3 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks
Action options	**Repair** - Select this check box to include a REPAIR clause, instructing Oracle to attempt to repair any errors found during the check. Deselect this check box to include a NOREPAIR clause.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

4 Click **Execute**. For information on the scheduling option, see Scheduling.

Check Index (DBArtisan)

The Check Index Wizard lets you test whether indexes are consistent with the data they index. The Wizard issues warning messages when an inconsistency is found.

Run the **Check Index** action after a conditional restart or a point-in-time recovery on all tablespaces where you suspect that the indexes may not be consistent with the data. Also run the wizard before executing the **Check Dat**a action to ensure that the indexes used by the Check Data Wizard are valid (see Check Data (DBArtisan)). This is especially important before using the Check Data with DELETE YES.

When checking an auxiliary table index, the **Check Index** Wizard verifies that each LOB is represented by an index entry, and that an index entry exists for every LOB.

Platform Availability

- DB2 z/OS

To Check Index/Data Consistency

1 Initiate a **Check Index** action against a tablespace or index. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Check Index Utility** wizard:

Step		Settings and tasks
1	**Select Index**	If you initiated the Check Index operation against one or more indexes, the target indexes are initially displayed. To check additional Indexes, click the **Add** button.
	Add	Opens the Indexes **Selector** dialog.
	Delete	Deletes selected index.
2	**Select tablespaces**	If you initiated the Check Index operation against one or more tablespaces, the target tablespaces are initially displayed. To check additional tablespaces, click the **Add** button.
	Add	Opens the Object Selector Dialog Box.
	Delete	Deletes selected tablespace.
3	**Do you want the utility to be restartable?**	Lets you specify to execute Wizard in restartable mode. You can restart it from the Datasource Navigator/Explorer.
	Work DDN	Lets you specify the name of a work data definition (DD) statement that corresponds to a data control block that contains the same name. Lets you select a previously saved dataset template or click **Edit** to open the Dataset Template Editor.

Do you want to specify the device type for temporary data sets by DFSORT?	Lets you specify the device type for temporary data sets to be dynamically allocated by DFSORT.
Do you want to specify the sort number?	Lets you type the number of datasets to be dynamically allocated by the sort program. The range of the text field is 0 to 2147483647.

3 When ready, click **Finish**.

Check Tables

This action lets you issue a CHECK TABLE statement, checking one or more tables for errors. This operation returns a standard CHECK TABLE result set. Each returned message consists of **Table** (name), **Op** (with a value of CHECK), **Msg_type** (STATUS, ERROR, INFO, WARNING), and **Msg_text** components.

Platform Availability

- MySQL

To Check a Table for Errors

1 Initiate a **Check Tables** action against one or more tables. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in this wizard:

Step	Settings and tasks	
Action Options	**Changed**	Only check tables that have changed since the last check.
	Extended	Perform a full key lookup for all keys in each row.
	Fast	Check only tables that have not been properly closed.
	Medium	Scan rows to verify that deleted links are valid. Calculate a key checksum for the rows and verify using a calculated checksum for keys.
	Quick	Perform a full key lookup for all keys in each row.
Preview	Displays the DDL that will execute the object action. For more information, see Preview.	

3 Click **Execute**. For information on the scheduling option, see Scheduling.

The results set opens on the **Results** tab of an ISQL editor window. For more information, see Using the Results Editor.

Checkpoint

Running a Checkpoint lets you force all dirty pages for the current database to be written to disk. A dirty page is any data or log page which, after being read into the buffer cache, is modified but not yet written to disk. The Checkpoint statement saves time in a subsequent recovery by creating a point at which all modifications to data and log pages are guaranteed to have been written to disk. If the current database is in log-truncate mode, CHECKPOINT also truncates the inactive portion of the log.

Platform Availability

- SQL SVR, SYB ASE

Important Notes

The default permission for executing a checkpoint is the db_owner fixed database role.

To Run a Checkpoint Against One or More Databases

1 Initiate a **Checkpoint** action against one or more databases. For more information see Initiating an object operation.

2 Use the following table as a guide to working through the panels of the **Checkpoint** dialog

Step	Description
Action options	Displays the names of the database(s) you chose.
Preview	Displays the DDL generated to execute the Checkpoint operation. For more information, see Preview.

3 Use one of the **Schedule** or **Execute** buttons to execute the Checkpoint.

Checksum Tables

This action lets you issue a CHECKSUM TABLE statement, returning a live checksum.

Platform Availability

- MySQL

To Check a Table for Errors:

1 Initiate a **Checksum Tables** action against one or more tables. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in this wizard:

Step	Settings and tasks	
Action Options	**Extended**	Specifies that the whole table is to be read, row by row.
	Quick	Specifies that the live checksum is to be reported. For information on enabling a live checksum for a table, see Tables wizard (MySQL).
Preview	Displays the DDL that will execute the object action. For more information, see Preview.	

3 Click **Execute**. For information on the scheduling option, see Scheduling.

The results set opens on the **Results** tab of an ISQL editor window. For more information, see Using the Results Editor.

Coalesce

You can maximize the size of free space chunks in tablespaces to avoid the situation in which an object cannot acquire enough contiguous free space to accommodate its next extent size. Towards this goal, look for opportunities to coalesce adjacent blocks of free space into a single, larger block.

Oracle automatically coalesces adjacent free space chunks with a background process. However, it still supports the commands for coalescing free space manually. Depending on the size of the tablespace, coalescing its free space can take a long time. So determine when to perform this operation. It you coalesce immediately, the tablespace is locked.

Platform Availability

- ORCL

Important Notes

- You cannot coalesce on an UNDO tablespace.

To Coalesce a Tablespace

1 Initiate a **Coalesce** action against one or more tablespaces. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Coalesce** wizard:

Step	Settings and tasks
Action options	Verify display of the tablespaces to be coalesced.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation. For more information, see Dependencies.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

Compile

You can recompile specific objects by issuing the proper ALTER statement. The explicit recompilation of invalid objects eliminates the need for implicit run-time recompilation which, in turn, can cause run-time compilation errors and performance overhead. Recompile objects after you make changes to that object or dependent objects.

The following table lists the object types by DBMS for which the Compile operation is available:

DB2 LUW						Procedures			
Oracle	Functions	Java Sources	Materialized Views	Packages	Package Bodies	Procedures	Schema	Types and Type Bodies	Views

Note: The objects you can compile and the steps in compiling them differ according to the type of data source you are working against. Before proceeding, make sure you are familiar with the material in Compiling Oracle Objects.

To Compile an Object

1 Initiate a **Compile** action against one or more supported objects (see the table above). For more information see Initiating an object operation.

The **Compile** wizard opens. Options offered on the **Action Options** tab differ by object type. For example, no options are provided when compiling Java Sources, while a full range of dependent object-based and debug options are provided when compiling procedures.

2 Use the following table as a guide to working through the panels of the dialog box

Step		Settings and tasks	Availability
Action options	**Compile dependents**	If enabled, this option compiles statements for all objects referenced by the object being compiled. For example, if you compile a function that references a specific procedure and you select to compile the dependent objects, an ALTER COMPILE statement is created for that referenced procedure. If disabled, only the current object is compiled and the object's dependencies are ignored. This is the default setting.	DB2 LUW Procedures Oracle Functions, Packages, Procedures, Types, and Views
	Compile only invalid dependents	If you enabled the **Compile dependents** option, enabling this option compiles only invalid dependent objects - Creates ALTER COMPILE statements for only those objects that are currently invalid.	DB2 LUW Procedures Oracle Functions, Packages, Procedures, Types, and Views
	Compile system dependents	If you enabled the **Compile dependents** option, enabling this option compiles dependent system objects - Compiles all of the referenced objects with the debug option.	DB2 LUW Procedures Oracle Functions, Packages, Procedures, Types, and Views

	Compile with debug	Enabling this option instructs the Oracle PL/SQL compiler to generate and store the code for use in debugging sessions.	DB2 LUW Procedures Oracle Functions, Packages, Package Bodies, Procedures, Types, Type Bodies, and Views
	Keep specific name	When enabled, compilation will keep the current name.	DB2 LUW Procedures
Dependencies	Lets you review any dependencies before you proceed. For more information, see Dependencies.		
Preview	Displays the DDL generated to execute the Checkpoint operation. For more information, see Preview.		

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Compiling Oracle Objects

To recompile an object it must belong to your schema or you need ALTER ANY privileges on that object. You must also have appropriate privileges for any associated objects. Prior to compiling objects of a particular type, see the relevant topic below:

- Notes on Compiling Oracle Functions
- Notes on Compiling Oracle Java Sources
- Notes on Compiling Oracle Materialized Views
- Notes on Compiling Oracle Packages and Package Bodies
- Notes on Compiling Oracle Procedures
- Notes on Compiling Oracle Types and Type Bodies
- Notes on Compiling Oracle Views

Notes on Compiling Oracle Functions

You can recompile a function. Oracle first recompiles any invalid objects on which the function depends. In addition, it marks any objects that depend on the function as invalid.

To recompile a function that is part of a package, compile the package itself. An ALTER FUNCTION statement to compile a stand-alone function. However, you should not use the ALTER FUNCTION statement to individually recompile a function that is part of a package.

For more information, see **Compile**.

Notes on Compiling Oracle Java Sources

Oracle lets you compile a Java source. Oracle resolves references to other Java sources.

For more information, see **Compile**.

Notes on Compiling Oracle Materialized Views

You can compile materialized views. If a materialized view fails to revalidate after you recompile, that materialized view cannot be fast refreshed ON DEMAND or used for query rewrite.

For more information, see **Compile**.

Notes on Compiling Oracle Packages and Package Bodies

You can recompile a package and recompile all package objects together.

Recompiling a package compiles both the package specification and the package body by issuing two consecutive ALTER statements. However, only the ALTER statement is shown for the specification in the **Preview: Confirm Compile** dialog. You can recompile only the package body by explicitly compiling the package body itself.

When recompiling the entire package, Oracle recompiles the package even if it is invalid. However, if there are compilation errors, the package remains invalid and Oracle invalidates all dependent objects.

Recompiling only a package body does not invalidate objects that depend upon the package specification regardless of whether or not the package body has compilation errors.

For more information, see **Compile**.

Notes on Compiling Oracle Procedures

You can compile a procedure that is part of a package, by compiling the package itself. The ALTER PROCEDURE statement is used to compile a stand-alone procedure. However, you should not use the ALTER PROCEDURE statement to individually recompile a procedure that is part of a package.

For more information, see **Compile**.

Notes on Compiling Oracle Types and Type Bodies

You can recompile a type. Both the type specification and the type body are recompiled.

For more information, see **Compile**.

Notes on Compiling Oracle Views

You can recompile a view when you have altered an object referenced by that view.

When you recompile a view, Oracle invalidates all dependent objects.

For more information, see **Compile**.

Convert Tables

This action lets you build and submit an ALTER TABLE statement, specifying an ENGINE= option, specifying any available storage engine.

Note: For information on how to quickly convert a table to use the INNODB storage engine, see Rebuild Table.

Platform Availability

- MySQL

To Change the Storage Engine for a Table

1 Initiate a **Convert Tables** action against one or more tables. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in this wizard:

Step	Settings and tasks
Action Options	Use the **Convert To** dropdown to select an available storage engine.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Copy Object Names

The Copy Object Names functionality lets you copy and then paste object name(s) into other applications.

Platform Availability

- ORCL, SQL SVR

To Copy an Object Name to be Pasted into Another Application

1 On the DBArtisan Datasource Navigator/Explorer, expand nodes until the target object type is visible and then click the target object type.

2 Right-click on the specific database object or objects and select **Copy object name(s)**.

Copy Schema

This action lets you quickly make copies of a database schema and its corresponding objects.

Platform Availability

- DB2 LUW

To Copy a Schema

1 In the data sources pane, right-click a data source and select **Copy Schema** from the context menu.

2 Use the following table as a guide to understanding and choosing options the options in the Copying Schema dialog:

Pane	Options
Action Options	Lets you specify **Copying Schema**, **Target Schema**, specify a **Copy Mode** of COPY, COPYNO, or DDL, specify a **Log filename** (COPY and COPYNO modes only), select an **Object Owner**, and **Error table schema**, provide an **Error table name**, and specify a **Drop error table after action execution** options.
Tablespace mapping	Lets you specify source and target tablespace options.
Preview	Displays the DDL generated to execute the operation. For more information, see Preview.

3 **Execute** or **Schedule** the action.

Copy to Copy (DBArtisan)

Platform Availability

- DB2 z/OS

The **Copy to Copy** dialog lets you make image copies. Starting with either the local primary or recovery site primary copy, the **Copy to Copy** dialog can make up to three copies of the following types:

- Local primary
- Local backup
- Recovery site primary
- Recovery site backup

The copies are used by the RECOVER utility when recovering a table space or index space to the most recent time or to a previous time. These copies can also be used by MERGECOPY, UNLOAD, and possibly a subsequent Copy to Copy execution.

To Perform a Copy To Copy Operation

1 Initiate a **Copy To Copy** action against a tablespace or index. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Copy To Copy Utility** wizard:

Step		Settings and tasks
1	**Do you want the utility to be restartable?**	Lets you specify to execute Wizard in restartable mode. You can restart it from the Datasource Navigator/Explorer.
	Tape Stacking	Stacks similar output data sets as successive files on one logical tape volume, where a logical tape volume can consist of a multi-volume aggregate, if any of the resulting data sets on the tapes span volumes. Within one utility execution, output data sets are stacked on a logical tape volume of the same usage type. For example, local primary image copies are stacked separately from local backup image copies. Also lets you specify the Copy and Recovery dataset information once as variable names appended with generated values.
2-4	**Object Grid**	Displays the target tablespaces, indexspaces, or indexes. Your can select multiple items in the list. The Partition column is editable. You can add the partition numbers in this column, separated by commas.
	Add	For details, see Object Selector Dialog Box.
	Delete	Deletes the selected objects.
	Set Properties	For details, see Object Properties Dialog Box.

3 When ready, click **Finish**.

Object Properties Dialog Box

The **Object Properties** dialog opens when you click the **Set Properties** button against a selected object when performing a Copy To Copy operation.

The table below describes the options and functionality on this dialog box:

Option	Description
From	**Copy -** The particular image copy data set (dsn) as the input to COPYTOCOPY job. Specify the name in the **Dataset Name** field. **Last Copy -** The most recent image copy that was taken for the tablespace or index space to be the input to COPYTOCOPY job. This could be a full or incremental copy. **Last Full Copy -** The most recent full image copy that was taken for the object to be the input to COPYTOCOPY job. **Last In CR Copy -** The most recent incremental image copy that was taken for the object to be the input to COPYTOCOPY job.
Dataset Name	Only available if you selected a **From** Value of **Copy**.This field lets you specify a dataset name.
Would you like to specify the volume information	Only available if you selected a **From** Value of **Copy**. If you select Yes, you can specify a **Volume** and view the **Sequence Number**.
Local Site Primary Copy	Lets you specify a dataset name for the local site primary image copy. This relates to the copydsn1 parameter in the stored procedure.
Local Site Backup Copy	Lets you specify the device for the image copy. This relates to the copydevt1 parameter in the stored procedure.
Recovery Site Primary Copy	Lets you specify the primary space allocation amount for the image copy. This relates to the copyspace1 parameter in the stored procedure.
Recovery Site Backup Copy	Lets you specify a dataset name for the local site backup image copy. This relates to the copydsn2 parameter in the stored procedure.

Object Selector Dialog Box

The table below describes the options and functionality on the panel of the **Object Selector** dialog

Option	Description
Database Like	Lets you type the pattern for querying the tablespaces. You can type multiple words, separated by spaces.
Tablespace Like	Lets you type the pattern for querying the tablespaces. You can type multiple words, separated by spaces.
Creator Like	Lets you type the pattern for querying the creator. You can type multiple words, separated by spaces.
Index Like	Lets you type the pattern for querying the index. You can type multiple words, separated by spaces.
Match Case	Lets you specify match case.
Object Grid	Lets you select one or more rows in this list.
Query	Repopulates grid based on the object query.
Select All	Selects all rows in Object Grid.
Unselect All	Unselects all rows in Object Grid.

Create Alias

The **Create Alias** object action lets you create an alias to a selected object. The following table shows the object types for which this action is available:

	Materialized Query Tables	Sequences	Tables	Views
DB2 LUW	✓	✓	✓	✓
DB2 z/OS			✓	✓

To create an alias to one of the supported object types:

1 Initiate a **Create Alias** action against a supported object (see the table above). For more information see Initiating an object operation.

The Alias Wizard opens. For information on using the Alias editor, see the topic corresponding to the DBMS you are working against:

- Aliases Wizard (DB2 LUW)
- Aliases Wizard (DB2 Z/OS)

Create Clone

This action provides support for ADD CLONE option functionality, letting you create a clone on an existing base table. This creates a table with a cloned definition but with no data.

Note: The base table must reside on a universal tablespace. For more information, see Tablespaces Wizard (DB2 Z/OS).

Platform Availability

- DB2 z/OS version 9 ^

To Create a Clone for One or More Tables:

1 Initiate a **Create Clone** action against one or more tables. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Create Clone** wizard:

Step	Settings and tasks
Action Options	Lets you provide a **Clone Name**.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

The clone for a table is not displayed in the Datasource Navigator/Explorer listing of tables for a data source. To determine whether a table has a clone, see the **Dependencies** tab/panel on a Table editor or on the wizard for object actions such as Drop or Rename.

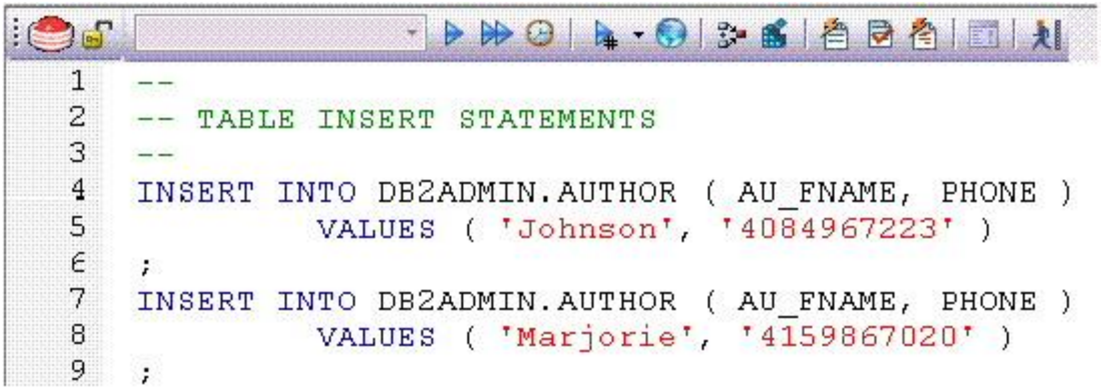
After creating a clone for a table, you can exchange data between the base table and the clone.

Related Information

- Exchange Data With Clone
- Drop Clone

Create Insert Statements

The Create Insert Statements dialog box lets you automatically create a set of Insert Statements for selected columns and records and based on the data in a table.



```
--
-- TABLE INSERT STATEMENTS
--
INSERT INTO DB2ADMIN.AUTHOR ( AU_FNAME, PHONE )
          VALUES ( 'Johnson', '4084967223' )
;
INSERT INTO DB2ADMIN.AUTHOR ( AU_FNAME, PHONE )
          VALUES ( 'Marjorie', '4159867020' )
;
```

Platform Availability

- DB2 LUW, DB2 z/OS, MySQL, ORCL, SQL SVR, SYB ASE

To Create Insert Statements Based on Existing Table Data

1 In the left pane of the application, select the Tables node.

2 In the right pane of the application, right-click a table, and then select **Create Insert Statements**.

The **Create Insert Statements** dialog opens.

3 In Columns, select columns to be included in the INSERT statement.

4 Optionally, in the Where box, type a WHERE clause.

5 Enable or disable **Include owner information in insert statements**.

6 Enable or disable **Set row count**, which lets you specify a row count, the number of rows in a table that were affected by the Insert statement executed against the table, or a view based on the table.

7 Click **OK**.

An ISQL window opens with a set of Insert statements corresponding to the criteria you provided and the table data. For more information, see Using the SQL Editor.

Create Like

The Create Like Editor lets you create a new object based on an existing object. The following table shows availability of the **Create Like** function for particular object types by DBMS:

DB2 LUW	DB2 Z/OS	ITB/FBD	MySQL	SQL Server	Oracle	Sybase ASE
Materialized Query Tables Tables	Tables	Tables	Tables	Logins Tables Users	Tables	Logins Tables Users

To Create an Object Bbased on Another Object

1 Initiate a **Create Like** action against a table. For more information see Initiating an object operation.

A Create Like wizard opens.

2 Provide a name for the new object.

3 Modify settings on each panel of the wizard as required.

For particular object types, tasks and settings on the panels of the Create Like wizard are similar to those for the object editors for that object type. For information, see:

- Materialized Query Tables Editor (IBM DB2 LUW)
- Tables Editor (IBM DB2 LUW)
- Tables Editor (IBM DB2 Z/OS)
- Tables editor (MySQL)
- Logins Editor (SQL Server)
- Tables Editor (SQL Server)
- Users Editor (SQL Server)
- Tables Editor (Oracle)
- Logins Editor (Sybase ASE)
- Tables Editor (Sybase ASE)
- Users Editor (Sybase ASE)

4 Click **Execute** to create the new object.

Create or Edit Java Source

The Java Editor lets you enter Java code. The table below describes the Java Editor toolbar options:

Option	Description
Lock/Unlock Connection	Click to lock or unlock connection.
Create	Click to open the **Create Options** dialog, which lets you select the options for creating the java source.
Errors	Click to split the workspace in half, displaying the error messages in the lower half of the workspace.

When you add a script in a new file, you not only choose a name for the file, but choose among create options. Finally, when you complete your script, you can Preview, Schedule or immediately Execute it.

Platform Availability

- ORCL

Create Synonym

You can perform an ad hoc Create Synonym operation against a specific object displayed in the Datasource Navigator/Explorer. For background information, see Synonyms.

Platform Availability

- ORCL

You can perform this operation against the following object types:

Functions	Materialized Views	Packages	Package Bodies
Procedures	Sequences	Tables	Views

To Create a Synonym for an Object

1 Expand the node corresponding to one of the object types supported for this operation, listed in the table above.

The right-hand pane is populated with all objects of that type.

2 Right-click the target object and select **Create Synonym** from the context menu.

3 A **Create Synonym** wizard opens.

For information on using the wizard, see Synonyms Wizard (Oracle).

Create View

Right-clicking one or more tables in the Navigator/Explorer and selecting **Create View** opens a wizard that lets you create a new view. For more information, see the following topics:

- Views Wizard (DB2 LUW)
- Views Wizard (DB2 Z/OS)
- Views Wizard (SQL Server)
- Views Wizard (Oracle)
- Views Wizard (Sybase ASE)

DBCC

The DBCC (Database Consistency Check) function box lets you:

- Specify single or multiple tables or indexes for validation.
- Perform database-wide validations.
- Perform object-level validations of databases.

Platform Availability

- SQL SVR, SYB ASE

To Perform a Database Consistency Check

1 Initiate a **DBCC** action against a database, table, or index, as follows:

- If you are checking consistency of a Sybase ASE or SQL Server database or a SQL Server table, right-click the object and select **DBCC** from the context menu, as appropriate.
- If you are checking consistency of a Sybase ASE table, right-click the table and select either **DBCC > Check Allocation**, **DBCC > Check Table**, **DBCC > Check Text**, or **DBCC > Rebuild Index** from the context menu, as appropriate.
- If you are checking consistency of a Sybase ASE index, right-click the index and select either **DBCC > Check Allocation**, or **DBCC > Check Index** from the context menu, as appropriate.
- If you are checking consistency of a SQL Server index, right-click the table and select either **DBCC > Check Fragmentation**, **DBCC > Update Usage**, **DBCC > Check Index**, or **DBCC > Show Statistics** from the context menu, as appropriate.

For more information see Initiating an object operation.

A dialog that that lets you specify DBCC options opens.

2 See the following topics as a guide to understanding and setting options on the dialog:

- DBCC for Microsoft SQL Server Databases
- DBCC for Microsoft SQL Server Tables
- DBCC for Microsoft SQL Server Indexes
- DBCC for Sybase ASE Databases
- DBCC for Sybase ASE Tables
- DBCC for Sybase ASE Indexes

3 Click **Execute**. For information on the scheduling option, see Scheduling.

DBCC for Microsoft SQL Server

The **DBCC (Database Consistency Check**) dialog lets you specify single or multiple tables or indexes for validation in Microsoft SQL Server. Use this dialog box to perform table-level or index-level validations of databases which are too large to undergo database-level DBCC operations in a time-efficient manner.

The **DBCC** dialog includes the following elements:

- A window displaying the target database objects
- A drop-down list of DBCC Operations
- Buttons for previewing the operation's SQL code, scheduling the operation, and executing the operation

For more information, see:

DBCC for Microsoft SQL Server Databases

DBCC for Microsoft SQL Server Tables

DBCC for Microsoft SQL Server Indexes

DBCC for Microsoft SQL Server Databases

The **DBCC** dialog for databases lets you perform database-wide validations. You should validate your databases as part of regular database maintenance to guard against corruption and failure. Microsoft SQL Server offers a set of DBCC commands to validate the integrity of your databases. Generally, you should perform these DBCC commands prior to dumping your databases to ensure that you are capturing clean backups of your databases.

The fundamental difference between the **DBCC** dialog for databases, tables and indexes is the content of the DBCC Operation drop-down list.

The table below describes the options and functionality on the **DBCC** dialog.

DBCC Operation	Description
Check Allocation	Executes a DBCC CHECKALLOC command. Makes sure that all data and index panels are correctly allocated and used. It reports on the amount of space allocated and used in the database. When checking allocation, you have the option to skip non-clustered indexes by selecting the Skip non-clustered indexes check box.
Check Catalog	Executes a DBCC CHECKCATALOG command. Checks for consistency in and between system tables.
Check Database	Executes a DBCC CHECKDB command. Verifies that all tables and indexes are properly linked, that indexes are in proper sorted order, that all pointers are consistent, that the data on each panel is reasonable, and that panel offsets are reasonable. When checking a database, you have the option to skip non-clustered indexes by selecting the Skip non-clustered indexes check box.
Check FileGroup	Executes a DBCC CHECKFILEGROUP command. Verifies that all tables and indexes for the specified filegroup are properly linked, that indexes are in proper sorted order, that all pointers are consistent, that the data on each panel is reasonable, and that panel offsets are reasonable. When checking filegroups, you have the option to skip non-clustered indexes by selecting the Skip non-clustered indexes check box.
Show Oldest Transaction	Executes a DBCC OPENTRAN command. Displays information on the oldest active transaction and the oldest distributed and non distributed replicated transactions, if any, within the specified database.
Update Usage	Executes a DBCC UPDATEUSAGE command. Reports and corrects the rows, used, reserved, and dpanels columns of the sysindexes table for any clustered indexes on objects of the type U (user-defined table) or S (system table).

For more information, see **DBCC.**

DBCC for Microsoft SQL Server Tables

The **DBCC** dialog for tables lets you perform table-level validations of databases. The fundamental difference between the **DBCC** dialog for tables and indexes is the content of the DBCC Operation drop-down list.

The table below describes the options and functionality on the **DBCC** dialog.

Option	Description
Check Current Identity Value	Checks the current identity value for the target objects, correcting values if needed depending on parameter specifications. Identity columns created with a NOT FOR REPLICATION clause in either the CREATE TABLE or ALTER TABLE statement are not corrected by this operation.
Check Fragmentation	Displays the target table's data and index fragmentation information, determining whether the table is heavily fragmented. When a table is heavily fragmented, you can reduce fragmentation and improve read performance by dropping and recreating a clustered index (without using the SORTED_DATA option). Doing so reorganizes the data, resulting in full data pages. To adjust the level of fullness, use the Rebuild Index operation’s FILLFACTOR option. When INSERT, UPDATE, and DELETE statements fragment tables, they usually do so with unequal distribution across the entire database so that each page varies in fullness over time, forcing additional page reads for queries that scan part or all of a table.
Check Table	Checks the linkages and sizes of text, ntext and image pages for selected tables. For the data, index, text, ntext, and image pages of the target tables, this operation also checks that index and data pages are correctly linked, indexes are in their proper sorted order, pointers are consistent, the data on each page is reasonable, and the page offsets are reasonable. DBCC CHECKTABLE requires a shared lock on all tables and indexes in the database for the duration of the operation. However, DBCC CHECKTABLE does not check the allocations of pages in the specified table (for this, use DBCC CHECKALLOC). To perform DBCC CHECKTABLE on every table in the database, use DBCC CHECKDB.
Check Text/Image Allocation	Checks the allocation of text, ntext, or image columns for a table. In later versions of Microsoft SQL, use DBCC CHECKTABLE to check the integrity of the data, index, text, ntext, and image pages for the target table.
Pin Table	Pins target tables in memory so that they are not flushed when Microsoft SQL Server needs space to read in new pages. DBCC PINTABLE is best used for keeping small, frequently referenced tables in memory. Pinning a large table can consume a large portion of the buffer cache, leaving inadequate memory to service other tables in the system. A pinned table that is larger than the buffer cache itself can fill the entire cache, necessitating a shut down of the system by a sysadmin user, who must then restart Microsoft SQL Server and unpin the table. Pinning too many small tables can result in a similar problem.
Rebuild Index	Dynamically rebuilds one, multiple, or all indexes for a table in the target database, allowing indexes which enforce either primary key or unique constraints to be rebuilt without need for dropping and recreating. This operation is not supported for use on system tables.
Unpin Table	Marks target tables as unpinned, rendering their pages flushable from the buffer cache if space is needed to read in a new page from disk.

Update Usage	Reports and corrects inaccuracies in the sysindexes table (which can result in incorrect space usage reports by the sp_spaceused system stored procedure) and corrects the rows, used, reserved, and dpages columns of the sysindexes table for tables and clustered indexes. If there are no inaccuracies in sysindexes, DBCC UPDATEUSAGE returns no data. Use this operation to synchronize space-usage counters. Executing this operation on large tables or databases can require some time, so it should typically be used only when you suspect incorrect values returned by sp_spaceused.

For more information, see **DBCC**.

DBCC for Microsoft SQL Server Indexes

The following Database Console Commands are supported, letting you perform index-level validations of databases.

Option	Description
Check Fragmentation	Lets you build and submit a DBCC SHOWCONTIG command, to display fragmentation information for the selected index, letting you determine whether the table is heavily fragmented. When a table is heavily fragmented, you can reduce fragmentation and improve read performance by dropping and recreating a clustered index. For details, see Rebuild Index (SQL Server).
Update Usage	Lets you build and submit a DBCC UPDATEUSAGE command, reporting on and correcting pages and row count inaccuracies that can result in incorrect space usage reports. The **Count Rows** control lets you add a WITH COUNT_ROWS argument, dictating that the row count column be corrected as necessary. The **No Informational Messages** control lets you add a WITH NO_INFOMSGS argument, suppressing all informational messages.
Check Index	Lets you build and submit a DBCC CHECKTABLE command, verifying the integrity of pages and structures of the table or indexed view. Disabling the **Show Informational Messages** control adds a WITH NO_INFOMSGS argument, suppressing informational messages.
Show Statistics	Lets you build and submit a DBCC SHOW_STATISTICS command, displaying current query optimization statistics for the owning table or indexed view.

For more information, see **DBCC**.

DBCC for Sybase ASE

The **DBCC** (Database Consistency Check) dialog lets you specify single or multiple databases, tables or indexes for validation in Sybase ASE. Use this dialog box to perform table-level or index-level validations of databases which are too large to undergo database-level DBCC operations in a time-efficient manner.

The **DBCC** dialog includes the following elements:

- A window displaying the target database objects
- A drop-down list of DBCC Operations

- Buttons for previewing the operation's SQL code, scheduling the operation, and executing the operation

For more information, see:

DBCC for Sybase ASE Databases

DBCC for Sybase ASE Tables

DBCC for Sybase ASE Indexes

DBCC for Sybase ASE Databases

The **DBCC** dialog for databases lets you perform database-wide validations. The fundamental difference between the **DBCC** dialog for databases, tables and indexes is the content of the DBCC Operation drop-down list.

The table below describes the options and functionality on the **DBCC** dialog.

Option	Description
Check Allocation	Checks the allocation and use of all pages in the target database.
Check Catalog	Checks for consistency in and between system tables in the target database.
Check Database	Checks the allocation and structural integrity of all the objects in the target database.
Check Storage	Checks the target database for allocation, OAM page entries, page consistency, text valued columns, allocation of text valued columns, and text column chains. The results of this operation are stored in the dbccdb database.
Database Repair	Drops a damaged database.

For more information, see **DBCC**.

DBCC for Sybase ASE Tables

The **DBCC** dialog for tables lets you perform table-level validations of databases. The fundamental difference between the **DBCC** dialog for tables and indexes is the content of the DBCC Operation drop-down list.

The table below describes the options and functionality on the **DBCC** dialog.

Option	Description
Check Allocation	Checks the database to see that every page is correctly allocated, and that no allocated page is unused. Use TABLEALLOC frequently (daily) to check page linkages in the Adaptive Server before performing a database dump to ensure the integrity of the dumped data.
Check Table	Checks the linkages and sizes of text, ntext and image pages for selected tables. For the data, index, text, ntext, and image pages of the target tables, this operation also checks that index and data pages are correctly linked, indexes are in their proper sorted order, pointers are consistent, the data on each page is reasonable, and the page offsets are reasonable. DBCC CHECKTABLE requires a shared lock on all tables and indexes in the database for the duration of the operation. However, DBCC CHECKTABLE does not check the allocations of pages in the specified table (for this, use DBCC CHECKALLOC). To perform DBCC CHECKTABLE on every table in the database, use DBCC CHECKDB.
Check Text	Upgrades text values after you have changed an Adaptive Server's character set to a multibyte character set.
Rebuild Index	Dynamically rebuilds one, multiple, or all indexes for a table in the target database, allowing indexes which enforce either primary key or unique constraints to be rebuilt without need for dropping and recreating. This operation is not supported for use on system tables.

DBCC Operation Options

Additional options are offered for selected operations which you can specify to further customize a database consistency check. The table below describes each option:

Option	Description
Error Option	Click **Fix Error** to have any allocation errors fixed. You must put your database in single-user mode to fix errors, so specify this option during times of low usage.
Job Scope	Select Optimize to produce a report based on the allocation pages listed in the object allocation map (OAM) pages for the table. It does not report and cannot fix unreferenced extents on allocation pages that are not listed in the OAM pages. The optimized option is the default. Select Full to perform the equivalent of a table-level CHECKALLOC, reporting all types of allocation errors. Select Fast to produce an exception report of pages that are referenced but not allocated in the extent. Fast does not produce an allocation report.
Update Index Option	Click this check box to skip non-clustered indexes when updating index options.

For more information, see **DBCC**.

DBCC for Sybase ASE Indexes

The **DBCC** dialog for indexes lets you perform index-level validations of databases. Unlike the **DBCC** dialog for tables, this **DBCC** dialog offers only one option on the DBCC Operation drop-down list: Check Allocation. This option checks the specified database to see that all pages are correctly allocated and that no allocated page is unused.

The table below describes the options and functionality on the **DBCC** dialog.

Option	Description
DBCC Option	Checks the specified database to see that all pages are correctly allocated and that no page that is allocated is not used.
Error Option	Any allocation errors detected are fixed. You must put your database in single-user mode to fix errors, so specify this option during times of low usage
Job Scope	Produces a report based on the allocation pages listed in the object allocation map (OAM) pages for the table. It does not report and cannot fix unreferenced extents on allocation pages that are not listed in the OAM pages. The optimized option is the default. A full job is the equivalent to a table-level CHECKALLOC, reporting all types of allocation errors. A fast job does not produce an allocation report, but produces an exception report of pages that are referenced but not allocated in the extent.

For more information, see **DBCC**.

Deallocate Unused Space

The **Deallocate Unused Space** dialog lets you deallocate space from clusters, indexes, and tables. You can also deallocate unused space from table partitions and subpartitions. When you find that allocated space is not being used, you can free that space for use by other objects by explicitly deallocating space. Oracle releases the freed space to the user quota for the tablespace in which the deallocation occurs.

Oracle deallocates unused space from the end of the object toward the high water mark. In other words, Oracle frees space starting with the space that would have been used last. If an extent is completely contained in the space to be deallocated, then the whole extent is freed. If an extent is only partially contained in the space to be deallocated, then Oracle shrinks that extent to the size of the used space up to the high water mark, and frees the unused space in the extent.

If you are deallocating unused space from and index and the index is range-partitioned or hash-partitioned, Oracle deallocates unused space from each partition in the index. If an index is a local index on a composite-partitioned table, Oracle deallocates unused space from each of the subpartitions in the index.

Tip: You can verify that the deallocated space is freed by going to the **Space** tab in the appropriate editor.

Platform Availability

- ORCL

To Deallocate Unused Space from an Index or Table

1 Initiate a **Deallocate Unused Space** action against one or more Clusters, Indexes, Primary Keys, Tables, or Unique Keys. For more information see Initiating an object operation.

The **Deallocate Unused Space** dialog opens.

2 Use the following table as a guide to understanding and specifying options on this dialog:

Option	Description
Specify the number of bytes above the high-water mark that the objects will have after deallocation. If no value is specified, all unused space will be freed.	If you do not specify an amount of unused space and the high water mark is above the size of INITIAL and MINEXTENTS, then all of the unused space is freed. If the high water mark is less than the size of INITIAL or MINEXTENTS, then all unused space above MINEXTENTS is freed. If you specify an amount of unused space and the remaining number of extents is smaller than MINEXTENTS, then the MINEXTENTS value changes to reflect the new number. If the initial extent becomes smaller as a result of the deallocation, the INITIAL value changes to reflect the new size of the initial extent.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Delete Statistics

Lets you create and issue a DELETE STATISTICS statement, allowing you to delete statistics for a table or for specified columns of a table.

Platform Availability

- SYB ASE

To Delete Statistics for a Table

1 Initiate a **Delete Statistics** action against one or more tables. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Delete Statistics** wizard:

Step	Settings and tasks
Action options	Displays the names of the tables you selected and if appropriate, lets you select a specific **Partition Name**.
Columns (Sybase ASE 15 only)	**All columns** - lets you delete statistics for all columns in the table. **Specific columns** - select this option and then select check boxes associated with each specific column for which statistics are to be deleted.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation. For more information, see Dependencies.
Preview	Preview the DDL generated for the operationn. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Describe

RIght-clicking a table, view, procedure or function in the Navigator/Explorer and selecting **Describe** opens a window that lets you view columnar information (for tables and views) or input parameter information (for procedures and functions).

Detach Database

The **Detach Database** object action removes the database from the server but leaves the database intact within the data and transaction log files that compose the database. These data and transaction log files can then be used to attach the database to any instance of Microsoft SQL Server, including the server from which the database was detached. This makes the database available in exactly the same state it was in when it was detached.

Platform Availability

- SQL SVR

To Detach Data and Transaction Files for a Database

1 Initiate a **Detach Database** action against a database. For more information see Initiating an object operation.

The **Detach Database(s)** dialog opens.

2 To skip the UPDATE STATISTICS operation when detaching the database, select the **Skip Checks** check box for the target database(s).

Tip: This option is useful for databases that are to be moved to read-only media.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Related Information

- Attach Database

Disable Index

The **Disable Index(es)** action lets you mark an index as disabled or inactive and unavailable for use.

Platform Availability

- SQL SVR

Note: On SQL Server this action can also be applied to primary keys and unique keys.

To Disable an Index

1 Initiate a **Disable Index(es)** action against one or more indexes. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the wizard:

Step	Settings and tasks
Action options	Displays the name of the index, primary key, or unique key being disabled.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation. For more information, see Dependencies.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Disable Job

You can enable or disable any job queue. Job Queues are built-in mechanisms that let you schedule a variety of SQL-based or command-line driven tasks.

Platform Availability

- ORCL

To Disable a Job Queue

1 Initiate a **Disable Job** action against a job queue. For more information see Initiating an object operation.

Tthe **Disable** dialog opens.

2 Click **Execute**. For information on the scheduling option, see Scheduling.

Related Information

Enable Job (Job Queue)

Run Job (job queues).

Disable Keys

This action lets you build and submit an ALTER TABLE statement that specifies the DISABLE KEYS option. This action tells MySQL stop updating nonunique indexes.

Platform Availability

- MySQL

To Stop Updates to Nonunique Indexes

1 Initiate a **Disable Keys** action against one or more tables. For more information see Initiating an object operation.

The **Disable Keys** dialog opens.

2 Use the following table as a guide to understanding and modifying settings in this wizard:

Step	Settings and tasks
Action Options	Lets you review the tables you selected.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Note: You can subsequently recreate missing indexes. For details, see Enable Keys.

Disable/Enable Triggers

These actions let you build and submit ALTER TABLE statements that specify ENABLE TRIGGER or DISABLE TRIGGER options.

Platform Availability

- SQL SVR, SYB ASE

To Enable or Disable Triggers for a Table

1 On the Database Navigator/Explorer, select the **Tables** node.

2 On the Database Navigator/Explorer, right-click one or more selected tables and select **Enable Triggers** or **Disable Triggers**.

The **Enable Triggers** or **Disable Triggers** dialog opens, as appropriate.

3 Click **Execute**.

Disconnect ASM Connection (DBArtisan)

This action disconnects a direct connection to an ASM instance and reverts to the default datasource login.

Note: For information on connecting directly to an ASM instance, see Manage ASM Connection (DBArtisan).

Platform Availability

- ORCL 10g

To Disconnect from an ASM Instance

1 On a connected Oracle datasource, expand the **Instance** node.

2 Right-click the **ASM** node and select **Disconnect ASM Connection** from the context menu.

Drop

The **Confirm Drop** dialog lets you drop one or more database objects and remove their definition from the system catalog.

Platform Availability

- DB2 LUW, DB2 z/OS, MySQL, ORCL, SQL SVR, SYB ASE, SYB IQ

To Drop an Object

1 Initiate a **Drop** action against one or more objects. For more information see Initiating an object operation.

The **Confirm Drop** dialog opens on the **Action Options** panel.

2 On the **Action Options** panel, select any options specific to the object type or configuration of the object you are dropping.

3 Navigate to the **Dependencies** pane to view database objects that are dependent on the object you are dropping.

Note: For Microsoft SQL Server, when dropping Logins, you also have the option to delete corresponding user objects.

Note: For Microsoft SQL Server, when dropping Databases, you also have the option to delete the backup and restore history for the database.

4 Navigate to the **Preview** pane to verify that the correct DDL was created, and if necessary navigate backward and modify your choices.

5 Click **Execute** to drop the object.

Drop Automatic Storage Path(s)

This action lets you remove tablespace storage paths on a datasource and subsequently rebalance any affected tablespaces. It builds and submits:

- An ALTER DATABASE DROP STORAGE ON... statement specifying the storage paths to be removed
- Optionally, one ALTER TABLESPACE... REBALANCE statement for each tablesapce impacted by the ALTER DATABASE DROP STORAGE ON... statement

Platform Availability

- DB2 LUW 9.7^

Prerequisites or Setup Tasks

This object action is only available for datasources configured for autostorage and that have at least two autostorage paths configured.

To drop storage paths on a datasource

1 Initiate a **Drop Automatic Storage Path(s)** action against a datasource. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in this wizard:

Step	Settings and tasks
Path(s) Selection	**Path:** Select the check box associated with each storage path that is to be dropped. This object action will not allow you to select all storage paths on a datasource. At least one path must be left active. **Rebalance the affected tablespaces:** Selecting this check box generates an ALTER TABLESPACE... REBALANCE statement for each tablesapce affected by the storage paths you selected to drop.
Preview	Displays the DDL that will execute the object action. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

For related information, see Rebalance.

Drop By Category

This action lets you build and submit one or more calls to the **outln_pkg.drop_by_cat** subprogram, letting you drop outlines belonging to specified categories.

Platform Availability

- ORCL

To Drop One or More Stored Outlines of a Particular Category

1 Initiate a **Drop By Category** action against one or more stored outlines. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in this **Drop By Category** wizard:

Step	Settings and tasks
Action Options	Select the categories to drop.
Preview	Displays the DDL that will execute the object action. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Related Information

- Change Category
- Reassign By Category

Drop Clone

If you have created a clone from a base table, you can drop that clone. To determine whether a table has a clone, see the **Dependencies** tab/panel on a Table editor or on the wizard for object actions such as Drop or Rename.

Platform Availability

- DB2 z/OS version 9 ^

To Drop the Clones of One or More Tables

1 Initiate a **Drop Clone** action against one or more tables. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Confirm Drop Clone** wizard:

Step	Settings and tasks
Action Options	Lets you review the action you initiated.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Related Information

- Create Clone
- Exchange Data With Clone

Drop Java

This action lets you make use of the `dropjava` tool, deleting schema objects corresponding to Java files.

Platform Availability

- ORCL

To Drop Schema Objects Assocaited with a Java File

1 Initiate a **Drop Java** action against one or more Java Classes. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings on the **Drop Jave Objects** dialog:

Setting	Description
Who Owns the Java Object	Select the owner of objects to be dropped.
File to be dropped	Use these controls to build a list of Java files for which associated objects are to be dropped.
Drop Synonym	Drops any synonyms defined for objects being dropped.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Related Information

- Load Java

Drop Login Trigger

The **Drop Login Trigger** action lets you build and submit an **sp_modifylogin** or ALTER LOGIN call that specifies a NULL login script option. This removes the configured login trigger for the login, the default database stored procedure that executes each time the user successfully logs in.

Note: If a login has a login trigger configured, the specified stored procedure name is visible in the Login editor. For details, see Logins Editor (Sybase ASE).

Platform Availability

- SYB ASE

To Remove the Currently Configured Login Trigger for a Login

1 On the DBArtisan Datasource Navigator/Explorer, expand the data source nodes until the **Logins** node is visible and then select the **Logins** node.

2 Right-click a login and select **Login Trigger Actions > Drop Login Trigger** from the context menu. The **Drop Login Trigger** dialog opens.

3 Use the following table as a guide to understanding and modifying settings in the wizard:

Step	Settings and tasks
Action options	Displays the login you selected to run the action against.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation. For details, see Dependencies.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

Related Information

- Add/Modify Login Trigger

Drop Materialized Query Table

The **Drop Materialized Query** action lets you convert a materialized query table to a regular table.

Platform Availability

- DB2 LUW

To Convert a Materialized Query Table to a Regular Table

1 Initiate a **Drop Materialized Query Table** action against one or more materialized query tables. For more information see Initiating an object operation.

The **Confirm MQT Conversion to Regular** dialog opens.

2 Click **Execute**. For information on the scheduling option, see Scheduling.

Drop Unused

This action lets you build and submit one or more calls to the **outln_pkg.drop_unused** subprogram, letting you drop outlines that have not been used since being created.

Platform Availability

- ORCL

To Drop One or More Unused Stored Outlines

1 Initiate a **Drop Unused** action against one or more stored outlines. For more information see Initiating an object operation.

The **Drop Unused** dialog opens.

2 Use the following table as a guide to understanding and modifying settings in this wizard:

Step	Settings and tasks
Preview	Displays the DDL that will execute the object action. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

This action lets you build and submit an ALTER TABLE... DROP UNUSED COLUMNS statement, letting you remove columns currently marked as unused.

Platform Availability

- ORCL

To Drop the Clones of one or more Tables

1 Initiate a **Drop Unused Columns** action against one or more tables. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Drop Unused Columns** wizard:

Step	Settings and tasks
Action Options	Displays the tables you selected
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Edit Data

Right-clicking a table and selecting Edit Data lets you edit table data. For details, see Data Editor.

Enable/Disable (Oracle Jobs)

This action builds and submits a DBMS_SCHEDULER.DISABLE or DBMS_SCHEDULER.ENABLE call, letting you enable or disable a job or program.

Platform Availability

- ORCL 10g ^

To enable or disable a job or program

1 In DBArtisan, on a connected Oracle datasource, expand the **Instance** and **Scheduler** nodes and select the **Jobs** or **Programs** node.

2 Right-click a job or program and select **Enable/Disable** from the context menu.

3 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks
Action options	Select the **Enabled** check box to enable the job or program. Leave the **Enabled** check box deselected to disable the job or program.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

4 Click **Execute**. For information on the scheduling option, see Scheduling.

Enable/Disable Filetable(s)

This action lets you build and submit an ALTER TABLE statement, specifying either an ENABLE FILETABLE_NAMESPACE or DISABLE FILETABLE_NAMESPACE argument. This enables or disables system-defined constraints on a FileTable.

Platform Availability

- SQL SVR

To enable or disable system-defined constraints on a FileTable

1 Initiate an **Enable/Disable Filetables** action against one or more tables. For more information see Initiating an object operation.

The **Enable/Disable Filetable(s)** dialog box opens.

2 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks	
Action options	**Enabled**	If selected, an ALTER TABLE with an ENABLE FILETABLE_NAMESPACE argument is generated. If unselected, an ALTER TABLE with a DISABLE FILETABLE_NAMESPACE argument is generated.
Dependencies	Lets you review the objects potentially impacted by this action. For more information, see Dependencies.	
Preview	Preview the DDL generated for the operation and when ready, use the **Schedule** or **Execute** button to perform this action.	

3 Finally, use the **Execute** button to create the object.

For information on setup of filegroups, FileTable, and Filestream, see the following topics:

- Databases Wizard (SQL Server)
- Tables Wizard (SQL Server)

Enable/Disable Job (DBArtisan - SQL Agent Job)

This action builds and submits an **sp_update_job** call with an **@enabled** argument. It lets you change the enabled/disabled status of a SQL Agent job.

Platform Availability

- SQL SVR

To Change the Enabled/Disabled Status of a Job

1 Initiate an **Enable/Disable Job** action against one or more jobs. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Enable/Disable jobs** wizard:

Step	Settings and tasks
Action Options	Select the **Enable** check box to enable the selected jobs. Deselect the **Enable** check box to disable the selected jobs.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Enable/Disable Stored Outline Auto-creation/Auto-use

There are two object actions that can be performed against one or more stored outlines, that dictate whether execution plans are automatically generated or used:

Enable/disable stored outline auto-creation	Lets you specify whether Oracle to create a stored outline for every query submitted to the server.
Enable/disable stored outline auto-use	Lets you specify whether Oracle uses stored outlines to generate execution plans.

Platform Availability

- ORCL

To Specify Whether Stored Outlines are Automatically Created and Used

1 Initiate an **Enable/disable stored outline auto-use** or**Enable/disable stored outline auto-creation** action against one or more stored outlines. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the wizard:

Step	Settings and tasks	
Action Options	**Automatically create stored outlines** or **Automatically use stored outlines**	Lets you enable or disable the feature associated with your selection.
	Category	Lets you select an outline category. This option is only available if you enabled the feature.
	No Override	If selected, outlines will not override. This option is only available if you enabled the feature.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation.	
Preview	Preview the DDL generated for the operation. For more information, see Preview.	

3 When ready, click **Execute**.

Enable Job (Job Queue)

This action lets you enable or disable any job queue. Job Queues are built-in mechanisms that let you schedule a variety of SQL-based or command-line driven tasks.

Platform Availability

- ORCL

To Enable a Job Queue

1 Initiate an **Enable Job** action against a job queue. For more information see Initiating an object operation.

The **Enable** dialog opens.

2 Click **Execute**. For information on the scheduling option, see Scheduling.

Related Information

- Disable Job
- Run Job (job queues)

Enable Keys

You can issue an ALTER TABLE statement that specifies the ENABLE KEYS option. This action tells MySQL to recreate missing indexes.

Platform Availability

- MySQL

To Recreate Missing Indexes

1 Initiate an **Enable Keys** action against one or more tables. For more information see Initiating an object operation.

The **Enable Keys** dialog opens.

2 Use the following table as a guide to understanding and modifying settings in this wizard:

Step	Settings and tasks
Action Options	Lets you review the tables you selected.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Note: You can also instruct MySQL to stop updating nonunique indexes. For details, see Disable Keys.

Enable Recycle Bin

This action builds and submits an ALTER SYSTEM SET "_recyclebin"=TRUE or ALTER SYSTEM SET "_recyclebin"=FALSE statement, letting you enable or disable the Oracle Recycle Bin.

This action lets you enable or disable the Oracle Recycle Bin, building and submitting one of the following version-specific statement variations:

- **Oracle 10g R1** - ALTER SYSTEM SET "_recyclebin" TRUE / FALSE
- **Oracle 10g R2** - ALTER SYSTEM SET RECYCLEBIN ON / OFF SCOPE=BOTH
- **Oracle 11g** - ALTER SYSTEM SET RECYCLEBIN ON / OFF DEFERRED SCOPE=BOTH (note that in this case, the change takes effect after reconnecting to the datasource)

When the Recycle Bin is enabled, dropped objects are stored in the Recycle Bin until explicitly purged.

To Enable or Disable the Recycle Bin

1 In DBArtisan, right-click an Oracle datasource node, and select **Command > Enable Recycle Bin** from the context menu.

2 Use the following table as a guide to understanding and modifying settings in this wizard:

Panel	Settings and tasks
Action Options	Select the **Enable** check box to enable Recycle Bin functionality. Deselect the **Enable** check box to disable the feature.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

Note: In general, you can determine the current enabled/disabled state of the Recycle Bin by opening the **Enable Recycle Bin** dialog. If the **Enable** check box is selected, the Recycle Bin is currently Note however, that the **Enable Recycle Bin** action reads the current state of the Recycle Bin by querying the v$parameter object. On some Oracle 10g R1 servers, the "_recyclebin" parameter does not appear listed in v$parameter – it is hidden. As a result, for this specific case, the value of the check box when opening the **Enable Recycle Bin** dialog will always be TRUE, regardless of the actual value.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Related Information

- Recycle Bin (for a support summary including topics such as displaying Recycle Bin contents)
- Flashback Recycle Bin Entry
- Purge Recycle Bin Entry
- Purge Recycle Bin

Error

The Error message displays a warning. After reading the warning, click **Close** to continue.

Estimate Size

The **Estimate Size** dialog for tables and indexes lets you estimate how large a table or index will become given a row growth projection. The results let you proactively plan your space-related object needs.

Platform Availability

- ORCL, SYB ASE

To Estimate Size for a Table or Index

1 Initiate an **Estimate Size** action against one or more tables or indexes. For more information see Initiating an object operation.

The **Estimate Size** dialog opens.

2 Use the following topics as a guide to estimating size and using the other options offered on this dialog:

- Estimating Oracle index or table sizes
- Estimating Sybase ASE index or table sizes

Estimating Oracle index or table sizes

The **Estimate Size** dialog remains open until you manually close it. This lets you run several estimations, varying the method and various metrics. Each estimation run is a three-step process:

1 Select a size estimation method:

- **Use current row data from current statistics** - The table or index must have been previously analyzed for statistics to be obtained. Note that the **Analyze Index(es)** and **Analyze Table(s)** buttons can be used to initiate this action. For more details, see Analyze.
- **Use table/index definition source** - The maximum row length for the index or table will be calculated. Note that this usually produces much higher space estimates.
- **Manually enter estimation metrics** - Manually enter in data for space estimates if the objects have no data or if the index definition source method is not desired.

2 Depending on the method you chose, provide additional metrics:

- **Percent Free**
- **Row Size**
- **Number of rows**

3 Click the **Estimate Size** button and view the estimate in the **Estimated Index Size** box.

Other options include:

- Use the **Report** button to generate an HTML report of results

- Use the **Save As** button to save a result grid file
- Use the **Owner** and **Name** controls to select another table or index for size estimates.
- Use the **Add Index** or **Add Table** button to add another table or index to your size estimates. Similarly, use the **Remove Index** or **Remove Table** button to remove a table or index from the dialog.

Estimating Sybase ASE index or table sizes

The **Estimate Size** dialog remains open until you manually close it. This lets you run several estimations, varying the available metrics. Each estimation run is a two-step process:

1 Provide the metrics for your estimate:

- **Fill Factor** (indexes only) - Specify a percentage of how full each index page can become.
- **Number of rows** - Specify the number of rows for your size estimate.

Note: Optionally you can use the **Update Statistics** button to open a dialog that lets you generate new statistics for the index or table. For details, see Update Statistics.

2 Click the **Estimate Size** button and view the estimate in the **Estimated Index Size** box.

Other options include:

- Use the **Report** button to generate an HTML report of results
- Use the **Save As** button to save a result grid file
- Use the **Database**, **Owner**, **Table Name**, and **Index Name** controls to select another table or index for size estimates.
- Use the **Add Index** or **Add Table** button to add another table or index to your size estimates. Similarly, use the **Remove Index** or **Remove Table** button to remove a table or index from the dialog.

Exchange Data With Clone

This action lets you issue an EXCHANGE statement with the DATA BETWEEN TABLE *table1* AND *table2* syntax, letting you swap data between a base table and its clone. To determine whether a table has a clone, see the **Dependencies** tab/panel on a Table editor or on the wizard for object actions such as Drop or Rename.

Platform Availability

- DB2 z/OS version 9 ^

To Exchange Data Between One or More Tables and their Clones

1 Initiate a **Clone Actions > Exchange Data With Clone** action against one or more tables. For more information see Initiating an object operation.

The **Exchange Data With Clone** dialog opens.

2 Use the following table as a guide to understanding and modifying settings in this wizard:

Step	Settings and tasks
Action Options	Lets you review the action you initiated.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Related Information

- Create Clone
- Drop Clone

Execute

The **Execution** dialog lets you execute extended procedures, functions, and procedures.

Platform and object type availability

	DB2 LUW	DB2 z/OS	ITB/FBD	MySQL	Oracle	SQL Server	Sybase ASE
Extended procedures							✓
Functions					✓		
Procedures	✓	✓	✓		✓	✓	✓

To Compile an Extended Procedure, Function, or Procedure

1 In the DBArtisan Datasource Navigator/Explorer, select a valid object node. Objects of that type are shown in the right pane.

2 Right click the target object and select **Execute** from the context menu.

Note: If the extended procedure, function, or procedure takes input parameters, you are prompted to provide values. For more information, see Working With Execution Parameters.

The Results editor opens, showing either results of a successfully executed extended procedure, function, or procedure, or showing error messages if execution was not successful. For details, see Using the Results Editor.

Working With Execution Parameters

If you initiate an **Execute** action against an extended procedure, function, or procedure that takes input parameters, a dialog prompts you to provide parameter values and otherwise manage parameters.

The **Procedure Execution** dialog lets you:

- Save input parameters as *.prm files to preserve specific input parameter configurations.
- Open *.prm files to save the effort of reentering specific input parameters.
- Reset parameters to their default setting.

The table below describes the options and functionality of the **Procedure Execution** dialog:

Option	Description
Parameter	Specify the required input parameters in this window. If input parameters are not required for the execution of the target procedure, a message displays in this window, stating that the procedure "has no input parameters. Press execute to run it."
Open	Click to open an **Open** dialog, from which you can open an existing *.prm file. The saved parameters immediately populate the dialog box upon opening.
Save	Click to save the values of your input parameters as a *.prm file. You can reopen a saved *.prm file from this dialog box at any time.
Reset	Click to reset the parameters in the Parameter window to their default values.
Default	Select to gather default information from the data dictionary. ORACLE ONLY: You can not specify non-default parameters after specifying a default parameter.
Continue	Click to execute the procedure once you have entered values for all required parameters in the Parameter window.

For more general information on the **Execute** operation, see **Execute**.

Extract

You can extract the statements required to create existing objects into an Interactive SQL window.

Note: For information on setting preferences for this action, see DDL Extract Options.

Platform Availability

- DB2 LUW, DB2 z/OS, MySQL, ORCL, SQL SVR, SYB ASE, SYB IQ

To Extract the DDL for an Object

1 Initiate an **Extract** action against one or more objects. For more information see Initiating an object operation.

The DDL Editor opens with the DDL to create the selected objects. For more information, see Using the DDL Editor.

Extract Data as XML

This function allows you to take SQL data, extract it, and make it available as XML data. The XML Editor Filter allows you to pick and choose among columns and designate rows and then creates the For XML statement that enables the operation. The resulting XML document is created and presented in an active XML Editor. At this point the document can be saved in XML format.

Caution: To use the **Extract Data as XML** feature in Sybase ASE, you must first purchase and install an XML Services license from Sybase.

Platform Availability

- ORCL 9i ^, SQL SVR, SYB ASE 12.5.1 ^

To Open the XML Editor Filter

1 Initiate an **Extract Data as XML** action against a table. For more information see Initiating an object operation.

An **XML Editor Filter** dialog opens.

2 In the **Columns** area, select the check boxes corresponding to the columns with data you want to extract.

3 Optionally, in the **Where** box, type a WHERE clause to qualify the records that will be extracted.

Note: As you select options, inspect the **Select** box to verify the query you are creating.

4 Use the table below as a guide to understanding and specifying the optional DBMS-specific settings in this dialog:

DBMS	Setting or group	Description
Sybase ASE	Refer to Sybase online documentation for details onXML mapping information. For assistance, see Accessing Third Party Documentation.	
Oracle	**Row Set Tag**	Identify the XML element name you want to use to replace the Row Set tag.
	Row Tag	Identify the XML element names you want to use to replace the Row tag.
	Max Rows.	The maximum number of rows to fetch

SQL Server	**XML Mode**	**AUTO** mode returns query results as nested XML elements. For any table in the From clause with a column in the Select clause, it is represented as an XML element. When you select **RAW** mode, each row in the query result set is transformed into an XML element with the generic row identifier. The XML attribute name will be the same as the column name for non-null columns.
	XML Options	**XML DATA** specifies that an XML data schema will be returned. **ELEMENTS** specifies that columns will be returned as subelements--otherwise they are mapped to XML attributes. This is an option only if AUTO is selected. When **BINARY BASE64** is selected, any binary data is encoded in base-64 format. You must specify this option in RAW mode. It is the default for AUTO.

5 When ready, click **OK**. An XML representation of the extracted data opens in an XML editor.

Flashback Recycle Bin Entry

This action lets you build and submit a FLASHBACK TABLE... TO BEFORE DROP statement against a table in the Recycle Bin. This recovers the table and restores it to its state immediately before the drop.

Note: The Datasource Explorer entry for a Recycle Bin entry has a **Can Undrop** field that indicates whether a **Flashback Recycle Bin Entry** action can be performed against the item

Platform Availability

- ORCL 10g^

To Recover a Table Currently in the Recycle Bin

1 In the DBArtisan Datasource Navigator/Explorer, under an Oracle **Schema** node, select the **Recycle Bin** node. Contents of the Recycle Bin are displayed.

2 Right-click one or more selected tables in the Recycle Bin and select **Flashback Recycle Bin Entry** from the context menu. The **Flashback Recycle Bin Entry** dialog opens.

3 Use the following table as a guide to understanding and modifying settings in the wizard:

Step	Settings and tasks
Action options	Displays the table entries you selected to restore.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation. For details, see Dependencies.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

4 Click **Execute**. For information on the scheduling option, see Scheduling.

Related Information

- Recycle Bin (for a support summary including topics such as displaying Recycle Bin contents)
- Enable Recycle Bin
- Purge Recycle Bin Entry
- Purge Recycle Bin

Flashback Table

This action builds and submits a FLASHBACK TABLE statement, letting you restore a table to a state at some specified time in the past.

Note: Before using this object action, consult Oracle documentation for information on restrictions, required permissions, and details on System Change Numbers and Restore Points. For more information, see Accessing Third Party Documentation.

Platform Availability

- ORCL 10g ^

Note: Available options differ by the version of Oracle that you are working against.

To Restore a Table to Some Previous State

1 Initiate a **Flashback Table** action against a table. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings on the dialog:

Panel		Settings and tasks
Action Options	**Type**	Lets you select a specific FLASHBACK TABLE clause: **BEFORE DROP** - If the table is currently in the Recycle Bin, the table will be restored to its state immediately before being dropped. **SCN** - restores the table to a time corresponding to a specified System Change Number. **TIMESTAMP** - restores the table to a time corresponding to a specified timestamp value. **RESTORE POINT** - (10g R2 ^) restores the table to a time corresponding to a specified name associated with an SCN.
	SCN	If you selected a **Type** value of SCN, type a System Change Number.
	Timestamp	If you selected a **Type** value of TIMESTAMP, use this control to provide a timestamp.
	Restore Point (10g R2 ^)	If you selected a **Type** value of RESTORE POINT, select a restore point.
	Enable Trigger	If you selected a **Type** value of SCN, TIMESTAMP, or RESTORE POINT, selecting this check box overrides the default behavior, ensuring that triggers remain enabled during the flashback process.
	Rename To	If you selected a **Type** value of BEFORE DROP, specify a new name for the table being retrieved from the Recycle Bin.
Dependencies	For details on using this tab, see Dependencies.	
Preview	Displays the DDL that will execute the object action. For details, see Preview.	

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Related Information

Flashback Recycle Bin Entry

Flush Cache

This action lets you clear all dynamic SQL in the cache for a package and forces IBM DB2 for Linux, Unix, and Windows to recompile the SQL the next time it is called.

Platform Availability

- ○ DB2 LUW

To Flush the Cache for a Package

1 Initiate a **Flush Cache** action against a package. For more information see Initiating an object operation.

The **Flush Package Cache** dialog opens.

2 Click **Execute**. For information on the scheduling option, see Scheduling.

Flush Tables

This action lets you build and submit a FLUSH statement, specifying a TABLE or TABLES option. This flushes the specified tables and removes all query results from the query cache.

Platform Availability

- MySQL

To Flush One or More Tables

1 Initiate a **Flush Tables** action against one or more stored tables. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in this wizard:

Step	Settings and tasks
Action Options	Displays the tables selected for this flush operation.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Free (Packages)

You can delete one or more packages.

Platform Availability

- DB2 z/OS

To Delete Packages

1 Initiate a **Free Packages** action against one or more packages. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Confirm Free** wizard:

Step	Settings and tasks
Action Options	Displays the names of the packages you chose to free. This panel also lets you select a **Order Drop By Selection** setting, which dictates whether drop statements are generated in the order objects are selected.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Free Plan

In DBArtisan, this action lets you drop plans associated with packages or DBRMs.

To Drop a Plan

- In DBArtisan, click the **Free** button on the **Plans** tab of the DBRMs editor. For more information, see DBRMs Editor (DBArtisan - IBM DB2 Z/OS).
- In DBArtisan, click the **Free** button on the **Plans/Packlists** tab of the Packages editor. For more information, see Packages Editor (IBM DB2 Z/OS).

For more information on dropping database objects, see Drop.

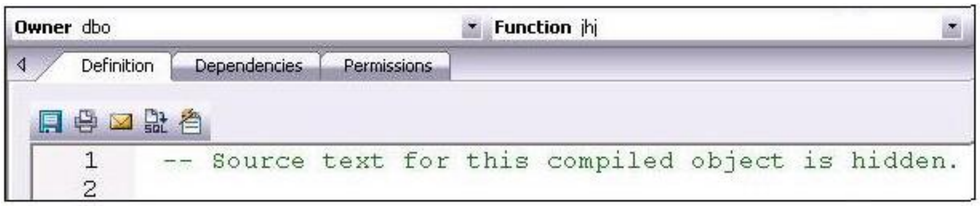
Hide Text

This action lets you build and submit **sp_hidetext** calls against one or more objects. This allows you to hide the source text for particular object types:

Caution: The **Hide Text** action is irreversible. Once hidden, source text cannot be made viewable or otherwise retrievable again.

Note: For details on the specific elements hidden, consult Sybase documentation. See Accessing Third Party Documentation.

When viewing an object after hiding text, hidden elements occupying entire tabs or individual property boxes, are annotated with a **Source text for this compiled object is hidden** message.



```
1    -- Source text for this compiled object is hidden.
2
```

Hide Text can be applied both to individual objects and to objects containing or owning other objects. Supported object types for which **Hide Text** can be applied to individual objects, and the resulting application-specific hidden elements in object editors are as follows:

Check constraints	**Check Condition** box, **Definition** tab.
Defaults	**Value** box, **Properties** tab.
Extended procedures	**Library name** box, **Properties** tab.
Functions	**Definition** tab.
Procedures	**Definition** tab.
Rules	**Restriction** box, **Properties** tab.
Triggers	**Definition** tab.
Views	**Definition** tab.

Object-containing or object-referencing object types to which **Hide Text** can be applied are as follows:

Databases	Hides the source text of all compiled objects in the selected database.
Tables	Hides the source text of all check constraints, defaults, and triggers defined on the selected table.
Users	Hides the source text of all compiled objects owned by the selected user.

Caution: Because of the potentially broad scope of hiding text for users and databases, you should exercise extreme care when applying **Hide Text** to those object types.

Platform Availability

- SYB ASE

To Permanently Hide the Source Text for One or More Objects

1 For DBArtisan, in the Datasource Navigator/Explorer, expand nodes until the node corresponding to one of the target object types is visible, and select that node. Objects of that type are shown in the right pane.

2 Right-click one or more selected objects, and then select **Hide Text** from the context menu. The **Confirm Hide Text** dialog opens.

3 Use the following table as a guide to understanding and modifying settings in the wizard:

Step	Settings and tasks
Action options	Displays the name of the objects you selected.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation. For more information, see Dependencies.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

4 Click **Execute**. For information on the scheduling option, see Scheduling.

DBArtisan Datasource Explorer views for supported objects show a **Hidden** property that indicates whether **Hide Text** has been applied to the object. **Y** indicates that source text for this compiled object has been irreversibly hidden.

Import Data From File

RIght-clicking a table in the Navigator/Explorer and selecting **Import Data From File** opens a dialog that lets you import table data from a file. For details, see Import Data.

Kill (DBArtisan)

This action issues a DBMS-specific command (KILL or ALTER SYSTEM KILL SESSION, for example) required to terminate a blocking lock, lock, or process. The following table shows the DBMS platform and object types for which the **Kill** action is available:

	DB2 LUW	DB2 z/OS	MySQL	ORCL	SQL SVR	SYB ASE
Blocking Locks				✓	✓	✓
Locks			✓	✓	✓	✓
Processes	✓		✓	✓	✓	✓

To kill a blocking lock, lock, or process

1 In the Navigator/Explorer, expand the **Performance** node and then select the **Blocking Locks**, **Locks**, or **Processes** node.

2 In the Navigator/Explorer, right click a blocking lock, lock, or process and select **Kill** from the context menu.

3 When prompted to verify, click **Yes**.

Load Java

Before you can call Java stored procedures, you must load them into the Oracle database and publish them to SQL.

Note: This functionality is available for Oracle only.

Note: For related information, see Drop Java.

The Java Load Wizard lets you:

- Select the owner of the Java object and the files to load.
- Select options for the loading of the files.
- Select Resolver options.

For more information, see Java Load Wizard - Panel 1.

Java Load Wizard - Panel 1

The table below describes the options of the first panel of the Java Load Wizard.

Option	Description
Who owns the Java Object?	Lets you select the owner of the Java object.
Select files to be loaded	Select a file, and then click **Add**.

Java Load Wizard - Panel 2

The table below describes the options of the second panel of the Java Load Wizard.

Option	Description
When do you want the Java files to be resolved?	Lets you specify when the source file is loaded as a source schema object, the source file is compiled, class schema objects are created for each class defined in the compiled .java file, and the compiled code is stored in the class schema objects.
Select the Encoding Options	Lets you specify the encoding of the .java file.
Grant Access to the following users	Lets you select one or more users.

Java Load Wizard - Panel 3

The table below describes the options of the third panel of the Java Load Wizard.

Option	Description
Other Load Options	OPTIONAL: Lets you select options.
Add Resolver Options	Lets you specify the objects to search within the schemas defined. **Add** - Click to open the **Select a Resolver Option** dialog to add a new resolver option in the list. **Edit** - Click to open the **Resolver Edit** dialog to modify a resolver option. **Remove** - Select one or more resolver option and click to delete.

Lock

The **Lock Table** dialog lets you lock tables to prevent other users from reading or updating the table data. Locking a table saves the time of locking each row of the table that needs to be updated. Lock mode options preventing other users from viewing or modifying table data, or allowing other users to view but not modify table data. Locks are released at the end of a transaction.

Platform Availability

- DB2 LUW

To Lock a Table

1 Initiate a **Lock** action against one or more tables. For more information see Initiating an object operation.

The **Lock Table** dialog opens.

2 Select a Lock Mode option:

- Share - Lets other users view but not modify the table data.
- Exclusive - Prevents other users from viewing or modifying the table data.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Lower High Water Mark

Against automatically-managed tablespaces, this action lets you build and submit an ALTER TABLESPACE... REDUCE statement, specifying that existing containers be reduced by a specified size.

Against database-managed tablespaces, you build and submit an ALTER TABLESPACE... LOWER HIGH WATER MARK statement followed by an ALTER TABLESPACE... REDUCE statement, explicitly lowering the high water mark before reducing the size of the tablespace.

Note: Before using this object action, consult DB2 LUW documentation for information on automatic and non-automatic storage, storage reclaimasbility, as well as database versus system-managed tablespaces. For more information, see Accessing Third Party Documentation.

Platform Availability

- DB2 LUW 9.7 ^

To Lower the High Water Mark for a Tablespace

1 Initiate a **Lower High Water Mark** action against a tablespace. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and choosing options in the **Lower High Water Mark** dialog:

Pane	Options
Action Options	Against automatically-managed tablespaces, this pane lets you provide the REDUCE option value for the generated ALTER TABLESPACE statement. Two options are provided. Selecting **MAX** specifies that the high water mark be lowered by moving the maximum number of extents to the beginning of the tablespace. Specifying a **Size** and **Unit** specifies the numeric value by which the tablespace is to be reduced through extent movement. Against database-managed tablespaces, this pane only allows you to specify that **All Containers** are to be decreased by a specified **Size** (in Pages, KB, MB, or GB **Unit**s). If you want to specify individual containers, use the **Containers** panel.
Containers	This tab is only available for database-managed tablespaces. As well, the functionality provided depends on whether you selected the **All Containers** option on the **Action Options** pane. With **All Containers** selected, this panel lets you specify an ON DBPRATITIONNUM clause on the ALTER TABLESPACE... REDUCE statement generated, specifying partitions for the container operations. Click the **Partitions** button, select one or more partitions on the **Select Partitions** dialog, and click **OK**. With **All Containers** deselected, this panel lets you specify an ON DBPRATITIONNUM clause and also lets you specify a database container clause on the ALTER TABLESPACE... REDUCE statement generated. For each container to be included in the databae container clause, click the Add Container button, select a container from the dropdown, and select a **Size** and **Unit** (Pages, KB, MB, or GB) for the container.
Preview	Displays the DDL generated to execute the operation. For more information, see Preview.

3 **Execute** or **Schedule** the action.

Manage ASM Connection (DBArtisan)

For working against ASM disk groups, DBArtisan lets you toggle the login between the general datasource and a specific connection to the associated ASM instance.

Note: When registering an Oracle datasource that has an ASM instance, you can use the **Connect as** control to specify the SYSASM privilege. This give you full administrative privilege for the ASM instance. For details, see Registering Datasources.

You can create or edit disk groups using the basic datasource connection. For information on object actions that require you to connect directly to the ASM instance, see the following topics:

- Check Disk Group (DBArtisan)
- Mount Disk Group (DBArtisan)
- Rebalance Disk Group (DBArtisan)
- Undrop Disk (DBArtisan)

The **Manage ASM Connection** action lets you toggle between the two connection types.

Platform Availability

- ORCL 10g ^

To Toggle the Login Between the Datasource and the Specific ASM Instance

1 On a connected Oracle datasource, expand the **Instance** and **ASM** node.

2 Right-click the **Disk Groups** or **ASM** node and select **Manage ASM Connection** from the context menu. A **Manage ASM Instance Connection** dialog opens, prompting you to choose a connection type. Take one of the following actions:

- To connect directly to the ASM instance, deselect the **Use existing connection to DB instance** check box, provide the connection details and logon credentials.
- To use the default connection to the associated datasource, select the **Use existing connection to DB instance** check box.

3 Click **OK**.

Note: The **Manage ASM Instance Connection** dialog also opens automatically on the first object action against a disk group after connecting to an Oracle datasource.

When connected to the ASM instance you can revert to the default datasource connection with a single step. For details, see Disconnect ASM Connection (DBArtisan).

Merge Copy (DBArtisan)

The Merge Copy function makes a merged copy of the data sets within a table space. You can merge incremental copies of a tablespace to make one incremental copy, or you can merge incremental copies with a full image copy to make a new full image copy.

Platform Availability

- DB2 z/OS

To Create a Merge Copy

1 On the Datasource Navigator/Explorer, expand the Tablespaces node.

DBArtisan displays the Tablespaces.

2 Right-click the target tablespace and select MergeCopy from the context menu.

DBArtisan opens the **MERGECOPY Utility** wizard.

3 Use the following table as a guide to understanding and modifying settings in the wizard:

Step	Settings and tasks	
1	**Do you want the utility to be restartable?**	Yes or No
	Tape Stacking Properties	Yes or no. When you select Yes, you can then set the dataset **Properties**.
	Do you want to merge all incremental copies with the full image copy to make a new full image copy?	Self-evident
	WorkDDN	Here you are setting the dataset information for the working data definition name.
2	**Select tablespaces**	lets you work with the list of tablespaces.

4 When ready, click **Finish**.

Migrate (DBArtisan)

Right-clicking one or more objects and selecting **Migrate** from the context menu opens the Migration Wizard, letting you migrate an object definition to another supported DBMS. For information on using the wizard, see Using the Schema Migration Wizard (DBArtisan).

Mirror (DBArtisan)

The **Mirror** object action lets you build and submit a DISK MIRROR command, creating a mirror that immediately takes over when the primary device fails.

Platform Availability

- SYB ASE

To Create a Database Device Mirror

1 Initiate a **Mirror** action against a database device. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Mirror** wizard:

Step	Settings and tasks	
Action Options	Displays the database device you selected and lets you work with the following settings:	
	Mirror Name	Lets you specify the full path name of the database mirror device that is to be the secondary device.
	Serial Writes	Lets you specify whether serial writes to the devices are enforced
	Contiguous (VMS only)	If the mirror is a file rather than a device, this option forces the file to be created contiguously.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.	
Preview	Displays the DDL that will execute the object action. For details, see Preview/.	

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Modify Recovery (DBArtisan)

The Modify Recovery Wizard lets you delete records from SYSIBM.SYSCOPY, related log records from SYSIBM.SYSLGRNX, and entries from the DBD. You can delete records based on a specific age or based on the date the record was written.

You should run the wizard regularly to clear outdated information from SYSIBM.SYSCOPY and SYSIBM.SYSLGRNX. These tables, and particularly SYSIBM.SYSLGRNX, can become very large and take up considerable amounts of space. By deleting outdated information from these tables, performance for processes that access data from these tables can be improved.

Platform Availability

- DB2 z/OS

To Run the Modify Recovery Utility

1 Initiate a **Modify Recovery** action against a tablespace. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Modify Recovery Utility** wizard:

Step		Settings and tasks
1	**Select tablespaces**	Displays the available tablespaces.
	Add	Opens the Object Selector Dialog Box.
	Delete	Deletes the selected items.
2	**Delete history record(s) by**	**Age** lets you delete history records by age, specifying a **Number of Days** between 1 and 32767. **Date** lets you delete all records written before a specified **Date** in eight digit format yyyymmdd (e.g. 20000218).

3 When ready, click **Finish**.

Modify Statistics (DBArtisan)

The Modify Statistics Wizard lets you delete unwanted statistics history records from the statistics catalog tables. You can delete statistics history records that were written before a specific date, or records of a specific age. You can delete the records for an entire tablespace, index space, or index.

You should run the wizard regularly to clear outdated information from the statistics history catalog tables. Deleting outdated information from the statistics history catalog tables, improves the performance of processes that access data from those tables.

Platform Availability

- DB2 z/OS

To Run the Modify Statistics Utility

1 Initiate a **Modify Statistics** action against a tablespace. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Modify Statistics** wizard:

Step		Settings and tasks
1	**Select tablespaces**	Displays the available tablespaces.
	Add	Opens theObject Selector Dialog Box.
	Delete	Deletes the selected items.
2	**Select indexes**	Displays the available indexes.
	Add	Opens the Index Selector Dialog Box.
	Delete	Deletes the selected items.
3	**Select indexspaces**	Displays the available indexspaces.
	Add	Opens the **Indexspaces Selector** dialog.
	Delete	Deletes the selected items.
4	**Delete history record(s) by**	**Age** lets you delete history records by age, specifying a **Number of Days** between 1 and 32767. **Date** lets you delete all records written before a specified **Date** in eight digit format yyyymmdd (e.g. 20000218).
	Delete options	Lets you choose among **Delete All statistics rows**, **Delete Access path statistics rows**, and **Delete Space tuning statistics rows** options

3 When ready, click **Finish**.

Mount Disk Group (DBArtisan)

This action builds and submits an ALTER DISKGROUP... MOUNT or ALTER DISKGROUP... DISMOUNT FORCE/NOFORCE statement. This lets you manually make disk groups available or unavailable to all database instances running on the same node.

Note: Before this object action is available you must have a direct connection to the ASM instance. For details, see Manage ASM Connection (DBArtisan).

Platform Availability

- ORCL 10g ^

To Mount or Dismount a Disk Group

1 On a connected Oracle datasource, expand the **Instance** and **ASM** nodes.

2 Right-click the **Disk Groups** node and select **Mount Disk Group** from the context menu.

3 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks
Action options	**All** - Selected, an ALTER DISKGROUP ALL... statement is generated, letting you the mount or dismount operation to occur in all disk groups. **Dismount** - select this check box to make the disk group unavailable. Deselect this check box to have the disk group mounted. **Force** - only available with **Dismount** selected. If **Force** is selected, ASM will dismount the disks even if some files in the disk group are open. If deselected, a NOFORCE option is used. **Restricted** (available as of Oracle 11GR1) - When selected, a RESTRICTED clause is included in the generated ALTER DISKGROUP statement, mounting the disk in single-instance exclusive mode, making the disk group unusable by any ASM client. When unselected, a NORMAL clause is included in the generated ALTER DISKGROUP statement, mounting the disk group in shared mode, and other ASM instances and clients can access the disk group.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

4 Click **Execute**. For information on the scheduling option, see Scheduling.

Move Log

You can move a transaction log from one device to another.

Platform Availability

- SYB ASE

To Move a Database Transaction Log to Another Device

1 Initiate a **Move Log** action against one or more databases. For more information see Initiating an object operation.

The **Move Log** dialog opens.

2 From the **New Device** dropdown, select the defined device to which you want to move the transaction log.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Move Table

The **Move Table** action lets you move data in an active table into a new, identically-named table, leaving the data online and available for use. This action lets you build and submit one or more SYSPROC.ADMIN_MOVE_TABLE calls. A single call, corresponding to the MOVE command, lets you execute all move operations in a single step. Multiple calls correspond to use of the INIT, COPY, REPLAY, VERIFY, SWAP, CLEANUP or CANCEL commands, and let you execute the move operations in individual steps.

Note: Before using this object action, consult DB2 LUW documentation for information on the ADMIN_MOVE_TABLE procedure, specifically the options and parameters used by the procedure. For more information, see Accessing Third Party Documentation.

Platform Availability

- DB2 LUW 9.7 ^

To move table data

1 Initiate a **Move Table** action against a table. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and choosing options the options in the dialog:

Pane		Categories and Option Descriptions
Action Options	Method	The **Use Target Table** option lets you select from the two methods available with use of the ADMIN_MOVE_TABLE procedure. Select the **Use Target Table** check box if you have already created the target tabl , and are using this action to move the data to that table. Leave the **Use Target Table** check box unselected to specify tablespaces used by the new table and have this action create the table before moving data to the new table.
	Tablespace	These options are only available with the **Use Target Table** check box deselected. They let you optionally specify new data, index, and LOB tablespaces for the target table.
	Target Table	This option is only available with the **Use Target Table** check box selected. It lets you specify an existing table to be used as the target table during the move.
	Operations	The **Generate...** options correspond to the ADMIN_MOVE_TABLE Operation input parameters. These can be used in one of two mutually exclusive ways. You can generate a single ADMIN_MOVE_TABLE call specifying MOVE to have all operations performed at one time. Alternatively, you can generate one or more ADMIN_MOVE_TABLE calls corresponding to each of the selected INIT, COPY, REPLAY, VERIFY, SWAP, CLEANUP, or CANCEL options, to have the table moved one step at a time.

	MOVE, INIT, COPY, REPLAY, VERIFY, SWAP, CLEANUP, and CANCEL Options	A set of options is displayed for each selected **Generate...** option. Depending on the **Generate...** selection, KEEP, COPY_USE_LOAD, COPY_WITH_INDEXES, FORCE, NO_STATS, COPY_STATS, NO_AUTO_REVAL, REORG, NO_TARGET_LOCKSIZE, CLUSTER, NON_CLUSTER, and LOAD_MSGPATH options may be available. Keep in mind that CLUSTER and NON_CLUSTER options are mutually exclusive and that LOAD_MSGPATH is only available when COPY_USE_LOAD is selected.
Preview	Displays the DDL generated to execute the operation. For more information, see Preview.	

3 Click the **Execute** or **Schedule** button.

Next Used Filegroup

This action builds and submits an ALTER PARTITION SCHEME... NEXT USED statement, letting you alter the designated NEXT USED filegroup for a partition scheme.

Platform Availability

- SQL SVR

To Change a Partition Scheme's Next Used Filegroup

1 Initiate a **Next Used Filegroup** action against a partition scheme. For more information, see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks
Next Used Filegroup	Select the new filegroup from the **Filegroup Name** dropdown.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Object Properties

This action submits a `sp_iqhelp` procedure call against a selected object of one of the following types:

Domains **Events Functions Procedures**

System Tables Tables Views

Standard `sp_iqhelp` output, providing information about the object, is opened in a Results Editor.

Platform Availability

- SYB IQ

To display sp_iqhelp information for an object

1 Initiate an **Object Properties** action against an object of one of the types in the table above. For more information see Initiating an object operation.

The output opens in a Results Editor.

Open (DBArtisan - Logs and Error logs)

This action displays the contents of error logs or SQL Server Agent-specific log files.

Platform Availability

- SQL SVR

To display the contents of a log file

1 In the Navigator/Explorer, expand the **Server** node and if opening a SQL Server Agent log, expand the **SQL Server Agent** node.

2 In the Navigator/Explorer, select the **Logs** or **Error Logs** node.

3 In the Navigator/Explorer, right click an ierror log or log and select **Open** from the context menu.

The contents of the file are displayed in the right-hand pane.

Open Control Files (DBArtisan)

The control file contains information about the associated database that is required for the database to be accessed by an instance, both at startup and during normal operation.

Note: Only the Oracle database server can modify a control file's information; no user can edit a database's control file directly.

The **Control File Information** dialog lets you view statistics (**Type**, **Record Size**, **Records Total**, **Records Used**, **First Index**, **Last Index**, and **Last Recid**) on the following control file entries:

- ACM OPERATION
- ARCHIVED LOG
- BACKUP CORRUPTION
- BACKUP DATAFILE
- BACKUP PIECE
- BACKUP SET
- BACKUP REDOLOG
- BACKUP SPFILE
- CKPT PROGRESS
- COPY CORRUPTION
- DATABASE
- DATABASE BLOCK CORRUPTION
- DATABASE INCARNATION
- DATAFILE
- DATAFILE COPY
- DATAFILE HISTORY
- DELETED OBJECT
- FILENAME
- FLASHBACK LOG
- FOREIGN ARCHIVED LOG
- GUARANTEED RESTORE POINT
- INSTANCE SPACE RESERVATION
- LOG HISTORY
- MTTR
- OFFLINE RANGE
- PROXY COPY
- RECOVERY DESTINATION
- REDO LOG
- REDO THREAD
- REMOVABLE RECOVERY FILES
- RESTORE POINT
- RMAN CONFIGURATION
- RMAN STATUS
- STANDBY DATABASE MATRIX
- TABLESPACE
- TEMPORARY FILENAME
- THREAD INSTANCE NAME MAPPING

Platform Availability

- ORCL

To View the Control File for a Database

1 On the Navigator/Explorer, under **Storage** expand the Control Files node.

DBArtisan displays the Control Files.

2 In the right pane of the application, right-click a control file, and then select **Open**.

DBArtisan opens the **Control File Information** dialog.

Optimize Tables

You can issue an OPTIMIZE TABLE statement, reclaiming unused space and defragmenting the data file.

Platform Availability

- MySQL

To Optimize Table Storage

1 Initiate an **Optimize Tables** action against one or more tables. For more information see Initiating an object operation.

The **Optimize Tables** dialog opens.

2 Use the following table as a guide to understanding and modifying settings in this wizard:

Step	Settings and tasks
Action Options	Lets you review the tables you selected.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Place

The **Placement** dialog lets you place tables and indexes on different segments. From a performance standpoint it is not recommended to have a table and its supporting indexes on the same device or disk segment. It is also good to have more frequently accessed indexes and tables grouped together on higher speed devices, if possible.

Platform Availability

- SYB ASE

To Place an Index or Table on a Segment

1 Initiate a **Place** action against one or more tables or indexes. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Place** wizard:

Step	Settings and tasks
Action options	Use the **Segment** dropdown to specify the segment on which you can place objects, the default, logsegment or system.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation. For more information, see Dependencies.
Preview	Preview the DDL generated for the operationn. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Population status

This action builds and submits an ALTER FULLTEXT INDEX statement with a START {FULL|INCREMENTAL|UPDATE} POPULATION or STOP | PAUSE | RESUME POPULATION argument, letting you start, stop, pause or resume population of the full-text index.

Note: Before using this action, consult Microsoft SQL Server documentation for detailed information on topics such as timestamp column restrictions on the INCREMENTAL option and the interaction of these options with the background update index and auto change tracking features. For more information, see Accessing Third Party Documentation.

Platform Availability

- SQL SVR

To Control Population of a Full-text Index

1 Initiate a **Population Status** action against one or more full-text indexes. For more information, see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks
Action Options	Select an option from the **Action** dropdown: START FULL - every row of the table is retrieved for full-text indexing, even rows that have already been indexed. START INCREMENTAL - only the rows modified since the last population are retrieved. This option only works properly against tables with a timestamp column. START UPDATE - only insertions, updates, or deletions processed since the last change-tracking index update. Change-tracking population must be enabled on the table, but the background update index or the auto change tracking should be disabled. STOP - stops a paused population or population in progress. PAUSE - pauses a FULL population in progress RESUME - resumes a paused FULL population.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Purge Recycle Bin

This action builds and submits a PURGE RECYCLEBIN, PURGE DBA_RECYCLEBIN, PURGE TABLESPACE *tablespacename*, or PURGE TABLESPACE *tablespacename* USER *username* statement. This lets you select one of the following purge Recycle Bin options:

- Purge all objects belonging to the current user
- Purge all objects belonging to all users
- Purge all objects residing on a specified tablespace
- Purge all objects residing on a specified tablespace and belonging to a specified user

Platform Availability

- ORCL 10g ^

To Purge Objects in the Recycle Bin Tables and Associated Objects

1 In DBArtisan, expand the **Schema** node, right-click the **Recycle Bin** node and select **Purge Recycle Bin** from the context menu.

2 Use the following table as a guide to understanding and modifying settings on the **Confirm Purge Recycle Bin** wizard:

Panel		Settings and tasks
Action Options	**Purge**	Select a PURGE option type: RECYCLEBIN purges all objects belonging to the current user. DBA_RECYCLEBIN purges objects system wide. TABLESPACE purges objects residing on a specific tablespace.
	Tablespace	If you selected a **Purge** value of TABLESPACE, select the tablespace for which all residing objects will be purged. Optionally, use the **User** control to further qualify the objects that will be purged.
	User	If you selected a **Purge** value of TABLESPACE, you can use this control to restrict purging to only those tablespace-residing objects belonging to a particular user.
Preview	Displays the DDL that will execute the object action. For details, see Preview.	

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Related Information

- Recycle Bin (for a support summary including topics such as displaying Recycle Bin contents)
- Enable Recycle Bin
- Flashback Recycle Bin Entry
- Purge Recycle Bin Entry

Purge Recycle Bin Entry

This action builds and submits a PURGE INDEX or PURGE TABLE statement, letting you permanently delete tables or indexes from the Oracle Recycle Bin. Once purged, the tables or indexes and any associated objects that were contained in the Recycle Bin are no longer retrievable.

Note: The Datasource Explorer entry for a Recycle Bin entry has a **Can Purge** field that indicates whether a **Purge Recycle Bin Entry** action can be performed against the item

Platform Availability

- ORCL 10g

To Permanently Purge Selected Tables or Indexes Currently in the Recycle Bin

1 In the DBArtisan Datasource Navigator/Explorer, under an Oracle **Schema** node, select the **Recycle Bin** node. Contents of the Recycle Bin are displayed.

2 Right-click one or more selected tables or indexes in the Recycle Bin and select **Purge Recycle Bin Entry** from the context menu. The **Confirm Purge Recycle Bin Entry** dialog opens.

3 Use the following table as a guide to understanding and modifying settings in the wizard:

Step	Settings and tasks
Action options	Displays the table and index entries you selected to purge.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation. For details, see Dependencies.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

4 Click **Execute**. For information on the scheduling option, see Scheduling.

Note: When dropping tables on a datasource with the Recycle Bin feature enabled, the **Drop** action has an option that lets you immediately purge the table.

Related Information

- Recycle Bin (for a support summary including topics such as displaying Recycle Bin contents)
- Enable Recycle Bin
- Flashback Recycle Bin Entry
- Purge Recycle Bin

Quiesce

You can quiesce at the instance or database level. For details, see the following topics:

- Quiesce (Database)
- Quiesce (Instance)

Quiesce (Database)

This action lets you restrict user access to a database.

Platform Availability

- DB2 LUW

Note:Quiesce Database is not supported for a DB2 8 server from an IBM DB2 for Linux, Unix, and Windows client or for an IBM DB2 for Linux, Unix, and Windows 7 server from an IBM DB2 for Linux, Unix, and Windows client.

To Quiesce a Database

1 On the Navigator/Explorer, right-click a datasource node and select **Command > Quiesce** from the context menu. Provide login credentials if prompted.

2 Use the following table as a guide to understanding and modifying settings on the **Quiesce Database** dialog:

Setting	Description
Connections	Lets you select an **Immediate** quiesce or **Defer.**

3 Click **Execute**.

Related Information

- Unquiesce

Quiesce (Instance)

This action lets you restrict user access to an instance.

Platform Availability

- DB2 LUW

To Quiesce an Instance

1 On the Navigator/Explorer, right-click the **Instance** node.

2 Select **Command > Quiesce** from the context menu. Provide login credentials if prompted.

3 Use the following table as a guide to understanding and modifying settings on the **Quiesce Instance** dialog:

Setting	Description
For user	Select this radio button to restrict a user and then type the user name in the associated box.
For group	Select this radio button to restrict a group and then type the group name in the associated box.
Connections	Lets you select an **Immediate** quiesce or **Defer.**

4 Click **Execute**.

Related Information

- Unquiesce

Quiesce Tablespaces (DBArtisan)

The Quiesce Tablespaces action lets you build and submit a call to the online QUIESCE utility with a TABLESPACES option, establishing a quiesce point on a table space.

Platform Availability

- DB2 z/OS

To Quiesce a Tablespace

1 On the Datasource Explorer, select the **Tablespaces** node or in the Datasource Navigator expand the node.

DBArtisan displays the Tablespaces.

2 Right-click one or more selected tablespaces, and then select **Quiesce Tablespaces** from the context menu.

DBArtisan opens the **Quiesce Utility Wizard** dialog.

3 Optionally, deselect **Write To Disk** so that changed pages are not written to disk.

4 Click **Finish**.

Related Information

- Recover (DBArtisan)

Reassign By Category

This action lets you reassign the category of stored outlines in Oracle.

Outlines are a set of results for the execution plan generation of a particular SQL statement. When you create an outline, plan stability examines the optimization results using the same data used to generate the execution plan. That is, Oracle uses the input to the execution plan to generate an outline, and not the execution plan itself.

Platform Availability

- ORCL

To Reassign a Stored Outline to a Different Category

1 Initiate a **Reassign By Category** action against one or more stored outlines. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Reassign By Category** wizard:

Step	Settings and tasks
Action options	Displays the outlines you selected. Select a new category for the outline or outlines from the **Category** dropdown.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Related Information

- Change Category
- Drop By Category

Rebalance

This action lets you build and submit one or more ALTER TABLESPACE... REBALANCE statements. This creates containers on recently added storage paths and drops containers from storage paths that are in DROP PENDING state.

Platform Availability

- DB2 LUW 9.7^

Prerequisites or Setup Tasks

This object action is only available for tablespaces that meet the following criteria:

- The tablespace must use automatic storage. That is, in the Tablespace Editor, the **Use Automatic Storage** property is selected.For more information see Tablespaces Editor (IBM DB2 LUW).
- In the Tablespace Editor, the **Type** property specifies LARGE or REGULAR.

To rebalance a tablespace

1 Initiate a **Rebalance** action against one or more tablespaces.

2 Use the following table as a guide to understanding and modifying settings in this wizard:

Step	Settings and tasks
Action Options	Displays the tablespaces selected for this operation.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Related Information

- See Drop Automatic Storage Path(s).

Rebalance Disk Group (DBArtisan)

This action builds and submits an ALTER DISKGROUP... REBALANCE statement with an optional POWER option. This lets you manually rebalance the disk group, distributing datafiles evenly across drives.

Note: Before this object action is available you must have a direct connection to the ASM instance. For details, see Manage ASM Connection (DBArtisan).

Platform Availability

- ORCL 10g

To Manually Rebalance a Disk Group

1 On a connected Oracle datasource, expand the **Instance** and **ASM** nodes.

2 Right-click the **Disk Groups** node and select **Rebalance Disk Group** from the context menu.

3 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks
Action options	**Power** - If deselected, ASM executes rebalance operations at the power rate set by the ASM_POWER_LIMIT initialization parameter. If selected, you can specify the power rate using the **Power value** control. **Power value** - If **Power** is selected, specify a value between **0** and **11**. A **0** value stops the rebalance operation and at the other extreme, **11** lets ASM execute the rebalance as quickly as possible. **Wait** (available as of Oracle 11GR1) - Lets you specify a WAIT/NOWAIT clause, letting you specify when during the rebalance that control is returned to the user.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

4 Click **Execute**. For information on the scheduling option, see Scheduling.

Rebind Packages

The **Rebind Package** dialog lets you update the best access path for SQL statements when the contents of a package changes.

Tip: If the physical storage of a package is changed or dropped, rebinding updates the path of the SQL statements.

Platform Availability

- DB2 LUW, DB2 z/OS

To Rebind One or More Packages

1 Initiate a **Rebind Packages** action against one or more packages. For more information see Initiating an object operation.

The **Rebinding Package** dialog opens.

2 Click **Execute**. For information on the scheduling option, see Scheduling.

Rebind Plans

This action lets you update the best access path for SQL statements when the contents of a plan change.

Tip: If the physical storage of a plan is changed or dropped, rebinding updates the path of the SQL statements.

Platform Availability

- DB2 z/OS

To Rebind a Plan

- In DBArtisan, click the **Rebind** button on the **Plans** tab of the DBRMs editor. For more information, see DBRMs Editor (DBArtisan - IBM DB2 Z/OS).
- In DBArtisan, click the **Rebind** button on the **Plans/Packlists** tab of the Packages editor. For more information, see Packages Editor (IBM DB2 Z/OS).

Rebuild (Full-text Catalogs)

This action builds and submits a basic ALTER FULLTEXT CATALOG *catalogname* REBUILD statement, with no additional options or arguments. This lets you rebuild the entire catalog by deleting the existing catalog and building a new one.

Platform Availability

- SQL SVR

To Rebuild a Full-text Catalog

1 Initiate a **Rebuild** action against one or more full-text catalogs. For more information, see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks
Action options	Displays the catalogs you selected.
Dependencies	Lets you review the objects potentially impacted by this action. For more information, see Dependencies.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Rebuild Index

See the following topics for DBMS-specific instructions on rebuilding indexes:

- Rebuild Index (DBArtisan - DB2 z/OS)
- Rebuild Index (Oracle)
- Rebuild Index (SQL Server)

Rebuild Index (DBArtisan - DB2 z/OS)

The Rebuild Index Wizard reconstructs indexes from the table that they reference.

To Rebuild One or More Indexes

1 On the Datasource Explorer, select the **Indexes** node or in the Datasource Navigator expand the node.

2 Right-click the target index select **Rebuild Index** from the context menu.

The **Rebuild Index Utility** wizard opens.

3 Use the following table as a guide to understanding and modifying settings in the wizard:

Panel	Option	Description
1	Select Indexes	Lets you select an index to rebuild.
	Add	For more information, see Index Selector Dialog Box.
	Delete	Deletes the selected index(es).
2	Select tablespaces	Lets you select the target tablespace.
	Add	For more information, see Table Selector Dialog Box.
	Delete	Deletes the selected items.
3	Do you want the utility to be restartable?	Select to make the utility restartable.
	Do you want to reuse the db2 managed dataset?	Default is No.
	WorkDDN	Lets you select a work ddn template from the list or click Edit to specify the dataset information.
	Do you want to specify to sort device type?	Default is No.
	Do you want to specify the sort number?	Default is No.
	Do you want to specify the sort keys?	Default is No.

4	Do you want to specify the stats online?	Lets you specify the stats online.
	Do you want to output message to SYSPrint?	The default is no.
	Do you want to specify the correlation stats option?	Lets you specify correlation stats on the next panel of the wizard.
	Do you want to force aggregation or rollup processing to be done even though some parts do not contain data?	The default is Default.
5	KeyCard	Collects all of the distinct values in all of the 1 to n key column combinations for the specified indexes. n is the number of columns in the index.
	Frequent Item	Lets you add items to the grid.
	Add	For more information, see Add Frequent Item Dialog Box.
6	Update Statistics	Lets you specify update statistics.
	History Statistics	Lets you specify history statistics.

Rebuild Index (Oracle)

The **Rebuild Indexes** dialog lets you rebuild an index that has become fragmented. Rebuilding an index is a good alternative to coalescing an index because you can move the index to a different tablespace and change both tablespace and storage parameters while eliminating fragmentation. However, rebuilding an index has a higher cost than coalescing an index. These same qualities also make rebuilding an index a viable alternative to dropping an index then re-creating it.

As a rule of thumb, check indexes for rebuilds when their level (or tree depth) reaches four or greater, or many deleted leaf rows are found. The **Rebuild Indexes** dialog can also be used to easily move an index from one tablespace to another.

Important Notes

- If you are rebuilding a function-based index, the index is enabled when the rebuild is finished.
- You cannot rebuild a partitioned index. You must rebuild each partition or subpartition individually.

The table below describes the options and functionality on the **Rebuild** dialog.

Option	Description
New Tablespace	Defaults to the tablespace which currently includes the index. To change the tablespace containing the index, choose a new tablespace from the list.
Logging	Recoverable - The creation of the index logs in the redo log file. Non-Recoverable - The creation of the index is not logged in the redo log file.
Use Parallel Processes	Performs processes for the sequential execution of a SQL statement in parallel using multiple parallel processes. One process, known as the parallel execution coordinator, dispatches the execution of a statement to several parallel execution servers and coordinates the results from all of the server processes to send the results back to the user. NOTE: Only available for Oracle with the Parallel Server option. NOPARALLEL execution - Select this if you are concerned that the cost of synchronizing parallel processes will impede the throughput of data.
Order	Reverse - Instructs Oracle to store the bytes of the index block in reverse order and to exclude the ROWID when rebuilding the index. No Reverse - Instructs Oracle to store the bytes of the index block in normal order when rebuilding the index.

To Rebuild an Oracle Index

1 Initiate a **Rebuild** action against one or more indexes. For more information see Initiating an object operation.

The **Rebuild Indexes** dialog opens.

2 To move the index to a new tablespace, click the **New Tablespace** list and then click the new tablespace.

3 In the **Logging** box, click:

- The **Recoverable** option button to make the operation log in the redo file.
- The **Non-Recoverable** option button if you do not want the operation logged in the redo file.

4 If you are using Parallel Server, select the **Parallel Server** check box and:

- Type a value indicating the number of query server processes that should be used in the operation in the **Degree** box.
- Type a value indicating how you want the parallel query partitioned between the Parallel Servers in the **Instances** box.

5 In the **Order** box:

- Click the **Reverse** option button to rebuild the index to store the bytes of the index block in reverse order.
- Click the **No Reverse** option button to rebuild the index to store the bytes of the index block in order.

6 Click **Execute**. For information on the scheduling option, see Scheduling.

Rebuild Index (SQL Server)

The **Rebuild Indexes** dialog lets you rebuild an entire index, primary key, or unique key or a single partition of those objects. Depending on your choice, a number of REBUILD WITH clause options are available.

To Rebuild an Index

1 Initiate a **Rebuild Index** action against one or more indexes, primary keys, or unique keys. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Rebuild Index** wizard:

Step	Settings and tasks
Action options	To rebuild a single partition, specify a **Partition number** and optionally, provide **Sort in tempdb** and **MaxDOP** property values used to create the REBUILD WITH clause. To rebuild the entire index, primary key, or unique key, DO NOT provide a **Partition Number**, and optionally, provide **Pad Index**, **Sort in tempdb**, **Ignore Duplicate Key**, **Statistics no recompute**, **Online**, **Allow Row Locks**, **Allow Page Locks**, **MaxDOP**, and **Fill Factor** property values used to create the REBUILD WITH clause. For more information on these properties, see Indexes Wizard (SQL Server).
Dependencies	Lets you review the objects potentially impacted by this action.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Rebuild Outlines

You can rebuild one or more stored outlines in a single operation.

Platform Availability

- ORCL

To Rebuild a Stored Outline

1 Initiate a **Rebuild Outlines** action against one or more stored outlines. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Rebuild Outlines** dialog:

Step	Settings and tasks
Action options	Displays the stored outlines you selected.
Dependencies	Lets you review the objects potentially impacted by this action.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Rebuild Table

This action builds and submits an ALTER TABLE statement with an ENGINE=InnoDB option.

Platform Availability

- MySQL

To Rebuild a Table

1 Initiate a **Rebuild Table** action against one or more tables. For more information, see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Rebuild Table** dialog:

Step	Settings and tasks
Action options	Displays the tables you selected.
Dependencies	Lets you review the objects potentially impacted by this action. For more information, see Dependencies.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Recompile

You can recompile one or more tables. Recompilation causes each procedure and trigger that uses the target table to be recompiled the next time it runs.

The queries used by procedures and triggers are optimized only once, when they are compiled. As you add indexes or make other changes to your database that affect its statistics, your compiled procedures and triggers may lose efficiency. By recompiling the procedures and triggers that act on a table, you can optimize the queries for maximum efficiency.

Platform Availability

- SQL SVR, SYB ASE

To Recompile One or More Tables

1 Initiate a **Recompile** action against one or more tables. For more information see Initiating an object operation.

The **Recompile Tables** dialog opens.

2 Click **Execute**. For information on the scheduling option, see Scheduling.

Recover (DBArtisan)

This action lets you build and submit a call to the RECOVER utility, recovers data to the current state or to a previous point.

Platform Availability

- DB2 z/OS

To Recover Data to the Current State or to Some Previous State

1 On the Datasource Explorer, select the Tablespaces or Indexes node or in the Datasource Navigator expand one of those nodes.

2 Right-click a tablespace or index and then select **Recover** from the context menu.

The **RECOVER Utility Wizard** dialog opens.

3 Complete the panels of the wizard and when complete click **Finish**.

Recover Abort (DBArtisan)

This action issues a RECOVER INDOUBT... ACTION ABORT command against an indoubt thread, cancelling the thread.

Platform Availability

- DB2 z/OS

To cancel an indoubt thread

1 In the Navigator/Explorer, expand the **Performance** node.

2 If using the Datasource Explorer select the **Indoubt Threads** node or if using the Datasource navigator expand that node.

3 Right click an indoubt thread and select **Recover Abort** from the context menu.

4 When prompted to verify, click **Yes**.

See Also

- Recover Commit (DBArtisan)

Recover Commit (DBArtisan)

This action issues a RECOVER INDOUBT... ACTION COMMIT command against an indoubt thread, committing the thread

Platform Availability

- DB2 z/OS

To commit an indoubt thread

1 In te Datasource Navigator/Explorer, expand the **Performance** node.

2 If using the Datasoure Explorer select the **Indoubt Threads** node or if using the Datasource Navigator expand that node.

3 Right click an indoubt thread and select **Recover Commit** from the context menu.

4 When prompted to verify, click **Yes**.

See Also

- Recover Abort (DBArtisan)

Redistribute (DBArtisan)

DBArtisan lets you redistribute the data in a nodegroup. The Redistribute action implements the REDISTRIBUTE NODEGROUP and lets you specify UNIFORM, USING DISTFILE, USING TARGETMAP, -CONTINUE, or ROLLBACK parameters.

Platform Availability

- DB2 LUW

To Redistribute the Data in a Nodegroup

1 If using the Datasource Explorer select the Nodegroups node or if using the Datasource Navigator expand that node.

DBArtisan displays the nodegroups.

2 Select one or more nodegroups, right-click the selected objects, and then select **Redistribute** from the context menu.

DBArtisan opens the **Redistribute Nodegroup** dialog.

3 Select one of the following methods:

- Uniform
- Using Dist File (provide the distinct file name in the corresponding box)
- Using Target Map (provide the target file name in the corresponding box)

4 Click **Execute**. For information on the scheduling option, see Scheduling.

Refit (DBArtisan)

DBArtisan lets you refit a database device. If you are restoring a damaged master database, or have added devices since the last database or transaction log dump, then refit a device after reinitializing. Refitting a device ensures that the information in sysusages and sysdatabases is consistent with the information in sysdevices.

Platform Availability

- SYB ASE

To Refit a Database Device

1 Initiate a **Refit** action against one or more database devices. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Refit** wizard:

Step	Settings and tasks
Action options	Displays the database devices you selected for refitting.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation.
Preview	Preview the DDL generated for the operation.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Refresh Table

The **Refresh Table** dialog lets you reload materialized query tables that have been defined with refresh options.

Platform Availability

- DB2 LUW

To Reload a Materialized Query Table

1 Initiate a **Refresh Table** action against one or more materialized query tables. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings on the **Refresh Materialized Query Table** dialog:

Pane	Description
Action Options	Select an **Online Option** of ALLOW NO ACCESS, ALLOW READ ACCESS or ALLOW WRITE ACCESS. Enable or disable **Query Optimization**. Select an INCREMENTAL or NOT INCREMENTAL **Refresh Option.**
Preview	Preview the DDL generated by the options you chose. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

For procedures on restoring a damaged master database, consult the Commands Reference Manual.

Refresh Materialized View

This action lets you build and submit a REFRESH MATERIALIZED VIEW statement, to replace the contents of a materialized view.

Platform Availability

PSTGRS

Note: This action is only available against views created with the Is Materialized control selected. For more information, see Views Wizard (PostgreSQL).

To replace the contents of a materialized view

1 Initiate an **Refresh Materialized View** action against a view. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks
Action options	If **With No Data** is selected, a WITH NO DATA clause is included in the generated DDL. No new data is generated and the materialized view will be in an unscannable state.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

3 Finally, use the **Execute** button to create the object.

For related information, see Backup Certificate.

Reinitialize (DBArtisan)

The reinitialize function allows you to keep a database device on the Sybase ASE server open while resetting the device to its state immediately after it was opened.

Platform Availability

- SYB ASE

To Reinitialize a Database Device

1 Initiate a **Reinitialize** action against a database device. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings on the **Reinitialize** dialog:

Pane	Description
Action Options	Displays the database device you selected to reinitialize.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation. For more information, see Dependencies.
Preview	Preview the DDL generated by the options you chose. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Rename

This action lets you rename an object. In general, all referenced or referring objects are updated to reflect the new name.

The following table outlines object type support for the operation for all supported DBMS:

	DB2 LUW	DB2 z/OS	ITB/FBD	MySQL	Oracle	SQL SVR	Sybase ASE
Blob filters			✓				
Check constraints	✓	✓	✓		✓	✓	✓
Databases						✓	✓
Defaults							✓
Domains			✓				
Exceptions			✓				
Extended procedures							✓
Foreign keys	✓	✓	✓		✓		✓
Generators			✓				
Indexes	✓		✓		✓		✓
Materialized Views					✓		
Primary keys	✓		✓		✓		✓
Procedures			✓		✓		✓
Roles			✓				
Rules							✓
Sequences	✓				✓		
Shadows			✓				
Stored outlines					✓		
Synonyms					✓		
Tables	✓	✓		✓	✓		✓
Tablespaces	✓				✓		
Unique keys	✓		✓		✓		✓
Views			✓		✓		

The following table notes the exceptions and provides prerequisite tasks to be performed before renaming an object.

DBMS	Notes and restrictions on renaming
Microsoft SQL Server	Microsoft SQL Server lets you rename a database if you own it. Before renaming a database, set it to single-user mode. Microsoft SQL Server will not rename a table if it is referenced within the body of other objects that call it, such as tables, triggers or views. As a result, renaming a table can result in broken dependencies with other objects. Also, Microsoft SQL Server does not let you rename System Tables.
IBM DB2 for OS/390 and z/OS	You can rename a primary key if the underlying table has only one owner. The rename operation does not rename the table if it is referenced within the body of other objects, such as tables, triggers or views, that call it. As a result, renaming a table can result in broken dependencies with other objects.
Sybase ASE	Before renaming a database, set it to single-user mode. System indexes can not be renamed. The rename operation does not rename the stored procedure if it is referenced within the body of other objects, such as another stored procedure, that call it. As a result, renaming a stored procedure can result in broken dependencies with other objects. The rename operation does not rename the table if it is referenced within the body of other objects, such as tables, triggers or views, that call it. As a result, renaming a table can result in broken dependencies with other objects. The rename operation does not rename the view if it is referenced within the body of other objects, such as stored procedures, triggers or other views, that call it. As a result, renaming a view can result in broken dependencies with other objects. Databases with a **Type** of ARCHIVE or with the **In-memory** property selected cannot be renamed. For more information, see Databases (Sybase ASE) - Properties.

To Rename an Object

1 Initiate a **Rename** action against a supported object (see the table above). For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Rename** dialog:

Step	Settings and tasks	
Rename	**Name**	Provide the new name for the object.
Dependencies	Review the referring and referred objects for which naming will be automatically resolved when you execute the renaming operation. For more information, see Dependencies.	
Preview	Preview the DDL generated for the operation. For more information, see Preview.	

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Reorganize /Reorg

Reorganize or Reorg options are offered against the following DBMS platform/object types:

- Reorganizing IBM DB2 for Linux, Unix, and Windows Objects
- Reorganizing Oracle Objects
- Reorganize (SQL Server indexes, primary keys, and unique keys)
- Reorganize (SQL Server Full-text Catalogs)
- Reorganize (Sybase ASE Indexes and Tables)
- Reorg Index (DB2 z/OS)
- Reorg (DB2 z/OS Tablespaces)

Reorganizing IBM DB2 for Linux, Unix, and Windows Objects

Reorganize Dialog Box (One Table)

The table below describes the options and functionality on the **Reorganize** dialog.

Step	Description
Action Options	○ Choose a **Reorg Type** of table or all indexes of the table. ○ Optionally, choose an **Index Schema** and specify an **Index name**. ○ Enable or disable **Inplace Reorg** and if you enable, choose an **Inplace reorgmode** of Start, Stop, Pause, or Resume. ○ Select an **Access mode** to control read and write access. ○ Select an **Index reorgmode** to clean up empty pages, delete after cleaning up empty pages, or convert to a type 2 index. ○ If you did not enable **Inplace Reorg**, select a **Tablespace**. ○ Select a **Long Tablespace**. ○ Enable or disable the following options: **Index scan**, **Reorglong field and LOB data**, **Reset Dictionary**, **No truncate table**, and **Reorganize all partitions**.
Partitions	If you did not enable **Reorganize all partitions**, select the partitions to reorganize.
Preview	Preview the DDL code generated from your choices. For more information, see Preview..

Reorganize Dialog Box (Multiple Tables)

The table below describes the options and functionality on the **Reorganize** dialog.

Option	Description
Temporary Tablespace	Associates a temporary tablespace with the table's tablespace. You can select another tablespace from the list.

Reorganize (SQL Server indexes, primary keys, and unique keys)

The **Reorganize Indexes** dialog lets you reorganize an entire index, primary key, or unique key or a single partition of that object. It also lets you specify a LOB_COMPACTION option.

Note: You cannot reorganize an index (primary key, or unique key) that has an **Allow Page Locks** property set to FALSE. For information on setting index properties, see Indexes Editor (SQL Server).

To reorganize an index

1 Initiate a **Reorganize** action against an index, primary key, or unique key. For more information see Initiating an object operation.

The **Reorganize Index** wizard opens.

2 Use the following table as a guide to understanding and modifying settings in the wizard:

Step	Settings and tasks
Action options	To reorganize a single partition, specify a **Partition number** and optionally, specify **Lob Compaction** used to create the REORGANIZE WITH clause. To reorganize the entire index, primary key, or unique key, DO NOT provide a **Partition Number**, and optionally, specify **Lob Compaction** used to create the REORGANIZE WITH clause.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation. For more information, see Dependencies.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Reorganizing Oracle Objects

The **Reorganize** dialog lets you reduce query processing time against tables.

To Reorganize an Oracle Table

1 Initiate a **Reorganize** action against a table. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Reoganize** dialog:

Panel	Group	Setting and description
Action Options	**Reorganize method**	**Online -** If selected, reorganization is carried out online, using DBMS_REDEFINITION procedures. DBMS_REDEFINITION is Oracle's online table reorganization package and is available for Oracle 9i. During an online reorganization, the table can still be accessed using DML statements like SELECT and UPDATE. If this check box is unselected, reorganization is carried out using a series of simple ALTER TABLE statements. This method is intended for offline usage.
	Tablespace	If you want to move the table(s) to a new tablespace, select the new tablespace from this list.
	Data Block Storage	Type ALTER TABLE property values for the **Percent Free** (PCTFREE property), **Percent Used** (PCTUSED property), **Initial Transactions** (INITTRANS property), and **Max Transactions** (MAXTRANS property).
	Extents	Type an ALTER TABLE MOVE STORAGE property value for the **Initial Extent** (INITIAL) property). Type ALTER TABLE STORAGE property values for the **Next Extent** (NEXT property), **Percent Increase** (PCTINCREASE property), **Minimum Extents** (NEXT property), and **Maximum Extents** (MAXEXTENTS property).
	Freelists	Use the **Free Lists** and **BufferPool** (DEFAULT, KEEP, RECYCLE) settings to provide values for the FREELISTS and BUFFER_POOL components of a STORAGE clause. Use the **Free List Groups** setting to provide a value for the FREELIST GROUPS component of a STORAGE clause
	Parallel Query	Use the **Parallel Degree** setting to provide a value for the DEGREE component of a PARALLEL clause. This specifies the number of servers used in processing operations.
	Physical	Logging **-** When disabled, a NOLOGGING clause is added to the ALTER TABLE statement.
Dependencies	For details on using this tab, see Dependencies.	
Preview	For details on using this tab, see Preview.	

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Reorganize (SQL Server Full-text Catalogs)

This action builds and submits a basic ALTER FULLTEXT CATALOG *catalogname* REORGANIZE statement, with no additional options or arguments. This lets you merge smaller indexes into a larger, master index.

To reorganize a full-text catalog:

1 Initiate a **Reorganize** action against one or more full-text catalogs. For more information, see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks
Action Options	Displays the catalogs you selected.
Dependencies	Lets you review the objects potentially impacted by this action. For more information, see Dependencies.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Reorganize (Sybase ASE Indexes and Tables)

The **Reorganize** dialog lets you reduce query processing time against tables. This functionality is available for both tables and indexes.

For more information, see

- Reorganizing Sybase ASE Tables
- Reorganize Sybase ASE Indexes

Reorganizing Sybase ASE Tables

This action lets you build and submit one of four REORG subcommands: FORWARDED_ROWS, RECLAIM_SPACE, COMPACT, or REBUILD. REORG optimizes the use of table space and improves performance.

The **Reorganize Table** dialog lets you reduce the query processing time against a table by reorganizing the table to ensure that space is properly allocated to it. For lengthy reorganization processes, this dialog box also lets you execute a process in increments, lets you resume an incomplete process, and lets you specify the duration of each increment. For more information, see Incremental Reorganizations.

Tip: Frequent update activity on a table can cause data rows to migrate and to chain over multiple data pages. Chained or forwarded rows can degrade performance because more physical reads are required to access a row of data. Consequently, you should monitor chained rows regularly to spot performance bottlenecks before they become severe. In addition, altering physical storage parameters can lead to fragmentation of space on your data pages, which also results in reduced performance levels.

You should consider reorganizing a table if you are experiencing slow performance due to:

- A large number of chained or forwarded rows on your data pages
- A large amount of fragmentation in your data pages

Note: You can reorganize tables in Sybase ASE versions 12 and 12.5.

The table below describes the options and functionality on the **Reorganize Table** dialog:

Option	Description
Compact	Lets you reclaim space and undo row forwarding. Minimizes interference with other activities by using multiple small transactions of brief duration. Each transaction is limited to eight pages of reorg processing. These three commands also provide resume and time options that allow you to set a time limit on how long a reorg runs and to resume a reorg from the point at which the previous reorg stopped, making it possible to use a series of partial reorganizations at off-peak times to reorg a large table. For information on result options, see Incremental Reorganizations.
Reclaim Space	Lets you reclaim unused space resulting from deletions and row-shortening updates on a page. Minimizes interference with other activities by using multiple small transactions of brief duration. Each transaction is limited to eight pages of reorg processing. These three commands also provide resume and time options that allow you to set a time limit on how long a reorg runs and to resume a reorg from the point at which the previous reorg stopped, making it possible to use a series of partial reorganizations at off-peak times to reorg a large table. For information on result options, see Incremental Reorganizations.
Rebuild	Lets you undo row forwarding and reclaim unused page space. It also rewrites all rows to comply with the target table's clustered index, writes rows to data pages to comply with space management setting changes (via sp_chgattribute), and drops and re-creates all the target table's (or tables') indexes. Reorg rebuild holds an exclusive table lock for its entire duration. On a large table this can be a significant amount of time. However, reorg rebuild accomplishes everything that dropping and re-creating a clustered index does and takes less time. In addition, reorg rebuild rebuilds the table using all of the table's current space management settings. Dropping and re-creating an index does not use the space management setting for reservepagegap. In most cases, reorg rebuild requires additional disk space equal to the size of the table it is rebuilding and its indexes.
Undo Row Forwarding	Lets you undo row forwarding, a process that occurs when an update increases a row's length in a data-only-locked table such that the row is too large to fit on its original page.
Options	Start at the point where a previous reorg left off - Select to resume a previously initiated but incomplete partial reorganization. Then specify the duration for which you want the resumed reorganization to continue before stopping again. This box is disabled for the rebuild command.

Incremental Reorganizations

If target tables are too long to reorganize in one session, you can reorganize them in increments over multiple sessions by specifying a maximum duration for each session. After tables are reorganized for the specified duration, the operation stops until you resume it again from the Options box of the **Reorganize Table** dialog. The Options box lets you specify to resume a previously initiated but incomplete partial reorganization. It also lets you specify the duration for which you want a resumed reorganization to continue before stopping again. The Option box is disabled for the rebuild command.

Note: The duration you specify refers to elapsed time, not CPU time

In the option box, if you select the check box without specifying a duration, the reorg executes at the point where the previous reorg stopped and continues to the end of the target tables. If you clear the check box and specify a duration, the reorg starts at the beginning of the target tables and continues for the specified number of minutes. If you select the check box and specify a duration, the reorg runs from the point where it last left off, and continues for the specified number of minutes.

Note: If you reorganize a table using one command (Compact, Reclaim Space, or Undo Forwarding) for a specified duration, you cannot resume the process from its resume point using a different command. For example, you cannot compact a table for an hour, and then reclaim space on the remainder of the table. A resumed reorganization process must utilize the same command from start to finish. Selecting a different command begins a new reorganization process.

Caution: While this option lets you reorganize a large table in multiple manageable pieces, any updates to the table between reorganization runs can cause pages to be skipped or processed more than once.

Reorganize Sybase ASE Indexes

The **Reorganize Index** dialog lets you reduce the query processing time against a table by running a reorg rebuild command on the target index.

This operation:

- Undoes row forwarding and reclaim unused page space
- Rewrites all rows in the table to comply with the table's clustered index
- Writes rows to data pages to comply with space management setting changes (via sp_chgattribute)
- Drops and re-creates the table's indexes

Reorg rebuild holds an exclusive table lock for its entire duration. On a large table this can be a significant amount of time. However, reorg rebuild accomplishes everything that dropping and re-creating a clustered index does and takes less time. In addition, reorg rebuild rebuilds the table using all of the table's current space management settings. Dropping and re-creating an index does not use the space management setting for reservepagegap. In most cases, reorg rebuild requires additional disk space equal to the size of the table it is rebuilding and its indexes.

Reorg Index (DB2 z/OS)

This action lets you build and submit a REORG INDEX utility call.

Note: Before using this action consult IBM documentation for details on REORG INDEX utility options. For online access to DB2 documentation, see Accessing Third Party Documentation.

To Reorganize an Index

1 Initiate a **Set Default** action against a tablespace. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in this wizard. Note that some settings are only available based on other selections.

Setting	Description
Select Indexes	On opening, the list displays the index you chose to reorganize. Optionally, you can add more indexes to the list that are to be reorganized by clicking **Add**. Adding indexes using the **Index Selector** dialog is a two-step process. Use the **Database Like**, **Index Creator**, **Index Like**, and **Match Case** controls to provide a search criteria pattern and then click **Query** to display qualified indexes in the **Index Selector** list. Then select one or more indexes from the list, click **Add**, and close the dialog. To remove a selected index from the list to be reorganized, select the index in the list and click **Delete**.
Do you want the utility to be reusable	Select this check box to make the utility restartable.
Share Level	Lets you select share level that allows users to view but not modify the table data: REFERENCE, CHANGE, or None.
Deadline	Lets you specify a deadline for the switch phase of reorganization to start. If it is estimated that the switch phase will not start by the deadline, the reorganization terminates. **None** - Specifies that there is no deadline for the read-only iteration of log processing. **Timestamp** - Specifies the deadline for the switch phase to start processing. *labeled-duration-expression* - Click the ... button to open a dialog that lets you build a custom CURRENT_TIMESTAMP or CURRENT_DATE value.
Drain Specification	Lets you specify that DB2 drains the write claim class after the delay. The number of log records, and thus the estimated time, for a future iteration of log processing will be 0.
Fast Switch	Keeps the data set name unchanged and updates the catalog to reference the newly reorganized data set.
Maxro (only available with a **Share Level** of CHANGE)	Lets you DEFER or specify the maximum amount of time for the last iteration (read-only access iteration) of log processing.
Drain (only available with a **Share Level** of CHANGE)	Lets you specify drain behavior (WRITERS or ALL) at the end of the log phase after the MAXRO threshold is passed and when the last log iteration is to be applied:
Long Log (only available with a **Share Level** of CHANGE)	Lets you provide a LONGLOG value of CONTINUE, DRAIN, or TERM
Delay (only available with a **Share Level** of CHANGE)	Lets you specify the minimum interval between the time REORG sends the LONGLOG message to the console and the time REORG performs the action specified by the **Long Log** setting.
Timeout (only available with a **Share Level** of CHANGE)	Lets you specify a timeout option of ABEND or TERM.
Leaf dist limit	Lets you specify the value that is to be compared to the LEAFDIST value for the specified partitions of the index in SYSIBM.SYSINDEXPART. If any LEAFDIST value exceeds the specified **Leaf dist limit**, REORG is performed or, if you specify REPORTONLY, recommended.

Report	If selected, REORG is only recommended.
Unload	Specify a continue or terminate value of CONTINUE, ONLY, or PAUSE.
Do you want to specify the stats option	Specifies that statistics are to be collected and either reported or stored in the DB2 catalog.
Do you want to output message to SYSPRINT	Specifies that a set of messages is to be generated to report the collected statistics and output to SYSPRINT.
Do you want to specify the correlation stats option	Lets you specify a correlation-stats-spec for statistics reporting.
Do you want to force aggregation or rollup processing to be done event though some parts do not contain data	Specifies whether statistics are to be aggregated or rolled up when RUNSTATS is executed.
Work DDN	Lets you specify a DD name or a TEMPLATE name from a previous TEMPLATE control statement.
Would you like to preformat	Lets you specify that remaining pages are to be preformatted up to the high-allocated RBA in the index space.

3 When ready, click **Finish**.

Reorg (DB2 z/OS Tablespaces)

The Reorganize Tablespace Wizard reorganizes a tablespace to improve access performance and reclaim fragmented space. The wizard can also reorganize a single partition or range of partitions of a partitioned table space. You can specify the degree of access to the data during reorganization, and optionally collect inline statistics.

Completing the Reorganize Tablespace Wizard

To complete the Reorganize Tablespace Wizard, do the following:

1 Initiate a **Reorg** action against a tablespace. For more information see Initiating an object operation.

The **Reorganize Tablespace** Wizard opens.

2 Complete the wizard panels, and then click **Finish**.

The Preview dialog opens.

Reorganize Tablespace Wizard - Panel 1

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

Option	Description
Do you want the utility to be restartable?	Select to make the utility restartable.
Database	Displays the list of databases in the system. You can select the tablespace database or click the filter button to limit the list of databases to a string pattern.
Tablespace	Displays the list of tablespaces.
Partition	Lets you specify a partition or a range of partitions.
Reuse	Select to reuse this tablespace.

Reorganize Tablespace Wizard - Panel 2

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

Option	Description
Log	List of the sequence of events after you execute the wizard.
Sort Data	Sorts columns alphanumerically in single result sets.
No Sysrec	Select Yes for no sysrec.
Sortkeys	Select Yes for sortkeys. Specifies that index keys are to be sorted and built in parallel during the SORTBLD phase to improve performance.

Reorganize Tablespace Wizard - Panel 3

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

Option	Description
Share level	Lets you select share level that allows users to view but not modify the table data: None, Reference, or Change.
Deadline	Lets you specify a deadline for the switch phase of reorganization to start. If it is estimated that the switch phase will not start by the deadline, the reorganization terminates. None - Specifies that there is no deadline for the read-only iteration of log processing. Timestamp - Specifies the deadline for the switch phase to start processing. Click the button to open the Deadline Expression Builder dialog.
Drain Specification	Lets you specify that DB2 drains the write claim class after the delay. The number of log records, and thus the estimated time, for a future iteration of log processing will be 0.
Fast Switch	Keeps the data set name unchanged and updates the catalog to reference the newly reorganized data set.

Deadline Expression Builder

The table below describes the options and functionality on the **Deadline Expression Builder** dialog

Option	Description
Current Date	Lets you select today as the basis of the deadline, click + or -, type the number of years, months, days, hours, minutes, seconds and microseconds.
Current Timestamp	Lets you select the current timestamp as the basis of the deadline, click + or -, type the number of years, months, days, hours, minutes, seconds and microseconds.

Reorganize Tablespace Wizard - Panel 4

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

Option	Description
Local Site Primary	Lets you specify the Local Site Primary.
Local Site Backup	Lets you specify the Local Site Backup.
Recovery Site Primary	Lets you specify the Recovery Site Primary.
Recovery Site Backup	Lets you specify the Recovery Site Backup.

Reorganize Tablespace Wizard - Panel 5

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

Option	Description
Off Pos Limit	The specified value is compared to the result of the calculation (NEAROFFPOSF + FAROFFPOSF) × 100 / CARDF for the specified partitions in SYSIBM.SYSINDEXPART for the explicit clustering indexes for every table in the specified table space. Alternatively, the values in SYSINDEXPART are checked for a single non-partitioned table space, or for each partition if you specified an entire partitioned table space as the target object. If at least one calculated value exceeds the OFFPOSLIMIT value, REORG is performed or recommended. NOTE: This option is valid for non-LOB table spaces only.
Ind Ref Limit	The specified value is compared to the result of the calculation (NEARINDREF + FARINDREF) × 100 / CARDF for the specified partitions in SYSIBM.SYSTABLEPART for the specified table space. Alternatively, the values in SYSTABLEPART are checked for a single non-partitioned table space, or for each partition if you specified an entire partitioned table space as the target object. If at least one calculated value exceeds the INDREFLIMIT value, REORG is performed or recommended. NOTE: This option is valid for non-LOB table spaces only.
Report Only	The reorganization is only recommended, not performed.
Unload	Specifies whether the utility job is to continue processing or end after the data is unloaded. Continue - Specifies that, after the data has been unloaded, the utility continues processing. Pause - Specifies that after the data has been unloaded, processing ends. Only - Specifies that after the data has been unloaded, the utility job ends and the status in SYSIBM.SYSUTIL corresponding to this utility ID is removed. External - Specifies that after the data has been unloaded, the utility job is to end and the status in SYSIBM.SYSUTIL corresponding to this utility ID is removed.

Reorganize Tablespace Wizard - Panel 5

Note: This panel is displays only if the share level is Change.

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

Option	Description
Table Owner	Displays the schema names in the system.
Table	Displays all tables for the selected schema.
Maxrow	Lets you specify the maximum amount of time for the last iteration of log processing. During that iteration, applications have read-only access. Type an integer value or select DEFER. DEFER Specifies that the iterations of log processing with read/write access can continue indefinitely.
Drain	Lets you select Writers or All. Specifies that the write claim class drains after the delay. The number of log records, and thus the estimated time, for a future iteration of log processing will be 0.
Long Log	Lets you specify the action is performed (after sending the LONGLOG message to the console) if the number of log records that are processed during the next iteration is not sufficiently lower than the number of log records that were processed during the previous iterations. Continue - Specifies that performing reorganization continues. Term - Specifies that reorganization terminates after the delay. Drain - Specifies that the write claim class drains after the delay. The number of log records, and thus the estimated time, for a future iteration of log processing will be 0.
Delay	Lets you type a lower bound for the interval between the time when REORG sends the LONGLOG message to the console and the time when REORG performs the action specified by the LONGLOG parameter.
Timeout	Lets you select timeout option: Abend - Abnormal end of task Term - Termination

Reorganize Tablespace Wizard - Panel 6

Note: This panel is displays only if the share level is Reference or Change.

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

Option	Description
Local Site Primary Copy	Lets you specify the Local Site Primary Copy.
Local Site Backup Copy	Lets you specify the Local Site Backup Copy.
Recovery Site Primary Copy	Lets you specify the Recovery Site Primary Copy.
Recovery Site Backup Copy	Lets you specify the Recovery Site Backup Copy.

Reorganize Tablespace Wizard - Panel 7

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

Option	Description
Off Pos Limit	The specified value is compared to the result of the calculation (NEAROFFPOSF + FAROFFPOSF) × 100 / CARDF for the specified partitions in SYSIBM.SYSINDEXPART for the explicit clustering indexes for every table in the specified table space. Alternatively, the values in SYSINDEXPART are checked for a single non-partitioned table space, or for each partition if you specified an entire partitioned table space as the target object. If at least one calculated value exceeds the OFFPOSLIMIT value, REORG is performed or recommended. NOTE: This option is valid for non-LOB table spaces only.
Ind Ref Limit	The specified value is compared to the result of the calculation (NEARINDREF + FARINDREF) × 100 / CARDF for the specified partitions in SYSIBM.SYSTABLEPART for the specified table space. Alternatively, the values in SYSTABLEPART are checked for a single non-partitioned table space, or for each partition if you specified an entire partitioned table space as the target object. If at least one calculated value exceeds the INDREFLIMIT value, REORG is performed or recommended. NOTE: This option is valid for non-LOB table spaces only.
Report Only	The reorganization is only recommended, not performed.
Unload	Specifies whether the utility job is to continue processing or end after the data is unloaded. Continue - Specifies that, after the data has been unloaded, the utility continues processing. Pause - Specifies that after the data has been unloaded, processing ends. Only - Specifies that after the data has been unloaded, the utility job ends and the status in SYSIBM.SYSUTIL corresponding to this utility ID is removed. External - Specifies that after the data has been unloaded, the utility job is to end and the status in SYSIBM.SYSUTIL corresponding to this utility ID is removed.

Reorganize Tablespace Wizard - Panel 6

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

Option	Description
Do you want to keep the dictionary?	A collection of language-related linguistic information used during text analysis, indexing, retrieval, and highlighting of documents in a particular language. NOTE: Available only if Continue or Pause option was specified for Unload on the previous panel.
Do you want to specify statistics?	Lets you include statistics for the tablespace in the SQL statement. Enabled only if Continue or Pause option was specified for Unload on the previous panel.
Do you want to output message to SYSPRINT?	Lets you specify to output message to SYSPRINT.
Do you want to force aggregation or rollup processing to be done even though some parts do not contain data?	Lets you specify to process with forced aggregation, gathered into a mass, or rollup.

Reorganize Tablespace Wizard - Statistics Tables

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

Option	Description
Select Tables	Lets you select tables (and columns) to collect statistics for. The sample column is editable and you can type an integer value into this column.
Add	Opens the Table Selector Dialog Box.
Set Properties	Lets you customize the selection of columns in the selected tables.

Reorganize Tablespace Wizard - Statistics Indexes

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

Option	Description
Select Index	Lets you specify the indexes and the correlation values for collecting the statistics. The sample column is editable and you can type an integer value into this column.
Add	Opens the Index Selector Dialog Box.
Set Properties	Lets you specify the correlation information for each index.

Reorganize Tablespace Wizard - Panel 7

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

Option	Description
Punch DDN	Lets you select punch DDN from the list. Contains generated LOAD statements that let you reload the discarded data. Required if Unload External option is chosen on Panel 5.
Discard DDN	Lets you select discard DDN from the list.
Unload DDN	Lets you select DD name of the unload data set. Required unless NOSYSREC on Panel 2 or Share level CHANGE Panel 3 is selected.
Work DDN	Lets you select the DD statement for a temporary data set used for intermediate output.

Reorganize Tablespace Wizard - Panel 7

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

Option	Description
Discard	Lets you select discard DDN from the list.
No Pad	No a character is added to fill empty space.
Add	Opens the Table Selector Dialog Box.
Set Condition	Lets you customize the selection of columns in the selected tables.

Reorganize Tablespace Wizard - Panel 8

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard

Option	Description
Would you like to specify the device type?	Lets you specify the device type.
Do you want to specify the sort number?	Lets you type the number of datasets to be dynamically allocated by the sort program. The range of the text field is 0 to 2147483647.
Preformat	Lets you specify that the remaining pages are preformatted up to the high allocated RBA in the table space and index spaces associated with the table specified in table-name. The preformatting occurs after the data has been loaded and the indexes are built.

Reorganize Tablespace Wizard - Panel 9

The table below describes the options and functionality on this panel of the Reorganize Tablespace Wizard:

Option	Description
Punch DDN	Lets you select punch DDN from the list. Contains generated LOAD statements that let you reload the discarded data. Required if Unload External option is chosen on Panel 5.
Discard DDN	Lets you select discard DDN from the list.
Unload DDN	Lets you select DD name of the unload data set. Required unless NOSYSREC on Panel 2 or Share level CHANGE Panel 3 is selected.
Work DDN	Lets you select the DD statement for a temporary data set used for intermediate output.

Condition Dialog Box

The **Condition** dialog lets you type free form condition text.

Note: The correctness of the condition text is not tested.

Repair Tables

This action builds and submits a REPAIR TABLE statement, letting you repair a corrupted table. This operation returns a standard REPAIR TABLE result set. Each returned message consists of **Table** (name), **Op** (with a value of REPAIR), **Msg_type** (STATUS, ERROR, INFO, WARNING), and **Msg_text** components.

Platform Availability

- MySQL

To Repair a Possibly Corrupted Table

1 On the Datasource Navigator/Explorer, expand nodes until the **Tables** node is visible.

2 In DBArtisan, if using the Datasource Explorer select the **Tables** node or if using the Datasource Navigator expand that node.

4 Right-click one or more selected tables and select **Repair Tables** from the context menu.

The **Repair Tables** dialog opens.

5 Use the following table as a guide to understanding and modifying settings in this wizard:

Step	Settings and tasks	
Action Options	**Quick**	Only the index tree is fixed.
	Extended	Creates the index row by row instead of creating one index at a time with sorting.
	Use FRM file	Recreates the `.MYI` file header using information in the `.frm` file.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.	
Preview	Displays the DDL that will execute the object action. For more information, see Preview.	

6 Click **Execute**. For information on the scheduling option, see Scheduling.

The results set opens on the **Results** tab of an ISQL editor window. For more information, see Using the Results Editor.

Report

DBArtisan offers two basic object reporting options:

- You can generate a simple summary of all objects of a given type for a single database. For each object, the report contains similar information to that shown in the Datasource Explorer for that object type. The report opens in a built-in HTML browser. You can find data in the report, save it, or print the report.
- You can generate a detailed report of one or more objects of a given type for a single database. For each object, the report contains information similar to that available in the object editor for that object type. In addition to opening in the built-in HTML browser, the report file is written to an .htm file.

Platform Availability

- DB2 LUW, DB2 z/OS, MySQL, ORCL, SQL SVR, SYB ASE

To Generate a Summary Report on all Objects of a Selected Type in DBArtisan

1 On the Datasource Explorer, select a node corresponding to the object type you want to report on or on the Datasource Navigator expand that node.

2 Right-click an object, and then select **Report** from the context menu.

The **Generate Report** dialog opens.

3 Select the **Summary Report** option and click **OK**.

To Generate a Detail Report on One or More Objects

1 In DBArtisan, on the Datasource Navigator/Explorer, select the target object node.

2 Right-click one or more selected objects, and then select **Report** from the context menu.

3 If using DBArtisan, select the **Detail Report** option and click **OK**.

The **Report** dialog opens.

4 In **Report Home Page File Name**, type the report name or click **Browse** to locate the report.

5 Use the **Append date and time stamp to report file name** and **Save report in date specific folders** controls to ensure that the disk files are not overwritten each time you generate reports against the specified file name.

6 In **Report Title**, type the title that is to appear in the report.

7 If you selected multiple objects, use the **Report Options** controls to specify a **Single HTML File for All Objects** or **Separate HTML Files for Each Object**.

8 Click **Execute**.

The report opens in the built-in browser and writes a copy to disk.

Resize (DBArtisan)

The **Resize** dialog lets you dynamically increase the size of database devices without having to add and configure additional database devices.

Platform Availability

- SYB ASE 12.5.01 ^

To Resize a Database Device by a Specified Number of Pages

1 Initiate a **Resize** action against one or more database devices. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Resize** wizard:

Step	Settings and tasks	
Action Options	**Measurement Unit** and **Size**	Lets you specify the amount, in pages, kilobytes, megabytes, gigabytes, or terabytes, by which to increase the device size.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation. For more information, see Dependencies.	
Preview	Preview the DDL generated for the operation and when ready, use the **Schedule** or **Execute** button to perform this action. For more information,see Preview.	

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Restart (Object Action)

The **Restart Sequence** dialog lets you restart a sequence, starting at the value specified in the sequence definition or starting at a specified value.

Platform Availability

- DB2 LUW

To Rebuild a Stored Outline

1 Initiate a **Restart** action against a sequence object. For more information see Initiating an object operation.

The **Restart Sequence** dialog opens.

2 Select one of the following **Restart Sequenceat** options:

- **Originating Value** use the value specified in the object definition
- **This value** lets you specify a custom start value

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Restart Sequence(s)

This action builds and submits an ALTER SEQUENCE... RESTART/RESTART WITH statement. This lets you reset the current value to either the initial value specified when the sequence was created, or reset it to a specified value.

Platform Availability

- SQL SVR

To reset the value of a sequence

1 Initiate a **Restart Sequence(s)** action against one or more sequences. For more information, see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks	
Action options	**Start With**	An integer that will be returned as the next value of the sequence object. The integer must be within the range specified for the sequence. If no value is provided, sequence numbering restarts with the initial value specified when the sequence was created.
Dependencies	Lets you review the objects potentially impacted by this action. For more information, see Dependencies.	
Preview	Preview the DDL generated for the operation. For more information, see Preview.	

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Resynchronize

This action executes an ALTER DATABASE... FOR PROXY_UPDATE statement, forcing resynchronization of proxy tables within the proxy database.

Note: This functionality is available for Sybase ASE 12.5.

To manually resynchronize proxy tables

1 Initiate a **Resynchronize** action against one or more databases. For more information, see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks
Action options	Lists the database or databases you selected to resynchronize.
Preview	Lets you preview the DDL generated for the operation.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Role Activation

You can activate or deactivate roles for the current session. The **Role Activation** action builds and submits one or more SET ROLE commands.

Platform Availability

- ORCL, SYB ASE

To Enable or Disable One or More Roles in the Current Session

1 Right-click on a connected Sybase ASE or Oracle datasource node and select **Role Activation** from the context menu. The **Role Activation** dialog opens.

2 Use the following table as a guide to understanding and modifying settings in the wizard:

Panel	Settings and tasks
Role Activation	The list contains entries for each role that has been granted to the current login. For information on granting/assigning roles to a login or user, see Users Editor (Oracle) and Logins Editor (Sybase ASE). To activate a role, select the **Active** check box associated with the role. If the role is configured to require a password, provide it in the **Password** box. To deactivate a role, deselect the **Active** check box associated with the role.
Preview	Preview the DDL generated for the operation and when ready, click **Execute**.

3 Click **Execute**.

Related Information

- Role Wizard (DBArtisan - Sybase ASE)

Rotate Log (DBArtisan)

This action signals the log-file manager to immediately switch to use of a new output file.

Platform Availability

PSTGRS

To Drop an Object

1 Initiate a **Rotate Log** action against the Logs node of a PostgreSQL datasource node. For more information see Initiating an object operation.

2 When prompted to verify, click **Yes**.

Run Job (Oracle Jobs)

This action builds and submits a DBMS_SCHEDULER.RUN_JOB call, letting you run a job immediately.

Platform Availability

- ORCL 10g ^

To run a job immediately

1 In DBArtisan, on a connected Oracle datasource, expand the **Instance** and **Scheduler** nodes and select the **Jobs** node.

2 Right-click a job and select **Run Job** from the context menu.

3 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks
Action options	Select the **Use current session** check box to provide a USE_CURRENT_SESSION attribute value of TRUE. Deselect the **Use current session** check box to provide a use_current_session attribute value of FALSE.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

4 Click **Execute**. For information on the scheduling option, see Scheduling.

Run Job (job queues)

You can immediately run any defined job queue, regardless of its current schedule. Job Queues are built-in mechanisms that let you schedule a variety of SQL-based or command-line driven tasks.

Platform Availability

- ORCL

To Run a Job

1 Initiate a **Run Job** action against a job queue. For more information see Initiating an object operation.

The **Run** dialog opens.

2 Click **Execute**. For information on the scheduling option, see Scheduling.

Related Information

- Enable Job (Job Queue)
- Disable Job.

Runstats Index (DBArtisan)

The Runstats Index Wizard gathers statistics on indexes. If no statistics are available, DB2 makes fixed default assumptions which may result in inefficiencies.

To ensure the effectiveness of the paths selected, you can run the wizard:

- After a table is loaded.
- After an index is physically created.
- After running **Rebuild Index** or **Reorg Index** without collecting statistics.

For more information, see Rebuild Index (DBArtisan - DB2 z/OS) and Reorg Index (DB2 z/OS).

- Before running **Reorg Index** with the OFFPOSLIMIT, INDREFLIMIT, or LEAFDISTLIMIT options. For more information, see Reorg Index (DB2 z/OS).

Platform Availability

- DB2 z/OS

To Gather Statistics on an Index

1 On the Datasource Navigator/Explorer, expand the **Tablespaces** or **Indexes** node.

DBArtisan displays objects of the selected type.

2 Right-click the target tablespace and select **RunStats Index** from the context menu.

DBArtisan opens the **RunStats Index Utility** wizard.

3 Use the following table as a guide to understanding and modifying settings in the wizard:

Step		**Settings and tasks**
1	**Select Index**	If you initiated the RunStats Index operation against one or more indexes, the target indexes are initially displayed. To check additional Indexes, click the **Add** button.
	Add	Opens the **Indexes Selector** dialog.
	Delete	Deletes selected index.
	Set Properties	Opens the Set Correlation Option Dialog Box for the selected row.
2	**Select tablespaces**	If you initiated the RunStats Index operation against one or more tablespaces, the target tablespaces are initially displayed. To check additional tablespaces, click the **Add** button.
	Add	Opens the Object Selector Dialog Box.
	Delete	Deletes selected tablespace.

	Set Properties	Opens the Set Correlation Option Dialog Box for the selected row.
3	**Access level while RUNSTATS is running**	Lets you select an access level, either Readonly or Change.
	Do you want to output message to SYSPRINT?	Lets you specify to output message to SYSPRINT.
	Do you want to force aggregation or rollup processing to be done even though some parts do not contain data?	Lets you specify to process with forced aggregation, gathered into a mass, or rollup.
4	**Update Statistics**	Lets you update the catalog statistics.
	History Statistics	Lets you update the history statistics.

4 When ready, click **Finish**.

Set Correlation Option Dialog Box

The table below describes the options and functionality on the **Set Correlation Option** dialog:

Option	Description
KeyCard	Collects all of the distinct values in all of the 1 to n key column combinations for the specified indexes. n is the number of columns in the index.
Frequent Item	Lets you add items to the grid.
Add	Opens the Add Frequent Item Dialog Box

Add Frequent Item Dialog Box

The table below describes the options and functionality on the **Add Frequent Item** dialog:

Option	Description
Number of Columns	Lets you type the number of columns. Default is 1.
Count	Lets you type the count. Default is 10.

Runstats Tablespace (DBArtisan)

The Runstats Tablespace Wizard gathers statistics on tablespaces, indexes, and columns. This information is recorded in the DB2 catalog and uses it to select access paths to data during the bind process. The wizard lets you evaluate database design and helps you decide to reorganize tablespaces or indexes.

Platform Availability

- DB2 z/OS

To Perform a RunStats Operation Against a Tablespace

1 On the Datasource Explorer, expand the **Tablespaces** node or in the Datasource Navigator expand that node.

Objects of the selected type are displayed.

2 Right-click the target tablespace and select **RunStats Tablespace** from the context menu.

The **RunStats Tablespace Utility** wizard opens.

3 Use the following table as a guide to understanding and modifying settings in the wizard:

Step	Settings and tasks	
Preliminary options	**Database Name**	Lets you select a database.
	Tablespace Name	Lets you select a tablespace.
	Partition Number	Lets you specify on which the command needs to be run.
	Do you want to runstats on all tables in this tablespace?	Default is Yes.
	Do you want to runstats on all indexes in this tablespace?	Default is Yes.
Table sample and key card details (Panel is displays only if you selected **Yes** to either of the **all indexes...** or **all tables...** options on the first panel)	**Table Sample**	Lets you type a table sample. Default is 25. You can enter any value between 1 and 100.
	KeyCard	Collects all of the distinct values in all of the 1 to n key column combinations for the specified indexes. n is the number of columns in the index.
	Add	Opens the Add Frequent Item Dialog Box
	Delete	Deletes selected item(s.)

Table selection (Panel is displays only if you selected **No** to the **all tables...** options on the first panel)	**Select Tables**	Lets you select the tables in the table that runstats will be run on. The Sample column is blank initially for each row. It is editable with values between 1 - 100. The Indicator column is blank initially for each new row that is added. However if you set the custom columns for a row (via set properties button), this column shows an asterisk (*).
	Add	Opens the Table Selector Dialog Box.
	Delete	Deletes the selected table(s.)
	Set Properties	Lets you customize the selection of columns in the selected table(s.)
Index selection (Panel is displays only if you selected **No** to the **all indexes...** options on the first panel)	**Select Indexes**	Lets you select index(es) for the runstats operation.
	Add	Opens the Select Column for Table Dialog Box.
	Delete	Deletes selected index(es).
	Set Properties	Opens the Set Correlation Option Dialog Box.
Runstats options	**Access level while RUNSTATS is running**	Lets you select an access level, either Readonly or Change.
	Do you want to output message to SYSPRINT?	Lets you specify to output message to SYSPRINT.
	Do you want to force aggregation or rollup processing to be done even though some parts do not contain data?	Lets you specify to process with forced aggregation, gathered into a mass, or rollup.
Statistics	**Update Statistics**	Lets you update the catalog statistics.
	History Statistics	Lets you update the history statistics.

4 When ready, click **Finish**.

Select Column for Table Dialog Box

The table below describes the options and functionality on the **Select Column for Table** dialog:

Option	Description
Column Name	Displays the column name.
Included	Select to include column.
Select All	Click to select all columns.
Unselect All	Click to deselect all columns.

Table Selector Dialog Box

The table below describes the options and functionality on the panel of the **Table Selector** dialog

Option	Description
Tablespace Like	Displays the tablespace.
Table Creator	Displays the table(s) creator of the tablespace that was selected on Panel 2.
Table Name	Displays the table(s) of the tablespace that was selected on Panel 2.
All Columns	Selects all columns.
Table Selector	Lets you select one or more rows per table, up to 10.
Query	Repopulates Table Selector based on the tablespace query.
Select All	Selects all rows in Table Selector.
Unselect All	Unselects all rows in Table Selector.
Add	Click to add the table to the wizard.

Index Selector Dialog Box

The table below describes the options and functionality on the panel of the **Index Selector** dialog

Option	Description
Index Creator	Displays the index(es) creator of the tablespace that was selected on Panel 2.
Index	Displays the index(es) of the tablespace that was selected on Panel 2.
Index Selector	Lets you select one or more rows in this list.
Query	Repopulates Index Selector based on the tablespace query.
Select All	Selects all rows in Index Selector.
Unselect All	Unselects all rows in Index Selector.
Add	Click to add the index to the wizard.

Schema (Object Action)

The **Schema** dialog lets you view key properties of the columns in a table or view.

Platform Availability

- DB2 LUW, DB2 z/OS, MySQL, ORCL, SQL SVR, SYB ASE

To View a Column Summary for a Table or View

1 Initiate a **Schema** action against a table or view. For more information see Initiating an object operation.

Select * From

This action lets you retrieve all data contained in tables or views. Results are displayed in a row and column grid.

Platform Availability

- DB2 LUW, DB2 z/OS, MySQL, ORCL, SQL SVR, SYB ASE

To Select All Data in One or More Tables or Views

1 Initiate a **Select * From** action against one or more tables or views. For more information see Initiating an object operation.

For each selected table or view, a two-tabbed window opens:

- The **Results** tab displays the data selected from the table, in a grid similar to a spreadsheet. Right-click and toolbar options offer many of the functions provided in the Results editor. For more information, see Using the Results Editor.
- The **Query** tab displays the full query generated for the operation. Right-click and toolbar options offer many of the functions provided in the ISQL editor. For more information, see Using the SQL Editor.

Set Default

This function lets you set a tablespace as the default tablespace. Users created without a specified default tablespace will be assigned this default tablespace. If no default tablespace is set, users created without a specified tablespace will have their default tablespace set to SYSTEM.

Platform Availability

- ORCL 10g ^

To Set a Tablespace as the Default

1 Initiate a **Set Default** action against a tablespace. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Set Default** wizard:

Step	Settings and tasks
Action options	Verify that the panel displays the tablespace to be set as default.
Dependencies	Review the referring and referred objects that will be automatically resolved when you execute this operation. For more information, see Dependencies.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

Set Integrity

DB2 SET INTEGRITY statement support lets you perform tasks such as taking tables into and out of set integrity pending state, placing tables into full access state, and pruning the contents of one or more staging tables.

Note: This operation should not be performed without an in-depth understanding of the SET INTEGRITY statement. For online access to DB2 documentation, see Accessing Third Party Documentation.

Platform Availability

- DB2 LUW

To Open the Set Integrity Wizard

1 On the Navigator/Explorer, expand a DB2 database, and click or expand the **Table** node.

2 Select the appropriate table(s), right-click, and choose Set Integrity from the menu. You can select more than one table by clicking CTRL + the tables you want.

OR

Select the appropriate table, click **Command** on the toolbar, and then choose **Set Integrity** from the drop-down menu.

The table below lists all fields you can see in the Set Integrity Wizard. Depending on the options you choose, you may not see them all.

Required Field	Description
Tables	
Tables	The list of one or more tables you selected appear.
Integrity Option	
Integrity Option	**OFF**--When selected, tables have their foreign key and check constraints, and column generation disabled and so are put in a pending state. This also means materialized query or staging tables are not immediately refreshed and are put in a pending state. **TO DATALINK RECONCILE PENDING**--When selected, DATALINK integrity is disabled and tables are put in a check pending no access state. Dependent and descendant tables are not affected. **IMMEDIATE CHECKED**--This turns on a table's integrity checking turned on. Any checking that was deferred is carried out. **FULL ACCESS**--Tables become fully accessible as soon as the SET INTEGRITY statement executes. **PRUNE**--This is appropriate only for staging tables. The content of the staging table is pruned and set to an inconsistent state. If any table in the table-name list is not a staging table, an error is returned. **UNCHECKED**--Allows you turn on some or all integrity checking but the table will not be checked for integrity violations. This can affect data integrity.
Table Readability/Cascade/Descendent Types	
Specifies the readability of the table while it is in check pending state:	**NO ACCESS**--The table(s) are put in a check pending no access state so read/write access to the table is prohibited. **READ ACCESS**--The table(s) are put in a check pending read state. This allows read access to the non-appended portions of any tables.
Specifies whether to be immediately cascaded to all descendents	**CASCADE IMMEDIATE**--The check pending state for foreign key constraints is immediately extended to descendant foreign key constraints or to materialized query or staging tables. **CASCADE DEFERRED**--Only the selected tables are put in the check pending state. Descendant foreign key, materialized query, or staging tables remain unchanged.
Descendent Types	**Materialized Query Tables**--When selected, the check pending state is immediately cascaded to only descendant materialized query tables. **Foreign Key Tables**--When selected, the check pending state is cascaded immediately only to descendant foreign key tables. **Staging Tables**--When selected, the check pending state is cascaded immediately only to descendant staging tables.
Check Appended Portion?	
Do you want to check on the appended portion (if any) of the table?	**Default/Yes/NoForce Generated**--If you do not specify this generated column current values will be compared to the computed value of the expression as if an equality check constraint existed. If this is specified, generated columns are computed only for the appended portion. **Prune**--Possible only for staging tables. When you check this, the contents of the staging table are pruned and the staging table is set to an inconsistent state. **Full Access**--When selected, tables will become accessible after the SET INTEGRITY statement executes.

Specify Exception Tables

List of Base Tables	Any row that is in violation of a foreign key or check constraint is copied to the exception table you select.

Integrity Options: IMMEDIATE UNCHECKED

IMMEDIATE UNCHECKED options	**Foreign Key**--These constraints are turned on when the table is removed from check pending status. **Check**--Check constraints are turned on when the table is removed from check pending status. **Datalink Reconcile Pending**--DATALINK integrity constraints are turned on when the table is removed from check pending status. **Materialized Query**--Immediate refreshing is turned on for a materialized query table when it is removed from a check pending state. **Generated Column**--When the table is removed from check pending status, generated columns are turned on. **Staging**--Immediate propagation is turned on for a staging table.
Do you want tables to become fully accessible after the SET INTEGRITY statement executes?	**Yes/No**

3 When ready, click the **Finish** button to preview and submit the generated DDL. For more information, see Preview.

Set Online/Offline

The **Set Database(s) Online/Offline** dialog lets you disable your databases to prevent access, and enable your databases to grant access through the **Datasource** menu.

Platform Availability

- SQL SVR, SYB ASE

Important Notes

For Sybase ASE, you can only set databases online.

To Set One or More Databases Online or Offline

1 Initiate a **Set Online/Offline** action against one or more databases. For more information see Initiating an object operation.

2 Use the following table as a guide to working through the panels of the **Set Online/Offline** dialog

Step	Settings and tasks	
Action options	**Set offline** (SQL Server only)	Set this to TRUE to set the database offline or set it to True to set the database online.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.	
Preview	Displays the DDL that will execute the object action. For details, see Preview.	

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Set UNDO

Set UNDO Tablespace dialog lets you dynamically set an UNDO tablespace if the tablespace is running in AUTO UNDO mode.

Platform Availability

- ORCL 9i ^

To Dynamically Set an UNDO Tablespace

1 Initiate a **Set Undo** action against a tablespace. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying the settings on the **Set Undo Tablespace** wizard:

Step	Functionality
Action Options	Displays the tablespace you selected to be set as an Undo tablespace.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Preview the DDL generated from your choices. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Shrink

For details on using Shrink operations to save or reclaim space, see the following topics:

- Shrinking SQL Server databases
- Shrinking Oracle rollback segments
- Shrinking Oracle tables or indexes

Shrinking SQL Server databases

You can reclaim space from a database that is too large.

To Shrink a SQL Server Database

1 Initiate a **Shrink** action against one or more databases. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying the settings on the **Shrink** wizard:

Step	Description	
Action Options	Displays the databases you selected and lets you work with the following settings:	
	Move data pages to beginning of file before shrink	Select to move data pages to the beginning of the file before the shrink.
	Release All Unused Space	Deselect to set the target free space to retain, and then in the Target Free Space to Retain (percent) box, type the new value of free space to retain. The new size for the database must be at least as large as the Minimum Allowable Size displayed in the Current File Size box.
	Target free space percent after shrink	Lets you specify the target free space percent after the shrink.
Preview	Preview the DDL generated from your choices. For more information, see Preview.	

3 Click **Execute**. For information on the other options, see Preview and Scheduling.

Shrinking Oracle rollback segments

You can shrink the size of rollback segments. The proper sizing of rollback segments is critical to their overall performance. Performance degrades whenever a rollback segment must extend, wrap or shrink in response to transaction loads. Ideally, you want to make the extents of rollback segments as small as possible while still ensuring that each transaction can fit into a single extent.

After an abnormally large transaction load, you might consider shrinking a rollback segment to eliminate unnecessary space. Oracle lets you shrink a rollback segment manually by a specific amount or back to its Optimal Size.

Important Notes

For Oracle 9 or later, Shrink is not available if auto-UNDO management is enabled.

To Shrink the Size of One or More Rollback Segments

1 Initiate a **Shrink** action against one or more rollback segments. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying the settings on the **Shrink** wizard:

Step	Description
Action Options	Displays the rollback segments you selected and lets you work with the following settings: **Specify the size...** If you do not provide a specific number the Rollback Segment uses the OPTIMAL value specified in the Storage clause. If an OPTIMAL value is not specified, the size defaults to the MINEXTENTS value of the Storage clause.
Preview	Preview the DDL generated from your choices. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Shrinking Oracle tables or indexes

You can shrink the size of tables or indexes.

Important Notes

Shrink is only available for tables in which the **Row Movement** property is enabled. For information on setting the **Row Movement** when creating or editing a table, see Tables Wizard (Oracle) and Tables Editor (Oracle).

To Shrink the Size of One or More Indexes or Tables

1 Initiate a **Shrink** action against one or more indexes or tables. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying the settings on the **Shrink** wizard:

Step	Description
Action Options	Displays the objects you selected and lets you work with the following settings: **Compact** Enable **Compact** to restrict shrinking to defragmenting the segment space and compact rows. **Cascade** Enable **Cascade** to simultaneously shrink all dependent objects.

Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Preview the DDL generated from your choices. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Start Database

The **Start Database** dialog lets you start a database:

- When the database that has been stopped with a **Stop Database** dialog. For more information, see Stop Database.
- After a tablespace, partition, or index has been placed in group buffer pool RECOVER pending status (GRECP) or if pages have been put on the logical page list (LPL) for that object.

Depending on the specified options, the database can be made available for read-only processing, read-write processing, or utility-only processing. In a data sharing environment, the command can be issued from any DB2 on the group that has access to the database.

Platform Availability

- DB2 z/OS

To Start a Database

1 In DBArtisan, Datasource Explorer, select the **Databases** or **Tablespaces** node or in the Datasource Navigator expand one of those nodes.

2 Right-click the target object and select **Start Database** from the context menu.

The **Start Database** dialog opens.

3 Use the following table as a guide to understanding and settings options on this dialog:

Option	Description
Database Grid	The Partition column is editable and can include one or more unique numeric values separated by commas. Initially the Partition column is blank. No validation is made for the correctness of partition numbers so make sure that the partitions exist.
Access	Lets you select access options of **Read/Write**, **Read only**, **Utility** only or **Force**.

4 Click **Execute**. For information on the scheduling option, see Scheduling.

Stop Database

The **Stop Database** dialog lets you stop a database, optionally letting you allow running applications access until their next COMMIT.

Platform Availability

- DB2 z/OS

To Stop a Database

1 On the DBArtisan Datasource Explorer, select the **Databases** or **Tablespaces** node or in the Datasource Navigator expand one of those nodes.

2 Right-click the target object and select Stop Database from the context menu.

The **Stop Database** dialog opens.

3 Use the following table as a guide to understanding and settings options on this dialog:

Option	Functionality
Database Grid	The Partition column is editable and can include one or more unique numeric values separated by commas. Initially the Partition column is blank. No validation is made for the correctness of partition numbers so make sure that the partitions exist.
Stop at COMMIT	Enabling this option sends the STOP DATABASE statement.with an AT(COMMIT) clause, allowing running applications to continue access until their next commit.

4 Click **Execute**. For information on the scheduling option, see Scheduling.

Start/Stop Job (DBArtisan - SQL Agent Job)

This action builds and submits an **sp_start_job** or **sp_stop_job** call for one or more SQL Agent jobs. It lets you immediately start or halt execution of a job.

Platform Availability

- SQL SVR

To Immediately Start or Halt Execution of One or More Jobs

1 Initiate a **Start/Stop** action against one or more jobs. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Start/Stop jobs** wizard:

Step	Settings and tasks
Action Options	Select the **Start** check box to start the selected jobs. Deselect the **Start** check box to stop the selected jobs.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Stop Job

This action builds and submits a DBMS_SCHEDULER.STOP_JOB call, letting you stop a job immediately.

Platform Availability

- ORCL 10g ^

To stop a job

1 In DBArtisan, on a connected Oracle datasource, expand the **Instance** and **Scheduler** nodes and select the **Jobs** node.

2 Right-click a job and select **Stop Job** from the context menu.

3 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks
Action options	Select the **Force Stop** check box to provide a FORCE attribute value of TRUE forcing the job to terminate immediately. Deselect the **Use current session** check box to provide a FORCE attribute value of FALSE.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

4 Click **Execute**. For information on the scheduling option, see Scheduling.

Switch Logfile (DBArtisan)

The Switch Logfile action issues an ALTER SYSTEM SWITCH LOGFILE statement. This forces a log switch to make the currently active group inactive and available for maintenance activities.

Platform Availability

- ORCL

To Force a Log Switch

1 Select or expand the Storage node and select the **Redo Log Groups** node

2 Right-click one or more selected objects and select Switch Logfile from the context menu.

The **Preview: Switch Logfile** dialog opens.

3 Click **Execute**.

Related Information

- Take Checkpoint (DBArtisan)

Switch Online

The **Switch Online** dialog lets you access a tablespace by switching it online after the parent container(s) have been recovered or restored.

Platform Availability

- DB2 LUW

To Switch a Tablespace Online:

1 Initiate a **Switch Online** action against one or more tablespaces. For more information see Initiating an object operation.

The **Switch Online** dialog opens.

2 Click **Execute**. For information on the scheduling option, see Scheduling.

Take Checkpoint (DBArtisan)

The Take Checkpoint action issues an ALTER SYSTEM CHECKPOINT statement. This forces a fast database checkpoint.

Platform Availability

- ORCL

To Force a Fast Database Checkpoint

1 Select or expand the **Storage** node and select the **Redo Log Groups** node

2 Right-click one or more selected redo log groups and select Take Checkpoint from the context menu.

The **Preview: Take Checkpoint** dialog opens.

3 Click **Execute**.

Related Information

Switch Logfile (DBArtisan)

Transfer Ownership

An object owner or a user with SECADM authority can transfer ownership of an object to another user. The new user is automatically granted the same privileges as the former owner. Ownership can be transferred on an object-by-object basis or you can transfer all objects currently owned by an individual user.

Platform Availability and object type support

- DB2 LUW 9.1 ^

The following table lists the object types that support ownership transfer:

Aliases	Check Constraints	Event Monitors	Foreign Keys
Functions	Indexes	Primary Keys	Procedures
Materialized Query Tables	Schema	Sequences	Structured Types
Tables	Tablespaces	Triggers	Unique Keys
User Datatypes	Views		

To Transfer Ownership of an Object

1 Initiate a **Transfer Ownership** action against one or more supported objects (see table above). For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Transfer Ownership** wizard:

Step	Settings and tasks
Action options	If you initiated the ownership transfer against a user, select the specific objects belonging to that user that are to be assigned to a new user. Use the **New Owner** dropdown to select a defined user as the new owner of the selected object or objects.
Preview	Preview the DDL generated for the operation. For more information, see Preview.

Truncate

This action lets you quickly delete the rows of one or more tables or clusters. Truncating a table is a faster alternative to deleting all of its rows.

Caution: If you truncate a table, all the rows are deleted. These rows are not logged as individual drops and cannot be recovered from a transaction log or other type of log.

Platform Availability

- DB2 LUW, DB2 z/OS, MySQL, ORCL, SQL SVR, SYB ASE

To Truncate a Table or Cluster

1 Initiate a **Truncate** or **Truncate Table** action against one or more tables or clusters. For more information see Initiating an object operation.

The **Truncate** wizard/dialog box opens.

2 For help with DBMS-specific settings and additional information, see the following topics:

- Notes on truncating Oracle objects
- Notes on truncating Sybase ASE objects
- Notes on truncating IBM DB2 z/OS objects

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Notes on truncating Oracle objects

You can truncate any table or cluster in their schema or, if you have the DROP ANY TABLE system privilege, you can truncate any table in any schema.

Observe the following when truncating tables or clusters:

- Before truncating a cluster or table containing a parent key, disable all referencing foreign keys existing in different tables.
- Truncating a cluster or table does not generate any rollback information and commits immediately.
- Oracle alters the storage parameter NEXT to the size of the last extent deleted from the segment.
- Oracle automatically deletes all data in the table's indexes and any materialized view direct-load INSERT information associated with a truncated table.
- If the table is not empty, all associated nonpartitioned indexes and all partitions of associated global partitioned indexes are marked unusable.
- You cannot truncate a hash cluster nor can you truncate individual tables in a hash cluster or an index cluster.

When you truncate a table or cluster, you can specify whether space currently allocated for the table is returned to the containing tablespace or if it is returned to the system. The table below describes the Truncate dialog options available when truncating an Oracle table or cluster:

Option	Description
Drop Storage	Select if you want the freed extents returned to the system where they can be used by other objects.
Reuse Storage	Select if you want the space to remain allocated to the table or cluster you have just truncated.

Notes on truncating Sybase ASE objects

Truncate against a Sybase ASE table can be applied to the entire table or to a single partition.

Note: You cannot truncate a table referenced by a foreign key constraint. Instead, use a DELETE statement without a WHERE clause.

Tip: When you truncate a table, Sybase ASE removes all rows from the target table, but retains the table structure (its indexes, columns, constraints, etc.). The counter used by an identity for new rows is reset to the seed for the column. To retain the identity counter, use a DELETE statement instead of TRUNCATE. To remove the target table definition and its data, use a DROP TABLE statement.

The following table shows the settings on the Truncate Table dialog:

Option	Description
Partition Name (only available with a single table selected)	Selecting a partition name from the dropdown restricts the truncate operation to a single partition.

Notes on truncating IBM DB2 z/OS objects

When truncating an IBM DB2 z/OS table, the **Action Options** tab offers the following settings:

Setting	Description
Reuse Storage	Corresponds to the DROP STORAGE/REUSE STORAGE clause of a TRUNCATE TABLE statement. This setting specifies whether storage currently allocated to the table is reused or dropped.
Restrict When Delete Triggers	Corresponds to the RESTRICT WHEN DELETE TRIGGERS/IGNORE DELETE TRIGGERS clause of a TRUNCATE TABLE statement. When enabled, an error is returned if triggers are defined for the table. When disabled, triggers defined for the table are not activated by the Truncate operation.
Immediate	Corresponds to the IMMEDIATE clause of a TRUNCATE TABLE statement. If enabled, the truncate operation is executed immediately and cannot be undone. If disabled, a Rollback can undo the Truncate operation.

Unbind From Temporary Database

This action builds and submits a **sp_tempdb unbind** system procedure call. It lets you unbind a login from a temporary database.

Platform Availability

- SYB ASE

To Unbind a Login from a Temporary Database

1 Initiate a **Unbind From Temporary Database** action against one or more logins. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying settings in the **Unbind From Temporary Database** wizard:

Step	Settings and tasks
Action Options	Displays the logins you selected for unbinding.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Undrop Disk (DBArtisan)

This action builds and submits an ALTER DISKGROUP... UNDROP statement. This lets you cancel pending drops of disks in one or all disk groups.

Note: Before this object action is available you must have a direct connection to the ASM instance. For details, see Manage ASM Connection (DBArtisan).

Platform Availability

- ORCL 10g

To Cancel Pending Disk Drops

1 On a connected Oracle datasource, expand the **Instance** and **ASM** nodes.

2 Right-click the **Disk Groups** node and select **Undrop Disk** from the context menu.

3 Use the following table as a guide to understanding and modifying settings in the dialog:

Step	Settings and tasks
Action options	**Undrop all** - If unselected, this action applies only to the disk group you initiated the action against. If selected, the action is applied to disks in all disk groups on this ASM instance.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Displays the DDL that will execute the object action. For details, see Preview.

4 Click **Execute**. For information on the scheduling option, see Scheduling.

Unmirror (DBArtisan)

The Unmirror operation lets you execute a DISK UNMIRROR command against a database device, suspending disk mirroring. Options include specifying the primary or secondary device, and whether unmirroring is permanent or temporary.

Platform Availability

- SYB ASE

To Suspend Mirroring for a Database Device

1 Initiate an **Unmirror** action against a database device. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying the settings on the **Unmirror** wizard:

Step	Description
Action Options	Displays the database device you selected and lets you work with the following settings: **Side** Specifies whether to disable the PRIMARY device or the SECONDARY, or mirror, device. **Mode** Specifies RETAIN (unmirroring is temporary) or REMOVE (unmirroring is permanent).
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.
Preview	Preview the DDL generated from your choices. For more information, see Preview.

3 Click **Execute**. For information on the scheduling option, see Scheduling.

Unquiesce

The Unquiesce operation lets you restore user access to a single database or all databases of an instance previously quiesced.

Platform Availability

- DB2 LUW

Note: RUnquiesce Database for supported servers from IBM DB2 for Windows/Unix 7 or earlier clients is not supported.

To Restore User Access to a Database

1 Take one of the following actions, based on whether you are unquiescing a single database or all databases on an instance:

- Right-click a datasource node and select **Command > Unquiesce** from the context menu.
- Right-click the Instance node, select **Command > Unquiesce** from the context menu, and provide credentials if prompted.

The **Unquiesce Database** or **Unquiesce Database** dialog opens.

2 Click **Execute**.

Related Information

- Quiesce

Update Statistics

The **Update Statistics** dialog lets you update the statistics for an active table or index. As indexes grow and shrink in response to data modification, the accuracy of their statistics can deteriorate.

The following topics provide details on updating statistics by supported DBMS and object type:

- Updating statistics for tables or indexes (IBM DB2 for Linux, Unix, and Windows version 8.2.2)
- Updating statistics for views (IBM DB2 for Windows, Unix, and Linux)
- Updating statistics for databases, indexes, or tables (Microsoft SQL Server)
- Updating statistics for indexes or tables (Sybase ASE)

Updating statistics for tables or indexes (IBM DB2 for Linux, Unix, and Windows version 8.2.2)

To Update Statistics for an Index or Table

1 Initiate an **Update Statistics** action against one or more tables or indexes. For more information see Initiating an object operation.

The **Update Statistics** dialog opens.

2 Use the following table as a guide to understanding and modifying the settings in this dialog.

Tab	Options	Description
Table Options	**Update table statistics**	Updates table statistics.
	Distribution Options	**Do not collect column statistics** - Column statistics provide information that the optimizer uses to choose the best access plans for queries. **Collect column statistics on key columns only** - Collects column statistics on columns that make up all the indexes defined on the table. **Collect column statistics on all columns** - Collects column statistics for all columns. Column statistics provide information that the optimizer uses to choose the best access plans for queries. **Frequency** - Lets you specify the maximum number of frequency values to collect, between 1 and 32767. **Quantiles** - Lets you specify the maximum number of distribution quantile values to collect, between 1 and 32767.

	Column Options	**Do not collect distribution statistics** - Does not collect basic statistics or distribution statistics on the columns. **Collect distribution statistics on key columns only** - Collects both basic statistics and distribution statistics on key columns only. **Collect distribution statistics on all columns** - Collects both basic statistics and distribution statistics on all columns. For efficiency both of RUNSTATS and subsequent query-plan analysis, you might collect distribution statistics on only the table columns that queries use in WHERE, GROUP BY, and similar clauses. You might also collect cardinality statistics on combined groups of columns. The optimizer uses such information to detect column correlation when it estimates selectivity for queries that reference the columns in the group. **Exclude XML columns** - Lets you collect statistics on non-XML columns only. XML columns are not included in statistics collection.
Index Options	**Update index statistics**	Lets you enable and disable statistics updates for indexes and controls the following settings:
	Collect extended index statistics	Collects extended index statistics, the CLUSTERFACTOR and PAGE_FETCH_PAIRS statistics that are gathered for relatively large indexes.
	Collect sample statistics	A CPU sampling technique is used when compiling the extended index statistics. If the option is not specified, every entry in the index is examined to compute the extended index statistics.
Access Options	**Allow read only access during collection**	Allows read only access while the statistics are being updated.
	Allow read/write access during collection	Allows read and write access while the statistics are being updated.

3 Click **Execute**. For information on the **Preview** option, see Preview.

Updating statistics for views (IBM DB2 for Windows, Unix, and Linux)

For IBM DB2 for Windows, Unix, and Linux, you can update the statistics for a view.

To Update Statistics for a View

1 Initiate an **Update Statistics** action against one or more views. For more information see Initiating an object operation.

2 Use the following table as a guide to choosing options:

Pane	Options
Action Options	Lets you work with the following settings: **Column Statistics**- lets you enable or disable collection of Column Statistics update, **Distribution Statistics**, **Frequency**, and **Quantiles** - let you enable and disable collection of distribution statistics and provide num_freqvalues and num_quantiles RUNSTATS options **Allow Write Access** - lets you enable or disable write access during statistics collection.
Dependencies	Lets you view referencing or referenced objects. For more information, see Dependencies.
Preview	Lets you view the DDL generated for the operation. For more information, see Preview.

3 **Schedule** or **Execute** the statistics update.

Updating statistics for databases, indexes, or tables (Microsoft SQL Server)

You can update statistics so that Microsoft SQL Server performs the most efficient query possible. This feature updates statistical information on your database so that the query processor can determine the optimal strategy for evaluating a query. These statistics record the key values used for distribution in an database.

You can use the **Update Statistics** dialog if there is significant change in the key values in the database, if a large amount of data in an indexed column has been added, changed, or removed, or if a table has been truncated causing significant changes in the distribution of key values.

The **Update Statistics** dialog lets you specify tables and indexes for validation. This dialog box offers different update options depending on your version of Microsoft SQL Server.

Tip: Avoid updating statistics on your target tables during busy access periods. Microsoft SQL Server locks remote tables and indexes while reading data for update statistics.

For Microsoft SQL Server version 7 or later, the **Update Statistics** dialog lets you specify a full or a percentage of a full scan to be used for updating table or index statistics. It also lets you enable or disable future automatic recomputations of statistics. These recomputations are made at Microsoft SQL Server's discretion. When updating statistics for tables, this dialog box also lets you specify the type of statistics you require.

To Update Statistics for a Database, Index, or Table

1 Initiate an **Update Statistics** action against one or more databases, indexes, or tables. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying the settings in the **Update Statistics** dialog:

Step	Settings and tasks	
Action Options	**Scan Range** (tables and indexes only)	Full Scan - Select when you want index statistics on every available row. Sample Scan - Select when database size prohibits a full scan and you can afford to rely on statistics extrapolated from a sample of all available rows.
	Sample Count (tables and indexes only)	If you specified a **Scan Range** of **Sample Scan**, provide a count.
	Sample Unit (tables and indexes only)	If you specified a **Scan Range** of **Sample Scan**, specify either **%** or **Rows**.
	Statistics Type (tables only)	Index - Select if you only require statistics on the target tables' indexed columns. Columns - Select if you require statistics on the target tables in their entirety. All existing statistics - Select if you require statistics on the whole database.
	Statistics Recompute (tables and indexes only)	Select if you want Microsoft SQL Server to recompute and update the statistics for the index as part of its normal internal maintenance. Deselect if you want the scheduling of future recomputations to be solely your responsibility.
Dependencies (tables and indexes only)	Lets you view referencing or referenced objects.	
Preview	Lets you view the DDL generated for the operation. For more information, see Preview.	

3 Use the **Execute** or **Schedule** button to perform the operation.

Updating statistics for indexes or tables (Sybase ASE)

The **Update Statistics** dialog lets you specify tables and indexes for validation. This dialog box offers different update options depending on your version of Sybase ASE.

Tip: Avoid updating statistics on your target tables during busy access periods. Sybase ASE locks remote tables and indexes while reading data for update statistics.

To Update Statistics for a Database, Index, or Table

1 Initiate an **Update Statistics** action against one or more tables or indexes. For more information see Initiating an object operation.

2 Use the following table as a guide to understanding and modifying the settings in the **Update Statistics** dialog:

Step		Settings and tasks
Action Options	**Index** (tables only)	Enabling this option updates statistics for indexes of the table.
	Table (tables only)	Enabling this option updates table-specific statistics. Column statistics stored in sysstatistiics are not affected.
	Partition Name (available when a single index or table is selected)	The name of the partition to be updated.
	Step Values	Lets you specify the number of histogram steps.
	Consumers	Specifies the number of consumer processes to be used for a sort when a list of columns is provided and parallel query processing is enabled.
	Sampling Percent	Specifies the percentage of the column to be randomly sampled in order to gather statistics.
Columns	(available only when a single table is selected)	Lets you specify all columns or specific columns to be used for the operation.
Dependencies	Lists referring and referenced objects potentially impacted by the change. For details, see Dependencies.	
Preview	Lets you view the DDL generated for the operation. For more information, see Preview.	

3 Use the **Execute** or **Schedule** button to perform the operation.

View Contents (DBArtisan)

The **Backup Device Information** dialog lets you view the contents of a Dump or Backup Device which stores backups of databases and transaction logs.

Platform Availability

- SQL SVR

To View the Content of a Backup or Dump Device:

1 On the Datasource Explorer select the **Backup Devices** node or in the Datasource Navigator select that node.

2 Right-click a backup device and then select **View Contents**.

DBArtisan opens the **Backup Device Information** dialog.

View Job History (DBArtisan)

This action opens a dialog showing the execution history for a job.

Platform Availability

- SQL SVR

To View the Execution History for a Job

1 On the Datasource Navigator/Explorer, expand nodes until the **Server > SQL Server Agent > Jobs** node is visible.

2 Select or expand the **Jobs** node.

The jobs are displayed in the right pane of the Explorer window.

3 Right-click a job and select **View Job History** from the context menu.

The **Job History** dialog opens.

Coding Environments/Editors

DBArtisan provides coding environments that let you create, edit, and where appropriate, execute scripts containing for example, SQL and DDL. These windows are context sensitive to the type of script you are opening or extracting. For example, if you extract the schema for a table, DBArtisan opens a DDL Window containing the script. If you execute a script, a result window displays containing the results of your execution statement.

The following topics describe use of the DBArtisan coding environments:

- Common Word-processing Operations describes common editor options such as file and clipboard operations.
- Common Coding Environment Features describes coding environment options such as commenting and working with line numbers.
- Using the SQL Editor provides details on the SQL Editor.
- Using the DDL Editor provides details on using the SQL Editor to generate object-creation code.

Common Word-processing Operations

The following topics describe features available in all coding environments/editors:

- Clipboard Functions
- File operations
- Find/Replace Options in Coding Environments
- Regular Expressions Support
- Changing Case
- Printing Options

Clipboard Functions

The coding/scripting environments all offer the most common clipboard-related options involving selecting text, cutting, copying, and pasting text. The following topic describes the clipboard option offered:

- Cutting/Copying and Pasting Columns

Cutting/Copying and Pasting Columns

To cut or copy and then paste a column, do the following:

1 In the Editor window, position the pointer in front of the column of the target text.

2 Press ALT and simultaneously drag the pointer over the target column.

3 On the **Edit** menu, select **Copy**s or **Copy**.

4 In the Editor window, position the pointer where you want to paste the column.

5 On the **Edit** menu, select **Paste**.

File operations

The following topics provide key information on file options available in coding environments/editors:

- Opening Files
- Inserting a File Into an Open Script
- Saving and Closing Scripts

Opening Files

All text-based editors offer an **Open File** facility. It provides dedicated support for files of the following types:

- Script (**.sql**), data definition languge (**.ddl**), query (**.qry**), and results (**.rsl**) files
- Query Builder files (**.eqb**) - for more information see Query Builder.
- XML (**.xml**) and text (**.txt**) files
- Web (**.htm**, **.html**, **.asp**) files
- Procedure (**.prc**, **.pro**), VBScript (**.vbs**),
- JScript (**.js**), and Java (**.java**) files

To open a supported-type file

1 Select **File > Open**. An **Open File(s)** dialog opens.

2 Use the dialog to locate, select a file type, and select one or more files to open.

If you select a file type of **All Files**, you can open a file with a suffix not corresponding to one of the supported file types. One of two actions is taken:

- It automatically opens the file in the SQL Editor
- It prompts you for additional information on the file before opening

Note: The specific action taken is dictated by an Options Editor selection. For details, see ISQL Options - Main Tab.

If you selected the Options Editor setting to prompt, when you select an unknown file type for opening, a **What Type of File Is** dialog opens. The table below describes the options and functionality on the dialog:

Option	Description
The file is a general purpose SQL script	Select if the unknown file is a SQL script.
The file contains the DDL to create a database object or the file contains Oracle Anonymous PL/SQL.	Lets you select object type and owner and provide an object name.
Always open unknown files into a SQL window without prompting.	Select to hide What Type of File Is Dialog Box for future unknown file types.

An SQL Editor or DDL Editor opens, as appropriate. For details, see the following topics:

- Using the SQL Editor
- Using the DDL Editor

For more general information, see Coding Environments/Editors.

Inserting a File Into an Open Script

The ISQL Editor facilitates the reuse of SQL scripts by letting you insert an existing file into another script.

To insert a file at the current cursor location

1 In an open editor, place the cursor where you want to insert the file.

2 Select **Edit > Insert File**. Tthe **Insert File into Current File** dialog opens.

3 Use the dialog to locate, select a file type, and select the files to insert.

For more general information, see Coding Environments/Editors.

Saving and Closing Scripts

In DBArtisan, untitled scripts are named according to the following conventions:

Environment	Default File Name	Default File Suffix
SQL Editor	**SQL1**...**SQLn**	**.sql**
DDL Editor	The object name provided when invoking the DDL EDitor	**.sql**

Note: If you have set the **Auto-Save** feature in the Options Editor, a temporary copy of your scripts is automatically saved periodically as you work. For details, see ISQL Options.

To close a script

1 On the on the Main menu, click **Close**.

2 If you have not saved your script, you are prompted to save the file. Click **Yes** to save and **No** to close without saving.

Explicitly Saving a Script

With an ISQL window open, selecting **File > Save** or **File > Save As** opens a standard **Save As** dialog. It lets you save a new script, save an existing script with a new name and optionally with a different file suffix. The following **Encoding** schemes are supported:

ANSI	UNICODE	UTF-16LE (No BOM)
UTF-16 Big Endian	UTF-16BE (No BOM)	UTF-8
UTF-8 (No BOM)	UTF-32 Little Endian	UTF-32LE (No BOM)
UTF-32 Big Endian	UTF-32BE (No BOM)	

Automatic Save Operations

If you shut down the application or close an editor window while there are modified, unsaved scripts, you are prompted to save before completing the operation. The table below describes the options and functionality on the **Save Modified Documents** dialog

Option	Description
Save All	Click to save documents in all open ISQL windows.
Save Selected	Click to save selected documents.
Invert	Click to clear selection.
Save None	Click to not save documents and close the application.

Find/Replace Options in Coding Environments

Each text-based editor offers basic find and find/replace text functions.

To invoke the Find or Find/Replace facility

1 With a text editor active, select **Edit > Find** or **Edit > Replace**.

2 Use the following table as a guide to working with the resulting dialog:

Option	Description
Find What	Lets you type your search string.
Replace With (Replace only)	Lets you type the replacement text.
Match whole word only	Select to search for only the complete word.
Match Case	Select to make the search case-sensitive.
Regular expression	Indicates the the find string contains regular expressions. For more information, see Regular Expressions Support.
Wrap around search	Lets you search from the end of the script and back to the insertion point.
Direction	Lets you specify the direction you want to search. Click the **Up** or **Down** option button.
Find	Click to find the next occurrence of your search string.
Bookmark All (Find only)	Places a bookmark at each line containing the find string. For details, see Working with Coding Environment Bookmarks.
In Selection	Replaces all instance of the find string with the replace string within a selected block of text.
Replace Button (Replace only)	Click replace the current selection.
Replace All Button (Replace only)	Click to automatically find and replace all occurrences of your search string within the current window.

Selecting **Edit > Repeat Find** finds the next instance of the most recently used find string without opening the **Find** or **Replace** dialog.

Regular Expressions Support

Some features, such as file search, SQL Editor search, and Browser list filtering, support the use of regular expressions. This provides flexibility in in searches and other situations in which text string qualification is required.

Detailed documentation on regular expressions is beyond the scope of this document. For descriptions of syntax and metacharacters, consult third-party documentation on the regular expressions standard. The following table provides short descriptions of the most commonly used regular expression metacharacters used with DBArtisan features:

Metacharacter	Description
.	Matches any single character.
*****	Matches zero or more characters.
+	Matches one or more characters.
^	The start of a string.
$	The end of a string.
\b	Word boundary.
\d, \w and \s	Shorthand character classes matching digits, word characters (letters, digits, and underscores), and whitespace (spaces, tabs, and line breaks).
\D, \W and \S	Negated versions of the above that should only be used outside character classes.
[ ]	Matches any one character enclosed between the square brackets.
[^]	Matches any character NOT enclosed between the square brackets.
{}	Matches aspecified number of repetations. For example `ab{2,4}` matches a**bb**, a**bbb**, and **abbbb**.
|	OR.
\	Escape special character.
()	Tag expression.
\s	Space or Tab.

In the case of some specific features, regular expression usage must be enabled or configured. For more information, see the DBArtisan feature topics for details:

- Find/Replace Options in Coding Environments
- Find in Files
- Filtering columns in Database Object Displays (DBArtisan only)

Changing Case

When writing SQL scripts, you can change the letter casing of certain statements or lines of code. You can change case from lowercase to uppercase, or from uppercase to lowercase, using the case functions.

Changing Case

To change case, do the following:

1 Select one or more letters in your script.

2 On the **Edit** menu, select **Upper Case**.

OR

On the **Edit** menu, select **Lower Case**.

Printing Options

A number of standard print options are provided from text-based editors:

- Selecting **File > Print** opens a dialog that lets you print the contents of the editor with options such as choosing a printer, specifying a page range, and specifying the number of copies to print
- Selecting **File > Print Preview** opens a new window showing a printed page appearance for the current document
- Selecting **File > Print Setup** opens a dialog that lets you select a printer, select printing options, and specify preferences

Common Coding Environment Features

The following features are avaiable in all coding environments/editors:

- Working with Comments
- Working with Coding Environment Bookmarks
- GoTo Line x
- Using the Token Matching Capability

Working with Comments

You can comment and uncomment blocks of code using the commenting scheme appropriate to the DBMS platform you are using.

To comment out code

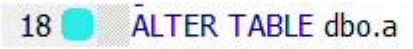
1 Select the block of code you want to comment out and then select **Edit > Comment Out**.

To uncomment code

1 Select a complete block of commented code or a single line of the block and then select **Edit > Undo Comment Out**.

Working with Coding Environment Bookmarks

Bookmarks are valuable navigation aids for jumping from one portion of a script to another. You can add bookmarks in important areas of your scripts, then jump back and forth between bookmarks. A blue dot is inserted in the gutter next to lines that you bookmark.



18 ALTER TABLE dbo.a

To set a bookmark on a line or clear an existing bookmark from a line

1 In the Editor window, position the pointer in front of the line and then select **Edit > Toggle Bookmark**.

To move the cursor back and forth among bookmarked lines

1 Select **Edit > Next Bookmark** or **Edit > Previous Bookmark**.

To clear all bookmarks

1 Select **Edit > Clear Bookmark**.

GoTo Line x

The **Goto** dialog lets you move to a specific line or column in your script.

To complete the **Goto** dialog, do the following:

1 On the **Edit** menu, select **Goto**. The **Go To** dialog opens.

2 In **Line Number**, type or select the target line number.

3 Click **OK**.

The cursor moves to the target line.

Using the Token Matching Capability

When you are working with a large script with multiple levels of embedded steps, compare left and right parentheses, curly braces, square brackets and BEGIN/END pairs to make sure that you have delimited your code properly.

To find matching tokens

1 Position the insertion pointer just to the left of the first token in a token pair you want to locate.

2 Click the **Match Token** button.

{→}

The pointer jumps to the next available token.

Using the SQL Editor

DBArtisan incorporates a powerful SQL scripting environment, the SQL Editor. The SQL Editor lets you write, debug, test and deploy solid SQL code for your database applications.

You can open multiple SQL windows. In addition, when you open an SQL file, it opens in an SQL Editor window. For more information, see Opening Files.

To open a SQL Editor window

1 Connect to the datasource that you want to write a script against and ensure that it is the active datasource. For more information, see Connected/Selected Datasource options.

2 On the **File** menu, select **New ISQL**.

An SQL Editor window opens in your current workspace.

The following topics provide details on functionality specific to the SQL Editor:

- SQL Editor Windows Basics
- Valid Content in the SQL Editor
- Automated Error Detection and Coding Aid Features
- Execution and Execution-related SQL Editor options
- Using the Results Editor
- Miscellaneous SQL Editor Tasks

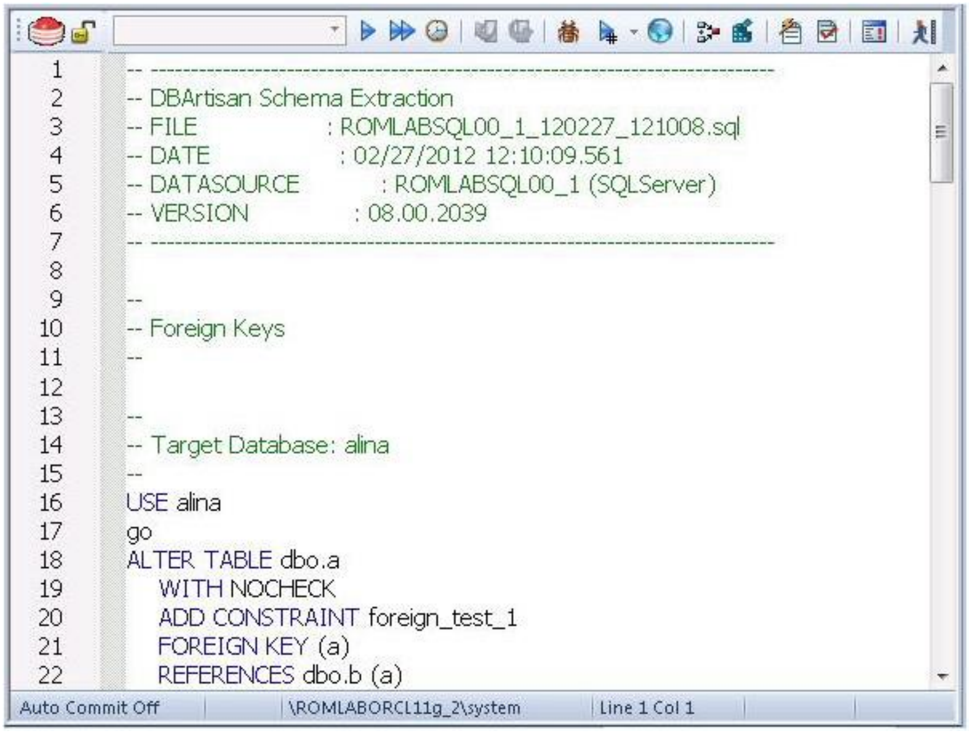
See the following topics for information that the SQL Editor shares with other DBArtisan environments:

- Common Word-processing Operations
- Common Coding Environment Features

Note: The **Tools** menu offers a number of code generation and execution utilities that can be used in conjunction with the SQL Editor. For details, see Tools.

SQL Editor Windows Basics

The SQL Editor consists of a shortcuts toolbar, an editing area, and a status bar.



```
-- ------------------------------------------------------------------------
-- DBArtisan Schema Extraction
-- FILE                  : ROMLABSQL00_1_120227_121008.sql
-- DATE                  : 02/27/2012 12:10:09.561
-- DATASOURCE            : ROMLABSQL00_1 (SQLServer)
-- VERSION               : 08.00.2039
-- ------------------------------------------------------------------------

--
-- Foreign Keys
--

--
-- Target Database: alina
--
USE alina
go
ALTER TABLE dbo.a
    WITH NOCHECK
    ADD CONSTRAINT foreign_test_1
    FOREIGN KEY (a)
    REFERENCES dbo.b (a)
```

For information on the toolbar and status bar, see the following topics:

- SQL Editor Toolbar Options
- ISQL Window Status Bar

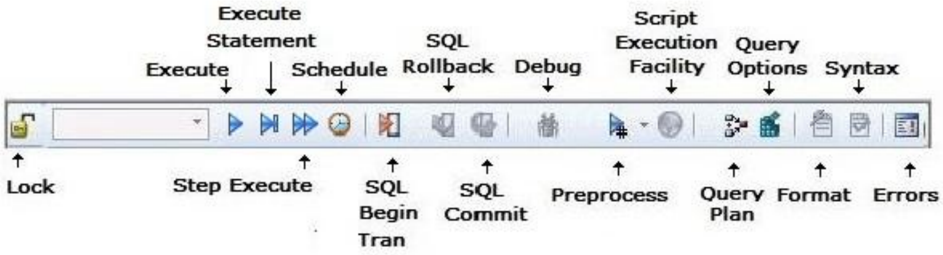
For information on using functionality available in the editing area, see the following topics:

- Common Word-processing Operations
- Common Coding Environment Features
- Valid Content in the SQL Editor
- Automated Error Detection and Coding Aid Features
- Execution and Execution-related SQL Editor options
- Miscellaneous SQL Editor Tasks

For related information, see Toggling Display of the Error Pane.

SQL Editor Toolbar Options

The SQL Editor toolbar is displayed above the editing window.

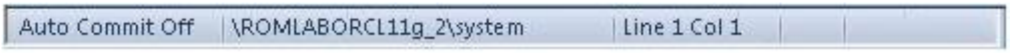


The table below describes the options of the ISQL Editor toolbar:

Option	Description
Lock - Unlock Connection	Lets you lock an ISQL Window to a specific datasource connection or unlock a currently locked window. For details, see Locking a SQL Editor Window to a Datasource.
Execute	Executes the script.
Execute Statement	Execute the statement at the cursor. For more information, see Executing the Current Statement.
Step Execute	Initiates step execution of the script. For more information, see Step Executing a Script.
Schedule	Opens a dialog that lets you schedule an SQL job. For more information, see Scheduling.
SQL Begin Transaction	Toggles the Begin Transaction status on and off.
SQL Rollback	Performs a rollback on the current transaction.
SQL Commit	Commits the current transaction.
Debug	Opens the Embarcadero SQL Debugger. For more information, see Embarcadero SQL Debugger.
Preprocess	Lets you set a preprocessing state for execution of the script. If you select **Pre-process Only**, subsequent use of the Execute button will result in the script being preprocessed without being executed. The script opens n a new editor window with all #define and #include substitutions made. If you select **Pre-process and Execute**, all #include and #define directives are carried out and the script is executed. No new editor window opens. For more information, see SQL Preprocessing: #define and #include.
Script Execution Facility	Provides a shortcut to opening the Script Execution Facility. For more information, see Script/File Execution Facilities.
Query Plan	Activates and deactivates Query Plan mode. For more information, see Using the Query Plan Facility.
Query Options	Provides a shortcut to opening the Query Options dialog. For more information, see Using the Query Plan Facility.
Format	Transforms spaghetti-style written SQL code into an easier read format.
Syntax Check	Initiates a syntax check. This is only necessary if SQL parsing is disabled and automatic syntax checking is therefore disabled. For more information, see Syntax Checking.
Errors	Toggles the **Errors** pane. For more information, see Toggling Display of the Error Pane.

ISQL Window Status Bar

The ISQL window Status bar is located below the editable area of the ISQL window.



It lets you view:

- Auto commit status (Oracle) - Automatically commits SQL statements as soon as the statements are run.
- Begin Transaction ON/OFF (SQL Server and Sybase ASE)
- Keyboard Emulation Mode - Lets you customize your editing environment to an emulation mode.

Tip: For Microsoft SQL Server and Sybase ASE, to set Begin Transaction status to “Yes”, on the ISQL Window toolbar, click the **SQL Begin Tran** button.

Tip: For Oracle, you can apply auto commit status changes to all open ISQL windows. You can set emulation and auto commit preferences. For details, see ISQL Options.

Valid Content in the SQL Editor

The SQL EDitor lets you create, edit, and execute scripts written in the SQL flavor native to each supported DBMS platform. The following topics cover additional content that can be included in your scripts.

- SQL Preprocessing: #define and #include
- ETStart and ETEnd tags
- Bind Variable Parameterization in Prepared Statements (Dynamic SQL)

For related information, see Automated Error Detection and Coding Aid Features and Execution and Execution-related SQL Editor options.

SQL Preprocessing: define and include

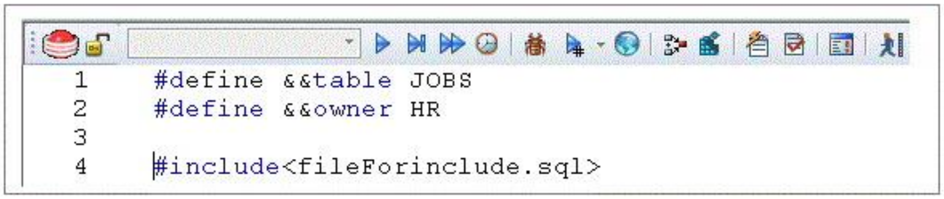
SQL Preprocessing: #define and #include

SQL preprocessing provides similar functionality to that provided by C language compiler directives. The ISQL Editor supports a simplified version of the following directives.

- **#include** provides a means to include the contents of a file in a script at the location of the directive
- **#define** provides a simple, global search and replace function within a script

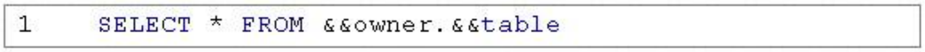
The following figure illustrates the result of basic preprocessing of a script, nesting of **#define** directives/references, and the notations required by the ISQL editor. The original script includes two **#define** directives and a **#include** reference to a one-line file named **fileForInclude.sql**. The referenced file includes two identifiers to be replaced with **#define** processing.

Script content before SQL preprocessing



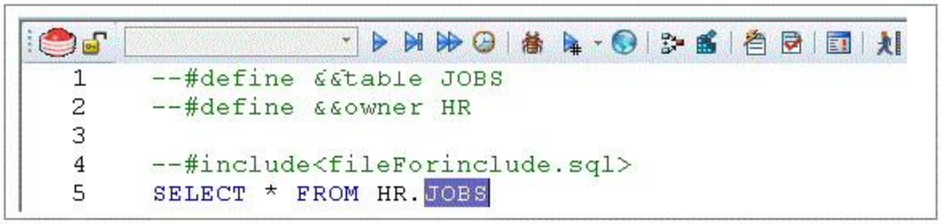
```
#define &&table JOBS
#define &&owner HR

#include<fileForinclude.sql>
```

Contents of fileForinclude.sql



```
SELECT * FROM &&owner.&&table
```

Script content produced by SQL preprocessing



```
--#define &&table JOBS
--#define &&owner HR

--#include<fileForinclude.sql>
SELECT * FROM HR.JOBS
```

The key steps in working with SQL preprocessing are:

- **Preparing the ISQL editor for #include and #define preprocessing** - Prior to using SQL preprocessing features, you should set the paths that will be searched in processing **#include** directives. For details, see Setting up DBArtisan to Preprocess #include directives.
- **Using #define and #include directives in scripts** - While the supported directives approximate typical C language #define and #include functionality, there are differences in functionality and required syntax. For example, identifiers in the **#define** directive and in all instances to be replaced, must be prefixed with two ampersand characters (#define &&PI 3.14159). For detailed information, see #include Functionality and Syntax and #define Functionality and Syntax.
- **Preprocessing and executing scripts containing #define and #include directives** - The ISQL Editor offers two preprocessing options. You can have a script preprocessed without being executed, opening the processed script in a new editor window with all **#define** and

#include substitutions made. This lets you view the processed script before execution or continue working with the processed SQL. Alternatively, you can have the script preprocessed and executed in a single step. For details, see Preprocessing and Executing Scripts Containing #define and #include Directives.

Setting up DBArtisan to Preprocess #include directives

In ISQL editor processing of a #include directive, the following locations are searched, in the following order, for the specified file:

1 The location specified on the **Datasource Properties** tab of the Datasource Registration Wizard/Editor. For details, see Registering Datasources.

2 The location specified on the Directories tab of the Options editor. For details, see Directories Options.

Note: For detailed information on setting options, see Specifying Application Preferences and Feature Options.

Before using SQL preprocessing, ensure that server side or server side and local search paths for include files are specified.

#include Functionality and Syntax

Support for the **#include** directive provides a means to include the contents of a file in a script at the location of the directive. For example, if a script contains the following:

#include mydeclarations.sql

then on preprocessing of the script, there are two effects:

- The line containing the **#include** directive is commented out before the script is sent to the database
- The text in the file mydeclarations.sql is placed in the script following the commented out line with the **#include** directive.

The **#include** directive is supported for simple file names only. Supported syntax of the **#include** directive for use in the ISQL editor, Procedure Object Editor, or Package Body Object Editor, is as follows:

#include <*filename.ext***>**

where:

- *filename.ext* is a simple filename and extension

Note: For those familiar with C compiler functionality, angle bracket and quoted forms are supported only indirectly. While **#include <***filename.ext***>** and **#include "***filename.ext***"** forms are valid, they are functionally equivalent to the **#include***filename.ext*. Using the angle bracket or quoted forms has no effect on locations searched for the target file.

Searches are performed in the locations specified in the setup for this feature. For details, see Setting up DBArtisan to Preprocess #include directives.

Error processing is as follows:

- If the preprocessor fails to include the specified file, it displays an error message noting the reason for the failure (such as the file does not exist, insufficient permissions on the file, or file too large). Preprocessing or execution of the script cannot continue until the error is corrected.
- If the file is found in the first search location specified in Setting up DBArtisan to Preprocess #include directives but cannot be opened (permission denied for example), no attempt will be made to locate the file in the second specified search location.

#define Functionality and Syntax

The **#define** directive provides a simple, global search and replace function within a script. For example, if a script contains the following:

#define &&PI 3.14159

then on execution of the script, there are two effects:

- All instances of **&&PI** in the script would be replaced by **3.14159** on execution of the script
- The line containing the **#define** directive is commented out before the script is sent to the database

The **#define** directive is supported for symbolic constants only. Supported syntax of the #define directive for use in the ISQL editor, Procedure Object Editor, or Package Body Object Editor, is as follows:

#define &&*IdentifierReplacement_text*

where:

- *Identifier* is any character string appearing in the script
- *Replacement_text* is the string that will replace all instances of the string specified by the *Identifier* argument. Valid values are strings, numbers or combinations consisting of the digits **0-9**, characters **a-z**, characters **A-Z**, and the underscore character.

Note: In addition to the actual **#define** directive appearing in a script, the ampersand notation is also required in all references that are to be replaced. References that are nor prefixed with ampersand characters are not processed.

Nested **#define** directives are also supported. For example if a script contains the following:

1. define &&myTable Clients

#define &&embtClients Embarcadero

#define &&tempTable New&&myTable

#define &&embtTempTable &&embtClients&&myTable

Select * from &&tempTable;

Select * from &&embtTempTable

then after preprocessing, the contents of the script would be as follows:

Select * from NewClients;

Select * from EmbarcaderoClients

Preprocessing and Executing Scripts Containing #define and #include Directives

The ISQL editor offer two preprocess/execute modes:

- The script can be simply preprocessed, opening the script in a new tab, with all **#define** and **#include** substitutions made. This lets you view your preprocessed script or continue with edits after preprocessing, before executing it in the new tab.
- The script can be preprocessed and executed in a single step.

Preprocessing or preprocessing/executing the script consists of selecting a mode and the executing the script. Preprocessing mode is controlled by the preprocessing dropdown on the ISQL editor toolbar.

To preprocess a script and have the preprocessed script opened in a new tab:

1 From the Preprocess dropdown, select **Pre-Process Only**. The Preprocess dropdown icon takes on a distinctive appearance to indicate **Pre-Process Only** mode.

2 On the ISQL toolbar, click the **Execute** button.

The script with all #define and #include replacements made, opens in a new ISQL editor.

Note: For details on error processing and specific handling of directives, see #include Functionality and Syntax and #define Functionality and Syntax.

To preprocess and execute a script in a single step:

1 From the Preprocess dropdown, select **Pre-Process and Execute**.

2 On the ISQL toolbar, click the **Execute** button.

The script executes. While the script sent to the server for execution includes all **#define** and **#include** substitutions, the new **Query** tab contains the original, unprocessed script.

For related information, see the following topics:

- See Script/File Execution Facilities for other script execution options
- See **SQL Preprocessing: #define and #include** for an introduction to SQL preprocessing in the ISQL editor

ETStart and ETEnd tags

With respect to functions and stored procedures, it can be necessary to have statements executed before and after creation of the procedure or function. This can be useful for example, if you need to create or drop temporary tables used by the function or procedure. Two tag pairs, **ETStart** and **ETEnd**, let you embed statements in the first comment block of a stored procedure or function. The following shows the expected syntax:

```
create procedure dbo.procname(@a numeric) as

/*

<ETStart>SQL Statement</ETStart>

<ETEnd>SQL Statement</ETEnd>

*/

begin

...
```

Bind Variable Parameterization in Prepared Statements (Dynamic SQL)

You can provide bind variable values when executing scripts containing prepared statements. Bind variables, or parameterized literals, can help optimize explain/execution plan usage by minimizing the requirement to "hard parse" queries or statements that differ by one or more literal values.

In scripts, DBArtisan recognizes named or unnamed variables that use the following notation:

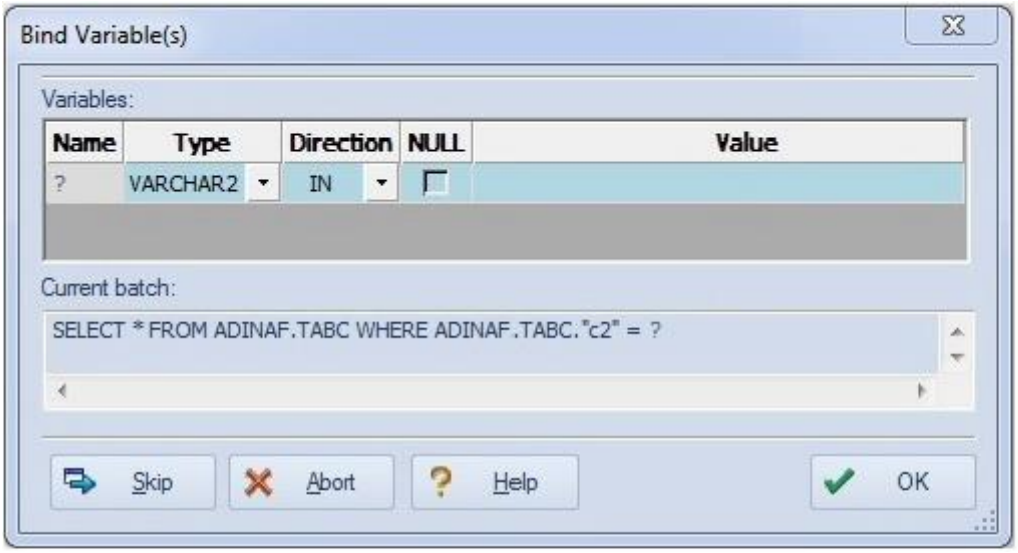
DBMS Platform/Driver Type	Notation
Oracle with native driver	:<*name*> or :<*number*>
Oracle With JDBC driver	?
All other platforms/drivers	?

For example:

- `SELECT * FROM MYTABLE WHERE MYCOLUMN = ?`
- `CALL MYPROC ( :a, :b, :c )`

Note: Support for bind variables is enabled on a editor window-by-editor window basis. Bind variable parameterization is enabled by setting the **Prepare Batch** Query Option, available on all DBMS platforms. For details, see Setting up the Execution Environment with Query options.

For an enabled SQL Editor window, when using execution options, whenever a statement containing bind variable notation is encountered, a **Bind Variables** dialog opens.



Name	Type	Direction	NULL	Value
?	VARCHAR2	IN	☐	

```
SELECT * FROM ADINAF.TABC WHERE ADINAF.TABC."c2" = ?
```

For each bind variable in the current statement, the dialog lets you specify:

- A type

  - An IN, OUT, or INOUT designation for elements such as function or procedure parameters or arguments, variables used in INTO clauses of an INSERT statements and variables used in RETURNING clauses of an Update statements.
  - A value

You can then execute, skip the statement, or abort execution for the currently running script.

Automated Error Detection and Coding Aid Features

Code is analyzed code as you add content to an ISQL editor session and offers the following automated features:

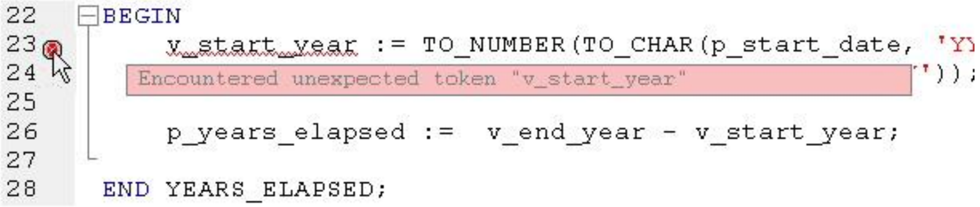
- Automatically flag syntactical errors in your scripts. For details, see Syntax Checking.
- Detect object name references to objects not present in the datasource index. For details, see Semantic Validation.
- Insert or replace object names, selected from suggestion lists, as you edit a script. For details, see Code Complete.
- Invoke common object operations such as **Open** and **Extract**, as well as object type-specific operations, from object names referenced in a script. For details, see Hyperlink Object Actions.
- Paste syntax for SQL statements, system functions calls, and other commonly used elements into an SQL Editor. For details, see Paste SQL Syntax.
- Generate and paste complete SELECT, INSERT, DELETE, UPDATE or EXEC statements or function call syntax into an SQL Editor. For details, see Paste SQL Statements.
- Create short text representations of larger text strings that can be used to quickly add the larger string to a script. For details, see Using Auto Replace Expressions.
- Make use of Embarcadero Team Server server metadata in the SQL Editor. For details, see Viewing Embarcadero Team Server Model Metadata in the SQL Editor.
- View details on errors. For details, see Toggling Display of the Error Pane.

Syntax Checking

The ISQL editor can perform on-the-fly or manually-initiated syntax checking. The syntax of the contents of an SQL Editor widow is checked against the SQL dialect native to the datasource to which the ISQL editor session is connected. Syntax checking can be performed regardless of whether an ISQL editor session is locked to a datasource.

For each syntax error detected, the script is annotates assistance is offered as follows:

- Line numbers for lines containing syntax errors are flagged with an error icon.
- Hovering the mouse over an error icon displays a tooltip with an error message
- The specific error in the line of code is underlined in red



```
BEGIN
    v_start_year := TO_NUMBER(TO_CHAR(p_start_date, 'Y
 Encountered unexpected token "v_start_year"          '));

    p_years_elapsed :=  v_end_year - v_start_year;

END YEARS_ELAPSED;
```

Note: A line-by-line listing of individual syntax errors is also available in the ISQL editor's Error pane. For more information, see [Using the DDL Editor](#).

Syntax checks can be configured to be performed automatically or you can require a syntax check to be initiated manually:

- **Automatic** - Automatic syntax checking is enabled from the Options Editor. For details, see [ISQL Options - Code Assist Tab](#).

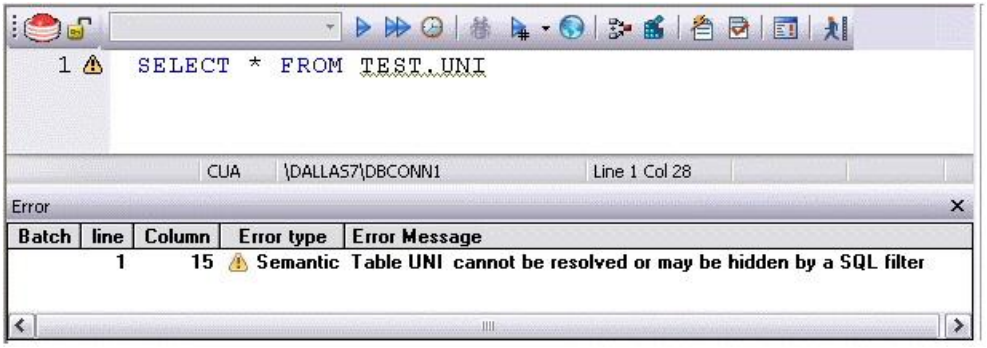
If the **Enable Real-time syntax checking** check box is selected, a syntax check is performed whenever there is an interval of 1.5 seconds or more between key strokes. Syntax error annotations persist until the error is corrected and the next automatic syntax check is executed.

- **Manual** - If automatic syntax checking is disabled, a syntax check is only performed when you click the **Syntax** button on the ISQL editor toolbar:

When automatic syntax checking is disabled, syntax error annotations persist until the error is corrected and you explicitly run another syntax check.

Semantic Validation

Semantic validation is an on-the-fly ISQL editor feature that verifies that object names are correctly specified. It ensures that the names of supported object types (columns, tables, synonyms, and views) present in a script match those for the connected datasource. Minimizing errors associated with typographical errors or references to obsolete object definitions, the names of supported object types are analyzed as you type, and an error condition is raised when it detects a name not present on the datasource.



```
1 ⚠ SELECT * FROM TEST.UNI
```

Batch	line	Column	Error type	Error Message
	1	15	⚠ Semantic	Table UNI cannot be resolved or may be hidden by a SQL filter

A semantic error can indicate one of the following conditions:

- The object name as specified in the script is not present on the database
- The referenced object is hidden by a SQL filter
- The schema information obtained to implement this feature is outdated. This can happen if other users or applications modify the schema during your ISQL editor session. For information on refreshing this information, see ISQL Options - Code Assist Tab.

When making use of this feature, always consider the following points

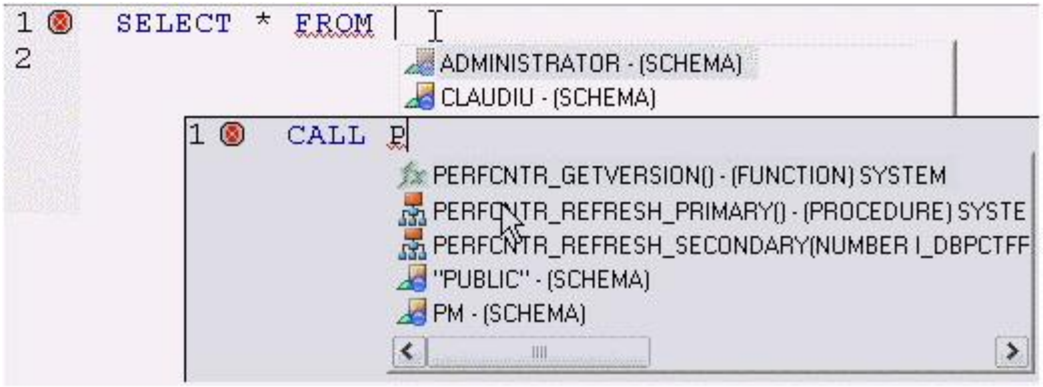
- Semantic validation is only available for objects and object types configured to be included in the datasource index. For more information, see ISQL Options - Code Assist - Advanced Tab on page 214.
- For SQL Server and Sybase ASE datasources, statements that contain temporary table names (that is, tables whose name begins with the # character) are ignored by the semantic validation feature.
- Semantic errors in a statement are not displayed for a statement that currently has outstanding syntax errors. For more information, see Syntax Checking.
- For SQL Server and Sybase ASE datasources, system table names (that is tables whose name begins with the # character) are ignored by the semantic validation feature.
- While semantic errors can be flagged with an ERROR severity level, they can be more easily distinguished from syntax errors by setting a severity level of WARNING. Similarly, if you do not want semantic errors flagged, you can set the severity level to IGNORE. For details, see ISQL Options - Code Assist Tab.
- Ensure that the ISQL Editor window is connected to a datasource. A datasource name is required in order to resolve an object name.

In the case of SQL Server and Sybase ASE, you must also select a database. The exception to this requirement is scripts that have a USE statement, specifying a database, appearing before any object name references.

Note: A line-by-line listing of individual semantic errors is also available in the ISQL editor's Error pane. For more information, see Using the DDL Editor.

Code Complete

Code Complete lets you quickly and accurately write DML and call\execute statements by providing a fast and intuitive lookup for database objects. Code Complete lets you select from a suggestion box that lists objects appropriate at the cursor location within a statement. Code Complete offers intelligent suggestions in providing simple object names or in constructing fully-qualified names.



```
SELECT * FROM
```

ADMINISTRATOR - (SCHEMA)
CLAUDIU - (SCHEMA)

```
CALL P
```

PERFCNTR_GETVERSION() - (FUNCTION) SYSTEM
PERFCNTR_REFRESH_PRIMARY() - (PROCEDURE) SYSTE
PERFCNTR_REFRESH_SECONDARY(NUMBER I_DBPCTFF
"PUBLIC" - (SCHEMA)
PM - (SCHEMA)

This feature becomes available at all points in INSERT, UPDATE, DELETE, SELECT, CALL, and EXEC statements, including all legal clauses, where a reference to the name of one of the following object types is valid:

- Columns
- Tables
- Views
- Functions
- Procedures
- Packages
- Synonyms

Before using the Code Complete feature, complete the following setup tasks:

- Ensure that the object types for which Code Complete is to be available, are being indexed in the datasource index. For details, see ISQL Options - Code Assist - Advanced Tab.
- By default, Code Complete is enabled and configured for automatic invocation. For information on enabling or disabling this feature and configuring auto-activation and other Code Complete preferences, see ISQL Options - Code Assist Tab.

When the Code Complete feature is enabled, it can be invoked in two ways:

- Automatically, if auto-activation is enabled. A Code Complete suggestion box is offered when you start typing an object name in a relevant location and then stop typing for an interval that exceeds the specified auto-activation delay.

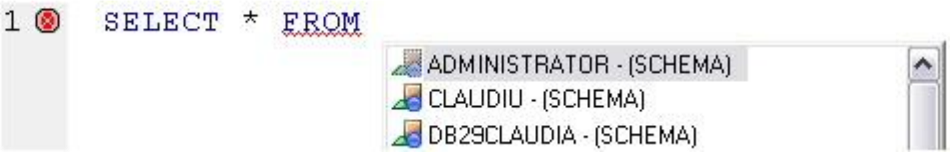
- ○ Manually, if auto-activation is disabled or if the specified auto-activation delay is sufficiently large, by pressing CONTROL+SPACE. or typing a period (.).

Note: By default, Code Complete is enabled and configured for automatic invocation. For information on enabling or disabling this feature and configuring auto-activation and other Code Complete preferences, see ISQL Options - Code Assist Tab.

Code Complete offers context sensitive suggestions based on the cursor location within the statement when Code Complete is invoked. For example:

- ○ **When invoked without any preceding text** - Code Complete will supply basic suggestions appropriate to that clause.

For example, suggestions for the table name in a FROM clause would help a user construct a qualified table or view name. Against a SQL Server datasource for example, this would include schemas and databases.

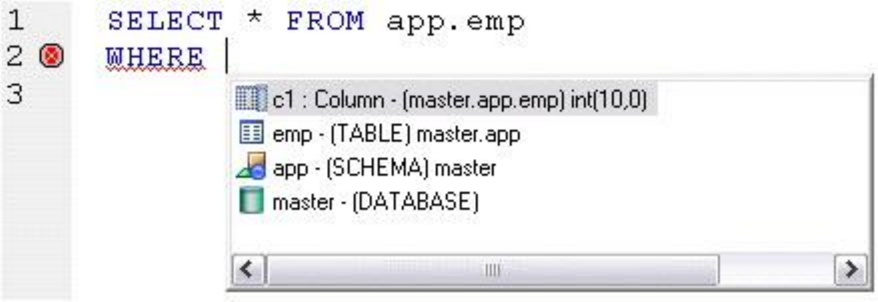


```
SELECT * FROM
```

ADMINISTRATOR - (SCHEMA)
CLAUDIU - (SCHEMA)
DB29CLAUDIA - (SCHEMA)

::

Suggestions for the column name in a WHERE clause in the same statement would contain the schemas, databases and tables\views referenced in the FROM clause as well as a listing of the applicable columns based on the referenced tables\views.

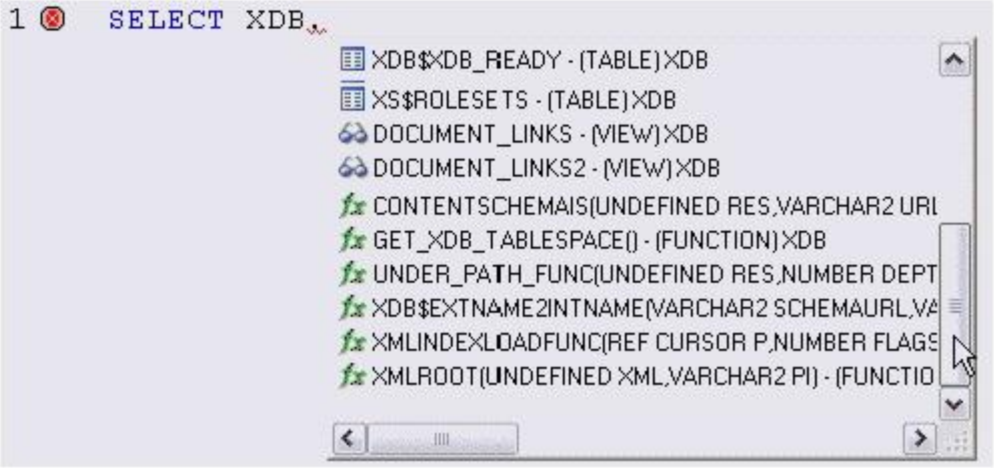


```
SELECT * FROM app.emp
WHERE
```

c1 : Column - (master.app.emp) int(10,0)
emp - (TABLE) master.app
app - (SCHEMA) master
master - (DATABASE)

::

- ○ **After the delimiter within an object name** - Code Complete offers a list that lets you select the next element to build a fully qualified object name.

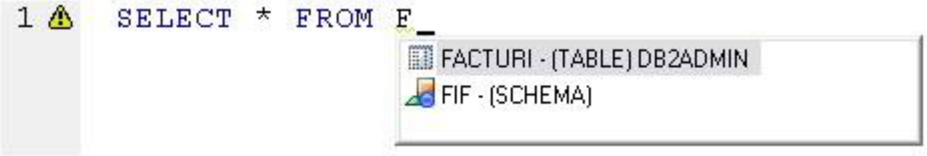


```
SELECT XDB.
```

XDB$XDB_READY - (TABLE) XDB
XS$ROLESETS - (TABLE) XDB
DOCUMENT_LINKS - (VIEW) XDB
DOCUMENT_LINKS2 - (VIEW) XDB
CONTENTSCHEMAIS(UNDEFINED RES,VARCHAR2 URI
GET_XDB_TABLESPACE() - (FUNCTION) XDB
UNDER_PATH_FUNC(UNDEFINED RES,NUMBER DEPT
XDB$EXTNAME2INTNAME(VARCHAR2 SCHEMAURL,VA
XMLINDEXLOADFUNC(REF CURSOR P,NUMBER FLAGS
XMLROOT(UNDEFINED XML,VARCHAR2 PI) - (FUNCTIO

::

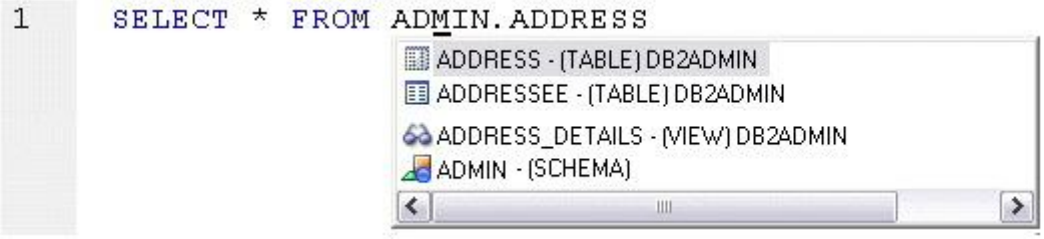
- **After typing one or more characters of an object name** - Code Complete offers a listing of the names of all object name elements, relevant to the cursor location within the object name, that start with the typed characters.



```
SELECT * FROM F
```

::

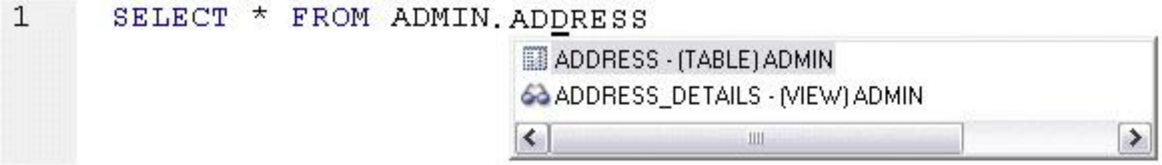
- **At any point within a partially-specified or complete object name** - Code Complete offers all selections appropriate to the location of the cursor.



```
SELECT * FROM ADMIN.ADDRESS
```

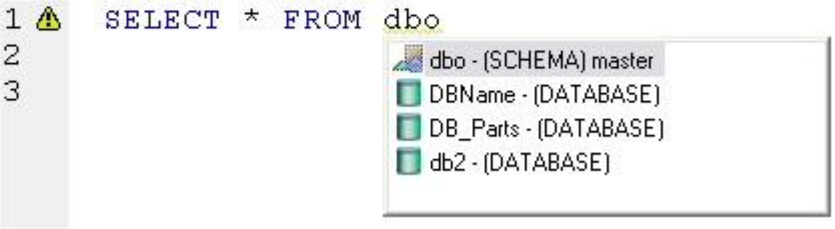
ADDRESS - (TABLE) DB2ADMIN
ADDRESSEE - (TABLE) DB2ADMIN
ADDRESS_DETAILS - (VIEW) DB2ADMIN
ADMIN - (SCHEMA)

::



```
SELECT * FROM ADMIN.ADDRESS
```

::



```
SELECT * FROM dbo
```

::

In addition, you can type additional characters while the suggestion list is available to further filter the items appearing in the list.

To insert or replace a suggestion:

1 Double-click an object name or object name element in the list. Alternatively, you can use the arrow keys and press ENTER.

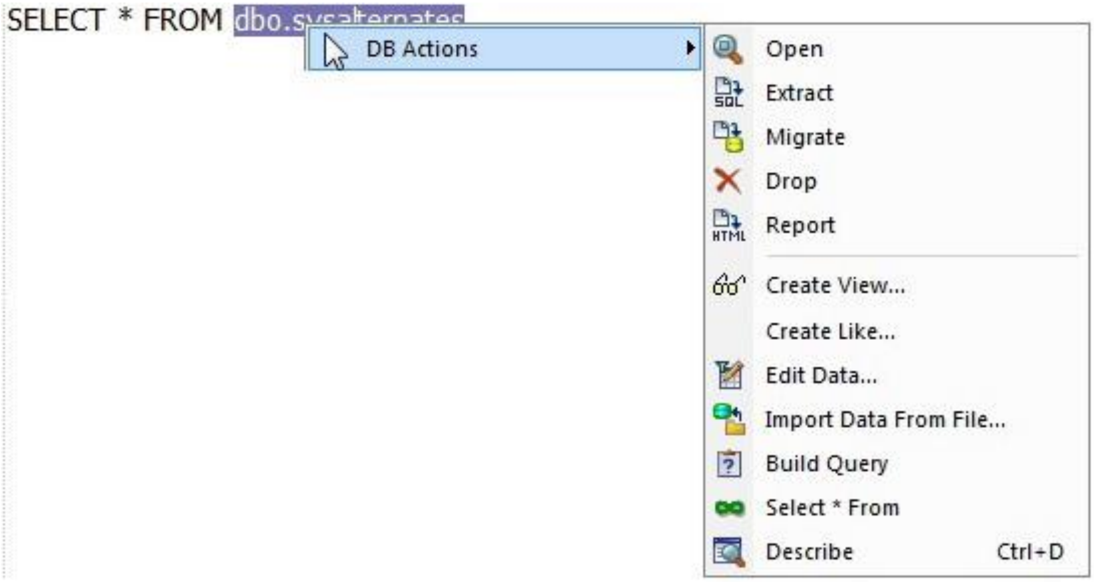
To dismiss the suggestion list without making a selection:

1 Click anywhere outside the suggestion list or press ESC.

Note: The schema information obtained to implement this feature can be outdated by the actions of other users or applications during an ISQL editor session. For information on refreshing this information, see ISQL Options - Code Assist Tab.

Hyperlink Object Actions

Hyperlinks let you invoke object actions against object names referenced in an ISQL Editor script. A configured keystroke combination (default, CTRL+right-click) opens a menu with commands specific to the supported object type:



Note: In addition to the object type-specific options, the context menu also includes SQL editor options typically available when right-clicking in the editor.

The following table shows the object action availability in DBArtisan for supported object type hyperlinks in the SQL editor:

Action (and notes)		Object type					
		Table	**View**	**Function**	**Procedure**	**Package**	**Package Body**
Build Query	For more information, see Query Builder	✓	✓				
Compile	For more information, see Script/File Execution Facilities					✓	✓
Create Like		✓					
Create View		✓					
Debug	For more information, see Embarcadero SQL Debugger			✓	✓		
Describe	For more information, see Describe Window	✓	✓	✓	✓		
Drop		✓	✓	✓	✓	✓	✓
Edit Data	For more information, see Data Editor	✓					
Execute	For more information, see Script/File Execution Facilities			✓	✓		
Extract		✓	✓	✓	✓	✓	✓
Import Data From File	For more information, see Import Data	✓					
Migrate	For more information, see Using the Schema Migration Wizard (DBArtisan)	✓	✓	✓	✓		
Open	For more information, see Modifying objects using editors	✓	✓	✓	✓	✓	✓
Report		✓	✓	✓	✓	✓	✓
Select * From		✓	✓				

Before using the Hyperlinks feature, complete the following setup tasks:

- Ensure that the object types for which Hyperlink object actions are to be available, are being indexed in the datasource index. For details, see ISQL Options - Code Assist - Advanced Tab on page 214.

- Ensure that the Hyperlink feature has been enabled, the appropriate keystroke combination configured, and the menu details selected. For details, see ISQL Options - Code Assist - Main Tab on page 211.

Once setup tasks are complete, the Hyperlinks feature is available in the SQL Editor.

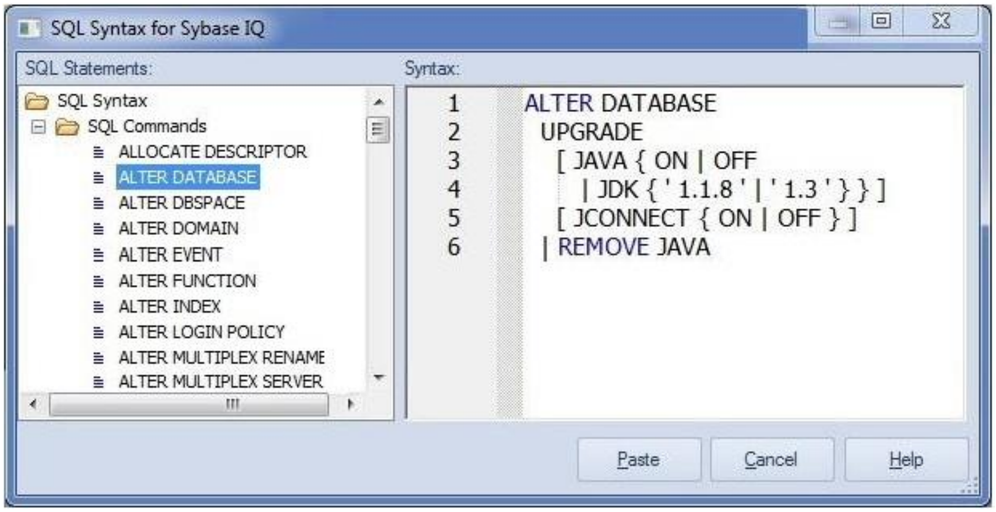
To invoke an object action using the Hyperlink feature

1 Use the configured keystroke combination anywhere within the text of an object name reference for a supported object type.

2 Select an object action from the context menu. For more information, see the online help for the selected action.

Paste SQL Syntax

The Paste SQL Syntax facility lets you paste syntax for SQL statements, system functions calls, and other commonly used elements into an SQL Editor.



```
ALTER DATABASE
  UPGRADE
    [ JAVA { ON | OFF
       | JDK { '1.1.8' | '1.3' } } ]
    [ JCONNECT { ON | OFF } ]
  | REMOVE JAVA
```

Once pasted into the SQL Editor, placeholders can be replaced with values specific to your implementation. The following table lists the categories of elements for which syntax is available for pasting on a DBMS-by-DBMS basis.

	DB2 LUW	DB2 z/OS	ITB/FBD *	MySQL	ORCL	PSTGRS	SQL SVR	SYB ASE	SYB IQ
Aggregate Functions			✓	✓			✓	✓	✓
Analytical Functions									✓
Character Functions	✓				✓				
Column Functions		✓							
Conversion Functions	✓				✓				
Datatype Conversion Functions			✓	✓			✓	✓	✓
Date/Time Functions	✓			✓	✓		✓	✓	✓
Encryption Functions				✓					
Group Functions	✓				✓				
HTTP Functions									✓
Mathematical and Number/Numeric Functions	✓			✓	✓		✓	✓	✓
Optimizer Hints					✓				
Other/Miscellaneous Functions (User, NVL, etc.)	✓				✓				✓
Scalar Functions		✓							
SQL Commands	✓	✓	✓	✓	✓		✓	✓	✓
String Functions				✓			✓	✓	✓
System Functions			✓	✓			✓	✓	✓
System Diagnostics								✓	
Text/Image Functions							✓		
Time Series Functions									✓

To paste SQL Syntax for an element into an SQL Editor

1 Open a SQL Editor against the DBMS platform you want work against. For details, see Coding Environments/Editors.

2 Place the cursor where you want to paste the SQL syntax.

3 Select **Edit > Paste SQL Syntax**.

4 In **SQL Statements**, select the element (such as an SQL statement or system function call) you want to paste. The syntax for the selected element is displayed in the **Syntax** pane.

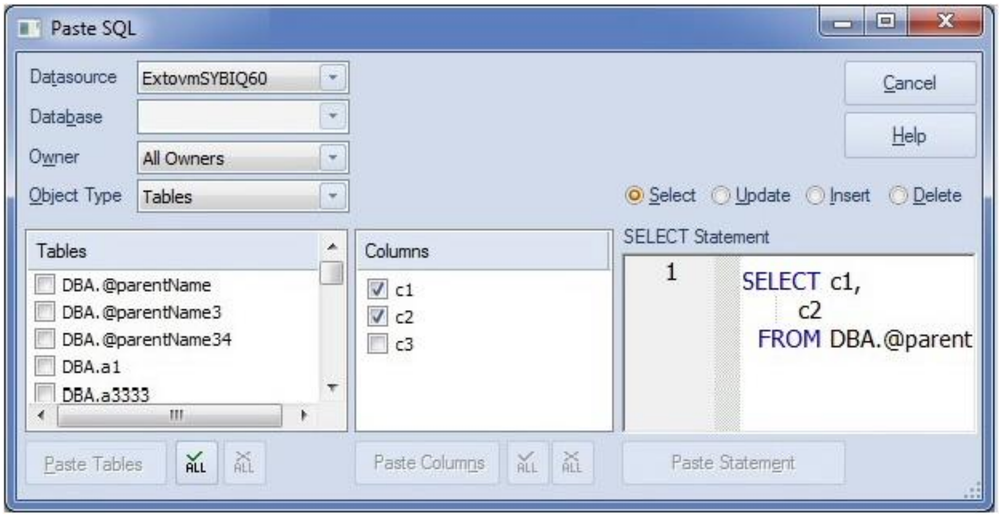
5 Click **Paste**.

Note: For information on generating and pasting complete SELECT, INSERT, UPDATE, or DELETE statements, see Paste SQL Statements.

Paste SQL Statements

The multi-purpose pasting facility offers the following:

- Build SELECT, UPDATE, INSERT, DELETE, or EXEC statements and subsequently paste into the SQL Editor. Function calls can also be generated and pasted.
- Paste comma-separated lists of defaults, functions, indexes, packages, procedures, rules, tables, triggers, user datatypes, or views into the SQL Editor. Similarly, you can paste lists of table or view column names, procedure parameter names, or function argument names into the SQL Editor.



Datasource: ExtovmSYBIQ60
Database:
Owner: All Owners
Object Type: Tables

● Select ○ Update ○ Insert ○ Delete

Tables
☐ DBA.@parentName
☐ DBA.@parentName3
☐ DBA.@parentName34
☐ DBA.a1
☐ DBA.a3333

Columns
☑ c1
☑ c2
☐ c3

SELECT Statement

```
1   SELECT c1,
           c2
      FROM DBA.@parent
```

To generate a SELECT, INSERT, UPDATE, DELETE or EXEC statement or a function call

1 Open a SQL Editor against the DBMS platform you want to work against. For details, see Coding Environments/Editors.

2 Select **Edit > Paste SQL Statement**. A **Paste SQL** dialog opens.

3 Use the **Datasource**, **Database** (if available), and **Owner** controls to specify the location and ownership of the object you want to generate the statement or call from.

4 From the **Object Type** list, select **Tables** or **Views** to generate a SELECT, UPDATE, INSERT, or DELETE, statement, **Procedures** to generate an EXEC statement, or **Functions** to generate a function call.

5 If you selected an **Object Type** of **Tables** or **Views**, select one of the **Select**, **Update**, **Insert**, or **Delete** radio buttons corresponding to the type of statement you want to generate and paste.

6 In the left list (which will be labelled **Functions**, **Procedures**, **Tables**, or **Views**), select the specific object on which the generated statement or function call will be based. The middle list is populated with the columns, arguments, or parameters for the left list object you selected.

7 In the middle list, select the specific columns, arguments, or parameters that are to appear in the statement or function call.

The right list is populated with the generated statement or function call.

8 If necessary, edit the right list, to provide valid argument values for example.

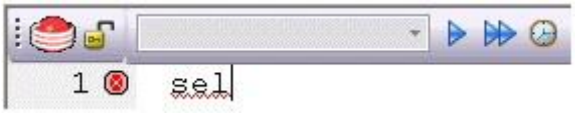
9 Click **Paste Statement** (or **Paste Syntax**).

At any time that the left or middle lists are populated, you can select multiple items in that list and click the associated **Paste**... button. This pastes a comma-separated list of those elements to the SQL Editor.

Using Auto Replace Expressions

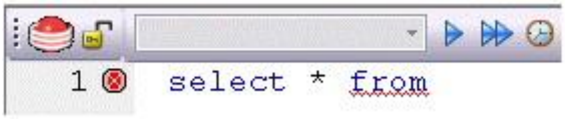
You can set up shortcuts consisting of a few characters that represent longer character strings. Instances of these Auto Replace expressions are automatically replaced by the replacement string on activation events such as typing SPACE, TAB, or RETURN. This feature is useful for creating shortcuts for one-line commands or SQL statement subsets, or even to detect and fix common typographical errors such as **teh** for **the**.

For example, consider an Auto Replace definition with an expression of **sel** to represent **Select * From**:



```
1 sel
```

If the associated activation event includes a SPACE, then on typing **sel** followed by typing SPACE, the following replacement occurs.



```
1 select * from
```

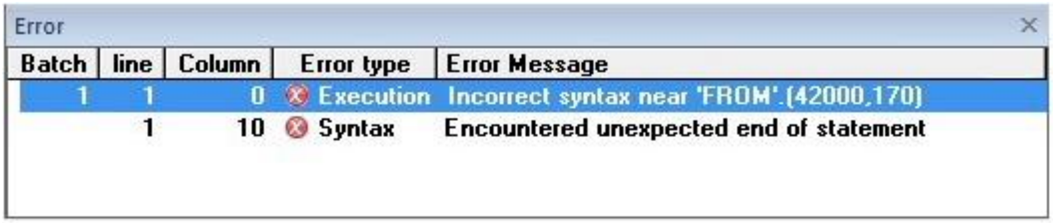
A default set of Auto Replace definitions are loaded at startup but you can also add, edit, and delete Auto Replace definitions. In addition, you can save sets of definitions to file and subsequently load specific Auto Replace definitions, allowing you to customize your templates to different platforms or development projects.

Before making use of this feature, you should be familiar with the following:

- Enabling the Auto Replace feature. For details, see Code Workbench.

Toggling Display of the Error Pane

You can view errors currently present in an ISQL editor window in a separate pane appearing below the main editor window.



Batch	line	Column	Error type	Error Message
1	1	0	Execution	Incorrect syntax near 'FROM'.(42000,170)
	1	10	Syntax	Encountered unexpected end of statement

The error pane provides a dynamic, line-by-line listing of the following error types:

- **Semantic** - errors in specified object names. Semantic problems persist in the Error pane until they are corrected. For more information, see Semantic Validation.
- **Syntactic** - syntax problems associated with the SQL dialect of the datasource to which the ISQL Editor session is connected. Syntax problems persist in the Error pane until they are corrected. For more information, see Syntax Checking.
- **Execution** - errors detected the last time the script was executed in the current ISQL editor window. Execution errors persist in the Error pane until they are refreshed by a subsequent execution.

A single menu command toggles the error pane open and closed.

To open or close the error pane:

1 On the **Query** menu, select **Show Errors**.

Execution and Execution-related SQL Editor options

The ISQL Editor lets you execute all or part of your SQL scripts. Unless you are executing large scripts that have multiple statements within them, or you need to view optimizer statistics, you execute most of your SQL scripts without options.

The following topic describe the execution related options available from the SQL Editor.

- Locking a SQL Editor Window to a Datasource
- Setting up the Execution Environment with Query options
- Executing Entire Scripts in the SQL Editor
- Executing Partial Scripts in the SQL Editor
- Executing the Current Statement
- Step Executing a Script
- Using the Query Plan Facility
- Canceling a Query

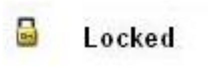
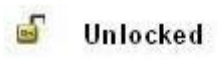
For related information, see Script/File Execution Facilities.

Locking a SQL Editor Window to a Datasource

When executing scripts or setting query options, all directives are issued to the currently active datasource. Since the selected datasource can be changed, locking the SQL Editor window to a datasource ensure that all directives are issued to the correct datasource and that sessions are set up to terminate properly.

Note: For information on the ways in which the selected datasource can change, see Connected/Selected Datasource options.

A SQL Editor window's locked/unlocked status is shown by the **Lock - Unlock Connection** button on the SQL Editor Toolbar. For information on the toolbar, see SQL Editor Windows Basics.

Lock - Unlock Connection button state	Description
	Manual session termination. The ISQL window will open a session and keep it open while the window remains open, unless the DBMS terminates the session due to a timeout, or the session is closed. The ISQL Window does not respond to datasource change events in other windows, and only uses the current connection for subsequent execution.
	Automatic session terminations. The ISQL window will execute each statement in a new session with no session state maintained between them. When explicitly unlocking an ISQL window, the connection can be returned to the connection pool or disconnected. For details, see the DBMS specific information under ISQL Options.

To toggle the datasource locked/unlocked status of a SQL Editor window

1 Click the **Lock - Unlock Connection** button.

Setting up the Execution Environment with Query options

A set of options corresponding to DBMS-specific query options are available. They let you customize the execution environment for an ISQL session with query-handling directives in areas such as performance, logging/reporting, and error handling.

See the following topics for descriptions of the options available for the DBMS you are working against:

- DB2 LUW Query options
- DB2 z/OS Query options
- MySQL Query options
- Oracle Query options
- SQL Server Query options
- Sybase ASE Query options
- Sybase IQ Query options
- Teradata Query options

If you do not change the default settings, a set of default query option settings are sent to the server, one set per DBMS platform. You can, however save a set of query option settings for each DBMS, and have that set of settings used as the default for ISQL sessions against that platform. As well, during any ISQL session, you can modify the query option settings for the current session.

The query options you select apply only to the current ISQL window instance. In addition, non-default query option settings you specify are not saved. You can however, save a set of query option settings to a file and subsequently open that file in other ISQL editor sections to have those settings applied to that session.

To specify query options:

1 With the SQL Editor open, select **Query Options** from the **Query** menu.

Once open, you have the following options:

- Set each option manually using the associated check box.
- Use the **Save** button to open a dialog that lets you save the current option query settings to an XML file. The dialog also has a **Load these query options as default** option that lets you use the settings in the saved file as the default query option settings for the current platform. Exercising this option sets the **Load Query Options** feature on the DBMS-specific **ISQL** tab of the Options editor to specify the file you saved. For more information, see ISQL Options.
- Use the **Load** button to open a dialog that lets you locate and open a previously-saved XML file containing query option settings.

- Use the **Reset** button to restore the query option defaults.

2 Click **OK** to set query options for the current ISQL window session.

Note: For information on conditions for which query options are sent to the server, see **Setting up the Execution Environment with Query options**.

When query options are sent to the server

Locked connections optimize sending query options to the server. Since an unlocked connection results in a new connection for each execution, query options must be sent with each execution. The following table outlines the specific cases:

Session unlocked	When you open an ISQL window, if you not lock the connection, all query options are sent to the server each time you execute a script, immediately before executing the actual script. As long as the connection remains unlocked, all query options are sent to the server at each execution. Similarly if you unlock a currently locked session, all query options are sent to the server at each subsequent execution. For information on locking sessions, see SQL Editor Toolbar Options.
Session locked	When you open an ISQL and lock the connection, all query options are sent to the server on the first execution. On subsequent executions, if you do not modify query option settings, no query options are sent to the server on subsequent executions. If you open **Query Options** dialog and modify options, on the next execution only the modified settings are sent to the server. These rules apply as long as the connection remains locked.

DB2 LUW Query options

The following settings are available when setting query options against a DB2 LUW datasource:

Query Option	Description
Batch Delimiter	The batch separator must be a viewable character and not a space, new line, or tab. The default (semi-colon) has been selected to ensure compatibility with the features of DBArtisan and the respective platform, but can be customized. Note: A custom delimiter works only from within an ISQL window and can't be used for extraction operations.
Check syntax when executing	TRUE/FALSE
Create Explain plan tables if required	If set to TRUE, Explain Plan tables are created, as necessary. If set to FALSE and you don't manually create tables, Explain Plan operations will fail.
Create explain plan tables on the SYSTOOLS schema	If set to TRUE, Explain Plan tables are created on the SYSTOOLS schema. If the tables already exist in the user's default schema, those tables will continue to be used. Refer to DB2 documentation for a listing of Explain Plan tables that must be deleted in order to use the SYSTOOLS option. If set to FALSE, Explain Plan tables are created under the user's default schema.
Max Errors Before Aborting	Select the maximum number of errors encountered before a script is aborted. Setting this value to zero disables the feature.
Row Count	When set to TRUE, a query is tereminated after returning the specified number of rows.
Run Script with batch execution	TRUE/FALSE
Isolation Level	Lets you set DB2 Isolation Levels of UNCOMMITED READ, RESET, CURSOR STABILITY, REPEATABLE READ, or READ STABILITY.
Prepare Batch	For the current SQL Editor window, setting this check box enables bind variable parameterization. This lets you provide named and unnamed variable values as a script executes. For details, see Bind Variable Parameterization in Prepared Statements (Dynamic SQL).

For more general information, see **Setting up the Execution Environment with Query options**.

DB2 z/OS Query options

The following settings are available when setting query options against a DB2 z/OS datasource:

Query Option	Description
Batch Delimiter	The batch separator must be a viewable character and not a space, new line, or tab. The default (semi-colon) has been selected to ensure compatibility with the features of DBArtisan and the respective platform, but can be customized. Note: A custom delimiter works only from within an ISQL window and can’t be used for extraction operations.
Check syntax when executing	TRUE/FALSE
Max Errors Before Aborting	Select the maximum number of errors encountered before a script is aborted. Setting this value to zero disables the feature.
Row Count	When set to TRUE, a query is terminated after returning the specified number of rows.
Run Script with batch execution	TRUE/FALSE
Prepare Batch	For the current SQL Editor window, setting this check box enables bind variable parameterization. This lets you provide named and unnamed variable values as a script executes. For details, see Bind Variable Parameterization in Prepared Statements (Dynamic SQL).

For more general information, see **Setting up the Execution Environment with Query options**.

MySQL Query options

The following settings are available when setting query options against a MySQL datasource:

Query Option	Description
Batch Delimiter	The batch separator must be a viewable character and not a space, new line, or tab. The default (semi-colon) has been selected to ensure compatibility with the features of DBArtisan and the respective platform, but can be customized. Note: A custom delimiter works only from within an ISQL window and can't be used for extraction operations.
Big Tables	When set to TRUE, allows big result sets by saving all temporary sets to file. This can slow queries.
Client Character	Default character set.
Insert ID	Choose a value to be used a following INSERT or ALTER TABLE statement when you supply an AUTO_INCREMENT value.
Interactive Timeout	28800 default
Last Insert ID	Set the value to be stored in the binary log when you use LAST_INSERT_ID() in a statement that updates a table.
Max Errors Before Aborting	Select the maximum number of errors encountered before a script is aborted. Setting this value to zero disables the feature.
Query Cache Type	The query is cached for ON or DEMAND.
Row Count	When set to TRUE, a query terminates after returning the specified number of rows.
SET Other Variables	Variables to be set at runtime.
SQL Auto IS NULL	When set to TRUE, enables you to find the last inserted row for a table.
SQL Big Selects	When set to TRUE, SELECT statements likely to take a very long time to execute will be aborted (i.e., where the number of rows examined exceeds the max join size)
SQL Big Tables	TRUE/FALSE
SQL Buffer Result	When set to TRUE, forces results from SELECT statements into temporary tables.
SQL Log Bin	When set to TRUE, allows logging to the binary log.
SQL Log Off	When set to TRUE, no logging is done to the general query log.
SQL Log Update	When set to TRUE, allows logging to the binary log.

SQL Low Priority Updates	When set to TRUE, gives table modifying operations lower priority than SELECT operations.
SQL Max Join Size	When set to TRUE, you can catch SELECT statements where keys are not used properly and that would probably take a long time. Set it if your users tend to perform joins that lack a WHERE clause, that take a long time, or that return millions of rows.
SQL Quote Show Create	When set to TRUE, table and column names will be quoted.
SQL Safe Updates	When set to TRUE, the query aborts UPDATE or DELETE statements that do not use a key in the WHERE clause or a LIMIT clause. This makes it possible to catch UPDATE or DELETE statements where keys are not used properly and that would probably change or delete a large number of rows
SQL Select	The maximum number of records that should be returned from SELECT statements.
SQL Warnings	Defines whether or not single row insert statements generate an information string in the event of a warning.
Transaction Isolation	Repeatable Read is the default. Read Committed, Read Uncommitted, and Serializable are the other options. Refer to MS SQL Query Options dialog box for an explanation.
Unique Checks	Performs uniqueness checks for secondary indexes of MyISAM tables.
Prepare Batch	For the current SQL Editor window, setting this check box enables bind variable parameterization. This lets you provide named and unnamed variable values as a script executes. For details, see Bind Variable Parameterization in Prepared Statements (Dynamic SQL).

For more general information, see **Setting up the Execution Environment with Query options**.

Oracle Query options

The following settings are available when setting query options against an Oracle datasource:

Category	Query Option	Description
Query Analysis	**Network Activity**	True/False
	Execution Information	True/False
	I/O Activity	True/False
	Sort Activity	TRUE/FALSE
	Index Activity	True/False
	Parse Activity	TRUE/FALSE

	Table Activity	True/False
	OS Activity	TRUE/FALSE
	Wait Activity	True/False
Miscellaneous	**Batch Delimiter**	The batch separator must be a viewable character and not a space, new line, or tab. The default ("/") has been selected to ensure compatibility with the features of DBArtisan and the respective platform, but can be customized. Note: A custom delimiter works only from within an ISQL window and can't be used for extraction operations.
	Check syntax when executing	TRUE/FALSE
	LONG Size Bytes	8,192 is the default
	Max Errors Before Aborting	Select the maximum number of errors encountered before a script is aborted. Setting this value to zero disables the feature.
	Row Count	When set to TRUE, a query terminates after returning the specified number of rows.
	Run Script with batch execution	TRUE/FALSE
	Prepare Batch	For the current SQL Editor window, setting this check box enables bind variable parameterization. This lets you provide named and unnamed variable values as a script executes. For details, see Bind Variable Parameterization in Prepared Statements (Dynamic SQL).

For more general information, see **Setting up the Execution Environment with Query options.**

SQL Server Query options

The following settings are available when setting query options against a SQL Server datasource:

Category	Query Option	Description
Query Analysis	**Show Plan**	When set to TRUE, reports data retrieval methods chosen by the Microsoft SQL Server query optimizer.
	No Count	Terminates the message indicating the number of rows affected by a Transact-SQL statement from being returned as part of the results.
	No Exec	When set to TRUE, compiles each query without executing it.
	Force Plan	When set to TRUE, processes a join in the same order as tables appear in the FROM clause of a SELECT statement only.

	Statistics I/O	Lets you display information regarding the amount of disk activity generated by Transact-SQL statements.
	Statistics Time	Displays the number of milliseconds required to parse, compile, and execute each statement.
	Parse Only	When set to TRUE, checks the syntax of each Transact-SQL statement and returns any error messages without compiling or executing the statement. When TRUE, makes Microsoft SQL Server only parse the statement. When FALSE, makes Microsoft SQL Server compile and execute the statement. Do not use Parse Only in a stored procedure or a trigger.
Arithmetic	**Ignore Overflow**	When set to TRUE, any overflow caused by a resulting value that is larger than a column's specified size is ignored.
	Abort On Overflow	If set to TRUE, queries will be aborted on encountering a value greater than the specified size.

SET Options		The **Send Set Options** setting dictates whether the remaining ANSI SQL Options in this category are sent to the server. The default for this option is set using the **Enable SET query options** setting on the ISQL tab of the Options editor. For details, see [ISQL Options](). If the **Send Set Options setting** is enabled, the remaining settings in this category let you specify the specific ANSI SQL options that are sent to the server: **ansi_defaults**, **ansi_nulls**, **ansi_null_dflt_on**, **QUOTED IDENTIFIER**, **cursor_close_on_commit**, **ANSI_PADDING**, **ANSI WARNINGS**, **IMPLICIT_TRANSACTIONS**, and **CONCAT_NULL_YIELDS_NULL**. The initial default values are hard-coded, not obtained from server settings.
Transactions	**Isolation Level**	Read Committed: Microsoft SQL Server default transaction isolation level. Specifies that shared locks are held while data is read to avoid dirty reads. You can change the data before the end of the transaction, resulting in nonrepeatable reads or phantom data. Read Uncommitted: The lowest level of transaction isolation. Transactions are isolated to ensure that physically corrupt data is not read. Applies dirty read, or isolation level 0 locking, which ensures that no shared locks are issued and no exclusive locks are honored. If set, it is possible to read uncommitted or dirty data values in the data can be changed and rows can appear or disappear in the data set before the end of the transaction. Repeatable Read: Places locks on all data used in a query, preventing other users from updating the data. Other users can insert new phantom rows into the data and are included in later reads in the current transaction. Concurrency is lower than Read Committed. Use this option only when necessary. Serializable: The highest level of transaction isolation. Transactions are completely isolated from one another. Places a range lock on the data set, preventing other users from updating or inserting rows into the data set until the transaction is complete. Concurrency is lower than Repeatable Read. Use this option only when necessary.
Miscellaneous	**Batch Delimiter**	The batch separator must be a viewable character and not a space, new line, or tab. The default ("go") has been selected to ensure compatibility with the features of DBArtisan and the respective platform, but can be customized. Note: A custom delimiter works only from within an ISQL window and can't be used for extraction operations.
	Max Errors Before Aborting	Select the maximum number of errors encountered before a script is aborted. Setting this value to zero disables the feature.
	Row Count	When set to TRUE, a query terminates after returning the specified number of rows.
	Run Script with batch execution	TRUE/FALSE
	Text Size	8,192 is the default
	Prepare Batch	For the current SQL Editor window, setting this check box enables bind variable parameterization. This lets you provide named and unnamed variable values as a script executes. For details, see [Bind Variable Parameterization in Prepared Statements (Dynamic SQL)]().

For more general information, see **Setting up the Execution Environment with Query options**.

Sybase ASE Query options

The following settings are available when setting query options against a Sybase ASE datasource:

Category	Query Option	Description
Query Analysis	**Show Plan**	When set to TRUE, reports data retrieval methods chosen by the Microsoft SQL Server query optimizer.
	No Count	Terminates the message indicating the number of rows affected by a Transact-SQL statement from being returned as part of the results.
	No Exec	When set to TRUE, compiles each query without executing it.
	Statistics I/O	Lets you display information regarding the amount of disk activity generated by Transact-SQL statements.
	Statistics Time	Displays the number of milliseconds required to parse, compile, and execute each statement.
	Statistics Subquery Cache	Displays the number of cache hits, misses, and the number of rows in the subquery cache for each subquery.
	Force Plan	When set to TRUE, processes a join in the same order as tables appear in the FROM clause of a SELECT statement only.
	Prefetch	When set to TRUE, enables large I/Os to the data cache. When set to FALSE, disables large I/Os to the data cache.
	Parse Only	When set to TRUE, checks the syntax of each Transact-SQL statement and returns any error messages without compiling or executing the statement. When TRUE, makes Microsoft SQL Server only parse the statement. When FALSE, makes Microsoft SQL Server compile and execute the statement. Do not use Parse Only in a stored procedure or a trigger.
Transactions	**Chained**	Invokes a begin transaction before the following statements: delete, insert, open, fetch, select, and update. You must still explicitly close the transaction with a commit.
	Isolation Level	0 1: Sybase default isolation level. Prevents dirty reads. 2: Prevents dirty and non-repeatable reads. 3: Prevents dirty and non-repeatable reads and phantoms. This level is equivalent to performing all selects with holdlock.
Arithmetic	**Ignore Overflow**	When set to TRUE, any overflow caused by a resulting value that is larger than a column's specified size is ignored.
	Abort On Overflow	If set to TRUE, queries will be aborted on encountering a value greater than the specified size.

	Abort On Truncation	Specifies behavior following a loss of scale by an exact numeric datatype during an implicit datatype conversion. When set to TRUE, a statement that causes the error is aborted but other statements in the transaction or batch continue to be processed. When set to FALSE, query results are truncated processing continues.
DBCC Traceflags	**Index Selection**	Valuable when tuning query performance.
	Join Selection	Valuable when tuning query performance.
	Output to Execution Window	TRUE/FALSE
	Output to Server Error Log	TRUE/FALSE
Miscellaneous	**ANSI NULL**	When set to TRUE, controls results of logical operations with NULL values.
	Set Quoted Identifier	TRUE/FALSE
	Batch Delimiter	The batch separator must be a viewable character and not a space, new line, or tab. The default ("go") has been selected to ensure compatibility with the features of DBArtisan and the respective platform, but can be customized. Note: A custom delimiter works only from within an ISQL window and can't be used for extraction operations.
	Row Count	When set to TRUE, a query terminates after returning the specified number of rows.
	Table Count	Sets the number of tables that Sybase ASE considers at one time while optimizing a join.
	Text Size	8,192 is the default
	Max Errors Before Aborting	Select the maximum number of errors encountered before a script is aborted. Setting this value to zero disables the feature.
	Run Script with batch execution	TRUE/FALSE

Text Formatting
(Sybase ASE 15)

These options let you generate diagnostic output in text format, using the Sybase ASE **set option** command. To use these options, you must be using the **sa_role** or have the **set tracing** permission. You can select ON, OFF BRIEF, or LONG standard Sybase ASE settings. If you are not using the **sa_role** or do not have the **set tracing** permission, use only the DEFAULT selection for each **Text Formatting** Set option is available. That setting has no effect. Using other selections without the required role or permission will result in an error on the next execution. For detailed information on these options and required permissions or roles, see the Sybase documentation. For access, see Accessing Third Party Documentation. Short descriptions message content displayed for Text Formatting **set option** command options are:

Show	A collection of details, where the collection depends on the choice of BRIEF, LONG, ON, or OFF.
Show Lop	Logical operators used.
Show Managers	Data structure managers used during optimization.
Show Props	Logical properties evaluated.
Show Parallel	Details of parallel query optimization.
Show Histograms	Processing of histograms associated with SARG/join columns.
Show Abstract Plan	Details of an abstract plan.
Show Search Engine	Details of the join-ordering algorithm.
Show Counters	Optimization counters.
Show Best Plan	Details of the best query plan selected by the optimizer.
Show Code Gen	Details of code generation.
Show Pio Costing	Estimates ofreads/writes from/to disk.
Show Lio Costing	Estimates of reads/writes from/to memory.
Show Pll Costing	Estimates relating to costing for parallel execution.
Show Elimination	Partition elimination.
Show Missing Stats	Details of useful statistics missing from SARG/join columns.

Prepare Batch	For the current SQL Editor window, setting this check box enables bind variable parameterization. This lets you provide named and unnamed variable values as a script executes. For details, see Bind Variable Parameterization in Prepared Statements (Dynamic SQL).

For more general information, see **Setting up the Execution Environment with Query options**.

Sybase IQ Query options

The following settings are available when setting query options against a Sybase IQ datasource:

Query Option	Description
Send SET Options	The **Send Set Options** setting applies to all settings except those in the bottom-most **Miscellaneous** category. It dictates whether the child options are sent to the server. The default for this option is set using the **Enable SET query option** setting on the ISQL tab of the Options editor. For details, see ISQL Options.
Isolation Level	Specifies the locking isolation level for Catalog Store tables.
Allow Nulls By Default	Sets the ALLOW_NULLS_BY_DEFAULT option, controling whether new columns created without specifying either NULL/NOT NULL are allowed to contain NULL values.
ANSI Null	Sets the ANSINULL option, controlling the interpretation of using = and != with NULL.
Conversion Error	Sets the CONVERSION_ERROR option, controlling reporting of data type conversion failures on fetching information.
Divide-by-zero Error	Sets the DIVIDE_BY_ZERO_ERROR option, controlling reporting of division by zero.
SET quoted_identifier	Sets the QUOTED_IDENTIFIER option, controlling the interpretation of strings enclosed in double quotes.
SQL Flagger Error Level	Sets the SQL_FLAGGER_ERROR_LEVEL option, controlling the behavior in response to any SQL code that is not part of the specified standard.
SQL Flagger Warning Level	Sets the SQL_FLAGGER_WARNING_LEVEL option, controlling the response to any SQL that is not part of the specified standard.
String R-Truncation	Sets the STRING_RTRUNCATION option, determining whether an error is raised when an INSERT or UPDATE truncates a CHAR or VARCHAR string.
T-SQL Variables	Sets the TSQL_VARIABLES option, controlling whether the **@** sign can be used as a prefix for Embedded SQL host variable names.
Disable RI Check	Sets the DISABLE_RI_CHECK option, allowing LOAD, INSERT, UPDATE, or DELETE operations to bypass the referential integrity check to improve performance.
Early Predicate Execution	Sets the EARLY_PREDICATE_EXECUTION option, controlling whether simple local predicates are executed before query optimization.
Extended Join Syntax	Sets the EXTENDED_JOIN_SYNTAX option, controlling whether queries with an ambiguous syntax for multi-table joins are allowed or reported as an error.
Index Advisor	Sets the INDEX_ADVISOR option, generateing messages suggesting additional column indexes that may improve performance of queries.
Index Advisor Max Rows	Sets the INDEX_ADVISOR_MAX_ROWS option, setting the maximum number of unique advice messages stored by the index advisor.
Infer Subquery Predicates	Sets the INFER_SUBQUERY_PREDICATES option, controlling the optimizer's inference of additional subquery predicates.

Join Optimization	Sets the the JOIN_OPTIMIZATION option, enabling or disabling the optimization of the join order.
Large Doubles Accumulator	Sets the LARGE_DOUBLES_ACCUMULATOR option, controlling which accumulator to use for SUM or AVG of floating-point numbers.
Minimize Storage	Sets the MINIMIZE_STORAGE option, minimizing use of disk space for newly created columns.
No Exec	Sets the NOEXEC option, generating the optimizer query plans instead of executing the plan.
Non ANSI NULL Varchar	Sets the NON_ANSI_NULL_VARCHAR option, controlling whether zero-length VARCHAR data is treated as NULLs for INSERT, lLOAD, and UPDATE operations.
On Charset Conversion Error	Sets the ON_CHARSET_CONVERSION_FAILURE option, controlling the action taken on a character conversion error.
Query Name	Sets the QUERY_NAME option, giving a name to an executed query in its query plan.
Query Detail	Sets the QUERY_DETAIL option, specifying whether additional query information is included in the Query Detail section of the query plan.
Query Plan	Sets the QUERY_PLAN option, specifying whether or not additional query plans are printed to the message file.
Query Plan After Run	Sets the QUERY_PLAN_AFTER_RUN option, printing the entire query plan after query execution is complete.
Query Plan as HTML	Sets the QUERY_PLAN_AS_HTML option, generating graphical query plans in HTML format.
Query Plan as HTML Directory	Sets the QUERY_PLAN_AS_HTML_DIRECTORY option, specifying the directory into which Sybase IQ writes HTML query plans.
Query Rows Returned Limit	Sets the QUERY_ROWS_RETURNED_LIMIT option, setting the row threshold for rejecting queries based on an estimated size of the result set.
Query TempSpace Limit	Sets the QUERY_TEMP_SPACE_LIMIT option, specifying the maximum estimated amount of temp space before rejecting a query.
Query Timing	Sets the QUERY_TIMING option, determining whether specific timing statistics are collected and displayed.
Batch Delimiter	A delimiting character used to separate multiple SQL statements within a statement batch.
Row Count	Lets you limit the number of rows returned by a query.
Max Errors Before Aborting	Lets you limit the number of errors returned before aborting a query.
Run Script With Batch Execution	When selected, scripts are executed statement by statement (batch delimited).

Prepare Batch	For the current SQL Editor window, setting this check box enables bind variable parameterization. This lets you provide named and unnamed variable values as a script executes. For details, see Bind Variable Parameterization in Prepared Statements (Dynamic SQL).

For more general information, see **Setting up the Execution Environment with Query options**.

Teradata Query options

The following settings are available when setting query options against a Teradata datasource:

Query Option	Description
Batch Delimiter	A delimiting character used to separate multiple SQL statements within a statement batch.
Row Count	Lets you limit the number of rows returned by a query.
Max Errors Before Aborting	Lets you limit the number of errors returned before aborting a query.
Run Script With Batch Execution	When selected, scripts are executed statement by statement (batch delimited).
Prepare Batch	For the current SQL Editor window, setting this check box enables bind variable parameterization. This lets you provide named and unnamed variable values as a script executes. For details, see Bind Variable Parameterization in Prepared Statements (Dynamic SQL).

For more general information, see **Setting up the Execution Environment with Query options**.

Executing Entire Scripts in the SQL Editor

You can execute the contents of an SQL editor window.

To execute an entire script

1 On the **Query** menu, select **Execute**.

The script starts to execute. If there are results, a **Results** tab opens letting you work with the results. For details, see Using the Results Editor.

Executing Partial Scripts in the SQL Editor

You can execute a selected portion of a script within an SQL editor window.

To execute a partial script

1 Ensure that partial script execution is enabled. For details, see ISQL Options - Main Tab.

2 Select the portion of the script to be executed.

3 On the **Query** menu, select **Execute**.

The selected statements execute. If there are results, a **Results** tab opens letting you work with the results. For details, see Using the Results Editor.

Executing the Current Statement

The **Execute Statement** command lets you execute an individual statement where statements in the script are terminated with a valid delimiter (**;** or **/**).

Note: The **Execute Statement** command is only available if the **Scan entire file for context** option in the Options Editor is selected. For details, see ISQL Options - Code Assist - Advanced Tab.

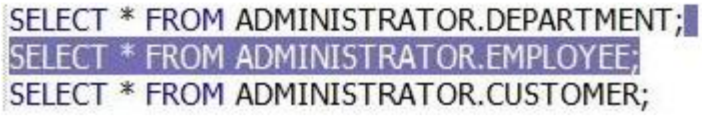
To execute a delimiter-terminated statement

1 Place the cursor anywhere within the statement.

2 From the **Query** menu, select **Execute Statement**.

The statement is executed and if results are returned, the Results Editor opens. For details, see Using the Results Editor.

As a visual aid, the executed statement is highlighted.



```
SELECT * FROM ADMINISTRATOR.DEPARTMENT;
SELECT * FROM ADMINISTRATOR.EMPLOYEE;
SELECT * FROM ADMINISTRATOR.CUSTOMER;
```

For similar options and context information, see Execution and Execution-related SQL Editor options.

Step Executing a Script

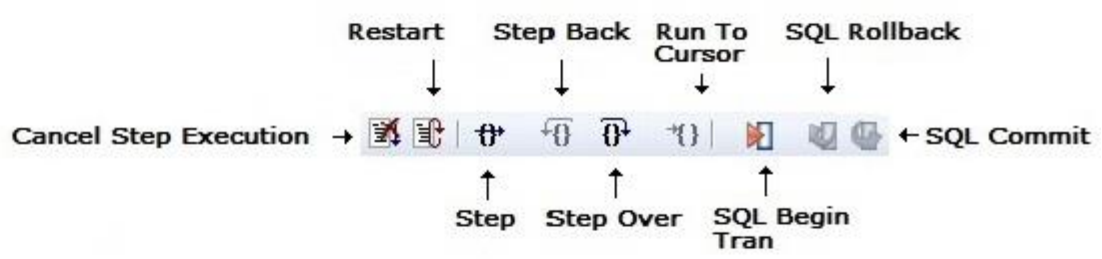
Step execution of scripts is an invaluable method to debug your scripts. The Step Execution facility lets you step through each batch in your script. While some batches can include many lines of code, some batches can consist of one line. The Step Execution facility parses the statements and moves from batch to batch during the step execution process, each step controlled by you clicking the step execution button.

To use the Step Execute Facility

1 Open a script. For details, see Opening Files.

2 On the **Query** menu, select **Step-Execute**.

The procedure starts to execute and displays errors at the bottom of the Editor window. A toolbar offering subsequent step execution actions opens:



Note: Each executing line is indicated with a yellow arrow in the gutter of the Editor window. The gutter is that gray area between the line numbers on the left and the code window. As you step through your batches, the arrow moves to indicate your current location.

3 Use the following table as a guide to subsequent actions you take:

Feature	
Step	Click the **Step** button to step into the next batch of code.
Step Back	Click the **Step Back** icon to step back to the most previous batch of code.
Step Over	Click the **Step Over** icon to jump over a batch to the next batch.
Run to Cursor	Click the **Run to Cursor** icon to execute all of the code between the beginning of the script to wherever you have inserted the pointer.
Cancel Step Execution	Click the **Cancel Step Execution** icon to change to regular execution mode.
SQL Begin Transaction	Toggles the Begin Transaction status on and off.
SQL Rollback	Performs a rollback on the current transaction.
SQL Commit	Commits the current transaction.

Keep the following in mind when using this feature:

ISQL Window Gutter	The ISQL Window Gutter is a vertical gray display bar located to the left of the ISQL window. It includes location indicators such as line numbers, error indicators, and bookmarks. The gutter is a quick visual cue to your current position in the script.
Script Line Numbers	Line numbers are included to let you navigate large scripts. Error messages in the output window indicate the line number where the error occurred.
Automatic Error Flagging	When using the Step Execution mode, errors are flagged with a red dot in the ISQL window gutter. The errors are flagged sequentially as they are stepped into.
Point and Click Error Navigation	Errors are displayed in the output window at the bottom of the screen and selects the errors as they occur. You can click each error and be taken directly to the line where that error occurred.

Using the Query Plan Facility

Each RDBMS platform lets you view the execution path that your SQL follows. For details, see the following topics:

- Viewing a Tree-based Query Plan (IBM DB2 for LUW, SQL Server, Sybase ASE)
- Viewing a Tree-based or Graphical Query Plan (Oracle)

Viewing a Tree-based Query Plan (IBM DB2 for LUW, SQL Server, Sybase ASE)

For IBM DB2 for Windows, Unix, and Linux, Microsoft SQL Server, and Sybase ASE, you can view a tree-based representation of a query plan. The Query Plan toolbar button is a toggle. Set it to enable the Show Plan mode.

The Query Plan window displays data for the estimated costs, number of rows, and bytes returned by each plan step.

Note: For IBM DB2 for Linux, Unix, and Windows, a tree view of statements and associated costs is included.

To view a tree-based representation of a query plan

1 Open a script.

2 On the **Query** menu, select **Query Plan.**

The Show Plan mode starts.

3 To generate the Show Plan in a separate result window, click **Execute**.

Viewing a Tree-based or Graphical Query Plan (Oracle)

For Oracle, you can view a graphical representation of a query plan. You can toggle the view between the graph-based view and a tree-based view, view details for each step, and work with a number of viewing options.

Note: The Options editor has a **Default Query Plan** setting that controls whether the default query plan display is tree-based or graphical. For details, see ISQL Options.

The Query Plan toolbar button is a toggle. Set it to enable the Show Plan mode.

To view a graphical representation of a query plan

1 Open a script.

2 On the **Query** menu, select **Query Plan**.

The Show Plan mode starts.

3 To generate the Show Plan in a separate result window, click **Execute**.

Options when working with the graphical Query Plan view include:

- Hovering the mouse over an execution step node to display detailed cost details for that step
- Right-clicking and choosing **Find Node** or **Find Next Node** to search large plans for nodes whose label contains a specified text string
- Right-clicking and choosing **Zoom In** or **Zoom Out**
- Right-clicking and choosing an **Orientation** sub-menu command to change the orientation of the view
- Right-clicking and choosing **Overview Window** to open a small window showing the entire plan
- Right-clicking and choosing **Save to File** to open a dialog that lets you save the graphical plan as a graphics file
- Clicking the **Query Plan** button to toggle between the graphical view and a tree-based view
- For multiple plans created rom the same script/ISQL window, using the dropdown at the top of the plan to change the plan displayed

Canceling a Query

The ISQL Editor lets you cancel a query while the rows are still being returned.

To Cancel a Query

1 On the Editor window tool bar, click **Cancel**. For more information, see SQL Editor Toolbar Options.

Note: This button is only enabled after a script has begun executing.

Using the Results Editor

The results of your SQL queries are displayed in the **Results** tab of each Editor Window, which captures result sets in a virtual data grid that accommodates large result sets. The data grid offers many of the features of a basic Windows spreadsheet, giving you a great deal of flexibility in editing the worksheet and formatting its contents.

Tip: For Oracle, REF CURSOR contents are displayed in the ISQL Window and **Results** tab.

You have many choices for navigating and viewing your SQL query results. The Results window includes functionality to set result window options, find and replace, export data to other products such as Microsoft Excel, and mail your results files.

For more information, see Configuring Result Set Windows.

Topics

- Results Window Toolbar Options
- Configuring Result Set Windows
- Configuring Results Window Grid Properties
- Exporting Data to Other Products
- Setting Result Windows to Read-only Mode
- Result Window Status Bar
- Mailing Result Sets
- Renaming and Closing Query Window Tabs
- Closing Result Window Tabs
- Saving and Closing Result Sets
- Editing Result Sets
- Formatting Result Sets
- Notes on XML Types and Unicode Display in the Results Editor

Results Window Toolbar Options

The table below describes the options of the Results window toolbar:

Option	Description
Lock	Lets you lock an ISQL Window to a specific datasource connection. When locked, the ISQL Window does not respond to datasource change events in other windows, and only uses the current connection for subsequent execution.
Options	Open the Options editor, letting you specify results window preferences. For more information, see Results (ISQL) Options.
ReExecute	Lets you execute the script again without switching to the Query tab.
Close	Lets you close the current results window.

Configuring Result Set Windows

Result set windows can be configured in a variety of ways. You can configure your result set windows to present result sets in multiple or single panels, attached or detached from the corresponding ISQL window. These options can be set globally to save you the hassle of setting options for each result window. Additionally, Result windows can be torn off and dropped onto an open area of the workspace to create separate windows. These are known as Tear Off Tab Windows.

You can set the Result Window to display result sets in any of the following modes:

- Single result sets in one attached tab window.
- Multiple result sets in one attached tab window.
- Single result sets in one separate unattached tab windows.
- Multiple result sets in one separate unattached tab windows.

To configure result set windows

1 On the **File** menu, select **Options**.

2 In the Options Editor, click the list, and then click **Results** or click the **Results** tab.

3 Refer to Results (ISQL) Options for details on how to set result windows options.

Configuring Results Window Grid Properties

You can alter the physical appearance in a results set window. Effects include items such as 3-D buttons, highlighted headings, and enabled or disabled row and column gridlines.

To configure the appearance of result set grids

1 On the **File** menu, select **Options**.

2 In the Options Editor, click the list, and then click **Grid Properties** or click the **Grid Properties** tab.

3 Refer to Grid Properties (Results window) Options for details on how to set result windows options.

Exporting Data to Other Products

You can export data from a result set to traditional spreadsheet products, such as Microsoft Excel. You can copy part or all of a result set and paste it into your traditional spreadsheet product by way of the Microsoft Windows Clipboard function. You can also save your result sets as:

- Tab delimited files,
- Comma separated files
- Pipe delimited ('|") files
- HTML
- Excel 2000 or higher
- XML
- User-specified delimited files

For more information, see Using the Results Editor.

Setting Result Windows to Read-only Mode

To set your result windows to read only mode to keep anyone from accidentally editing or altering a result set, do the following:

1 Select a Result window that you want to make read only.

2 On the **Edit** menu, select **Read Only**.

A check-mark is placed next to read only.

Note: The Read Only command is a toggle. When it is set, the formatting buttons on the **Edit** menu are not available.

For more information, see Using the Results Editor.

Result Window Status Bar

The Result Window Status Bar displays information about each Result window at the bottom of each window. You can display or hide the status bar by setting the Status Bar preference. This preference applies to all result windows.

Mailing Result Sets

If you have MAPI-compliant electronic mail software installed on your computer, then you can mail result sets to other users.

Mailing a Result Set

To mail a result set, do the following:

1 Open the **Message** dialog:

2 On the **File** menu, select **Send**.

The open **Message** dialog opens.

3 In the **Address** box, type the name of the addressee(s) and any other options.

The ISQL Editor automatically attaches a file containing your result set to the mail message.

4 Click **Send**.

The result set is sent to the specified addressee(s).

For more information, see Using the Results Editor.

Renaming and Closing Query Window Tabs

SQL windows are tabbed windows that can be closed or renamed.

To rename a **Query** tab, you need an open SQL window that includes an executed script. For information on how to execute scripts, see Execution and Execution-related SQL Editor options

Renaming a Query Window Tab

To rename a **Query Window** tab, do the following:

1 Right-click the **Query** tab on the SQL window, and then click **Rename**.

The **Rename Tab** dialog opens.

Note: The **Query** tab can be located on the top or bottom of the SQL window. You can set the location of the tab when configuring datasource options.

2 In the **New Name** box, type the name of the new Query window.

3 Click **OK**.

The name is changed and the **Rename Tab** dialog closes. The new name of the tab displays directly on the **Query Window** tab at the top of the window.

Closing a Query Window Tab

To close a **Query Window** tab, do the following:

1 At the top of the SQL window, right-click the **Query** tab, and then click **Close** or **Close All**.

The Query closes.

Closing Result Window Tabs

You can close tabbed Result set windows.

Closing a Result Window Tab

To close a **Result Window** tab, do the following:

1 On the **Result Window** tab tool bar, click **Close**.

OR

Right-click the **Results** tab, and then click **Close**.

The **Result Window** tab closes.

For more information, see Using the Results Editor.

Saving and Closing Result Sets

You can save your result sets using the standard Save and Save As functions. You can save multiple copies of the same result set and modify each copy to specific formatting requirements.

Saving Results

1 On the **File** menu, select **Save**. The **Save Results** dialog opens.

2 In the **File name** box, type the name of the result set.

3 From the **Save as type** dropdown, select a file type of **Excel 2000 or later .xls**, t**ab-delimited**, **comma-delimited**, **pipe-delimited**, **user-specified delimited .txt**, **HTML**, or **XML**.

4 From the **Encoding** dropdown, select an encoding scheme of ANSI, UNICODE, UTF-16LE (No BOM), UTF-16 Big Endian, UTF-16BE (No BOM), UTF-8, UTF-8 (No BOM), UTF-32 Little Endian, UTF-32LE (No BOM), UTF-32 Big Endian, or UTF-32BE (No BOM).

5 To include column titles, select **Include column titles when saving**.

6 If you chose **User Specified Delimiter**, in **User Specified Delimiter** type the delimiter.

7 Click **Save**.

Closing a Result Set

To close a result set, do the following:

1 On the **Main** menu, click **Close**.

The Result Set closes.

2 If you have not saved your result set, you are prompted to save the file. Click **Yes** to save and **No** to close without saving.

For more information, see Using the Results Editor.

Editing Result Sets

The Results Editor provides many ways to edit and customize your result windows. The Data Grid offers a host of features for manipulating, sorting and formatting data.

Topics

Cutting, Copying, and Pasting Cell Contents

The Result window supports standard cut, copy and paste functionality.

Cutting Cell Contents

To cut cell contents, do the following:

1 In the Results window, double click or tab to the target cell. A double line bounds the selected cell. You can also select text using standard text selection techniques.

2 On the **Edit** menu, select **Cut**.

Copying Cell Contents

To copy cell contents, do the following:

1 In the Results window, double click or tab to the target cell. A double line bounds the selected cell. You can also select text using standard text selection techniques.

2 On the **Edit** menu, select **Copy**.

Pasting Cell Contents

To paste cell contents, do the following:

1 In the Results window, double click or tab to the target cell. A double line bounds the selected cell.

2 On the **Edit** menu, select **Paste**.

For more information, see **Editing Result Sets**

Cutting, Copying, and Pasting Rows

You can perform standard cut, copy, and paste functions on rows, just as you can on individual cells.

Cutting Rows

To cut a row, do the following:

1 In the Results window, click the numbered row heading on the left side of the row.

2 On the **Edit** menu, select **Cut**.

Copying Rows

To copy rows, do the following:

1 In the Results window, double click or tab to the target row. A double line bounds the selected row. You can also select text using standard text selection techniques.

2 On the **Edit** menu, select **Copy**.

Pasting Rows

To paste rows, do the following:

1 In the Results window, double click or tab to the target row. A double line bounds the selected row.

2 On the **Edit** menu, select **Paste**.

For more information, see **Editing Result Sets**.

Cutting, Copying, and Pasting Columns

You can perform standard cut, copy, and paste functions on columns, just as you can on rows.

Cutting Columns

To cut columns, do the following:

1 In the Results window, click the column heading above the first row.

2 On the **Edit** menu, select **Cut**.

Copying Columns

To copy columns, do the following:

1 In the Results window, click the column heading.

2 On the **Edit** menu, select **Copy**.

Pasting Columns

To paste columns, do the following:

1 In the Results window, click the column heading above the first row to select the target column.

2 On the **Edit** menu, select **Paste**.

For more information, see **Editing Result Sets**.

Adding and Inserting Rows

You can add or insert rows to expand or rearrange your result sets.

Adding a Row

To add a row, do the following:

1 To add a row as the last row of the result set, position the pointer inside the result set.

2 On the **Edit** menu, select **Add Row**.

3 To add a row inside the result set, click the numbered row heading where you want to add a row.

4 On the **Edit** menu, select **Add Row**.

Inserting a Row

To insert a row, do the following:

1 To insert a row as the last row of the result set, position the pointer inside the result set.

2 On the **Edit** menu, select **Insert Row**.

3 To insert a row inside the result set, click the numbered row heading where you want to insert a row.

4 On the **Edit** menu, select **Insert Row**.

For more information, see **Editing Result Sets**.

Adding and inserting columns

You can add or insert columns to expand or rearrange your result sets.

Adding a Column

To add a column, do the following:

1 Position the pointer inside the result set.

2 Select Edit, Add Column from the main menu. The new column is added as the last column of the result set.

Inserting a Column

To insert a column, do the following:

1 Select the column where you want to insert a column.

2 Select Edit, Insert Column from the main menu. The new column is inserted to the left of the column that you selected.

For more information, see **Editing Result Sets**.

Deleting Rows and Columns

You can delete entire rows and columns to edit your result sets.

Deleting a Row

To delete a row, do the following:

1 Select the target row of data to delete.

2 On the **Edit** menu, select **Delete Row**.

Deleting a Column

To delete a column, do the following:

1 Select the target column of data to delete.

2 On the **Edit** menu, select **Delete Column**.

For more information, see **Editing Result Sets**.

Resizing Rows and Columns

Resizing rows and columns can aid readability of the spreadsheet contents or condense space for editing and formatting purposes.

Resizing Rows to Their Default Height

To resize rows to their default height, do the following:

1 Select one or more rows by clicking on the numbered row headings to the left of the rows.

2 Right-click the selected rows, and then click **Resize Rows**.

Resizing Rows to a New Height

To resize rows to a new height, do the following:

1 Select one or more rows by clicking the numbered row headings to the left of the rows.

2 Change the pointer to a solid horizontal bar with arrows on top and bottom by moving it to one of the upper or lower borders of the row heading.

3 Click and grab the row border and drag the pointer to enlarge or shrink the height of the row.

Resizing Columns to Their Default Widths

To resize columns to their default widths, do the following:

1 Select one or more columns by clicking the column headings.

2 Right-click the selected columns, and then click **Resize Columns**.

Resizing Columns to a New Width

To resize columns to a new width, do the following:

1 Select one or more columns by clicking the column headings.

2 Change the pointer to a solid horizontal bar with arrows on top and bottom by moving it to one of the upper or lower borders of the column heading.

3 Click and grab the column border and drag the pointer to enlarge or shrink the height of the column.

For more information, see **Editing Result Sets**.

Sorting Data

To order and organize data in a coherent manner, you can sort columns alphanumerically in single result sets.

Sorting Data

To sort data, do the following:

1 Double click the column header to sort the data in the column in ascending, alphanumeric order. To sort the column in descending order, double click the column header again.

Note: This option is not valid for multiple result sets.

For more information, see **Editing Result Sets**.

Formatting Result Sets

The ISQL Editor provides flexibility in formatting result sets, for analysis and reporting, from within a Result Window. Result sets can be formatted to best suit your purpose, whether it be sending via e-mail, printing, or exporting a file to other programs, such as Microsoft Excel. Some of these features change only the screen display of the results, while others allow you to format for printing.

Topics

- Changing the Displayed View
- Format Border Styles
- Format Font Styles
- Format Color Styles

For information on general appearance of results, see Configuring Results Window Grid Properties.

Changing the Displayed View

You can view results in a standard grid, in HTML format, or as flat ASCII. By default, when a result window is generated, the results are displayed according to the view option currently selected on the Results (ISQL) Options. You can change the display in the results window.

To change the result view on the fly:

1 RIght-click in the results window.

2 On the shortcut menu, choose one of the **View as Grid**, **View as HTML**, or **View as ASCII** options.

Format Border Styles

You can use the shortcut menu to alter border properties.

Completing the Format Styles Dialog Box

To complete the **Format Styles** dialog, do the following:

1 Right-click the **Result** data grid, and then click **Border**.

2 On the **Border** box, you can indicate whether or not a border should appear on the top, bottom, right, left, or for a range of cells by clicking the corresponding boxes.

3 To set a range of cells apart by setting a particular border style around that range, select the range in the result set before opening the **Format Styles** dialog. To select the **Range** property, click the range box.

4 In the **Type** box, you can select the type of line you want to border the cell or cells by clicking the corresponding boxes.

5 To select a color, click the **Color** list, and then click the border color.

6 Click **OK**.

Changes are saved and the **Format Styles** dialog closes.

For more information, see **Formatting Result Sets**.

Format Font Styles

You can use the shortcut menu to alter font properties. Selecting the Font command displays the **Format Styles** tabbed dialog.

Completing the Format Styles Dialog Box

To complete the **Format Styles** dialog, do the following:

1 Right-click the **Result** data grid, and then click **Font.**

The **Format Styles** dialog opens.

2 In the **Font** box, type or click the font you want to use.

3 In the **Style** box, type or click the font style you want to use.

4 In the **Size** box, type or click the size you want to use.

5 To make a line cross through the length of the text, in the **Effects** box, select the **Strikeout** check box.

6 To underline the text, in the **Effects** box, select the **Underline** check box.

7 To change the script style, click the **Script** list, and then click the new script style.

The Sample box displays the sample text of your selections

For more information, see **Formatting Result Sets**.

Format Color Styles

You can use the shortcut menu to alter color properties. Selecting the Color command displays the **Cell Properties** dialog.

Completing the Cell Properties Dialog Box

To complete the **Format Styles** dialog, do the following:

1 Right-click the **Result** data grid, and then click **Color.**

The **Cell Properties** dialog opens.

2 Change the **Text Color** and **Background Color** options.

For more information, see **Formatting Result Sets**.

Notes on XML Types and Unicode Display in the Results Editor

When viewing data in the Results grid, keep the following in mind:

- XML data types are supported for IBM DB2 for Windows, Unix, and Linux, Microsoft SQL Server, and Oracle. In the Results grid, XML data types are displayed as LOB content.
- Support for display of Unicode characters is provided as follows:
- IBM DB2 for Windows, Unix, and Linux V8 and V9: **character**, **clob**, **varchar**, and **longvarchar** types
- SQL Server 2005: **nchar**, **nvarchar**, **ntext**, and **nvarchar(max)** types
- Oracle 9i, and 10g: **NCHAR**, **NVARCHAR2** and **NCLOB** for non-Unicode UTF8 Character Set Instances and **NCHAR**, **NVARCHAR2**, **CHAR**, **VARCHAR2**, **LONG**, **NCLOB** and **CLOB** for Unicode UTF8 Character Set Instances
- Sybase ASE 12.5 and 15.2: **UNICHAR**,**UNIVARCHAR** and **UNITEXT** for non-Unicode UTF8 Character Set Instances and **UNICHAR**, **UNIVARCHAR**, **UNITEXT**, **NCHAR**, **NVARCHAR**, **CHAR**, **VARCHAR** and **TEXT** for Unicode UTF8 Character Set Instances

Miscellaneous SQL Editor Tasks

The following supplementary function is available from the SQL Editor:

- Emailing SQL Scripts

Emailing SQL Scripts

If you have MAPI-compliant E-mail software installed on your computer, then you can send SQL scripts to other users.

To send a SQL script

1 Open a SQL Editor and open, paste, or type a script. For more information, see Using the SQL Editor.

2 On the **File** menu, select **Send**. Your E-mail application opens and automatically attaches a file containing the script.

3 In the **Address** box, type the name of the addressee(s) and when ready click **Send**.

Using the DDL Editor

You use the DDL Editor when you want to create a script that is tied to an object type in your database. The DDL Editor opens containing a template script for the selected object type. Because the DDL Editor is directly tied to a database, database warning messages can be issued. For example, if you have a create table script which includes a DROP TABLE statement, the DDL Editor warns you about the existence of this statement and that you could lose existing table data.

Tip: Since you must drop a database object before you can recreate it, you can set the DDL Editor to automatically include DROP statements for specified objects. For more information, see the DBMS platform-specific topics under DDL Extract Options.

To create an object using the DDL EDitor

1 On the **File** menu, select **New > DDL Editor.**

2 Use the following table as a guide to completing the dialog:

Option	Description
Object Type	Lets you select the object type to which you want to attach the script.
Owner	Lets you type the name of the object owner for the object. The name of the owner connected to the current datasource is used as the default.
Object Name	Lets you type the name of the object type.

3 Click **OK**.

The DDL opens in the SQL Editor. A subset of SQL Editor functionality becomes available. For more information, see Using the SQL Editor.

Data, Schema, and Miscellaneous Utilities (DBArtisan)

DBArtisan incorporates a number of powerful utilities that let you create, edit and manage your development environment. Primarily schema and data-oriented, there also server configuration, session-specific, and process-specific utilities. The following table lists available utilities and their availability by DBMS platform.

	DB2 LUW	DB2 z/OS	MySQL	ORCL	PSTGRS	SQL SVR	SYB ASE
Archive Activity				✓			
Backup Control Files				✓			
Chained Row Analysis				✓			
Data Export	✓						
Data Load	✓	✓			✓	✓	✓
Data Pump Export				✓			
Data Pump Import				✓			
Data Unload	✓				✓	✓	✓
Database Backup			✓		✓	✓	✓
Database Export				✓			
Database Import				✓			
Database Manager	✓	✓	✓	✓		✓	✓
Database Monitor	✓	✓	✓	✓	✓	✓	✓
Database Recover/Restore			✓		✓	✓	✓
Free Space Deficit				✓			
Instance Configuration	✓	✓	✓	✓		✓	✓
Instance Parameter Configuration (DBArtisan - ORCL)				✓			
Maximum Extent Limits				✓			
Pinned Code				✓			
Resource Cost				✓			
Resource Limits							✓
Schema Extraction	✓	✓	✓	✓		✓	✓
Schema Migration	✓	✓	✓	✓		✓	✓
Schema Publication	✓	✓	✓	✓		✓	✓

Session Information				✓			
SQL Logfile	✓	✓	✓	✓		✓	✓
Top Sessions				✓			

Archive activity (DBArtisan - ORCL)

Oracle uses Redo logs to record changes made to a database during operation. These files, which are active or online during normal operation, are referred to as the online redo logfiles. Oracle uses these files to reapply changes made during the recovery process that were not written to datafiles before shutdown. You can operate an Oracle database in ARCHIVELOG or NOARCHIVELOG mode. If you choose ARCHIVELOG mode, then Oracle uses a background process to copy the contents of the online redo logfiles as archive redo logfiles in a separate area. ARCHIVELOG mode lets you perform online backups and database media recovery of your Oracle database.

The Archive Activity Monitor lets you to analyze the status, activity levels, proper sizing and performance of your archive redo logs. At a glance, you can determine the ARCHIVELOG mode, configuration and vital statistics about redo log performance. In particular, the Archive Activity Monitor lets you to assess whether or not the redo logfiles are sized properly. It is very important to choose the right size for online redo logfiles. If the redo logs are too small, the Log Writer must switch logfiles too often, which hurts performance. If the redo logs are too big, recovery times are longer.

The table below describes the Archive Activity Monitor options:

Category	Option	Description
Log Archiving	Log Mode	Whether or not ARCHIVELOG mode is active for the Oracle instance.
	Auto Archival	Whether or not the database has been configured to archive redo logs automatically.
	Destination	Directory where the archive redo logs are contained.
	Oldest Log Sequence	Oldest system change number. As a precaution, your backups should include all archived logfiles starting with the oldest log sequence.
	Current Log Sequence	Current system change number. It marks the start of the backup.
Online Redo Archive Activity	Redo Entries	Number entries made to the online redo logs by the Log Writer (LGWR) process.
	Avg Redo Size	Average size, in kilobytes, of a redo log entry.
Redo Log Buffer Contention	Entry Requests	Number of entries made to the redo log buffer by the Log Writer (LGWR) process.
	Entry Wait Times	Number of times that the Oracle Log Writer has had to wait before making an entry to the redo log buffer. This number is accumulated since the last time that the database was started. A high number of waits indicates that the log buffer size is too small. To increase the size, you need to modify the LOG_BUFFER parameter in the Oracle configuration file (e.g., INITORCL.ORA).

Redo Log Switches	Time	When the redo log switch occurred. If redo log switches occur frequently, the redo logs are probably too small. As a rule-of-thumb, redo log switches should not occur more frequently than once every 30 minutes. If they are occurring more frequently, then you should increase the size of the redo logs.
	Record ID in Controlfile	Displays ID number in the control file.

Opening the Archive Activity Dialog Box

1 On the **Utilities** menu, click **Archive Activity**.

DBArtisan opens the **Archive Activity** dialog.

Backup Control Files (DBArtisan - ORCL)

You should keep many copies of your control files to safeguard against disk failure or loss of the primary control file. DBArtisan provides a graphical interface for constructing the ALTER DATABASE command needed to backup your control files.

Using the Backup Control Files Utility

1 On the **Utilities** menu, click **Backup Control File**.

DBArtisan opens the Backup Controlfile Dialog Box.

Backup Controlfile Dialog Box

The table below describes the options and functionality available on the **Backup Controlfile** dialog

Option	Description
Backup Controlfile Name	Type the name of the backup control file in the box. To reuse an existing Back Up Controlfile, type the name of the existing file, and then select Reuse Existing Backup Controlfile. NOTE: When you type in the Backup Controlfile Name box, the check box is activated.
Reuse Existing Backup Controlfile check box	Select this check box if you want to reuse an existing file. The name of the file and the name entered in the Backup Controlfile Name box must be the same.
Trace SQL Statements	Click the option button that corresponds to how you want Trace SQL statements handled: None - No statements are traced. Reset Logs - Traces reset logs. No Reset Log - Does not trace reset logs.

Chained Row Analysis (DBArtisan - ORCL)

Updates can cause data rows to migrate and to chain over multiple data blocks. Chaining degrades performance because more physical reads are required to access a row of data. Consequently, you should monitor chained rows regularly to spot performance bottlenecks before they become severe.

To avoid chained rows, you can tune the Percent Free storage settings for that table or cluster. By increasing the Percent Free on each data block, you decrease the likelihood that an update causes a data row to expand over multiple data blocks.

If the row length is greater than the size of a data block, you cannot avoid chained rows.

DBArtisan provides a utility that lets you identify chained rows quickly by displaying chained rows for all or any schema in your Oracle database.

Analyzing Chained Rows

1 On the **Utilities** menu, select **Chained Row Analysis**.

DBArtisan opens the **Chained Row Analysis** dialog.

The table below describes the options and functionality on the **Chained Row Analysis** dialog:

Option	Description
Owner	Lets you select the owner for whom you want to scan for chained rows.
Refresh Button	Click to refresh the list.

Data Export (DBArtisan - DB2 LUW)

The Export Utility lets you use the Command Line Processor EXPORT command to export data. A wizard lets you generate two CLP calls:

- A CONNECT command to connect to the server
- An EXPORT command

Wizard controls let you specify a full set of EXPORT parameters.

Note: Before using this utility, you must specify the location of executables used by DB2. For details, see DB2 Utilities Options.

To export data from a DB2 LUW database

1 Connect to a DB2 LUW datasource and ensure that it is the currently selected datasource. For details, see Connected/Selected Datasource options.

2 Select **Utilities > Export** to open the **CLP Export** WIzard.

3 Use the following table as a guide to performing tasks as you pass through the wizard panels:

Panel	Option	Description
File Options	**Export file type**	Lets you select among **DEL (delimited ASCII format)**, **WSF (work sheet format)**, and **IXF (Integration Exchange Format)**, output file formats.
	Export file name	Lets you specify the name of the file to which data is to be exported.
	Messages file name	Lets you specify the name of the file to which warning and error messages occurring during an export operation will be written.
	IXF Specific File Options	If an **Export file type** of **IXF** was selected, this control lets you provide a **Code page of the data in the out put data set** (codepage=) value
	DEL Specific File Options	If an **Export file type** of **DEL** was selected, this group of controls lets you specify a set of type modifiers (corresponding to modifiers such as CHARDEL, COLDEL, DECPT, DECPLUSBLANK, NODOUBLEDEL, NOCHARDEL, STRIPLZEROS and TIMESTAMPFORMAT), specifying delimiter and general character conventions.
	WSF Specific File Options	If an **Export file type** of **WSF** was selected, this group of controls corresponds to valid type modifiers for worksheet (**1-4**) Lotus and DBCS formats.
General Options	**Separate the LOB data from the data file**	Selecting this check box adds a LOBSINFILE modifier and activates other LOB-related controls in the group.
	LOB Paths	Lets you build a list of LOBS TO clause values, specifying one or more paths to directories where LOB files are to be stored.

LOB Filenames	Lets you build a list of LOBFILE option values, specifying one or more files where LOBs are to be stored.
Place each LOB in a separate file	Adds a LOBSINSEPFILES modifier, specifying that each LOB value is to be written to a separate file.
XML Path	Lets you build a list of XML TO clause values, specifying one or more paths to directories where XML files are to be stored.
XML Filename	Lets you build a list of XMLFILE option values, specifying one or more XML file names.
Save XML schema information for all XML columns	Selecting this check box adds a XMLSAVESCHEMA modifier, specifying that XML schema information should be saved for all XML columns.
Write each XQuery Data Model (XDM) instance to a separate file	Selecting this check box adds a XMLINSEPFILES modifier, specifying that each XDM instance is written to a separate file
Write XDM instances without an XML declaration tag	Selecting this check box adds a XMLNODECLARATION modifier, specifying that XDM instances are written with no XML declaration tag.
Encode XML documents in the UTF-16 code page	Selecting this check box adds a XMLGRAPHIC modifier, specifying that exported XML documents are encoded in the UTF-16 code page regardless of the application code page or codepage file type modifier.
Use the application/the specified code page to write XDM instances	Selecting this check box adds a XMLCHAR modifier, specifying that

Source	When exporting data from a DB2 LUW database, this panel lets you specify the data to be exported. SELECT, XQUERY, HIERARCHY, and HIERARCHY STARTING options are provided. To specify a SELECT or XQUERY statement export, select the **select or xquery statement** radio button. In the **Specify the statement to use for export** box, provide text of the SELECT or XQUERY statement. Optionally, use the **Method N** controls (New, Delete, Move Item Up, Move Item Down) to provide a list of column names to be used in the output files. To specify a default traverse order, select the **hierarchy with the default traversal order** radio button. From the **Select the start table for the hierarchy** list, select the table that the export hierarchy is to start from. To specify a user-provided traverse order, select the **hierarchy with user specified traversal order** radio button. In the **Double-click on table...** list, double click on each table to be included in the traversal order list. The selected tables are displayed in the **Custom traversal order...** list. Use the controls associated with the **Custom traversal order...** list (Delete, Move Item Up, Move Item Down) to order the list. In the **Where clause** box, provide a condition specifying which rows are exported.

4 Finally, use the **Execute** button to perform the backup.

Data Load (DBArtisan)

The Data Load utility is available for a number of platforms:

- Data Load (DBArtisan - DB2 z/OS)
- Data Load (DBArtisan - DB2 LUW)
- Data Load (DBArtisan - SQL SVR, SYB ASE)
- Data Load (DBArtisan - PostgreSQL)

Data Load (DBArtisan - DB2 z/OS)

The DBArtisan Load Data Utility for DB2 z/OS enables you to load records into one or more tables in a tablespace without creating an intermediate exported file. If the tablespace already contains data, you can opt to add the new data to the existing data or to replace it. The data that's loaded is subject to the same editing or validation routines currently associated with the table and any field procedure associated with particular table columns. As output, you get a loaded tablespace or partition, a file of records that have been discarded, and an error report if you specify the Enforce Constraints option or if unique indexes were part of the load operation.

To use this utility, you need a sufficient set of privileges.

One of the following conditions must be met:

- You must own the table
- Have load privileges for the database
- DBAdmin or DBCTRL authority for the database.
- SYSCTRL or SYSADM authority.

Refer to IBM documentation for details on utilities. For assistance, see Accessing Third Party Documentation.

To Open the Load Data Utility Wizard

1 Connect to a DB2 OS390 database on the Navigator/Explorer tree.

2 Expand the database and click the **Tablespace** subnode.

3 In the right window, select a tablespace and click **Load Data** from the Command ⇩ drop-down menu.

4 Use the following topics as a guide to completing the pages of the wizard.

Initial Settings

Required Information	Description
Do you want the utility to be restartable?	The utility restart information will be included if a recover operation is required.
Tape Stacking	Select this if you are backing up this operation onto physical media. Click **Properties** to specify the particulars for the dataset as it's recorded by opening the Load Utility - Tape Stacking Dialog Box.
Do you want to specify a cursor?	If you want to specify a cursor you need to type the name of the cursor for the input data set in the field below. No is the default. Any cursor name you specify cannot exceed eight characters.
Specify the In DDN	The input template name. SYSREC is the default.
Would you like to preformat?	If you say yes, you are telling the operation to serialize at the tablespace level. This can inhibit concurrent processing of separate partitions. No is the default.

Resumption/ Replace/Statistics

Required Information	Description
Resume	If you say No, records load into an empty tablespace. This is the default. If you say Yes, records are loaded into a non-empty tablespace. Loading begins at the current end of data in the tablespace. If you specify the sharelevel as change, the before and after triggers for each row that's loaded are activated. The default sharelevel is none--this specifies that applications cannot concurrently access the tablespace during the load operation. When you select Yes, the remaining options on this panel are unavailable.
Replace	Here you indicate whether the tablespace and its indexes need to be reset to empty before records load. If you select Yes, you are telling the load operation to serialize at the tablespace level. No is the default.
Would you like to specify the statistics option?	Here you indicate whether you want to gather statistics for the tablespace. No is the default. If you select Yes, statistics are gathered on a base tablespace, but not an LOB tablespace. If you gather statistics on encrypted data, the results may not be useful.
Do you want to output message to SYSPRINT?	This is the output dataset for messages and printed output. Default is the default.
Do you want to force aggregation or rollup processing to be done even though some parts do not contain data?	Yes, No, or default. Default is the default. Here you specify whether you want statistics to aggregate or rollup when RUNSTATS is executed even in the event that some parts are empty.

Select Tables (for running statistics)

Here you are presented with a dialog box where you identify the table you want to run statistics on. Click **ADD** to open the **Table/Object Selector** dialog. Click **Set Properties** to open the **Select**

Column for Table dialog. Click **Delete** to remove a table you selected from the run statistics operation.

Select Tables to Load Data Into

Here you are presented with the dialog box where you identify the tables you want to load the data into. Click **ADD** to select one or more tables. Click **Set Properties** to open the Into Table Specification Dialog Box.

Select Indexes

Here you are presented with a dialog box where you identify the indexes you want to run statistics on. Click **ADD** to open the **Index/Object Selector** dialog. Click **Set Properties** to open the **Set Correlation option** dialog where you set the KeyCard and Frequent Item options. When you ADD Frequent Options, the **Add Frequent Options** dialog opens which identifies the number of columns and the count.

Update/History Statistics

Required Information	Description
Update Statistics	All: Indicates that all collect statistics are to be updated in the catalog. All is the default. Access Path: Indicates that updates are to be made only to the catalog table columns that provide statistics used for selecting the access path. Space: Indicates that updates are to be made only to the catalog table columns that provide statistics that help assess the status of a particular tablespace or index. None: Indicates no catalog tables will be updated. Note: This is enabled only when the Report option is set to Yes.
History Statistics	This records all catalog table inserts or updates to the catalog history tables. Default: The value specified on All: Indicates all collected statistics are to be updated in the catalog history tables. Access Path: Indicates that updates are to be made only to the catalog history table columns that provide statistics used for selecting the access path. Space: Specifies that only space-related statistics are to be updated in catalog history tables. None: Specifies that no catalog history tables are to updated with the statistics that have been collected.

Dictionary/Log

Required Information	Description
Keep dictionary	If you specify No, the load operation is prevented from building a new compression dictionary, thereby saving you the costs related to such a build operation. No is the default.
Reuse	When used with Replace (on an earlier panel of the wizard), this option indicates that the load operation should reset and reuse the datasets without deleting or refining them. No is the default.
Log	The Default specifies normal logging during the load operation and is equivalent to Yes. All loaded records will be logged. No specifies no logging of data in the load operation and disables the copy-pending option below.
LOAD is not to set the tablespace in the COPY-pending status	This, the NOCOPYPEND option will not turn off any copy pending options set prior the load operation. NOTE: Specify this option only if the data in this table can be recreated by another load operation if, for some reason, the data is lost. You may not be able to recover the tablespace if you don't take an image copy after the load operation. No is the default.

Enforce/Specify

Required Information	Description
Enforce	Specifies whether the load operation should enforce check and referential constraints. Constraints indicates constraints will be enforced. If the operation detects a constraint violation, the errant row is deleted and identified by a message. If you elect this option, and there are referential constraints, you must define sort input and output datasets. Default enforces the constraints. No specifies constraints are not enforced. It also puts the tablespace in the Check-pending status if at least one referential or check constraint exists.
Would you like to specify the Work DDN?	If you elected to enforce constraints, you must specify the work DDN. Here you establish DD names for the temporary sort input and output files. The default input is SYSUT1 and output is SORTOUT.
Would you like to specify the SORTKEYS?	This specifies that the index keys are to be sorted in parallel during the sort build phase to improve performance. If you want, you can specify an integer value to provide an estimate of the number of index keys that are to be sorted. The default is 0 if the input is on tape or a cursor.
Format	This selection is for the format of the input data. Unload specifies that the input record format is compatible with DB2 unload format. SQL/DS indicates the input record format is compatible with SQL/DS format.

Input/CCSID

Required Information	Description
Input data file type	Select between EBCDIC, ASCII, or Unicode.
CCSID(s)	These coded character set identifiers for the input file generally correspond to the input data file type. If that's not the case, the input data will be converted to the table CCSIDs before being loaded.
Do not accept substitution characters in a string	Specifies that the load operation should not accept substitution characters in the string. Use substitutions when a string is being converted from one data type to another. If you do not accept substitutions and the load operation determines that a substitution appears in the string because of a conversion, if discard processing is active, the record will be moved to the discard file. If discard processing is inactive, the load operation will terminate.
Float	S390 specifies that the load operation should expect floating point numbers are in the System/390 hexadecimal floating point format. IEEE specifies that the load operation should expect that floating point number are in IEEE binary floating point format. If you specify IEEE, the binary format floats will be converted to hexadecimal floating points. If there's a conversion error, the record goes in the discard pile.
Specify the error DDN	Here you specify the DD statement or template for a work dataset that's used during error processing. SYSERR is the default.

Map DDN/Discard DDN

Required Information	Description
Would you like to specify the Map DDN?	Specifies the DD statement or template for a work data set to be used in error processing. This is used to correlate the identifier of a table row with the input record that's causing the error. SYSMAP is the default.
Would you like to specify the Discard DDN?	Specifies the DD statement or template for a discard dataset that will hold copies for records that aren't loaded in the operation (e.g., a record with a conversion error). SYSDISC is the default.
Would you like to specify the maximum number of source records to be written on the discard data set?	Specify an integer with a value between 0 and 2146483647. If the maximum value is reached, the load operation abnormally terminates. When you enter 0, you are specifying that you do not want to set a maximum value and the entire input data set can be discarded.
Would you like to specify the sort device type?	Here you indicate, if available, the device type for temporary datasets that are to be dynamically allocated by DFSORT.

Specify Sort Number/CONTINUE-IF

Required Information	Description
Do you want to specify the sort number?	This is the number of temporary datasets that will be allocated by the sort application program.
Would you like to specify a CONTINUE-IF condition?	Yes indicates that you want to treat each input record as a portion of a larger record. If a record is true, the next record is concatenated with it before it is loaded.
Start/Finish	Specifies the column numbers of the first and last columns.
Hexadecimal	The byte-string value in the continuation field indicates that the next input record is a continuation of the current load record. Records with this byte-string value are concatenated until the value in the continuation field changes.
Condition	After the CONTINUE-IF completes, write a condition as X'byte-string' or 'character-string'.

Into Table Specification Dialog Box

At least one table INTO STATEMENT is required for each table that is to be loaded. Each Into Table Specification:

- Identifies the table that is to be loaded.
- Describes fields within the input record.
- Defines the format of the input dataset.

Note: All tables that are specified must belong to the same tablespace.

Required Information	Description
Page 1:	
Skip fields in the input data set that do not correspond to columns in the target table	Specifies whether or not the load operation should skip fields in the input dataset that do not correspond to tables in the target table.
Partition	An integer value that applies to the number of the partition where records are to be loaded. This is an option that is enabled only for partitioned tablespaces.
Would you like to preformat?	Specifies that any remaining pages should be preformatted up to the high-allocated RBA in the partition and its corresponding partitioning index space. Preformatting happens after the data is loaded and the indexes are built.
Load records into a non-empty partition	No loads records into an empty partition. No is the default. Yes loads records into a non-empty partition. If the partition is empty, you get a warning message, but the partition is loaded anyway.
Replace only the contents of the partition specified above, rather than the entire tablespace.	**Note:** You cannot use Load Replace with the partition integer replace option. You can either replace an entire tablespace using the load replace option or the single partition option here.
Reset and reuse DB2-managed data sets	When used with the replace option above, this specifies that the load operation should reset and reuse the DB2-managed datasets without deleting and redefining them.
Would you like to keep the dictionary?	This is an option only if a dictionary exists and the partition being loaded has the COMPRESS YES attribute. If the partition has the COMPRESS YES attribute but no dictionary exists, a dictionary is built and you get an error message.
Would you like to specify a cursor?	If desired, specify a cursor for the input dataset.
Select statement for cursor	
In DNN	Specifies the data definition statement or template for the input dataset for the partition. The default is SYSREC.
Discard DNN	Specifies the data definition statement or template for the partition. This dataset also holds copies of records that were loaded and removed. It must be a sequential data set.
Page 2:	
Do you want to set criteria for which records are loaded?	If you do not want to specify load criteria, all records in the input dataset are loaded, except for data that is beyond the range of the specified partition.
Do you want to use field selection criteria?	Yes or No.

SQL/DS	Table Owner: Select a table owner from the drop-down list. Table: This is the name of the table that the SQL/DS table is loaded into.
Field selection Criteria	The criteria describes a field and a character constant. A field must contain a character or graphic string. (When compared to a string constant, no datatype conversions are performed.) If the field and the constant are not the same length, the shorter of the two is padded with blanks before a comparison is made.
Fields	Click **Add** to open the Field Specification Dialog Box.

Field Specification Dialog Box

Required Information	Description
Field Name	The name of a field defined by a field specification. This can be any name you choose.
Would you like to specify a position?	Indicates where a field appears in the assembled load record. Start/End are the locations of the first and last field records. The first record is column 1.
Datatype	Pick the appropriate datatype from the drop-down list.
Length/Scale	Length is the length in bytes of the input field. Scale is the number of digits to the right of the decimal point and must be greater than or equal to 0. 0 is the default.
Would you like to add a condition that causes the db2 column to be loaded with NULL or it's default value?	NULLIF/CONDITION: Write a condition that causes the DB2 column to be loaded with NULL. DEFAULTIF/CONDITION: Write a condition that causes the DB2 column to be loaded with its default value. No

Set Dataset Info Dialog Box

Required Information	Description
Dataset Name	The name of the input dataset from which the table data should be loaded.
Template Name	
SMS: Management class	The management class name may not exceed 8 characters. This enables the dataset to be catalogued. MGMTCLAS name.
Storage class	The storage class name must be valid and cannot exceed 8 characters. STORCLAS name.
Data class	The data class name cannot exceed 8 characters. DATACLAS name.
Unit: Device Type	Specifies the device type or group name for the data set.
Count	
Volume: Private	A communications connection that's specific to DB2.
Sequence Number	
Count	
Serial no(s)	The serial number of the volume on which a new temporary or permanent data set resides.
Space: Unit	
Primary Allocation	
Secondary Allocation	
Release	
Label: Sequence Number	
Retention period	
Expiration date	

Load Utility - Tape Stacking Dialog Box

Tape stacking allows you to make four copies of the same data set simultaneously. DB2 OS390's Copy to Copy (DBArtisan) utility allows you to make multiple image copies as well.

Required Information	Description
Local Site Primary Copy	Name the data set where you want the output image copy to go at the local primary site. This relates to the copyddn1 parameter in the stored procedure.
Local Site Backup Copy	Name the data set where you want the output image copy to go at the local backup site. This relates to the copyddn2 parameter in the stored procedure.
Recovery Site Primary Copy	Name the data set where you want the output image copy to go at the recovery primary site. This relates to the recoveryddn1 parameter in the stored procedure.
Recovery Site Backup Copy	Identify the image set where you want the image copy to go at the recovery backup site. This relates to the recoveryddn2 parameter in the stored procedure.
In DDN	Identify the input data set that contains the table to be loaded.
Discard DDN	Identify the work data set that will hold copies of records not included in the load operation.

Data Load (DBArtisan - DB2 LUW)

DBArtisan offers a simple, six-panel wizard that lets you load data via DBArtisan's import utility. After defining the data import operation, you have the opportunity to save the job definition for future reuse.

Starting the Import Wizard

1 Connect to the datasource to which data is to be imported and ensure that it is the selected datasource. For details, see Connected/Selected Datasource options.

2 Select **Utilities > Data Load**. The **Data Import Wizard** opens.

3 Use the following table as a guide to providing details as you proceed through the wizard.

Step.Panel	Options/Description
1	Do you wish to initiate a new import operation or 'play back' a previously saved operation? New Data Load Operation - Select to define a new data load operation. Previous Data Load Operation - Select to open a previously defined schema extraction operation. In the Data Load Definition File Name box, type the location and name of the job file or click the browse button.
2	**What file has the data you wish to import?** - Lets you type the location and name of the file or click the browse button. **Specify the format of the file to be imported** - Lets you click the list and click the file format. Options Button - Click if you want to set the ASCII file type options for a delimited option. **What mode would you like to use for this import operation?** - Choose a mode: INSERT - The target table must exist and imported rows are inserted into the table without affecting existing rows in the table. INSERT_UPDATE - The target table must exist and have primary key constraint defined on it. Imported rows that match the existing row of data on which the primary key is defined are updated. New rows that do not match the primary key are inserted into the table. REPLACE - The target table must exist and all data is replaced with the contents of the import data file. CREATE - The table defined in the import data file is created and rows of data are inserted into the newly created table. You must use a *.IXF file to use this mode. REPLACE_CREATE - If the target table exists, the data will be replaced with the data contained in the import data file and the table definition remains unchanged. If the target table does not exist, the table defined in the import data file is created and rows of data are inserted into the newly created table. You must use a *.IXF file to use this mode. **What is the target object into which the data will be imported?** - Use one of the following: DATABASE- Lets you select the database. SCHEMA - Lets you select the schema. OBJECT - Lets you select the table or view.
3	**What method should be used for selecting columns for the export file?** - lets you specify a method: **Default**, **Position**, **Location**, or **Names**. **Indicate the columns into which data should be imported, specifying the relevant column selection attributes** - Lets you select the columns. Click **Edit** to open the **Column Position Attributes** dialog. **Include** - Select to include the column. **Position** - Lets you type the position.
4	**Would you like to retrieve large objects (LOB's) from separate files?** - specify LOB options: Add Button - Click to open the **Browse for Folder** dialog. Click the LOB path. Edit Button - Click to open the **Browse for Folder** dialog. Remove - Click to remove the LOB path. **Specify a path and name for the message file** - Type the directory location of the message file or click the **Browse** button to find the file.
5	How many of the file's records should be skipped before the import begins? - RESTART lets you type the value of the number of rows that should be skipped before the import operation begins. How many records should be imported before committing them to the database? - COMMENT RECORDS lets you type the value of the number of records that should be imported before they are committed to the database. **Use of Compound SQL may improve performance.** - STATEMENTS lets you type the value of the number, which cannot exceed 100, of the number of SQL statements per compound block.
6	**Do you wish to re-use this data unload operation?** - To re-use the operation, type the location and name of the data load definition file or click the **Browse** button. DBArtisan gives the operation a default extension of *.LOAD.
7	Review the data load operation description to verify its accuracy. To make any changes, click **Back** to scroll back to the appropriate panels of the wizard to make your corrections.

4 When ready, click **Execute**.

Data Load (DBArtisan - SQL SVR, SYB ASE)

DBArtisan offers a simple, six-step wizard that can load data via Microsoft SQL Server's BCP (bulk copy program) utility or via Sybase ASE's BCP (bulk copy program) utility. After defining the data import operation, you have the opportunity to save the job definition for future re-use.

Tip: For related information, see Data Unload (DBArtisan - SQL SVR, SYB ASE).

To load table data

1 Connect to the datasource where the data is to be loaded and ensure that it is the selected datasource. For details, see Connected/Selected Datasource options.

2 Select **Utilities > Data Load**. The **Data Load Wizard** opens.

3 The options and steps involved depend on the DBMS type and on the selections you make. Use the following table as a guide to providing details as you pass through the pages of the wizard.

Option	Description
Do you wish to initiate a new data unload operation or 'play back' a previously saved operation?	New Data Unload Operation - Select to define a new data unload operation. 'Playback' Previous Data Unload Operation - Select to use a previously defined schema extraction operation. Then in the Data Unload Definition File Name box type the location and name of the job file or click the browse button.
Which database do you want to load the data into?	Lets you click the list, and then select the server. Then click the list and select the database.
Which table do you want to load the data into?	Lets you click the list, and then select the table.
Use BCP Utility	Select to use the BCP (bulk copy program) utility.
Which rows do you want to import?	All Rows Range - Lets you type the range in the First Row and Last Row boxes.
What restrictions do you want to place on the import?	**Max Errors** or **Batch Size**.

The following options are available against SQL Server datasources.

Data File	Lets you type the directory location of the data file or click the browse button to find the file.
Format	To use a format file, type the directory location of the format file or click the browse button to find the file.
Error File	To output errors to an error file, type the directory location of the format file or click the browse button to find the file.
What character delimited the columns in the datafile?	Lets you click the list, and then click the character.
Do you wish to re-use this data load operation?	To re-use the operation, type the location and name of the data load definition file or click the browse button to find the file. DBArtisan gives the operation a default extension of *.LOAD.
Do you wish to DROP the index(es) on the target table BEFORE copying data?	Drop Index(es) Before Loading of Data - Select to drop the indexes on the target table before copying data. Recreate Index(es) After Loading of Data - Select to recreate these indexes after the data loads.

The following options are available against Sybase ASE datasources.

Row Terminator	Lets you enter a value for this common BCP option.
Packet Size	Lets you enter a value for this common BCP option.
Text/Image Size	By default, Sybase ASE only copies the first 32K of data in a text or image field. To override default size, type a new size.
Client Char Set	Lets you enter a value for this common BCP option.

Max Connections	Lets you specify the maximum connection of connections in the connection pool.
Show-fi	Provides the option to add columns for each functional index.
Hide-vcc	Provides the option to hide virtual computed columns

4 Review the data unload operation description to verify its accuracy. If you need to make any changes, click **Back** to scroll back to the appropriate panels of the wizard to make your corrections. When ready, click **Execute**.

Data Load (DBArtisan - PostgreSQL)

This wizard lets you build and submit a COPY... FROM... call, copying data from a file to a table.

Note: For information on how to unload table data to a file suitable for loading using the Data Load utility, see Data Unload (DBArtisan - PostgreSQL).

To load table data from a file

1 Connect to the datasource where the database to be loaded is located and ensure that it is the selected datasource. For details, see Connected/Selected Datasource options.

2 Select **Utilities > Data Load** to open the **Data Load** WIzard.

3 Use the following table as a guide to performing tasks as you pass through the wizard panels:

Step/Page	Option	Description
Playback	**New Data Load Operation**	Select to define a new backup operation.
	Previous Data Load Operation	Select to open a previously defined unload operation.
Table	**Which database do you want to load data into and Which table do you want to load data into**	Lets you fully qualify the database table to which data is to be loaded.
Options	**File Name**	Lets you provide the path and name of the file containing the data to be loaded.
	Format	Lets you specify whether the data file is in text, CSV, or Binary format.
	Encoding	Lets you specify either UTF8 or SQL-ASCII encoding.
	OIDS	When selected, specifies copying the OID for each row.
	Header	Specifies that the file contains a header line consisting of all column names.
	Delimiter	The character that separates columns within each line.
	Quote	Specifies the quote character for a Format of csv.
	Escape	Specifies the character that should appear before a QUOTE data character value for a Format of csv.
	Null String	Specifies the string that represents a NULL value.

	Columns to Import	Select the columns that are to be copied.
Save Operation	**Do you wish to re-use this data load operation**	Lets you save the options you selected in a data load definition file.
	Drop Index(es) BEFORE Loading of Data and Recreate Index(es) AFTER Loading of Data	Lets you decide whether you want indexes dropped and specify the timing of the DROP.
Summary	Provides a short summary of the operation options.	

4 Finally, use the **Execute** button to perform the backup.

Data Pump Export (DBArtisan - ORCL10g ^)

The Data Pump Export utility lets you build and submit a call to the **expdp.exe** utility to unload a dump file specifically formatted to be loaded using the **impdp.exe** utility. The wizard lets you specify a full set of **expdp.exe** parameters, which are also saved to file for convenient reuse. When ready to import, you can execute from the wizard, or independently using a command line call to **expdp.exe** specifying the saved parameters file.

Before using this utility:

- You must specify the location of the **expdp.exe** utility. For more information, see Oracle Utilities Options.
- Consult Oracle documentation for detailed information on use of the **expdp.exe** utility. For more information, see Accessing Third Party Documentation.
- You should also familiarize yourself with support for the Oracle Data Pump Import Utility. For details, see Data Pump Import Wizard (DBArtisan - ORCL10g ^).

To create and execute a call to the Data Pump Export utility

1 Connect to an Oracle datasource and ensure that it is the currently selected datasource. For details, see Connected/Selected Datasource options.

2 Select **Utilities > Data Pump Export** to open the Data Pump Export Wizard.

3 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- Data Pump Export Wizard - Playback.
- Data Pump Export Wizard - Export Data Options.
- Data Pump Export Wizard - Object Selection.
- Data Pump Export Wizard - Table Data Filter.
- Data Pump Export Wizard - File and Job Settings.
- Data Pump Export Wizard - Advanced Parameters.
- Data Pump Export Wizard - Summary.

Finally, use the Finish button to perform the export.

Data Pump Export Wizard - Playback

When exporting a dump file set using the Data Pump Export Wizard, this panel offers the following options:

Required Information	Description
New Data Pump Export	This initiates the process of creating a new export operation.
Use Existing Parameter File	If you want to replay an export operation, when you select the radio button, the browse function is enabled so you can find the **.txt** file you want to reuse.

Data Pump Export Wizard - Export Data Options

This wizard page lets you work with the following parameters:

Setting Group	Description
Export Mode	Lets you select a mode, specifying the dump file set content type to be exported: **Tables**, **Schemas**, **Full Dump File** (corresponds to FULL parameter), or **Tablespaces**.
Options	Select a **Data or DDL** value of ALL, DATA ONLY, or METADATA only to provide a CONTENT= parameter value, specifying the data to unload. Select a **Compression** value of ALL, DATA ONLY, or METADATA only to provide a COMPRESSION= parameter value, indicating what to compress before writing to the dump file set.

Data Pump Export Wizard - Object Selection

If you selected an **Export Mode** of **Tables**, **Schemas**, or **Tablespaces**, select the check boxes corresponding to the specific database objects to be exported.

For more general information, such as how to open this Wizard, see Data Pump Import Wizard (DBArtisan - ORCL10g ^).

Data Pump Export Wizard - Table Data Filter

This wizard page lets you include one or more QUERY= parameters, each specifying any valid QUERY= clause, such as a WHERE clause or ORDER BY clause. For each table that is to be filtered, click in the **QUERY Clause** field associated with the table and type a valid clause.

Note: Use the fields in the <ALL> row to filter the list.

For more general information, such as how to open this Wizard, see Data Pump Import Wizard (DBArtisan - ORCL10g ^).

Data Pump Export Wizard - File and Job Settings

This wizard page lets you work with the following parameters:

Setting Group	Description
Dump File	**Directory** - Select the directory object corresponding to the file system location where the dump file is to be stored. **File name** - Specify the full name of the dump file. Selecting **Overwrite existing dump files** adds a REUSE_DUMPFILES=Y parameter, ensuring that any existing dump files are overwritten. Selecting **Existing files will prevent export** adds a REUSE_DUMPFILES=N parameter.
Enable Logging	If **Enable Logging** is unselected, a NOLOGFILE=Y parameter is included with the utility call. To enable logging, select **Enable Logging** and if required, overwrite the default **Log File** name. This includes a LOGFILE= parameter with the utility call.
Job	Lets you provide a JOBNAME= parameter value that will be included with the utility call.

For more general information, such as how to open this Wizard, see Data Pump Import Wizard (DBArtisan - ORCL10g ^).

Data Pump Export Wizard - Advanced Parameters

The Data Pump Import Wizard builds and submits a call to the **expdp.exe** utility, specifying a minimal, required set of parameters, specified on preceding pages of the wizard. The **Advanced Parameters** page lets you provide any additional *parameter*=*value* pairs supported by the utility.

Note: For a listing of additional, supported parameters, see Data Pump Import/Export Parameters that can be Added Manually (DBArtisan).

- Use the **New** button to open the **Add Data Pump Parameter** dialog. Select a parameter name from the **Name** dropdown and in the **Value** box, provide the associated value.
- Use the **Edit** and **Delete** buttons to modify or remove selected parameters.

Data Pump Export Wizard - Summary

This wizard page provides a summary of the **expdp.exe** parameters you chose on previous pages of the wizard. It also displays:

- The location of the generated parameters file (**.txt**).
- Instructions on how to execute the export independently of the wizard, using the parameters file

For more general information, such as how to open this Wizard, see Data Pump Import Wizard (DBArtisan - ORCL10g ^).

Data Pump Import Wizard (DBArtisan - ORCL10g ^)

The Data Pump Import utility lets you build and submit a call to the **impdp.exe** utility to load an **expdp.exe**-produced dump file set to a target system. The wizard lets you specify a full set of **impdp.exe** parameters, which are also saved to file for convenient reuse. When ready to import, you can execute from the wizard, or independently using a command line call to **impdp.exe** specifying the saved parameters file.

Before using this utility:

- You must specify the location of the **impdp.exe** utility. For more information, see Oracle Utilities Options.
- Consult Oracle documentation for detailed information on use of the **impdp.exe** utility. For more information, see Accessing Third Party Documentation.
- You should also familiarize yourself with support for the Oracle Data Pump Export Utility. For details, see Data Pump Export (DBArtisan - ORCL10g ^).

To create and execute a call to the Data Pump Import utility

1 Connect to an Oracle datasource and ensure that it is the currently selected datasource. For details, see Connected/Selected Datasource options.

2 Select **Utilities > Data Pump Import** to open the Data Pump Import WIzard.

3 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- Data Pump Import Wizard - Playback.
- Data Pump Import Wizard - Import Data Options.
- Data Pump Import Wizard - Object Selection.
- Data Pump Import Wizard - Remapping.
- Data Pump Import Wizard - Import Options.
- Data Pump Import Wizard - Advanced Parameters.
- Data Pump Import Wizard - Summary.

4 Finally, use the **Finish** button to perform the import.

Data Pump Import Wizard - Playback

When importing a dump file set using the Data Pump Import Wizard, this panel offers the following options:

Required Information	Description
New Data Pump Import	This initiates the process of creating a new import operation.
Use Existing Parameter File	If you want to replay an import operation, when you select the radio button, the browse function is enabled so you can find the **.txt** file you want to reuse.

Data Pump Import Wizard - Import Data Options

This wizard page lets you work with the following parameters:

Setting Group	Description
Dump File	**Directory** - Select the directory object corresponding to the file system location of the dump file. **File name** - Specify the full name of the dump file.
Import Mode	Lets you select a mode, specifying the dump file set content type to be imported: **Tables**, **Schemas**, **Full Dump File** (corresponds to FULL parameter), or **Tablespaces**.
Options	Select a **Data or DDL** value of ALL, DATA ONLY, or METADATA only to provide a CONTENT= parameter valuer, specifying the data to unload.

Data Pump Import Wizard - Object Selection

If you selected an **Import Mode** of **Tables**, **Schemas**, or **Tablespaces**, select the check boxes corresponding to the specific database objects to be imported.

Data Pump Import Wizard - Remapping

This wizard page lets you specify one or more REMAP_SCHEMA or REMAP_TABLESPACE parameters. For each source schema or tablespace that is to be remapped, click to activate the associated **Target** dropdown and select the target schema or tablespace.

Data Pump Import Wizard - Import Options

This wizard page lets you work with the following parameters:

Setting or Group	Description
Action if table exists	Lets you specify a TABLE_EXISTS_ACTION parameter, telling the utility what to do if a table being created already exists. Provide a TABLE_EXISTS_ACTION= value of **Skip**, **Append**, **Truncate**, or **Replace**.
Skip Unusable indexes	If selected, a SKIP_UNUSABLE_INDEXES=y parameter is included with the utility call. If unselected, a SKIP_UNUSABLE_INDEXES=n parameter is included.
Overwrite existing datafiles	Lets you specify whether existing datafiles should be reused for tablespace creation. If selected, a REUSE_DATAFILES=y parameter is included with the utility call. If unselected, a REUSE_DATAFILES=n parameter is included.
Enable Logging	If **Enable Logging** is unselected, a NOLOGFILE=Y parameter is included with the utility call. To enable logging, select **Enable Logging** and if required, overwrite the default **Log File** name. This includes a LOGFILE= parameter withe the utility call.
Job	Lets you provide a JOBNAME= parameter value that will be included with the utility call.

Data Pump Import Wizard - Advanced Parameters

The Data Pump Import Wizard builds and submits a call to the **impdp.exe** utility, specifying a minimal, required set of parameters, specified on preceding pages of the wizard. The **Advanced Parameters** page lets you provide any additional *parameter***=***value* pairs supported by the utility.

Note: For a listing of additional, supported parameters, see Data Pump Import/Export Parameters that can be Added Manually (DBArtisan).

- Use the **New** button to open the **Add Data Pump Parameter** dialog. Select a parameter name from the **Name** dropdown and in the **Value** box, provide the associated value.
- Use the **Edit** and **Delete** buttons to modify or remove selected parameters.

Data Pump Import Wizard - Summary

This wizard page provides a summary of the **impdp.exe** parameters you chose on previous pages of the wizard. It also displays:

- The location of the generated parameters file (**.txt**).
- Instructions on how to execute the import independently of the wizard, using the parameters file

Data Unload (DBArtisan)

The Data Unload utility is available on the following platforms:

- Data Unload Wizard (DBArtisan - DB2 LUW)
- Data Unload (DBArtisan - SQL SVR, SYB ASE)
- Data Unload (DBArtisan - PostgreSQL)

Data Unload Wizard (DBArtisan - DB2 LUW)

DBArtisan offers a simple, wizard that lets you export table data. After defining the data export operation, you have the opportunity to save the job definition for future reuse.

Starting the Export Wizard

1 Connect to the datasource containing the data to be unloaded and ensure that it is the selected datasource. For details, see Connected/Selected Datasource options.

2 Select **Utilities > Data Unload**. The **Data Export Wizard** opens.

3 The options offered depend on the selections you make. Use the following table as a guide t completing the pages of the wizard.

Step	
New or playback action	Do you wish to initiate a new export operation or 'play back' a previously saved **operation?** New Export Operation - Select to define a new data unload operation. Previous Export Operation - Select to use a previously defined schema extraction operation. Then in the Export Definition File Name box type the location and name of the job file or click the browse button.
Datasource and format selection	**What datasource has the data you wish to export? -** Lets you click the list and select the datasource. **Specify the format for the data export file(s) to be generated. -** Lets you click the list and select the file format. **Option Button -** Available if you select a delimited option. **How would you like to specify the data to export? -** Enter a SELECT statement. Select tables/views from a list (one export file per table/view).
SELECT statement and LOB options	**Specify the data to be extracted from the database by entering a SELECT statement -** Enter the statement that will retrieve the row(s) or column(s) you want from one or more tables. Statements can be complicated. Go to http://msdn.microsoft.com/libraryand do a search on SELECT for help. **Would you like to store large objects (LOBs) in separate files? - When you choose to store LOBs in separate files, you will be given the chance to store them in files outside the database tablespaces.When you choose not to store your LOBs in separate files, they will exist in a table row in the database.**
Lob file details	If you specified that LOBs are to be stored in separate files, the following options are provided: **List paths where the LOB files are to be stored -** Add, Edit, or Remove files where you want LOBs to be stored. **List LOB file names -** Add, Edit, or Remove LOB file names.
Table selection details	**Owner -** Lets you select the tables and/or views you want to export or click the **Check All** button. **Count Rows -** Lets you update the table row counts. **Check All -** Lets you select all tables or views. **Uncheck All -** Lets you deselect all tables or views.
Output options	**Select a directory for export files (including and LOB files) -** Type the location and name of the export file or click the **Browse** button. **Specify a path and name for the message file -** Type the location and name of the message file or click the **Browse** button. **Do you wish to re-use this data unload operation? -** To re-use this data unload operation, in type the location and name of the data unload definition file or find the file by clicking the **Browse** button. DBArtisan writes a migrate definition file using the *.unload file extension that includes your job definition.
Options review	Review the data export operation description to verify its accuracy. If you need to make any changes, click **Back** to scroll back to the appropriate panels of the wizard to make your corrections.

4 When ready, click **Execute**.

Data Unload (DBArtisan - SQL SVR, SYB ASE)

DBArtisan offers a simple, five-panel wizard that can export table data via Microsoft SQL Server's BCP (bulk copy program) utility or Sybase ASE's BCP (bulk copy program) utility. After defining the data export operation, you can save the job definition for future re-use.

Tip: For related information, see Data Load (DBArtisan - SQL SVR, SYB ASE).

To unload table data

1 Connect to the datasource containing the data to be unloaded and ensure that it is the selected datasource. For details, see Connected/Selected Datasource options.

2 Select **Utilities > Data Unload**. The **Data Unload Wizard** opens.

3 The options and steps involved depend on the DBMS type and on the selections you make. Use the following table as a guide to providing details as you pass through the pages of the wizard.

4

Panel	Option	Description
1	**Do you wish to initiate a new data unload operation** or 'play back' a previously **saved operation?**	New Data Unload Operation - Select to define a new data unload operation. 'Playback' Previous Data Unload Operation - Select to use a previously defined schema extraction operation. Then in the Data Unload Definition File Name box type the location and name of the job file or click the browse button.
2	**In which directory should the unloaded files be placed?**	Lets you specify the directory by typing the location and name of the directory or find the directory by clicking the browse button.
	Which database has the data you wish to unload?	Lets you click the list, and then select the server. Then click the list, and then click the database.
	What character should delimit columns in the unloaded data files?	Lets you click the list, and then select the character.
3	The following options are available against SQL Server datasources.	
	Table Owner	Lets you select the target tables to unload.
	Upd Stats button	Select the target tables and click button to update the table row counts.
	Select All button	Click to select all tables.
	Unselect All button	Click to unselect all tables.
	The following options are available against Sybase ASE datasources.	

	Row Terminator	Lets you enter a value for this common BCP option.
	Packet Size	Lets you enter a value for this common BCP option.
	Text/Image Size	By default, Sybase only copies the first 32K of data in a text or image field. To override default size, type a new size.
	Client Char Set	Lets you enter a value for this common BCP option.
4	**Do you wish to re-use this data unload operation?**	To re-use this data unload operation, type the location and name of the data unload definition file or find the file by clicking the **Browse** button. DBArtisan writes a migrate definition file using the *.unload file extension that includes your job definition.

5 Review the data unload operation description to verify its accuracy. If you need to make any changes, click **Back** to scroll back to the appropriate panels of the wizard to make your corrections.

5 When ready, click **Execute**.

Data Unload (DBArtisan - PostgreSQL)

This wizard lets you generate a pgadmin export data call, dumping database data to a file.

To dump database data to a file

1 Connect to the datasource where the data to be unloaded is located and ensure that it is the selected datasource. For details, see Connected/Selected Datasource options.

2 Select **Utilities > Data UnLoad** to open the **Data UnLoad Wizard**.

3 Use the following table as a guide to performing tasks as you pass through the wizard panels:

Step/Page	Option	Description
Playback	**New Data Unoad Operation**	Select to define a new unload operation.
	Previous Data Unload Operation	Select to open a previously defined unload operation.
Unload Properties	**In which directory should the unload files be placed**	Lets you locate and select the path where the unload file will be written.
	What datasource has the data you wish to unload	Lets you select the datasource that is to be unloaded.
Unload Properties	Use the table list to select each specific table that is to be unloaded. The Schema dropdown can be used to reduce the number of tables displayed to only those objects within a selected schema.	
Unload Properties	**Row Separator**	Lets you specify whether simple linefeeds or carriage return/linefeed combinations are used to separate rows.
	Column separator	Lets you specify whether colons, semi-colons, or vertical bars are used to separate columns.
	Quote char	Lets you specify whether single quote or double quotes are used for quoting.
	Column names	Select this option to have the first line of the unload file contain column names.
	Encoding	Lets you specify whether you want to use simple, local character set encoding or whether to use the Postgres-specific UTF-8 encoding.
	Quoting	Lets you specify a quoting strategy of no quoting used, quoting on string columns only, or quoting on a ll columns.

Re-use	**Do you wish to re-use this data unload operation**	Lets you save the options you selected in a data unload definition file.
Summary	Provides a short summary of the operation options.	

4 Finally, use the **Execute** button to perform the unload.

Note: The data unloaded to file can be used in a subsequent load operation. For details, see Data Load (DBArtisan - PostgreSQL).

Database Backup (DBArtisan)

See the following topics for DBMS version-specific database backup wizards:

- Database Backup (DBArtisan - MySQL)
- Database Backup (SQL SVR)
- Database Backup (DBArtisan - SYB ASE)
- Database Backup (DBArtisan - PostgreSQL)

Database Backup (DBArtisan - MySQL)

The MySQL Database Backup WIzard lets you build and submit a `mysqldump` call, backing up databases to results files.

To back up a MySQL database

1 Connect to a MySQL datasource and ensure that it is the currently selected datasource. For details, see Connected/Selected Datasource options.

2 Select **Utilities > Database Backup** to open the Database Backup WIzard.

3 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- Playback panel - Database Backup Wizard
- Backup Type panel - Database Backup Wizard
- Databases panel - Database Backup Wizard
- Tables panel - Database Backup Wizard
- Where Clause panel - Database Backup Wizard
- Options panel - Database Backup Wizard
- Output File Type panel - Database Backup Wizard
- Normal Output File panel - Database Backup Wizard
- Datafile Page panel - Database Backup Wizard
- Summary panel - Database Backup Wizard

4 Finally, use the **Finish** button to perform the backup.

For information on restoring a database from results files, see Database Restore Wizard (MySQL).

Playback panel - Database Backup Wizard

When backing up a MySQL database, this panel offers the following options:

Required Information	Description
New Backup/Dump Operation	This initiates the process of creating a new backup or dump operation.
Enter the views SQL definition	If you want to replay a backup operation, when you select the radio button, the browse function is enabled so you can find the .DBDump file you want to reuse.

Backup Type panel - Database Backup Wizard

When backing up a MySQL database, this panel offers the following options:

Setting	Description
Backup databases	Lets you backup one or more entire databases.
Backup tables within a single database	Lets you backup one or more tables within a single database.

Databases panel - Database Backup Wizard

When backing up a MySQL database, this panel is only available if you selected the **Backup databases** option. For details, see Backup Type panel - Database Backup Wizard.

To specify the databases that are to be backed up

1 Select each check box corresponding to the name of a database you want to back up.

Tables panel - Database Backup Wizard

When backing up a MySQL database, this panel is only available if you selected the **Backup tables within a single database** option. For details, see Backup Type panel - Database Backup Wizard.

To specify the tables that are to be backed up

1 From the **Select tables from** list, select the database containing the tables you want to back up.

2 In the tables list, select each check box corresponding to the name of a table you want to back up.

Where Clause panel - Database Backup Wizard

When backing up a MySQL database, this panel is only available if:

- You selected the **Backup tables within a single database** option. For details, see Backup Type panel - Database Backup Wizard.
- You selected a single table from the **Tables** panel. For details, see Tables panel - Database Backup Wizard.

To dump only specific records

1 In the **WHERE clause** box, type a WHERE clause qualifying those records you want to dump.

Options panel - Database Backup Wizard

When backing up a MySQL database, this panel lets you set or select the `mysqldump` options dictating how the backup is to be performed.

Note: Before working with these options, consult MySQL documentation for information on the mysqldump program. For more information, see Accessing Third Party Documentation.

The following table shows the Database Backup Wizard settings on this panel, their `mysqldump` option equivalents, and a description of their effect.

Wizard Option	mysqldump Option Equivalent	Description
No Data	--no-data	Table contents are not dumped
Continue if Errors Occur	--force	Continues the dump even if SQL errors occur.
Flush Server Logs	--flush-logs	Flushes server logs before starting the dump.
Drop Table Statements	--add-drop-table	Adds a DROP TABLE statement before each CREATE TABLE statement.
Lock/Unlock Statements	--add-locks	LOCK TABLE and UNLOCK TABLE statements are generated for each table dump.
Lock All Tables	--lock-tables	Locks all tables before dumping them.
Allow Keyword Column Names	--allow-keywords	Allows creation of column names that contain keywords.
Complete Insert Statements	--complete-insert	Uses complete INSERT statements, including column names.
Use MySQL Table Options	--create-options	All MySQL-specific table options are included in CREATE TABLE statements.
Use Insert Delayed Options	--delayed-insert	Writes INSERT DELAYED statements instead of INSERT statements.
Disable Keys	--disable-keys	Each INSERT statement is surrounded by disable/enable key statements.
Use Extended Inserts	--extended insert	Generates multiple-row INSERT syntax that includes several VALUES lists.
Use Create Table Statements	--no-create-info	Scripts CREATE TABLE statements for each table.
Suppress Information	--skip-comments	Additional information such as program version, server version, and host, is not written to the dump file.
Exclude Create Database Statements	--no-create-db	Suppresses CREATE DATABASE statements.
Retrieve Single Rows	--quick	Rows are retrieved one at a time.
Use Quotes	--quote-names	Identifiers such as database, table, and column names are generated within quote characters.
Use Begin Statements	--single-transaction	Issues a BEGIN statement before dumping data from the server.

Use First-Slave Option	--first-slave	Locks all tables across all databases.
Use Master-Data Option	--master-data	Deselected, a CHANGE MASTER statement is written as an SQL comment.
Delete Binary Logs	--delete-master-logs	Deletes binary logs after the dump operation on a master replication server.
Suppress Set Names Statement	--skip-set-charset	Suppresses SET NAMES default_character_set statements.
Default Character Set	--default-character-set=charset	Uses the specified charset as the default character set.
Protocol	--protocol=	The connection protocol (TCP, SOCKET, PIPE, or MEMORY) to use.
Max Allowed Packet	max_allowed_packet=	The maximum size of the client/server communication buffer.
Net Buffer Length	net_buffer_length=	The initial size of the client/server communication buffer.
Compress	--compress	Compresses information sent between client and server if compression support is available.
ANSI	--compatible=ansi	These settings enable dump output compatibility modes.
MySQL 3.23	--compatible=mysql323	
MySQL 4.0	--compatible=mysql40	
PostgresSQL	--compatible=postrgreql	
Oracle	--compatible=oracle	
MSSQL	--compatible=mssql	
DB2	--compatible=db2	
MaxDB	--compatible=maxdb	
No Key Options	--compatible=no_key_options	
No Table Options	--compatible=no_table_options	
No Field Options	--compatible=no_field_options	

Output File Type panel - Database Backup Wizard

When backing up a MySQL database, this panel lets you select between standard output or use of the --tab option. It offers the following options:

Setting	Description
Create a normal output file	Produces standard output.
Create a tab-separated output file	For each dumped table, mysqldump creates a 'tbl_name.sql' file that contains the CREATE TABLE statement that creates the table, and a 'tbl_name.txt' that contains its data.

For more general information, such as how to open this Wizard, see Database Restore Wizard (MySQL).

Normal Output File panel - Database Backup Wizard

When backing up a MySQL database, this panel lets you provide results file options (--result-file=file and --xml). It offers the following options:

Setting	Description
Enter the path and filename of the results file	Lets you locate and provide/select a path and filename for the results file.

Prevent newline **"\n"** characters from being converted to **"\r\n"** carriage return/newline sequences **(recommended for Windows systems)**

Write dump output as well-formed XML

Datafile Page panel - Database Backup Wizard

When backing up a MySQL database, this panel is only available if

- You selected the **Backup tables within a single database** option. For details, see Backup Type panel - Database Backup Wizard.
- You selected the **Create a tab-separated tab file** option. For details, see Output File Type panel - Database Backup Wizard.

It offers the following options:

Setting	Description
Enter the directory where the file(s) will be stored	Lets you provide a --tab=path option value.
Fields Terminated By, **Fields Enclosed By**, **Fields Optionally Enclosed By**, **Fields Escaped By**, and **Lines Terminated By**	Lets you provide --fields-xxx and --lines--xxx option values for line and column separators.

Summary panel - Database Backup Wizard

When backing up a MySQL database, this panel summarizes the choices you made in previous steps setting up the operation.

Database Backup (SQL SVR)

This wizard lets you generate a BACKUP DATABASE script to be executed immediately, scheduled, or opened in the SQL Editor.

To create a database backup

1 Connect to the datasource where the database to be backed up is located and ensure that it is the selected datasource. For details, see Connected/Selected Datasource options.

2 Select **Utilities > Database Backup**. The **Backup Wizard** opens.

3 Use the following table as a guide to providing details as you pass through the pages of the wizard.

Step.Panel		Options/Description	
Backup Operation	**New Backup Operation**	Select to define a new backup operation.	
	Previous Backup Operation	Select to open a previously defined backup operation and use the associated controls to provide the location and name of the job file.	
Database Selection	Here you select the database you want to backup from a drop-down list and name the backup operation. You can keep the default name for the operation or replace it with one you type. The job description is optional and you can accept the default or write another.		
Backup Type	**Full Database**	Backs up the entire database. The size of the database is listed in parentheses.	
	Partial Database	Generates a BACKUP DATABASE statement that includes a READ_WRITE_FILEGROUPS option, dictating a partial backup of specified files and file groups.	
	Partial Incremental	Generates a BACKUP DATABASE statement that includes WITH DIFFERENTIAL and READ_WRITE_FILEGROUPS options, dictating that only specified files and file groups changed since last backup should be backed up.	
	Incremental Database	Generates a BACKUP DATABASE statement that includes a WITH DIFFERENTIAL option, dictating that only potions of the database changed since last backup should be backed up.	
	File and File Group	These controls are only available if the	Select to specify a file or file group.

	Transaction Log	Database option **trunc. log on chkpt.** is deselected. For more information, see Databases (SQL Server) - Options.	**Full Truncate**- Backs up the inactive portion of the transaction log. The size of the transaction log is listed in parentheses. **With No Log** (not available for SQL Server 2008 only) - Removes the inactive portion of the transaction log without making a backup of it. You should only use this command when you run out of room in the database and cannot issue DUMP TRAN WITH TRUNCATE_ONLY. **Full No Truncate** - Makes it possible to dump a transaction log even if the database is inaccessible. You can use this option when the data portion of your database is damaged and the transaction log resides on a separate device. **With Truncate Only** (not available for SQL Server 2008 only) - Removes the inactive portion of the transaction log without making a backup of it. You should use this command prior to backing up a database.
Backup Location	This panels lets you provide values used to create a TO argument for the BACKUP DATABASE statement generated for this operation. Use the **Add**, **Remove**, and **Contents** buttons to manage the disk devices, tape devices, and directory/filenames that are to be specified on the TO argument.		
Backup and Media Information	Here, you specify the backup set name, backup description, backup set password, media name, media description, and media name password.		
Backup Options	**Overwrite all backup sets on the device**	When selected, a WITH INIT option is generated with the BACKUP DATABASE statement, dictating that existing backup sets are to be overwritten. When deselected, a WITH NOINIT option is generated with the BACKUP DATABASE statement, dictating that this operation is appended to the media set. Selecting this option enables the **Format the entire media before backup operation** and **Specify when the backup set can be overwritten** controls	
	Format the entire media before backup operation	When selected, a WITH FORMAT option is generated with the BACKUP DATABASE statement, dictating that a new media set be created. When deselected, a WITH NOFORMAT option is generated with the BACKUP DATABASE statement, dictating that the existing media header and backup sets on the media volumes be preserved.	
	Specify when the backup set can be overwritten	Selecting this check box activates the **Expire Date** and **Retain Days** controls. The **Expire Date** control lets you add a WITH EXPIREDATE option to the generated BACKUP DATABASE statement, specifying when the backup set can be overwritten. The **Retain Days** control lets you add a WITH RETAINDAYS option to the generated BACKUP DATABASE statement, specifying the number of days before the backup set can be overwritten.	

Skip backup set expiration and media name checking	When selected, a WITH SKIP option is generated with the BACKUP DATABASE statement, disabling checking of expiration and name for the backup set. When deselected, a WITH NOSKIP option is generated with the BACKUP DATABASE statement, ensuring that the expiration date is checked before overwriting backup sets. This control is only enabled if **Format the entire media before backup operation** is deselected.
Copy-Only backup	When selected, a WITH COPY-ONLY option is generated with the BACKUP DATABASE statement, specifying that this backup is a copy-only backup, with no effect on normal backup operations.
Compression (SQL Server 2008 only)	You can select among **Use the default server setting,Compress** (generate a WITH COMPRESSION option), and **No Compress** (generate a WITH NO_COMPRESSION option)
Read and verify the integrity of the backup after finish	If selected, a RESTORE VERIFYONLY is generated to follow the BACKUP DATABASE statement, verifying but not restoring the backup operation.
Perform backup checksums	When selected, a WITH CHECKSUM option is generated with the BACKUP DATABASE statement, enabling backup checksums. When deselected, a WITH NO_CHECKSUM option is generated with the BACKUP DATABASE statement, explicitly disabling backup checksums.
Stop on checksum error	This control is only enabled if **Perform backup checksums** is selected. When selected, a CONTINUE_AFTER_ERROR option is generated with the BACKUP DATABASE statement, ensuring that the backup will continue to run even if errors such as invalid checksums or torn pages. are encountered. When deselected, a STOP_ON_ERROR option is generated with the BACKUP DATABASE statement, instructing the backup to terminate i a page checksum does not verify.
Rewind the tape before unload	When selected, a WITH REWIND option is generated with the BACKUP DATABASE statement, forcing release and rewind of the tape. When deselected, a WITH NOREWIND option is generated with the BACKUP DATABASE statement, dictating that the tape remain open after the backup completes.
Unload the tape after backup	This control is only enabled if **Rewind the tape before unload** is selected. When selected, a WITH UNLOAD option is generated with the BACKUP DATABASE statement, forcing an unload of the tape after rewind. When selected, a WITH NOUNLOAD option is generated with the BACKUP DATABASE statement, leaving the tape loaded after the backup operation completes.
No Recovery and **Standby file** (Standard SQL Server backup style only)	These controls correspond to the mutually exclusive STANDBY = *UNDO FILE NAME* (letting you back up the tail of the log and leave the database in a read-only and standby state) and NO RECOVERY (letting you back up the tail of the log and leave the database in RESTORING state) clauses.
I/O buffers	Lets you provide a WITH BUFFERCOUNT = value for the BACKUP DATABASE statement.

	Max transfer size	Lets you provide a WITH MAXTRANSFERSIZE = value for the BACKUP DATABASE statement.
	Report at each	Lets you provide a WITH STATS = value for the BACKUP DATABASE statement.
Backup Operation Summary	Here you indicate whether or not you want to save the operation, and if so, where the file should be saved. Also displayed are the elements of the function you are creating. You can go back to earlier panels if you want to make changes before the job runs.	
Action Selection	**Open the backup script in an ISQL editor**	The backup script will be opened into a new ISQL session. NOTE: This option is not available for Sybase backups.
	Perform the backup now	The backup will be executed immediately.
	Schedule the backup for later execution	The backup will be scheduled to run at a specified date/time.

4 When ready, click **Execute**.

Database Backup (DBArtisan - SYB ASE)

This wizard lets you generate and execute DUMP DATABASE or DUMP TRANSACTION commands.

To create a database backup

1 Connect to the datasource where the database to be backed up is located and ensure that it is the selected datasource. For details, see Connected/Selected Datasource options.

2 Select **Utilities > Database Backup**. The **Backup Wizard** opens.

3 The options offered depend on the sections you make as you progress through the pages of this wizard. Use the following table as a guide to providing details as you pass through the pages of the wizard.

Options/Description	
New Backup Operation	Select to define a new backup operation.
Previous Backup Operation	Select to open a previously defined backup operation and use the associated controls to provide the location and name of the job file.
What are the source Sybase server and database?	Select a server from the drop-down list and then the appropriate datasource.
What do you want to dump?	Database - Backs up the entire database. The size of the database is listed in parentheses. Transaction Log - Backs up the inactive portion of the transaction log. The size of the transaction log is listed in parentheses. Transaction Log With No Log - Removes the inactive portion of the transaction log without making a backup of it. You should only use this command when you run out of room in the database and cannot issue DUMP TRAN WITH TRUNCATE_ONLY. Transaction Log With Truncate Only - Removes the inactive portion of the transaction log without making a backup of it. You should use this command prior to backing up a database. Transaction Log With No Truncate - Makes it possible to dump a transaction log even if the database is inaccessible. You can use this option when the data portion of your database is damaged and the transaction log resides on a separate device.
What type of dump device do you want to use?	Choose among tape device, disk device, or file. When you select file, you need to type the file's name.
Where should messages about the dump operation be sent?	Your options are the Operator Console, Client, or Default.
Dump file	It's up to you to specify the name of the dump file.
What dump devices do you want to use?	Lets you select from available devices.
Do you want to specify any device handling options?	Lets you provide retention time and compression level details. If you selected a tape device, rewind and dismount options are provided.
Do you wish to re-use this dump operation?	To re-use the operation, type the location and name of the definition file or click the browse button to find the file. DBArtisan gives the operation a default extension of *.dbdump.

4 Review the data unload operation description to verify its accuracy. If you need to make any changes, click **Back** to scroll back to the appropriate panels of the wizard to make your corrections. When ready, click **Execute**.

Database Backup (DBArtisan - PostgreSQL)

This wizard lets you generate a pg_dump script, backing up one or more tables or views of a database.

To create a database backup

1 Connect to the datasource where the database to be backed up is located and ensure that it is the selected datasource. For details, see Connected/Selected Datasource options.

2 Select **Utilities > Database Backup** to open the **Backup Wizard**.

3 Use the following table as a guide to performing tasks as you pass through the wizard panels:

Panel	Option	Description
Playback	**New Backup Operation**	Select to define a new backup operation.
	Previous Backup Operation	Lets you specify the specific datasource and database where the tables and views to be backed up are located.
Datasource and Output Options	**What are the datasource and database**	Lets you specify the specific datasource and database where the tables and views to be backed up are located.
	Specify the format for the data file(s) to be generated	Lets you specify a -F (--format) output format option of Custom (custom-format archive suitable for input into Database Restore utility), Plain text (plain-text SQL script file), Directory (directory-format archive suitable for input into Database Restore utility), or Tar (TAR-format archive suitable for input into Database Restore utility).
	Compress ratio (not available if the Tar output format is selected)	Lets you provide a -Z option value of 0-9, specifying a pg_dump compression level.
	Encoding	Lets you provide a -E option value, specifying the character set encoding of the dump file.
	Rolename	Lets you provide a --role option value, instrucing pg_dump to issue a SET ROLE command on connection to the database.

Files	**Specify a path for the output file to be generated** and **Specify a path for the message file to be generated**	These settings lets you specify the paths for the output and message files as well as the optional file that lets you reuse the current backup operation.
Options	**Sections**	Lets you provide one or more --section option values (Pre-data. Data, Post-data) specifying that the selected sections are dumped.
	Type of Objects	Only data - corresponds to a -a (--data-only) option, specifying that only data be dumped. Only schema - corresponds to a -s (--schema-only) option, specifying that only object definitions be dumped. Blobs - corresponds to a -b (--sblobs) option, specifying that large objected be included in the dump.
	Don't save	Owner- corresponds to a -O (--no-owner) option, specifying no output commands to set ownership of objects to match the original database. Privilege - corresponds to a -X (--no-privileges) option, preventing dumping of access privileges. Tablespace - corresponds to a --no-tablespaces) option, specifying no output commands to select tablespaces. Unlogged table data - corresponds to a --no-unlogged-table-data option, specifying that the contents of unlogged tables are not dumped.
	Queries	Include CREATE DATABASE statement - corresponds to a -C (--create) option, specifying that the output begin with a command to create the database. Include DROP DATABASSE statement - corresponds to a -C (--clean) option, specifying that the commands to drop objects are specified before commands to create them.. Use Column Inserts - corresponds to a --column-inserts option, specifying that data is dumped as INSERT commands with explicit column names Use Insert commands - corresponds to a --inserts option, specifying that data is dumped as INSERT commands rather than COPY.
	Disable	Trigger- corresponds to a --disable-triggers option, specifying that the dump include commands to temporarily disable triggers on the target tables while data is reloaded. $ quoting-corresponds to a --disable-dollar-quoting option, specifying that dollar quoting is disabled for function bodies.
	Miscellaneous	Use SET SESSION AUTHORIZATION - corresponds to a --use-set-session-authorization option, specifying that object ownership is determined using SET SESSION AUTHORIZATION commands instead of ALTER OWNER commands. With OIDs - corresponds to a --oids option, specifying that object identifiers are dumped with every table. Verbose messages - corresponds to a --verbose option, specifying that verbose mode be used. Force double quotes on identifiers - corresponds to a --quote-all-identifiers option, forcing quoting of all identifiers.

Tables	**Select the tables and/or views to be exported**	Lets you select the tables that are to be dumped.
Summary	Provides a short summary of the dump operation options.	

4 Finally, use the **Finish** button to perform the backup.

The generated script can be used in a subsequent restore operation. For details, see Database Restore (DBArtisan - PostgreSQL).

Database Export (DBArtisan - ORCL)

DBArtisan provides comprehensive graphical support for the Oracle Export Utilities. DBArtisan offers a simple wizard that guides you through the process of extracting object definitions and table data from an Oracle database and creating binary export file. DBArtisan's Export Wizard ensures that the proper parameters and privileges are used, provides access to desired database information and resolves version compatibility differences.

The Export Wizard supports all four export modes:

- Table
- Users
- Full Database
- Point-in-time Recovery (available in Oracle8)

The benefits of using the Oracle Export Utilities range from moving data between different versions of Oracle quickly and efficiently, saving space on the database, to selectively backing up an entire or portion of a database using cumulative or incremental export functions.

Note: If you do not have the EXP_FULL_DATABASE role assigned to you, you may not be able to access all the functionality of the export wizard.

Starting the Database Export Wizard

1 Ensure that the path and name for the Oracle Export utility is specified in the **Oracle Utilities** tab of the Options Editor. For details, see Oracle Utilities Options.

2 Connect to the datasource where the database to be exported is located and ensure that it is the selected datasource. For details, see Connected/Selected Datasource options.

3 Select **Utilities > Database Export**. The **Data Export Wizard** opens.

4 Use the topics below as a guide to providing information as you walk through the pages of the wizard:

- Database Export Wizard - Panel 1
- Database Export Wizard - Panel 2
- Database Export Wizard - Panel 3
- Database Export Wizard - Panel 4
- Database Export Wizard - Panel 5
- Database Export Wizard - Panel 6
- Database Export Wizard - Panel 7
- Database Export Wizard - Panel 8

5 When ready, click **Execute**.

Database Export Wizard - Panel 1

The first panel of the Export Wizard lets you specify the source datasource, name an existing parameter file or use the existing Oracle default parameter file.

The table below describes the options and functionality on the first panel of the Data Export Wizard:

Option	Description
Which database has the data you wish to export?	Click the list, and then click the database.
Edit an existing parameter file	Type the location and name of the parameter file or click the browse button to find the file.
Default values	To use the Oracle default values for the parameter file, leave this box blank.

Database Export Wizard - Panel 2

The second panel of the Export Wizard lets you select the export mode that determines the database objects you will export.

- The full database mode makes available incremental, cumulative or complete export operations.
- The point-in-time export operation can export one or more tablespaces, which you can use in an import operation to recover a tablespace at a prior point in time without affecting the rest of the database.

The table below describes the objects, related objects and structures that are exported for each mode:

Mode	Object Types and Structures
Table	Table definitions, object types defined on the table, table data by partition, nested table data, indexes defined on the table, table constraints, table permissions, column and table comments, table analysis information, triggers that refer to the table, and auditing information.
User	Libraries, object types, database links, sequence numbers, cluster definitions, table definitions, object types defined on the table, table data by partition, nested table data, indexes defined on the table, table constraints, table permissions, table analysis information, column and table comments, private synonyms, stored procedures, functions, packages, auditing information, views, cluster analysis information, referential constraints, triggers, snapshots, and snapshot logs.
Full Database	Tablespace definitions, tablespace quotas, profiles, user definitions, roles, role privileges, system privileges, resource costs, rollback segment definitions, database links, sequence numbers, directories, libraries, object types, cluster definitions, password information, default and system auditing, object type tables, table definitions, table data by partition, nested table data, table indexes, table constraints, table privileges, table analysis information, column and table comments, auditing information, synonyms, views, stored procedures, functions, packages, triggers, cluster analysis information, snapshots, snapshot logs, refresh groups and children.
Point-In-Time (available in Oracle8)	Tablespaces.

The table below describes the options and functionality on the second panel of the Data Export Wizard:

Option	Description
Which mode would you like to use for this export?	Click the option button that corresponds to the export mode you want to implement: **Table**, **Use**r, **Full Database**, or **Point-in-Time Recovery**.

Database Export Wizard - Panel 3

The third panel of the Export Wizard lets you specify the objects you want to export based on the mode you selected in the second panel of the Export Wizard.

The table below describes the options and functionality on the third panel of the Data Export Wizard for Table mode:

Option	Description
Select the tables and/or partitions you would like to export	To display list of available tables, click the **Owner** list, and then click the owner.
Tables	In the box, select the check boxes that correspond to the tables you want to export. If a table has partitions, select the check boxes that correspond the partitions you want to export. To select all tables, click the **Select All** button.

The table below describes the options and functionality on the third panel of the Data Export Wizard for User mode:

Option	Description
Select the users whose objects you would like to export	To display list of available users, select the check boxes that correspond to the objects you want to export. To select all objects, click the **Select All** button.

The table below describes the options and functionality on the third panel of the Data Export Wizard for Full Database mode:

Option	Description
Would you like to perform an incremental export	An incremental operation exports all database objects that have changed since the last incremental, cumulative, or complete export operation. NOTE: To perform an incremental export operation, you must have the EXP_FULL_DATABASE role. A cumulative operation exports all database objects that have changed since the last cumulative or complete export. A complete operation exports all objects.
Do you want to records this export in the system tables that track exports?	To record the export operation in the SYS.INEXP, SYS.INCFIL, and SYS.INVID system tables, click the **Yes** button.

The table below describes the options and functionality on the third panel of the Data Export Wizard for Point-in-Time Recovery mode:

Option	Description
Select the tablespaces to be recovered	To display list of available tablespaces, select the check boxes that correspond to the tablespaces you want to recover. To recover all tablespaces, click the **Select All** button.

Database Export Wizard - Panel 4

The fourth panel of the Export Wizard lets specify:

- The objects you want to export.
- To view the progress meter.

The table below describes the options and functionality on the fourth panel of the Data Export Wizard:

Option	Description
Check the objects you would like to export	**Note:** This option is not available if you are performing a full database incremental export operation. The Oracle Export Utility automatically determines if a row has been changed since the last export operation. Select the check box that corresponds to the object(s) you want to export: To export indexes associated with the tables, select the Indexes check box. To export check constraints or referential integrity constraints, select the Constraints check box. To export object privileges, select the Grants check box. To export table data, select the Rows check box.
Rows/Dot	To view the progress meter which displays the status of rows being exported, in the box, type the value of the number of rows per dot you want displayed in the meter.

Database Export Wizard - Panel 5

The fifth panel of the Export Wizard lets you set the following options:

- The Export Path
- The Length of the File Record
- The Buffer Size

The table below describes the options and functionality on the fifth panel of the Data Export Wizard:

Option	Description
Would you like to perform a direct path export?	A direct path export extracts data much faster as it bypasses the SQL Command processing layer and saves data copies whenever possible.
Would you like to accept your platform's value for BUFSIZ as the length of the file record?	Lets you select the Yes or No option button. If you clicked the **No** option button, in the Specify length box, type the KB value of the length. The highest value you can specify is 64 KB.
Would you like to accept your platform's default value for the buffer size used to fetch rows?	**Note:** This option is only valid if you are performing a conventional path export. Lets you select the Yes or No option button. If you clicked the **No** option button, in the Specify size box, type the KB value of the length; the highest value you can specify is 64 KB.

Database Export Wizard - Panel 6

The sixth panel of the Export Wizard lets you specify:

- Handling storage of table data.
- Type of optimizer statistics to generate when the file is imported.

- Make the data remain unchanged during the export operation.

The table below describes the options and functionality on the sixth panel of the Data Export Wizard:

Option	Description
Would you like table data to be consolidated into one initial extent upon import?	Lets you select the No or Yes option button.
What type of database optimizer statistics should be generated upon import?	Lets you click the appropriate option button: **Estimate**, **Comput**e, or **None**.
Would you like to ensure that the data seen by this export operation does not change during execution?	**Note:** This option is not available for an incremental export operation. Lets you select the No or Yes option button. Click **Yes** to ensure that the data remains intact throughout the export operation.

Database Export Wizard - Panel 7

The seventh panel of the Export Wizard lets you specify names and locations for the export, parameter and logfiles.

The table below describes the options and functionality on the seventh panel of the Data Export Wizard:

Option	Description
Supply the export file name	Lets you type the full location and name of the export file. Export file names use the.dmp file extension. (for example, C:\Temp\EXPBerlin.dmp). If you expect to perform the same export operation again, type the location and name of the parameter file in the corresponding box. Parameter files are stored as.txt files. (for example, C:\Temp\EXPPARBerlin.txt).
Logfile	To capture informational and error messages, click the option button and type the location and name of the logfile in the corresponding box. Export logfile names use the.logfile extension. (for example, C:\Temp\EXPLOGBerlin.log).
None	If you do not want to capture informational and error messages, click the option button.

Database Export Wizard - Panel 8

Review the data export operation description to verify its accuracy.

Database Import (DBArtisan - ORCL)

DBArtisan provides comprehensive graphical support for the Oracle Import Utilities. DBArtisan offers a simple wizard that guides you through the process of extracting object definitions from an Oracle Export file and inputting them into the database. The Oracle Import utility only reads export files created with the Oracle Export utility. DBArtisan's Import Wizard ensures that the proper parameters and privileges are used, provides access to desired database information and resolves version compatibility differences.

The Import Wizard supports all four import modes:

- Table
- Users
- Full Database
- Point-in-time Recovery (available in Oracle8)

The Oracle Import utilities build database objects in the proper sequence to avoid dependency problems. Tables are created first, then table data is input into the proper tables, next indexes are built, then triggers are imported and finally integrity constraints are built.

Note: If you do not have the EXP_FULL_DATABASE role assigned to you, you may not be able to access all the functionality of the import wizard.

Starting the Import Wizard

1 Ensure that the path and name for the Oracle Import utility is specified in the **Oracle Utilities** tab of the Options Editor. For details, see Oracle Utilities Options.

2 Connect to the datasource where the database to be imported is located and ensure that it is the selected datasource. For details, see Connected/Selected Datasource options.

3 Select **Utilities > Database Import**. The **Data Import Wizard** opens.

4 Use the topics below as a guide to providing information as you walk through the pages of the wizard:

- Database Import Wizard - Panel 1
- Database Import Wizard - Panel 2
- Database Import Wizard - Panel 3
- Database Import Wizard - Panel 4
- Database Import Wizard - Panel 5
- Database Import Wizard - Panel 6
- Database Import Wizard - Panel 7
- Database Import Wizard - Panel 8

5 When ready, click **Execute**.

Database Import Wizard - Panel 1

The first panel of the Import Wizard asks you to specify the target datasource and to name an existing import parameter file or use the existing Oracle default parameter file.

The table below describes the options and functionality on the first panel of the Data Import Wizard:

Option	Description
Into which database would you like to import data?	Click the list, and then click the database.
Edit an existing parameter file	Type the location and name of the parameter file or click the browse button to find the file.
Default values	To use the Oracle default values for the parameter file, leave this box blank.

Database Import Wizard - Panel 2

The second panel of the Import Wizard lets you specify the export file from which the data should be imported. You must also select the import mode you want to use which determines the database objects that DBArtisan imports. The table below describes the objects, related objects and structures that are imported for each mode:

Mode	Object Types and Structures
Table	Table definitions, object types defined on the table, table data by partition, nested table data, indexes defined on the table, table constraints, table permissions, column and table comments, table analysis information, triggers that refer to the table, and auditing information.
User	Libraries, object types, database links, sequence numbers, cluster definitions, table definitions, object types defined on the table, table data by partition, nested table data, indexes defined on the table, table constraints, table permissions, table analysis information, column and table comments, private synonyms, stored procedures, functions, packages, auditing information, views, cluster analysis information, referential constraints, triggers, snapshots, and snapshot logs.
Full Database	Tablespace definitions, tablespace quotas, profiles, user definitions, roles, role privileges, system privileges, resource costs, rollback segment definitions, database links, sequence numbers, directories, libraries, object types, cluster definitions, password information, default and system auditing, object type tables, table definitions, table data by partition, nested table data, table indexes, table constraints, table privileges, table analysis information, column and table comments, auditing information, synonyms, views, stored procedures, functions, packages, triggers, cluster analysis information, snapshots, snapshot logs, refresh groups and children.
Point-In-Time (available in Oracle8)	Tablespaces.

The table below describes the options and functionality on the second panel of the Data Import Wizard:

Option	Description
Specify the export file from which you want to import data?	Lets you type the location and name export file to be used in the import operation, or find the file by clicking the **Browse** button.
Which database was used to create this export file?	Lets you choose the database used for export file creation from the list.
Which mode would you like to use for this export?	Lets you click the option button that corresponds to the import mode you want to implement. Table User Full Database Point-In-Time

Database Import Wizard - Panel 3

The third lets you specify which objects you want to import based on the mode you selected in the second panel of the Import Wizard.

The table below describes the options and functionality on the third panel of the Data Import Wizard for Table mode:

Option	Description
Select the tables and/or partitions you would like to import	Displays a list of available tables. Lets you click the **Owner** list, and then click the owner.
Tables	To In the box, select the check boxes that correspond to the tables you want to import. If a table has partitions, click the check boxes that correspond the partitions you want to import. To select all tables, click the **Select All** button.

The table below describes the options and functionality on the third panel of the Data Import Wizard for User mode:

Option	Description
Select the users whose objects you would like to import	To display list of available users, select the check boxes that correspond to the objects you want to import. To select all objects, click the **Select All** button.

The table below describes the options and functionality on the third panel of the Data Import Wizard for Full Database mode:

Option	Description
Would you like to perform an incremental import	An incremental operation imports all database objects that have changed since the last incremental, cumulative, or complete export operation.

Database Import Wizard - Panel 4

The fourth panel lets you specify options to import dependent objects, privileges and table data, and to create a file for index-creation commands.

The table below describes the options and functionality on the fourth panel of the Data Import Wizard:

Option	Description
Check the objects you would like to import	Select the check box that corresponds to the object(s) you want to export. To import object privileges, click the **Grants** check box. To import table data, select the Rows check box. To import indexes associated with the tables, select the Indexes check box.
Would you like to skip unusable indexes?	Lets you click the **Yes** or **No** option button.
Would you like specify a file to receive index-creation commands?	When this parameter is specified, index-creation commands for the requested mode are extracted and written to the specified file, rather than used to create indexes in the database. Tables and other database objects are not imported. If you clicked the **Yes** option button, type the location and name of the file or find the file by clicking the **Browse** button.

Database Import Wizard - Panel 5

The fifth panel of the Import Wizard lets you:

- Set the data rows transfer buffer size.
- Indicate if the import operation should commit after each array insert.
- Specify the option to re-use existing data files.
- Specify the option to ignore creation errors.

The table below describes the options and functionality on the fifth panel of the Data Import Wizard:

Option	Description
Would you like to accept your platform's default value for buffer size used to fetch rows?	If you select the No option button, in the Specify length box, type the KB value of the length. The highest value you can specify is 64 KB.
Should Import commit after each array insert?	Lets you select the No or Yes option button.
Would you like to reuse the existing datafiles making up the database?	Select Yes to set the reuse option in the datafile clause of the CREATE TABLESPACE command.
Should Import ignore object creation errors when attempting to create database objects?	Select Yes if you selected Table as your mode of import in panel 2, and the tables exist in the database in the target database.

Database Import Wizard - Panel 6

The sixth panel of the Import Wizard lets you specify:

- The length of the file record.
- How you want to implement SQL Analyze statements contained in the export file.
- To view the contents of the export file rather than import the database information.
- To view the progress meter.

The table below describes the options and functionality on the sixth panel of the Data Import Wizard:

Option	Description
Would you like to show the contents of the export file rather than import them?	Lets you select the No or Yes option button.
Would you like to view the progress meter as rows are imported	Lets you enter the number of rows per dot displayed in the meter box, type the value of the number of rows per dot you want displayed in the meter.
Would you like to accept your platform's value for BUFSIZ as the length of the file record?	If you select the No option button, in the Specify length box, type the KB value of the length; the highest value you can specify is 64 KB.
Would you like to accept your platform's default value for the buffer size used to fetch rows?	**Note:** This option is only valid if you are performing a conventional path export. If you click the **No** option button, in the Specify Size box, type the KB value of the length; the highest value you can specify is 64 KB.
Would you like Import to execute SQL Analyze statements found in the export file?	Lets you select the No or Yes option button.

Database Import Wizard - Panel 7

The seventh panel of the Import Wizard lets you specify names and locations for the parameter and logfiles. If you importing from an Oracle 6 export file, you can define the character set of the original file.

The table below describes the options and functionality on the seventh panel of the Data Import Wizard:

Option	Description
Character Set	For Oracle 6 export files, type the character set of the original file.
Logfile Button	Click to capture informational and error messages, and then type the location and name of the logfile in the corresponding box. Export logfile names use the *.logfile extension. (for example, C:\Temp\EXPLOGBerlin.log).
None Button	Click if you do not want to capture informational and error messages.
Parameter File	If you expect to perform the same export operation again, type the location and name of the parameter file in the corresponding box. Parameter files are stored as .txt files. (for example, C:\Temp\EXPPARBerlin.txt).

Database Import Wizard - Panel 8

Review the data import operation description to verify its accuracy.

Completing Panel 8

Review the import operation description to verify its accuracy.

1 If you need to make any changes, click **Back** to scroll back to the appropriate panels of the wizard to make your corrections.

2 If you are satisfied with the import operation, click **Execute**.

DBArtisan starts the Import operation.

Database Manager (DBArtisan)

DBArtisan lets you view and manage your databases through Database Manager. The Database Manager lets you:

- Add, modify, and delete database files.
- Add, modify, and delete transaction logfiles.
- Add or delete database options.
- View data space and transaction log usage.
- View DDL.

Opening the Database Manager

1 On the Datasource Explorer, select the **Databases** node.

DBArtisan displays the databases.

2 In the right pane of the Explorer window, click the database link(s).

3 On the **Utilities** menu, select **Database Manager.**

Database Monitor (DBArtisan)

The Database Monitor displays all processes connected to the server. The Database Monitor:

- Lets you scan databases for potential performance problems.
- Includes tabs that provide a wealth of information about your servers.
- Color-codes each process to reflect its state and let you identify problems quickly.

Tip: You can view some Database Monitor Information by clicking the **Performance** node of the Datasource Navigator/Explorer.

Important Notes

- Processes and Locks are results of DB2 API calls and are not located in the system catalog tables.
- The Database Monitor may display incomplete SQL due to limitations set by the DBMS. Platforms set limitations on the values returned by the commands that gather the information.

Opening the Database Monitor

1 On the **Utilities** menu, select **Database Monitor**.

DBArtisan opens the Database Monitor.

For more information, see:

- Color-coding
- Setting Scope Level in the Database Monitor
- Configuring the Database Monitor
- Refreshing the Database Monitor
- Database Monitor tabs

Color-coding

The Database Monitor color-codes displayed processes to reflect current state, to let you quickly spot problem conditions. For example, by default, process states are represented as follows:

Color	State
Yellow	Sleeping or Inactive
Green	Running
Red	Blocked by another process

You can use the Options Editor to customize the color scheme. For details, see DB Monitor Options (DBArtisan).

For more information, see Database Monitor (DBArtisan).

Setting Scope Level in the Database Monitor

Note: The functionality is available for IBM DB2 for z/OS and OS/390 only.

The Scope list lets you specify the scope of the element being monitored in a datasharing environment. Data Sharing lets one instance of the DB2 catalog to be shared equally among all members of the group.

The Scope list is available on the Traces, Functions, Procedures, and Utilities tabs.

To set the scope level, do the following:

1 On the Database Monitor, click the target tab.

2 On the Threads, Traces, Procedures and Functions tabs:

- In Scope, select Group for the monitoring results of all datasharing members in the set.

OR

- In Scope, select Local for the monitoring results for the member that your machine is are connected to.

DBArtisan displays the results in the Database Monitor.

3 On the Utility Tab:

- In Scope, select Group for the monitoring results of all datasharing members in the set.

OR

- In Scope, select Member for data by the member name.

DBArtisan displays the results in the Database Monitor.

Refreshing the Database Monitor

As an initial default, the Database Monitor display is refreshed every 90 seconds. There are two ways to change the rate at which displayed processes on a Database Monitor tab are refreshed:

- On-the fly, with a Database Monitor tab open, using the **Refresh Rate** control on the Database Monitor toolbar
- Setting a new initial value in the Options Editor. For details, see DB Monitor Options (DBArtisan).

1 On the Database Monitor tool bar, click the box to raise or lower the numbers, or type a new value in the box.

For more information, see Database Monitor (DBArtisan).

Configuring the Database Monitor

The Database Monitor (DBArtisan) lets you:

- Toggle to split the screen and display the target process SQL For details, see SQL Toggle.
- Open a SQL window and display the SQL. For details, see ISQL.
- View a Query Plan for SQL running for selected processes. For details, see Query Plan.
- Save custom column-widths on a DBMS platform basis. See Customizing column widths.

Customizing column widths

When you manually resize the width of a column on a Database Monitor tab, the new size is saved across startups of Database Monitor as well as DBArtisan for the current DBMS platform type.

SQL Toggle

DBArtisan lets you split the Database Monitor screen and display the SQL running for selected processes.

1 On the **Database Monitor** tool bar, click **Toggle**.

Note: To open a SQL window or view a Query Plan for SQL running on selected processes, you must have your Database Monitor Screen split to display SQL on the bottom.

For more information, see Configuring the Database Monitor.

ISQL

DBArtisan displays the SQL running for selected processes in a SQL Editor. You must turn on the SQL Toggle to display the target SQL in the split screen. To display the SQL in a separate window do the following:

1 On the Database Monitor, click the target process.

Note: To open a SQL Editor or view a Query Plan for processes, SQL Toggle must be on, displaying the target SQL in the split screen.

2 If SQL Toggle is not on, on the **Database Monitor** tool bar, click **Toggle**.

3 On the Database Monitor tool bar, click ISQL.

DBArtisan opens a SQL window with the target process SQL.

4 To execute the SQL, click the SQL Window tool bar, click **Execute**.

For more information, see Configuring the Database Monitor.

Query Plan

The Database Monitor lets you view the Query Plan for target processes. To view the Query Plan, do the following:

1 On the Database Monitor, select the target process.

Note: To open a SQL Editor or view a Query Plan for processes, SQL Toggle must be on, displaying the target SQL in the split screen.

2 If SQL Toggle is not on, on the **Database Monitor** tool bar, click **Toggle**.

3 On the Database Monitor tool bar, click **Plan**.

DBArtisan executes the SQL with query plan and opens a Results Editor with the results.

For more information, see Configuring the Database Monitor.

Session Details

Note: The functionality is available for Oracle only.

The Database Monitor lets you view user session details through the **User Session Information** dialog. Once you open the **User Session Information** dialog, you can view the SQL associated with the session.

The table below describes the tabs of the **User Session Information** dialog:

Tab	Description
Statistical Details	Lets you can scroll up and down to see parameters and values.
Waits	Displays either historic or current wait statistics and lets you display or filter out idle wait events. NOTE: Data returned for historic session waits is different than the data returned for current session waits.
Current SQL	Lets you view the SQL for this session.

Completing the Session Details Dialog Box

1 On the Database Monitor tool bar, click **Session**.

DBArtisan opens the **User Session Information** dialog.

2 On the **Statistical Details** tab, in the Session Parameters grid, you can scroll up and down to see parameters and values.

3 On the **Waits** tab, view select Historical Waits or Current Waits.

4 Select Include Idle Waits to display idle wait events.

5 On the **Current SQL** tab, view the SQL for this session.

6 When you finish viewing SQL, click **OK**.

DBArtisan returns to the **Database Monitor** dialog.

For more information, see Configuring the Database Monitor.

Database Monitor tabs

The following table describes the tabs available on the Database Monitor.

Tab	Description
Processes	The Database Monitor **Processes** tab displays all processes connected to the server and lets you scan databases for potential performance problems, view each process for potential performance problems, and view login time and seconds idle since last call. For more information, see Killing Processes.
Long Operations (ORCL)	Displays statistics from the Oracle `V$SESSION_LONGOPS` view or calculated from those base statistics. The primary criterion for an operation's inclusion in the `V$SESSION_LONGOPS` view is that the operation runs for more than six seconds. There are additional criteria for inclusion in the V$SESSION_LOGOPS view, including type of operation. Before working with this functionality, you should be familiar with the Oracle V$SESSION_LONGOPS view. For information on using Oracle documentation, see Accessing Third Party Documentation. The following columns are available for each operation listed on this tab: **SID** - The session identifier (SID). **START TIME** - The start time of the operation (START_TIME). **TIME REMAINING** - Estimated number of seconds before the operation completes (TIME_REMAINING). **PERCENT COMPLETE** - Completion status, expressed as a percentage (SOFAR / TOTALWORK) * 100). **UNITS SO FAR** - The units of work currently completed (SOFAR). **TOTAL WORK** - The total units of work for the operation (TOTALWORK). **UNIT OF MEASUREMENT** - The units of measurement (UNITS). **ELAPSED SECOND** - The number of seconds since the operation started (ELAPSED_SECONDS). **MESSAGE** - A statistics summary message (MESSAGE). **SERIAL #** - The Session serial number (SERIAL#). **OPERATION NAME** - A description of the operation (OPNAME). **TARGET OBJECT** - The object that the operation is carried out against (TARGET). **TARGET DESCRIPTION** - A description o the target object (TARGET_DESC). **LAST UPDATE TIME** - The time when statistics were last updated (LAST_UPDATE_TIME). **CONTEXT** - An application-provided value (CONTEXT). **USERNAME** - The identifier of the user executing the operation (USERNAME).
Threads (DB2 z/OS)	Displays the threads on the target datasource. For related information, see Setting Scope Level in the Database Monitor.
Locks	Displays all processes that are currently holding locks on an IBM DB2 database. Locks include: Shared Lock - A Shared Lock lets other users view and read objects concurrently. The Shared Lock owner or other concurrent processes cannot alter data in tablespaces, tables, or partitions. Exclusive Lock - DBArtisan promotes locks from Shared to Exclusive when a user decides to update an object. Once DBArtisan promotes the lock, the user must complete the update before the lock can return to shared mode; another user cannot demote a shared lock. Exclusive locks exclude other users from viewing updates before they are committed. There can only be one exclusive lock for a set of objects. The first thread gets the exclusive lock and holds it until the update is committed. Once the update is committed, the exclusive lock returns to shared mode. You cannot update a lock in DBArtisan.

Utilities (DB2 z/OS)	Displays wizards that are typically run from the command line and do not apply to a particular type of schema object. Some of these wizards are also available via the command menus for objects. The wizards include Check Data (DBArtisan), Check Index (DBArtisan), Copy to Copy (DBArtisan), Load Java, Merge Copy (DBArtisan), Modify Recovery (DBArtisan), Modify Statistics (DBArtisan), Rebuild Index (DBArtisan - DB2 z/OS), Reorganize /Reorg, Reorg (DB2 z/OS Tablespaces), Runstats Index (DBArtisan), Runstats Tablespace (DBArtisan), Unload, and Maintain Template. To run a wizard, select the command in the right pane of the application and click the **Execute** button on the Datasource Explorer toolbar. For related information, see Setting Scope Level in the Database Monitor.
Procedures (DB2 z/OS)	Displays the procedures on the target datasource. For related information, see Setting Scope Level in the Database Monitor.
Functions (DB2 z/OS)	Displays the functions on the target datasource. For related information, see Setting Scope Level in the Database Monitor.
Traces (DB2 z/OS)	Displays the traces on the target datasource.
Logs (DB2 z/OS)	Displays the logs on the target datasource.
Bufferpools	Displays the bufferpools on the target datasource.
All Locks (DB2 z/OS)	Displays all processes that are currently holding locks on a database.
Blocking Locks	Displays the locks that are blocked. Detect blocking locks as they occur, as they can escalate into dead lock situations which can force you to shut down and restart your server to clear the traffic.
Active Rollback Segment	Rollback or undo segments are used to guarantee transactional integrity. When a transaction has not been committed or rolled back, a rollback segment will be in use with live transactional data. The Active Rollback Segment tab displays information concerning active rollback segments.
Data Dictionary Cache	Displays key indicators in determining the performance of the shared pool. It shows how often object definitions are found in memory vs. having to read them in from disk. Because Oracle references the data dictionary many times when an SQL statement is processed, it is imperative that as much of this vital reference information be kept in RAM as possible.
Datafile I/O	Displays details concerning the physical I/O activity at the datafile level. Physical I/O consists of Oracle going to disk to gather or write data. Logical I/O refers to data access performed in memory. The database writer (DBWR) and log writer (LGWR) processes typically perform all the I/O work in the database against the physical datafile used to hold information The checkpoint and archive processes (CKPT and ARCH), also perform I/O work in the database.
General Statistics	Displays general statistics data, including names and values.

Hit Ratio By User	The database server reads and updates all data from a bufferpool because memory access is much faster than disk access. Data is copied from disk to a bufferpool as needed by the applications using the database. When the server needs to read/write data and the data is already in the bufferpool, no disk access is required. However, if the data is not in the bufferpool, it needs to be read from the disk, which is significantly slower process. The buffer pool hit ratio indicates the percentage of time that the database server did not need to load a page to service the request for a page.
Latches	Latches protect many memory structures in Oracle's SGA. They ensure that one and only one process at a time can run or modify any memory structure at the same instant. Much more restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism so either you get it or you do not and are forced to continually retry.
Library Cache	The library cache holds commonly used SQL statements - basically database code objects. A method for improving performance in Oracle is to encourage the reuse of SQL statements so expensive parse operations may be avoided. The library cache assists this tuning effort.
MTS Dispatcher Contention	Displays Microsoft Transaction Server data. DBArtisan lets you develop and deploy COM-based applications using Microsoft Transaction Server.
MTS Server Contention	Displays Microsoft Transaction Server data. DBArtisan lets you develop and deploy COM-based applications using Microsoft Transaction Server.
Open Cursors	Open cursors consume Oracle resources and have the potential to degrade performance, especially if the SQL code being used in the cursor is inefficient. The **Open Cursors** tab lets you quickly spot user accounts that have many cursors opened as well as the actual performance statistics for each opened cursor.
Parallel Queries	Displays the total number of queries eligible to be run in parallel.
SGA Detail	The Oracle System Global Area (SGA) is Oracle's memory structural area devoted to facilitating the transfer of data and information between clients and the Oracle database.
SGA Hit Ratios	The Oracle System Global Area (SGA) is Oracle's memory structural area devoted to facilitating the transfer of data and information between clients and the Oracle database.
SGA Summary	The Oracle System Global Area (SGA) is Oracle's memory structural area devoted to facilitating the transfer of data and information between clients and the Oracle database.
Tablespace I/O	Displays details concerning the physical I/O activity at the tablespace level. Physical I/O consists of Oracle going to disk to gather or write data. Logical I/O refers to data access performed in memory. The database writer (DBWR) and log writer (LGWR) processes typically perform all the I/O work in the database. Other processes like the checkpoint and archive processes (CKPT and ARCH) may also be used.
System I/O Summary (SQL SVR)	Displays a summary of I/O activities that occur at the system level.
Database I/O Summary (SQL SVR)	Displays an overview of I/O operations at the database level.

User I/O Summary (SQL SVR)	Displays an overview of I/O operations at the user operation level.
Waits (SQL SVR)	Displays a detailed view of all wait activity occurring on the server.
Top SQL Cache (SQL SVR)	Displays the top 100 objects in the SQL cache, ranked by use counts.
Last 50 Backups (SQL SVR)	Displays the last fifty backups performed by the Microsoft SQL Server and lets you reference hot backup details at the server level within DBArtisan.
File I/O Summary (SQL SVR)	Displays an overview of I/O operations at the database file level.
Top Twenty SQL (ORCL)	Lets you view the Top Twenty SQL, the top (worst) SQL running in the Oracle shared pool. For more information, see SQL Toggle, Query Plan, and ISQL.
Wait Events (ORCL)	Lets you analyze bottlenecks and discover what a database is waiting on.
Prepared Transactions (PSTGRS)	The Database Monitor **Prepared Transactions** tab displays identification details for transactions that have been prepared for two-phase commit. The following pg_prepared_xacts table columns are displayed for each transaction: transaction ID, global ID, date and time that the PREPARE TRANSACTION statement was executed, and the owner and database associated with the transaction. When this tab is active, the toolbar lets you **Commit** and **Rollback** a selected transaction.

For more general information, see Database Monitor (DBArtisan).

Killing Processes

If you have system administrator privileges, DBArtisan lets you kill processes from the Database Monitor.

1 On the **Utilities** menu, select **Database Monitor**.

DBArtisan opens the Database Monitor.

2 Click the target process to kill.

3 On the Database Monitor tool bar, click **Kill**.

4 Click **Yes**.

DBArtisan confirms you want to kill the process and closes the DBArtisan message.

For more information, see Configuring the Database Monitor.

Database Recover/Restore (DBArtisan)

Recover/Restore utilities are available for the following DBMS platforms:

- Database Restore (DBArtisan - SQL SVR)
- Database Restore (DBArtisan - SYB ASE)
- Database Restore Wizard (MySQL)
- Database Restore (DBArtisan - PostgreSQL)

Database Restore (DBArtisan - SQL SVR)

This Wizard lets you build and submit a RESTORE statement, selecting a restore type as well as other options.

To restore from backup

1 Connect to the datasource where the database to be restored is located and ensure that it is the selected datasource. For details, see Connected/Selected Datasource options.

2 Select **Utilities > Database Restore**. The **Recovery Wizard** opens.

3 Use the topics below as a guide to providing information as you walk through the pages of the wizard:

- Playback or New Operation
- Database Selection
- Recovery Completion
- Backup and Media Set Information
- Recovery Options
- Recover Operation Summary
- Action Selection

4 When ready, click **Finish.**

Playback or New Operation

Required Information	Description
New Recovery/Load Operation	This initiates the process of creating a new recovery operation.
Previous Recovery/Load Operation	If you want to replay a recovery operation, when you select the radio button, the browse function is enabled so you can find the .sqrc file you want to reuse.

Database Selection

Required Information	Description
Target Database	From the drop-down list, specify the target database for the recovery.
Restore Type	Elect to restore the database, filegroups or files, or a transaction log.
Backup Source	From Backup History - Find backup sets between the dates you specify and make your selection. From Device - Specify the device. Click **Add** or **Edit** to open the Recovery Wizard. Or, click **Remove** or **Remove All** to eliminate the device(s) from this operation.

Recovery Completion

Required Information	Description
Leave database operational	No more transaction logs can be recovered.
Leave database non-operational	You can restore more transaction logs.
Leave database read-only	You can restore more transaction logs.
Force restore over existing database	
Eject tapes (if any) after restoring each backup	Eject tapes after the backup is restored.
Point in time restore	Select the date and time from which you want the restore operation to take place.

Backup and Media Set Information

Here, for MS SQL, you specify the backup set name, backup set password, media name, and media name password. Any existing backup files are listed.

If you are creating at Leave database read-only operation, the Undo File option is enabled. Browse to find the file you want to undo.

Recovery Options

For MS SQL, the backup file is identified here, and if you are creating a new database from another database's backup, you can rename files in the Recover As box.

For Sybase ASE, you see the default backup file, which you can easily change. You also have the option, for tape device operations, to load the data with decompression.

Recover Operation Summary

You can elect to save the recovery operation. You can see the default location where the recovery operation will be saved or you can browse to another location. The summary panel displays the choices you have made to this point. Go back to make changes.

Action Selection

Option	Description
Open the recovery script in an ISQL editor	The recovery script will be opened into a new ISQL session. NOTE: This option is not available for Sybase ASE recoveries.
Perform the recovery now	The recovery will be executed immediately.
Schedule the recovery for later execution	The recovery will be scheduled to run at a specified date/time.

Database Restore (DBArtisan - SYB ASE)

This wizard lets you generate and submit a LOAD DATABASE statement, restoring a database and logs from tape, disk, or file.

To restore from backup

1 Connect to the datasource where the database to be restored is located and ensure that it is the selected datasource. For details, see Connected/Selected Datasource options.

2 Select **Utilities > Database Restore**. The **Recovery Wizard** opens.

3 Use the topics below as a guide to providing information as you walk through the pages of the wizard:

- Playback or New Operation
- Database Selection for Sybase ASE
- Recovery Completion for Sybase ASE
- Dump Device Selection for Sybase ASE
- Tape Handling Options for Sybase ASE
- Recovery Options
- Recover Operation Summary
- Action Selection

4 When ready, click **Finish**.

Playback or New Operation

Required Information	Description
New Recovery/Load Operation	This initiates the process of creating a new recovery operation.
Previous Recovery/Load Operation	If you want to replay a recovery operation, when you select the radio button, the browse function is enabled so you can find the .sqrc file you want to reuse.

Database Selection for Sybase ASE

Here you select the Sybase server and database you want to recover from the drop-down lists.

Recovery Completion for Sybase ASE

Required Information	Description
Which operation do you want to perform?	Load Database - Specifies that the complete database and transaction log should be restored. Load Transaction Log - Specifies that only the transaction log should be applied to the database. List Database - Validates a load database operation without actually performing the load. List Transaction Log - Validates a load transaction operation without actually performing the load.
Information type you want to see	**Header Only**, **Brief Description**, or **Detailed Description**.
If you are loading or listing a transaction log...	If you are loading the transaction log, in the If you are loading or listing a transaction log... box, type the specific log time you want to load.

Backup and Media Information for Sybase ASE

Required Information	Description
What type of dump device do you want to use?	Choose among: Tape or Disk Device: Existing options for either of these choices become apparent on the next panel. If none are available, you won’t see any devices. File: You must type the name of the file you want use
Where should messages about the load operation be sent?	Select among: **Default**, **Client**, or **Console**.
Dump File	You must type the name of the file where you want to send the load operation messages.

Dump Device Selection for Sybase ASE

If you elected to use a tape or disk device, here you select which specific tape/disk device you want to use. You can choose one or more of the devices that appear in the panel. If none appear, you have to go Back to pick another device type. You can specify tape or disk properties by clicking each device, and then clicking the **Attributes** button. If you click **Attribute**, the **Device Attribute Editor** dialog opens.

Tape Handling Options for Sybase ASE

You can opt to dismount after the operation completes or rewind and unload after the operation completes.

Recovery Options

For MS SQL, the backup file is identified here, and if you are creating a new database from another database’s backup, you can rename files in the Recover As box.

For Sybase ASE, you see the default backup file, which you can easily change. You also have the option, for tape device operations, to load the data with decompression.

Recover Operation Summary

You can elect to save the recovery operation. You can see the default location where the recovery operation will be saved or you can browse to another location. The summary panel displays the choices you have made to this point. Go back to make changes.

Action Selection

Option	Description
Open the recovery script in an ISQL editor	The recovery script will be opened into a new ISQL session. NOTE: This option is not available for Sybase ASE recoveries.
Perform the recovery now	The recovery will be executed immediately.
Schedule the recovery for later execution	The recovery will be scheduled to run at a specified date/time.

Database Restore Wizard (MySQL)

The MySQL Database Restore Wizard lets you build and submit a **mysqlimport** call, restoring a database from the data files generated by the Database Backup utility. For information on backing up a MySQL database, see Database Backup (MySQL).

To restore a MySQL database

1 Connect to a MySQL datasource and ensure that it is the currently selected datasource. For details, see Connected/Selected Datasource options.

2 Select **Utilities > Database Restore** to open the Database Import WIzard.

3 Use the following topics as a guide to setting properties and performing tasks as you pass through the wizard panels:

- **Playback** panel - for details, see Playback panel - Database Import Wizard.
- **Target Database** panel - for details, see Target Database panel - Database Import Wizard.
- **Select Datafiles** panel - for details, see Select Data Files panel - Database Import Wizard.
- **Options** panel - for details, see Options panel - Database Import Wizard.
- **Duplicate Key Options** panel - for details, see Duplicate Key Options panel - Database Import Wizard.
- **Summary** panel - for details, see Summary panel - Database Import Wizard.

4 Finally, use the **Finish** button to perform the restore.

Playback panel - Database Import Wizard

When restoring a MySQL database, this panel offers the following options:

Required Information	Description
New Database Import Operation	This initiates the process of creating a new restore operation.
Enter the views SQL definition	If you want to replay a restore operation, when you select the radio button, the browse function is enabled so you can find the .DBImport file you want to reuse.

Target Database panel - Database Import Wizard

When restoring a MySQL database, this panel lets you specify the database to be restored.

To set the database that is to be restored

- Select a database from the **Please choose the target database** list.

Select Data Files panel - Database Import Wizard

When restoring a MySQL database, this panel lets you build a list of dumped results files that are to be used to restore the database.

To set up a list of result files

- Use the New button to add a new entry to the list, and then use the browse button to locate and select a results file.
- Use the Delete button to remove a selected file from the list.
- Use the arrow buttons to change the position of a selected file in the list.

Options panel - Database Import Wizard

When restoring up a MySQL database, this panel lets you set or select the `mysqlimmport` options dictating how the backup is to be performed.

Note: Before working with these options, consult MySQL documentation for information on the mysqlimport program. For more information, see Accessing Third Party Documentation.

The following table shows the Database Backup Wizard settings on this panel, their `mysqldump` option equivalents, and a description of their effect.

Wizard Option	mysqldump Option Equivalent	Description
Protocol	--protocol=	Select the communication protocol (TCP, SOCKET, PIPE, or MEMORY) to be used for communication between client and server.
Compress	--compress	Compresses all information sent between the client and server.
Delete Table Data	--delete	Empties the table before importing from the results file.
Force Continue	--force	Force the restore to continue even if SQL errors occur.
Lock Tables	--lock-tables	Locks all tables for writing before processing results files.
Input File Location	--local	Indicate whether the results file is on the server (SERVER) or client (LOCAL).
Ignore Header Rows	--ignore-lines=	Indicate the number of rows at the start of the file that are to be ignored.
Fields Terminated By, **Fields Enclosed By**, **Fields Optionally Enclosed By**, **Fields Escaped By**, and **Lines Terminated By**	Lets you provide --fields... and --lines--terminated-by option values for line and column separators.	

Duplicate Key Options panel - Database Import Wizard

When restoring up a MySQL database, this panel lets you manage handling of input rows that duplicate existing rows on unique key values.

To specify duplicate row handling, select one of the following options

- Select **New Rows Replace Existing Rows** (--replace equivalent)
- Select **Ignore Input Rows That Duplicate Existing Rows** (--ignore equivalent)
- Select **Allow An Error To Occur When Duplicate Is Found And Ignore The Rest Of The Input File** (equivalent to mysqlimport execution with neither --replace or --ignore options)

Summary panel - Database Import Wizard

When restoring up a MySQL database, this panel summarizes the choices you made in previous steps setting up the operation.

Database Restore (DBArtisan - PostgreSQL)

This wizard lets you generate and submit a pg_restore utility operation, restoring one or more tables or views of a database from a previous backup.

Note: For information on generating a pg_dump backup script, see Database Backup (DBArtisan - PostgreSQL).

To restore a database from backup

1 Connect to the datasource where the database to be restored is located and ensure that it is the currently selected datasource. For details, see Connected/Selected Datasource options.

2 Select **Utilities > Database Restore** to open the **Restore Wizard**.

3 Use the following table as a guide to performing tasks as you pass through the wizard panels:

Panel	Option	Description
Playback	**New Backup Operation**	Select to define a new backup operation.
	Previous Backup Operation	Lets you specify a previously saved operation.
Datasource and Input options	**What are the destination datasource and database**	Lets you specify the specific datasource and database where the tables and views in the backup script are to be restored..
	Specify the format for the data file(s) to be generated	Lets you specify a -F (--format) archive format option of Custom or Tar or Directory.
	Number of jobs	Lets you provide a --j (--jobs) option value, specifing that the most time-consuming parts of the restore run as separate, concurrent jobs.
	Rolename	Lets you provide a --role option value, specify the role used to perform the restore.
Files	**Specify the data input file to be restored**, **Specify a path for the message file to be generated**, and **Do you wish to re-use this restore operation**	These settings lets you select the archive file used for the restore, specify the paths for the message file, and specify the path for the optional file that lets you reuse the current backup operation.
Options	**Sections**	Lets you provide one or more --section option values (Pre-data. Data, Post-data) specifying that the selected sections are restored.
	Type of Objects	Only data - corresponds to a -a (--data-only) option, specifying that only data be restored. Only schema - corresponds to a -s (--schema-only) option, specifying that only object definitions be restored.

	Don't save	Owner- corresponds to a -O (--no-owner) option, specifying no output commands to set ownership of objects to match the original database. Privilege - corresponds to a -X (--no-privileges) option, preventing restoration of access privileges. Tablespace - corresponds to a --no-tablespaces) option, specifying no output commands to select tablespaces.
	Queries	Include CREATE DATABASE statement - corresponds to a -C (--create) option, specifying that the database be created before restoring into it. Clean before restore - corresponds to a -C (--clean) option, specifying that the commands to drop objects are specified before commands to create them.. Use Column Inserts - corresponds to a --column-inserts option, specifying that data is dumped as INSERT commands with explicit column names Single transaction - corresponds to a -1 (--single-transaction) option, specifying that the restore be executed as a single transaction.
	Miscellaneous	Use SET SESSION AUTHORIZATION - corresponds to a --use-set-session-authorization option, specifying that object ownership is determined using SET SESSION AUTHORIZATION commands instead of ALTER OWNER commands. Exit on error - corresponds to a --exit-on-erroroption, specifying that the restore is aborted if an error is encountered. Verbose messages - corresponds to a --verbose option, specifying that verbose mode be used.
Objects	Use the object list to select each specific object that is to be restored. The Schema dropdown can be used to reduce the number of objects displayed to only those objects within a selected schema.	
Summary	Provides a short summary of the dump operation options.	

4 Finally, use the **Finish** button to perform the backup.

Free Space Deficit (DBArtisan - ORCL)

One of the central tasks of every Oracle administrator is to monitor the availability of free space in tablespaces. When a data segment (e.g., table, index or cluster) is created on a tablespace, it acquires an initial extent of a determined size. As it grows, the data segment will eventually need to acquire another extent to store additional data. If a data segment cannot acquire its next extent, all further transactions on it will fail.

The Free Space Deficit Monitor identifies any objects that will not be able to acquire their next extent. Whenever it identifies a problem data segment, you should take corrective action as soon as possible. To address the problem, you can:

1 Reduce the object's Next Extent Size from the appropriate Object Editor.

2 Increase the availability of free space in the tablespace by:

- Adding datafiles.
- Activating autoextend for the tablespace's datafiles. Autoextend allows the datafiles to grow to fill available room on the physical drive. This option is only available with Oracle.

The table below describes the objects:

Category	Information	Description
Tablespace	Name	Name of the tablespace.
	Largest Free Section	Largest contiguous section of free space in a tablespace.
Objects	Object Name	Name of any object that will not be able to acquire its next extent because it will be larger than the Largest Free Section.
	Object Type	Type of the object, such as a table, index or cluster, that will not be able to acquire its next extent.
	Next Extent	Specified size of the object's next extent. One of the options you have to correct the problem is to reduce the next extent size so that it is smaller than the Largest Free Section.

Opening the Free Space Deficit Dialog Box

1 On the **Utilities** menu, select **Free Space Deficit**.

DBArtisan opens the **Free Space Deficit** dialog.

The table below describes the options and functionality on the **Free Space Deficit** dialog:

Option	Description
Owner	Lets you click the owner for whom you want to analyze free space.

Instance Configuration (DBArtisan)

To let you configure and tune your database(s), DBArtisan offers a utility that lets you view and edit the configuration parameters for each server. These configuration parameters are stored in the system catalog and used when restarting the server. For detailed information on the meaning and purpose of each server configuration parameter, see the System Administration Guide and Performance and Tuning Guide.

Note: Many changes to instance configuration parameters require you to shutdown and to restart the server before the changes take effect.

1 On the **Datasources** menu, select **Configure**.

2 To modify a parameter, double-click the parameter or click **Edit**.

DBArtisan opens the **Edit Configuration** dialog.

Note: To modify a parameter, the parameter must have a Yes in the Session Modifiable or System Modifiable column of the Instance Parameter grid.

The table below describes the options and functionality on the **Edit Configuration** dialog:

Option	Description
New Value	Lets you type the value for the parameter.

Instance Parameter Configuration (DBArtisan - ORCL)

DBArtisan lets you configure and tune your Oracle databases. DBArtisan offers a utility that lets you view and edit the settings for each Oracle instance. DBArtisan derives its information from the system catalog. To reconfigure your Oracle instance, you must edit the parameter settings in the applicable configuration file (for example, INITORCL.ORA).

Note: After changing the settings, shut down and restart the database server for the changes to take effect.

Changing an Instance Parameter

1 On the **Datasources** menu, select **Configure**.

DBArtisan opens the **Instance Parameters** dialog.

2 To modify an instance parameter, double-click the target instance parameter or click the **Edit** button to open the **Edit Configuration** dialog.

Note: To modify an instance parameter, the parameter must have a Yes in the Session Modifiable or System Modifiable column of the Instance Parameter grid.

The table below describes the options and functionality on the **Edit Configuration** dialog:

Option	Description
New Value	Lets you type the value for the parameter.

Extracting Instance Configuration Parameters

DBArtisan lets you extract the statements for setting and displaying your instance configuration parameters.

1 On the Explorer, select the **Configuration** node.

DBArtisan displays the parameters in the right pane of the Datasource Explorer.

2 In the right pane of the Explorer, click the target parameter(s).

3 On the Explorer toolbar, click **Extract**.

OR

In the right-pane of the Explorer, right-click the parameters, and then click **Extract**.

DBArtisan extracts the definition to a text window.

Generating a Report on Configuration Parameters

DBArtisan lets you generate a detailed report containing information about your instance configuration parameters. The reports derive their contents from the information displayed in the right hand side of the Datasource Explorer. To facilitate the online publication of these reports, DBArtisan generates the report in Hypertext Mark-Up Language (HTML). You can view the report directly in DBArtisan's built-in HTML-browser.

Note: For more information on generating reports, see Schema Publication (DBArtisan).

1 On the Explorer, select the **Configuration** node.

DBArtisan displays the parameters in the right pane of the Datasource Explorer.

2 In the right pane of the Explorer, click the target parameter(s).

3 On the Explorer toolbar, click **Report.**

OR

In the right-pane of the Explorer, right-click the parameters, and then click **Report.**

DBArtisan opens the **Report** dialog.

The table below describes the options and functionality on the **Report** dialog:

Option	Functionality
Report Home Page File Name	To change the report home page file name, in the box, type the location and name of the file or find the file by clicking the **Browse** icon.
Report Title	To change the report title, in the box, type the location and name of the file or click the **Browse** button to find the file.

Maximum Extent Limits (DBArtisan - ORCL)

When an object reaches its maximum extent limit, it cannot acquire any new free space. This can cause critical problems in a production database. DBArtisan's Maximum Extent Limits for Oracle tool lets you quickly find and correct space problems.

Opening the Maximum Extent Limits Dialog Box

1 On the **Utilities** menu, select **Maximum Extent Limits**.

DBArtisan opens the **Maximum Extent Limits** dialog.

The table below describes the options and functionality on **Maximum Extent Limits** dialog.

Option	Description
Object Owner	Lets you select owner of the objects to display the Objects box.
Objects	Objects for the user account(s) that have reached their maximum extent limit.
Edit Button	To increase an object's maximum extent limit, edit the object and adjust its maximum extent property to a higher value or to unlimited.

Pinned Code (DBArtisan - ORCL)

For Oracle, DBArtisan now offers a Pinned Code Utility that lets you pin or unpin code objects in memory. This ensures that frequently accessed database stored procedures, triggers, functions, etc., are available in memory, which improves their overall execution performance.

Note: To create the DBMS_SHARED_POOL package log on as SYS and run the DBMSPOOL.SQL script that is found in the ORACLE_HOME\RDBMS\ADMIN directory.

Opening the Pin Code Dialog Box

1 On the **Utilities** menu, select **Pinned Code**.

DBArtisan opens the **Pin Code** dialog.

The table below describes the options and functionality on **Pin Code** dialog.

Option	Description
User	Lets you select the object user.
Pin Button	To pin an object, select the target object, and then click the **Pin** button.
UnPin Button	To unpin an object, select the target object, and then click the **UnPin** button.
Flush SGA Button	Click to flush the Oracle shared pool to remove fragmentation and any unwanted code objects from memory.

You must install the SYS.DBMS_SHARED_POOL package on your Oracle server to use this utility.

Confirm Shared Pool Flush

The **Confirmed Shared Pool Flush** dialog lets you confirm that you want to flush the Shared Pool. Flushing the Shared Pool causes all non-pinned code objects to be removed. After you flush the Shared Pool, all non-pinned objects are parsed and reloaded back into the pool on the next execution. This can extend first-time execution response times.

Resource Cost (DBArtisan - ORCL)

When you create a profile, you can assign a composite resource limit to it. The composite limit is a weighted average of four individual resource limits:

- Connect time
- Private SGA
- CPU per session
- Logical reads per session

The default setting is to weight each of these measures equally. You can alter the weights by issuing an ALTER RESOURCE COST command. DBArtisan provides a visual interface for constructing such a command.

Altering Resource Cost

1 On the **Utilities** menu, select **Alter Resource Cost**.

DBArtisan opens the **Alter Resource Cost** dialog.

The table below describes the options and functionality on the **Alter Resource Cost** dialog:

Option	Description
Weightings	In the box, specify the weightings for each resource limit by typing the weight value in the corresponding box: **Connect time**, **Private SGA**, **CPU per session**, or **Logical reads per sessio**n.
Enable Resource Limits	Select to To enable the resource limit changes immediately.

Resource Limits (DBArtisan - SYB ASE)

Resource limits are sets of parameters that let you maximize your datasource assets. By setting resource limits, you ca n prevent transactions and queries from congesting server resources. This ensures that you do not:

- Exceed estimated or actual input/output demands determined by the optimizer.
- Return a number of rows beyond a set limit.
- Exceed a defined elapsed time.

DBArtisan includes the **Resource Limits** dialog that lets you perform the following tasks:

- Showing Resource Limits
- Adding Resource Limits
- Editing Resource Limits
- Dropping Resource Limits

Opening the Resource Limits Dialog Box

You must connect to a datasource running Sybase ASE before you can open the **Resource Limits** dialog To open the **Resource Limits** dialog, do the following:

1 On the Datasource Navigator/Explorer, select the target datasource node.

2 On the **Utilities** menu, select **Resource Limits**.

DBArtisan opens the **Resource Limits** dialog.

The table below describes the columns in the Resource Limits table of the **Resource Limits** dialog

Column	Description
name	The login name for the resource limit.
appname	The application for the resource limit.
rangename	The time range in which the resource limit is enforced. The time range is a contiguous block of time.
rangeid	The value of the rangename.
limitid	The value and type of the resource limit.
limitvalue	The value of the resource limit.
enforced	The value and enforcement time for the resource limit.
action	The code number of the action for the resource limit. The codes numbers are: 1=Issue a warning 2=Abort the query batch 3=Abort the transaction 4=Kill the session
scope	The code number of the scope for the resource limit. The codes numbers are: 1=Query 2=Query batch 4=Transaction 6=Batch and transaction

Showing Resource Limits

DBArtisan displays all resource limits for the current datasource. In addition, DBArtisan lets you use the **Resource Limits** dialog to search for and display specific resource limits. In the **Resource Limits** dialog, you can specify search criteria for resource limits. DBArtisan displays the resource limits that meet your search criteria in the Resource Limits table of the **Resource Limits** dialog

Displaying Resource Limits

1 On the Datasource Navigator/Explorer, select the target datasource node.

2 On the **Utilities** menu, select **Resource Limits**.

DBArtisan opens the **Resource Limits** dialog.

3 Specify the search parameters. The table below describes the search parameters in the **Resource Limits** dialog:

Parameter	Description
Login list	Lets you specify the target login or logins.
Application list	Lets you specify an application or applications for the limit criteria.
Limit Time check box	Lets you specify a time range in which to enforce the resource limit.
Limit Time list	Lets you specify a start time for the resource limit. The start time must be within the proceeding 24 hours. To specify the start time, you must select the Limit Time check box.
Limit Day list	Lets you specify the day or day(s) for the resource limit.
Scope list	Lets you specify the scope of the target action(s). On the Scope list, the numbers next to each scope display in the scope column of the Resource Limits table.
Action list	Lets you specify the action to take place when a resource limit is exceeded. On the Action list, the numbers next to each action display in the action column of the Resource Limits table.

4 When you finish specifying search parameters, click **List Limits**.

DBArtisan lists the resource limits meeting the target criteria in the Resource Limits table.

Tip: To sort the display of the Resource Limits list, click a column heading.

Adding Resource Limits

DBArtisan offers a simple graphical wizard that guides you through the process of creating a resource limit without requiring your to know any of the underlying commands.

1 On the Datasource Navigator/Explorer, select the target datasource node.

2 On the **Utilities** menu, select **Resource Limits**.

DBArtisan opens the **Resource Limits** dialog.

3 In the **Resource Limits** dialog, click **Add**.

DBArtisan opens the Resource Limit Wizard - Panel 1

Resource Limit Wizard - Panel 1

The first panel of the Resource Limit Wizard lets you identify the objects to which the resource limit will apply.

The table below describes the options and functionality on the first panel of the Resource Limit Wizard:

Option	Description
Add Button	Click to open the Add/Edit Resource Limit Object Dialog Box.
Edit Button	Click to open the Add/Edit Resource Limit Object Dialog Box.

Add/Edit Resource Limit Object Dialog Box

The table below describes the options and functionality on the **Add/Edit Resource Limit Object** dialog:

Option	Description
Login	Lets you click the list, and then click the target user login.
Application	Click the list, and then click the target application.

Resource Limit Wizard - Panel 2

The second panel of the Resource Limit Wizard lets you specify:

- The type of resource limit.
- The value of the resource limit type.
- The time during which the limit will be enforced.

The table below describes the options and functionality on the second panel of the Resource Limit Wizard:

Option	Description
Select the type of resource limit	Lets you specify the type of resource limit by clicking one of the following option buttons: **Input/output cost**, **Elapsed time**, or **Row count**.
Limit Value box	**Note:** You must type the value for the limit to continue.
Time Range	Lets you click the list, and then click the target named range.
Modify, Add, and Edit buttons	Click to add or edit a named time range. DBArtisan opens the Named Time Ranges Dialog Box.
Pre-execution	To specify the execution times during which the limit will be enforced, select the target check box(es):
Execution	To specify the execution times during which the limit will be enforced, select the target check box(es):

Named Time Ranges Dialog Box

A time range is a contiguous set of days. For example if you designate a Friday start day and a Monday end day, the time range includes the weekend.

The table below describes the options and functionality on the **Named Time Ranges** dialog:

Option	Description
Name	In the box, type a name for the new time range.
Start time	In the box, specify the start time.
End time	In the box, specify the end time.
Start day	In the box, specify the beginning of the time range.
End day	In the box, specify the end of the time range.

Resource Limit Wizard - Panel 3

The third panel of the Resource Limit Wizard lets you specify the scope of the resource limit and the action to perform when the limit is exceeded. The available scopes are dependent on the type of resource limit you selected on the previous panel of the wizard.

The table below describes the types and available scopes:

Resource Limit Type	Available Scope
Input/output cost	Query
Elapsed time	Query batch, Transaction, and Query batch and transaction
Row count	Query

The table below describes the options and functionality on the third panel of the Resource Limit Wizard:

Option	Description
Scope	Click one of the option buttons to specify the scope of the resource limit: Query Query batch Transaction Query batch and transaction
Action	Click one the following option buttons to specify the action to take when the resource limit is exceeded: **Issue a warning**, **Abort the query batch**, **Abort the transaction**, or K**ill the session**.

Resource Limit Wizard - Last Panel

The last panel of the Resource Limit Wizard lets you:

- Preview the SQL for function that adds the resource limit.
- Execute the SQL for function that adds the resource limit.

Completing the Last Panel

1 Review the SQL.

2 Click **Execute**.

Editing Resource Limits

DBArtisan lets you edit resource limits. To edit resource limits, do the following:

1 On the Datasource Navigator/Explorer, select the target datasource node.

2 On the **Utilities** menu, select **Resource Limits**.

DBArtisan opens the **Resource Limits** dialog.

3 In the Resource Limits grid, select the target resource limit.

4 Click **Edit**.

DBArtisan opens the Resource Limit Editor.

5 To modify the limit value, in the Limit Value box edit the value.

6 To change the type of action, click a different option button.

7 Click **OK**.

Dropping Resource Limits

DBArtisan lets you drop resource limits. To drop resource limits, do the following:

1 On the Datasource Navigator/Explorer, select the target datasource node.

2 On the **Utilities** menu, select **Resource Limits**.

DBArtisan opens the **Resource Limits** dialog.

3 In the Resource Limits grid, select the target resource limit.

4 Click **Drop**.

DBArtisan opens a confirm drop message.

5 Click **Yes**.

Schema Extraction and Migration (DBArtisan)

DBArtisan lets you migrate schema and table data between databases on the same or different servers. Similarly, you can build and store a schema extraction script. Both features employ a wizard that walks you through the process and make use of similar options.

For detailed information, see the following topics:

- Using the Schema Extraction Wizard (DBArtisan)
- Using the Schema Migration Wizard (DBArtisan)
- Common Schema Migration and Extraction Options (DBArtisan)
- Migration and Object Restrictions/requirements (DBArtisan)

Using the Schema Extraction Wizard (DBArtisan)

The Schema Extraction wizard walks you through the process of building an extraction script, executing it for storage of the database or object schema in a file. The basic process includes choosing the server and database objects for extractions and specifying options that dictate how the extraction script is built. This utility allows you to:

- Extract and archive entire databases or sets of database objects.
- Generate the complete definition of any database object type.
- Reverse-engineer schema, and also place the SQL statements in their proper sequence, regardless of the objects you select.
- Create a new schema extraction operation.
- Reuse a previously defined schema extraction operation.
- Specify the target object types to extract.
- Specify the dependent object types automatically extracted for any object type.
- Set the general script type and options.
- Preview the definition of your schema extraction operation.

Platform Availability

DB2 LUW	DB2 z/OS	ITB/FBD *	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
✓	✓		✓	✓	✓	✓	

To build and execute a schema extraction script

1 Connect to the datasource where you want to extract schema. For details, see Connecting to Datasources.

2 On the Datasource Navigator/Explorer select the target database or database object.

3 On the **Utilities** menu, select **Schema Extraction**. The Extraction Wizard opens.

4 Use the following table as a guide to understanding and setting options in the wizard:

Panel	Tasks and Settings	Description
Playback	**Perform new extraction** and **Playback an existing extraction**	Lets you start a new extraction or select an existing extraction script you want to replay from the drop-down list. NOTE: When the script opens, you can make changes to the operation you are replaying. The changes you make will overwrite the current script.
Connection	**Source Datasource**	Lets you select the datasource from which you want to extract schema.
Catalogs (only displayed for multiple database DBMS datasources)	**Source Database**	Lets you select one or more databases from which you want to extract schema.
Object Selection	**Schemas**	Opens a dialog that lets you narrow the choice of candidate objects by selecting only those associated with specified schema.
	Object Types	Lets you select the object types that will be extracted. You can use the expand/collapse icons to hide and show the dependent object types for each object. As you select object types, all objects of that type (that satisfy the **Schemas** criteria) are made available for selection in the **Objects** list. Selected dependent object types are also made available in the **Objects** list. For information on setting the dependent object type automatically selected to be extracted along with each object type, see DDL Extract Options.
	Objects	Lets you select the specific objects that will be extracted. For each object selected, you can also select the specific dependent objects that are to be extracted.
Options	This panel lets you choose migration/extraction options and view an **Example Preview** script that is updated as you choose options. It also lets you save your options choices as a template, with an option to use the saved template as the default. For a detailed descriptions of the options presented on this panel, see Common Schema Migration and Extraction Options (DBArtisan).	
Summary	Presents the following options:	
	Output script to file	Lets you output the script performing the extraction to a file with the specified path and name.
	Save existing operation	Lets you save a copy the choices you made on the wizard panels. On subsequent schema extractions, you can then playback the existing extraction, optionally changing option settings, in creating an extraction script. You specify the path and name of the file in which to save the extraction operation.
	Click **Finish** when ready to proceed with the extraction.	

Execute	Displays the status of the extraction with details on the number of objects retrieved, the number of errors detected, and the elapsed time. On completion, the schema definition is stored in the file you specified on the previous panel. Filter options (Show All, Show Errors, and Show Warnings) let you display specific types of messages generated during the extraction process. When the extraction completes, you can also use the following options:

Continue	Opens the extraction script in the ISQL Editor. For more information, see Coding Environments/Editors.
Report	Opens a detailed report on the results of the schema extraction.

Using the Schema Migration Wizard (DBArtisan)

DBArtisan offers robust cross-DBMS-platform object migration capabilities that can migrate schema and table data between databases on the same or different servers. DBArtisan's migration capabilities automatically resolve differences between DBMS platforms, alleviating the responsibility of knowing the syntax and object variations inherent in database platforms. The Schema Migration Wizard sequences the creation of objects in the proper order to eliminate dependency errors. It also has the intelligence to use the fastest method available for copying table data.

The Schema Migration Wizard lets you:

- Create a new schema migration operation.
- Re-use a previously defined schema migration operation.
- Specify a source and a target datasource.
- Specify server object types, database object types, and general migration options.
- Select specific objects and to set certain migration options for each object type.
- Specify the dependent object types automatically migrated for any object type.
- Specify object ownership.
- Specify if the migration operation is to be saved for reuse.

Caution: Schema migration jobs created in earlier versions of DBArtisan that were saved with a .migrate file extension must be re-created in DBArtisan 8.1 or later. Only migration jobs with a .sav extension can be reused.

The table below describes the object types that can be migrated to different supported platforms and the utility used to accomplish the migration operation:

Source	Target	Object Type	Data Out Method	Data In Method
DB2	DB2	All Object Types & Data	DB2 Export API Call	DB2 Import API Call
	SQL Server	Tables, data, views, indexes, users	SQL SELECT	BCP Utility
	Sybase ASE	Tables, data, views, indexes, users	SQL SELECT	BCP Utility
	Oracle	Tables, data, views, indexes, users	SQL SELECT	SQL * Loader Utility
SQL Server	SQL Server	All Object Types & Data	BCP Utility	BCP Utility
	Sybase ASE	All Object Types & Data except functions	BCP Utility	BCP Utility

	Oracle	Tables, data, views, indexes, logins	SQL SELECT	SQL * Utility
	DB2	Tables, data, views, indexes, logins	SQL SELECT	DB2 Import API Calls
Oracle	SQL Server	All Object Types & Data	BCP Utility	BCP Utility
	Sybase ASE	Tables, data, views, indexes, logins	BCP Utility	BCP Utility
	Oracle	All Object Types and Data	SQL SELECT	SQL * Utility
	DB2	Tables, data, views, indexes, logins	SQL SELECT	SQL * Utility
Sybase ASE	Sybase ASE	All Object Types & Data	BCP Utility	BCP Utility
	SQL Server	All Object Types & Data except functions	BCP Utility	BCP Utility
	Oracle	Tables, data, views, indexes, logins	SQL SELECT	SQL * Utility
	DB2	Tables, data, views, indexes, logins	SQL SELECT	DB2 Import API Calls

To copy objects between servers, the Schema Migration Wizard establishes separate connections to both the source and target servers. Based on the copy options you choose, it then migrates schema from the source to the target database. If the bulk copy command is used when copying table data, a temporary BCP file is built in the temp directory of your computer.

Note: Before migrating, ensure that any object requirements, restrictions, or restrictions are satisfied. For details, see Migration and Object Restrictions/requirements (DBArtisan).

To Open the Schema Migration Wizard

1 Connect to the datasource from which you want to migrate schema. For details, see Connecting to Datasources.

2 On the Datasource Navigator/Explorer, select the target database or database object.

3 On the **Utilities** menu, select **Schema Migration**.

The Migration Wizard opens.

4 Use the following table as a guide to understanding and setting options in the wizard:

Panel	Tasks and Settings	Description
Playback	**Perform new migration** and **Playback an existing migration**	Lets you start a new migration or select an existing migration script you want to replay from the drop-down list. NOTE: When the wizard opens, you can make changes to the operation you are replaying.
Connection	**Source Datasource** and **Target Datasource**	Lets you select the datasource from which you want to migrate the schema and the datasource to which you want to migrate.
Catalogs (only displayed for multiple database DBMS datasources)	**Source Database** and **Target Database**	Lets you select a source and target database.
Object Selection	**Schemas**	Opens a dialog that lets you narrow the choice of candidate objects by selecting only those associated with specified schema.
	Object Types	Lets you select the object types that will be migrated. You can use the expand/collapse icons to hide and show the dependent object types for each object. As you select object types, all objects of that type (that satisfy the **Schemas** criteria) are made available for selection in the **Objects** list. Selected dependent object types are also made available in the **Objects** list. For information on setting the dependent object type automatically selected to be extracted along with each object type, see DDL Extract Options.
	Objects	Lets you select the specific objects that will be extracted. For each object selected, you can also select the specific dependent objects that are to be extracted.
Options	This panel lets you choose migration options and view an **Example Preview** script that is updated as you choose options. It also lets you save your options choices as an template, with an option to use the saved template as the default. For a detailed descriptions of the options presented on this panel, see Common Schema Migration and Extraction Options (DBArtisan).	
Summary	This panel presents the following options:	
	Output script to file	Lets you specify the name and path of a file to store the output script.
	Save existing operation	Lets you save a copy the choices you made on the wizard panels. On subsequent schema migrations, you can then playback the saved migration settings, optionally changing option settings. You specify the path and name of the file in which to save the migration operation.
	Click **Finish** when ready to proceed with the extraction.	

Execute	Displays the status of the migration with details on the number of objects retrieved and created, the number of errors detected, and the elapsed time. Filter options (Show All, Show Errors, and Show Warnings) let you display specific types of messages generated during the migration process. When the extraction completes, you can also use the following option:	
	Continue	Opens the extraction script in the ISQL Editor. For more information, see Coding Environments/Editors.
	Report	Opens a detailed report on the results of the schema migration.

Topics

- Executing Statements Before or After Creation of Functions or Procedures

Executing Statements Before or After Creation of Functions or Procedures

During migration, it can be necessary to have statements executed before and after creation of the procedure or function. This can be useful for example, if you need to create or drop temporary tables used by the function or procedure. DBArtisan supports the use of two tag pairs, ETStart and ETEnd, that let you embed statements in the first comment block of a stored procedure or function. The following shows the expected syntax:

```
create procedure dbo.procname(@a numeric) as

/*

<ETStart>SQL Statement</ETStart>

<ETEnd>SQL Statement</ETEnd>

*/

begin

...
```

Common Schema Migration and Extraction Options (DBArtisan)

When extracting or migrating schema, the wizards present a set of options that let you specify how the extraction or migration script is created. The following table describes these options.

Note: Depending on the DBMS you are working with, the types of objects you choose to migrate/extract, and the specific migration or extraction operation, not all options are available.

Tasks and Settings	Description
Group Dependencies	If set to True, objects that are dependent on each other are placed together in the script. For example, Table A would be followed by all objects dependent on Table A. If set to **False,** all objects of the same type are extracted simultaneously.
Script... options	Depending on the DBMS you working with and the types of objects you selected, these options let you specify whether you want to include Create statements, Drop statements, System Privileges, Object Privileges, and If Not Exists statements in the script.
Script Owner	True/False to include the owner in the scripts or not.
Owner	Retain Owner/Change Owner
Change Owner To	If you opted to change the owner, choose the new owner to use when scripting options from the drop-down list.
Copy Owner (Migration only)	Lets you retain the owner in the extraction/migration.
Overwrite Existing (Migration only)	Choose one of the following options: **Overwrite Existing**: objects are overwritten with no prompt. Objects on the target datasource are dropped before new objects are created **Prompt Whether To Overwrite**: The user will be prompted before the object is dropped and re-created. (**Note:** this will be done on an object-by-object basis). **Do Not Overwrite Existing**: If the object exists on the target datasource, no attempts is made to recreate it on the target. **Ignore Overwrite**: (default) Artisan will try to create the migrated objects, without checking the existence on the target datasource.
Generate Script (Migration only)	If set to FALSE, the generated DDL is executed on the target datasource. If set to TRUE, DBArtisan generates a script which could be used to migrate the schema at a later date.

For context information, see the following topics:

- Using the Schema Extraction Wizard (DBArtisan)
- Using the Schema Migration Wizard (DBArtisan)

Migration and Object Restrictions/requirements (DBArtisan)

Some objects have additional requirements, exceptions, or options. Before migrating a database object, check your database platform for additional information:

- Migrating IBM DB2 for Windows/Unix Objects
- Migrating IBM DB2 for OS/390 and z/OS Objects
- Migrating Microsoft SQL Server Objects
- Migrating Oracle Objects
- Migrating Sybase ASE Objects

Migrating IBM DB2 for Windows/Unix Objects

The table describes the condition(s) that need to be satisfied on the target database to migrate objects:

Object	Condition(s) on Target Database
Aliases	The login must already exist on the target server. The user must already exist in the target database.
Indexes	The base tables must already exist in the target database.
Nodegroups	The base tables must already exist in the target database.
Procedures	Any objects referenced by the stored procedure, including tables, views and other stored procedures, must already exist in the target database. The owner of the stored procedure must already exist in the target database.
Tables	Any referenced user datatypes must already exist in the target database. Any defaults referenced by the table must already exist in the target database. Any rules referenced by the table must already exist in the target database. If the tables have foreign key references to tables not included in the migration, these tables must already exist in the target database.
Tablespaces	The container mappings and directories must already exist on the target server.
Triggers	The base tables and their owners must already exist in the target database. Any objects referenced by the trigger, including other tables, views and other stored procedures, must already exist in the target database.
User Datatypes	The owner of the user datatype must exist in the target database.
Users	The database objects must already exist in the target database.

Migrating IBM DB2 for OS/390 and z/OS Objects

The table describes the condition(s) that need to be satisfied on the target database to migrate objects:

Object	Condition(s) on Target Database
Aliases	The login must already exist on the target server. The user must already exist in the target database.
Indexes	The base tables must already exist in the target database.
Procedures	Any objects referenced by the stored procedure, including tables, views and other stored procedures, must already exist in the target database. The owner of the stored procedure must already exist in the target database.
Tables	Any referenced user datatypes must already exist in the target database. Any defaults referenced by the table must already exist in the target database. Any rules referenced by the table must already exist in the target database. If the tables have foreign key references to tables not included in the migration, these tables must already exist in the target database.
Tablespaces	The container mappings and directories must already exist on the target server.
Triggers	The base tables and their owners must already exist in the target database. Any objects referenced by the trigger, including other tables, views and other stored procedures, must already exist in the target database.
User Datatypes	The owner of the user datatype must exist in the target database.
Users	The database objects must already exist in the target database.
Views	Any objects referenced by the view, including tables and other views, must already exist in the target database. The owner of the view must already exist in the target database.

Migrating Microsoft SQL Server Objects

The table describes the condition(s) that need to be satisfied on the target database to migrate objects:

Object	Condition(s) on Target Database
Aliases	The login must already exist on the target server. The user must already exist in the target database.
Defaults	The owner of the default must exist in the target database. Any bound user datatypes must exist in the target database. Any bound table columns must exist in the target database.
Groups	To migrate any object privileges granted to groups, the database objects must already exist in the target database.
Linked Servers	Links do not migrate. So, after you migrate a linked server, open the Linked Server Editor and create new linked server logins. For more information, see Linked Servers Editor (DBArtisan - SQL Server).
Procedures	The user who owns the procedure(s) must already exist in the target database. Any objects referenced by the procedure(s) must already exist in the target database.
Remote Servers	DBArtisan lets you migrate the schema for one or more remote servers between cross-platform DBMSs. Specifically, you can migrate remote servers between different versions of Microsoft SQL Servers and Sybase ASE. To migrate remote servers, DBArtisan establishes separate connections to both the source and target servers and copies the DDL between the source and target databases.
Roles	To migrate any object privileges granted to roles, the database objects must already exist in the target database.
Rules	The owner of the rule must exist in the target database. Any bound user datatypes must exist in the target database. Any bound table columns must exist in the target database.
Segments	To migrate segments successfully, all referenced database fragments must already exist in the target database.
Tables	Any referenced user datatypes must already exist in the target database. If you are copying segment placement also, then the segments must already exist in the target database. Any defaults referenced by the table must already exist in the target database. Any rules referenced by the table must already exist in the target database. If the tables have foreign key references to tables not included in the migration, these tables must already exist in the target database.
Triggers	The base tables and their owners must already exist in the target database. Any objects referenced by the trigger, including other tables, views and other stored procedures, must already exist in the target database.
User Datatypes	The owner of the user datatype must exist in the target database. Any bound defaults and rules must exist in the target database.
Users	The logins must already exist in the target server. The database objects must already exist in the target database.
Views	Any objects referenced by the view, including tables and other views, must already exist in the target database. The owner of the view must already exist in the target database.

Migrating Oracle Objects

The table describes the condition(s) that need to be satisfied on the target database to migrate objects:

Object	Condition(s) on Target Database
Clusters	The user who owns the cluster(s) must already exist in the target database. The tablespace on which the cluster(s) are placed must already exist in the target database.
Directories	To migrate directories successfully, the object must exist in the target database.
Functions	The user who owns the function(s) must already exist in the target database. Any objects referenced by the function(s) must already exist in the target database.
Indexes	The tables on which the indexes are being built must already exist in the target database. The user who owns the indexes must already exist in the target database. The tablespace on which the indexes are being placed must already exist in the target database.
Libraries	The object type must exist on the target server.
Materialized Views	The user who owns the materialized view(s) must already exist in the target database. Any objects referenced by the materialized view(s) must already exist in the target database.
Materialized View Logs	The user who owns the materialized view log(s) must already exist in the target database. Any objects referenced by the materialized view log(s) must already exist in the target database.
Packages	The user who owns the package(s) must already exist in the target database. Any objects referenced by the package(s) must already exist in the target database.
Procedures	The user who owns the procedure(s) must already exist in the target database. Any objects referenced by the procedure(s) must already exist in the target database.
Rollback Segments	The user who owns the rollback segment(s) must already exist in the target database. The tablespaces where the rollback segment(s) are to be placed must already exist in the target database.
Sequences	The user who owns the sequence(s) must already exist in the target database. Any objects referenced by the sequence(s) must already exist in the target database.
Synonyms	The user who owns the synonym(s) must already exist in the target database. The base objects referenced by the synonym(s) already exist in the target database.
Tables	Any referenced user datatypes must already exist in the target database. If you are copying tablespace placement also, then the tablespaces must already exist in the target database. If the tables have foreign key references to tables not included in the migration, these tables must already exist in the target database. NOTE: Index-Organized tables are coded to automatically migrate as though Copy Table Constraints in Alter Form is not selected, and Include Table Constraints is selected.
Triggers	The user who owns the trigger(s) must already exist in the target database. The base triggers must already exist in the target database. Any objects referenced by the trigger(s) must already exist in the target database.
Types	The user who owns the object type(s) must already exist in the target database. The base object types must already exist in the target database. Any objects referenced by the object type(s) must already exist in the target database.

Users	The default and temporary tablespaces assigned to the user(s) must already exist in the target database. The profiles assigned to the user(s) must already exist in the target database. Any tablespaces on which quotas are defined must already exist in the target database. Any roles granted to the user(s) must already exist in the target database.
Views	Any objects referenced by the view, including tables and other views, must already exist in the target database. The owner of the view must already exist in the target database.

Migrating Sybase ASE Objects

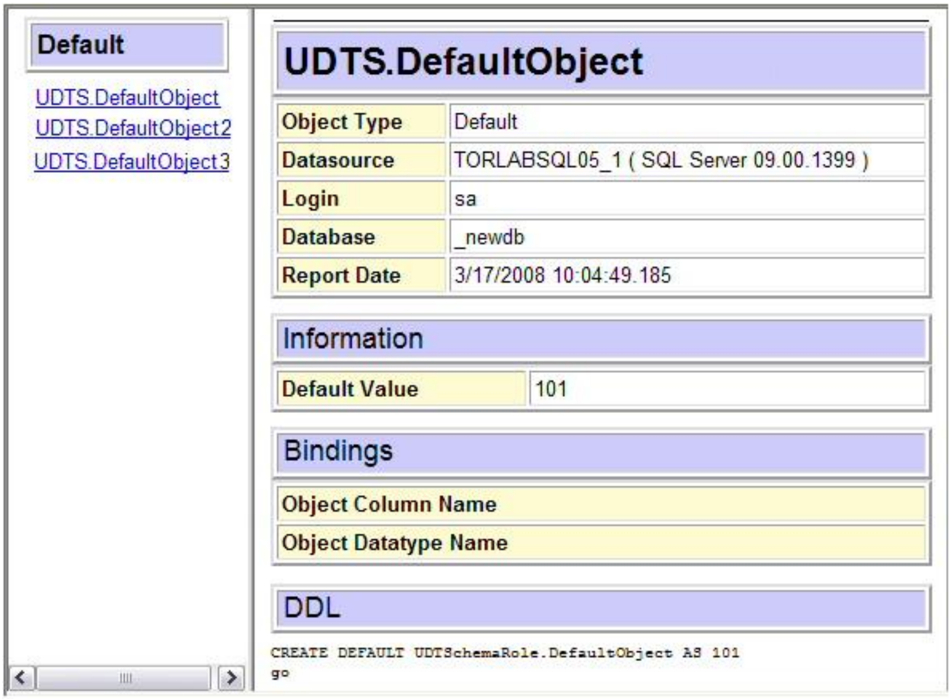
The table describes the condition(s) that need to be satisfied on the target database to migrate objects:

Object	Condition(s) on Target Database
Aliases	The login must already exist on the target server. The user must already exist in the target database.
Defaults	The owner of the default must exist in the target database. Any bound user datatypes must exist in the target database. Any bound table columns must exist in the target database.
Groups	To migrate any object privileges granted to groups, the database objects must already exist in the target database.
Procedures	The user who owns the procedure(s) must already exist in the target database. Any objects referenced by the procedure(s) must already exist in the target database.
Remote Servers	DBArtisan lets you migrate the schema for one or more remote servers between cross-platform DBMSs. Specifically, you can migrate remote servers between different versions of Sybase ASE and Microsoft SQL Servers.
Roles	To migrate any object privileges granted to roles, the database objects must already exist in the target database.
Rules	The owner of the rule must exist in the target database. Any bound user datatypes must exist in the target database. Any bound table columns must exist in the target database.
Segments	To migrate segments successfully, all referenced database fragments must already exist in the target database.
Tables	Any referenced user datatypes must already exist in the target database. If you are copying segment placement also, then the segments must already exist in the target database. Any defaults referenced by the table must already exist in the target database. Any rules referenced by the table must already exist in the target database. If the tables have foreign key references to tables not included in the migration, these tables must already exist in the target database. NOTE: DBArtisan no longer changes the table lock scheme from ALLPAGES back to DATAROWS on the target server when using BCP.exe version dated after December 1997.
Triggers	The base tables and their owners must already exist in the target database. Any objects referenced by the trigger, including other tables, views and other stored procedures, must already exist in the target database.
User Datatypes	The owner of the user datatype must exist in the target database. Any bound defaults and rules must exist in the target database.
Users	The logins must already exist in the target server. The database objects must already exist in the target database.
Views	Any objects referenced by the view, including tables and other views, must already exist in the target database. The owner of the view must already exist in the target database.

Schema Publication (DBArtisan)

The Schema Publication Wizard lets you generate HTML reports documenting the details of your schema, databases, and objects, which you can share over an intranet or the Internet. DBArtisan uses HTML Templates to create the reports which you can manually edit to create customized reports.

Schema publication produces a two-pane report. The left-hand pane lets you navigate through all objects contained in the report while the right-hand pane shows details for an individual object. THe following example shows a report on three Microsoft SQL Server default objects, displaying details for a default named **UTDS.DefaultObject**.



Default

UDTS.DefaultObject
UDTS.DefaultObject2
UDTS.DefaultObject3

UDTS.DefaultObject

Object Type	Default
Datasource	TORLABSQL05_1 (SQL Server 09.00.1399)
Login	sa
Database	_newdb
Report Date	3/17/2008 10:04:49.185

Information

Default Value	101

Bindings

Object Column Name
Object Datatype Name

DDL

```
CREATE DEFAULT UDTSchemaRole.DefaultObject AS 101
go
```

The Schema Publication Wizard lets you:

- Generate reports for all or any level of your database.
- View, find, and print the HTML reports directly in DBArtisan with the built-in HTML browser.
- Create a new schema publication operation.
- Re-use a previously defined schema publication operation.
- Select a datasource for which you want to generate an HTML report.
- Select the server object type and database object type for which you want to publish a schema report and lets you customize the report.
- Select specific objects and to set certain script options for each object type.
- Specify the report directory, HTML file name, and report title.

- Save the schema publication operation for re-use.
- Preview the definition of your schema publication operation.

Platform Availability

DB2 LUW	DB2 z/OS	ITB/FBD *	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
✓	✓		✓	✓	✓	✓	

To create a schema report

1 Connect to the datasource for which you want to produce a schema report. For details, see Connecting to Datasources.

2 On the **Utilities** menu, select **Schema Publication**.

3 Use the following table as a guide to understanding and setting options in the wizard:

Panel	Tasks and Settings	Description
Playback	**Perform new publication**	Select to define a new schema publication operation.
	Playback an existing publication	Lets you enable use of a previously defined publication operation and provide the path and file name of the operation.
Connection	Lets you select the datasource to be published.	
Catalogs (SQL Server, MySQL, and Sybase ASE datasources only)	Lets you select the target database.	
Object Selection	**Schemas**	Opens a dialog that lets you narrow the choice of candidate objects by selecting only those associated with specified schema.
	Object Types	Lets you select the object types and database object types that will be published.
	Objects	As you select types from the **Object Types** list, an entry corresponding to that type is added to the **Objects** list. To publish only specific objects of that type, expand the list for that type and select the check boxes corresponding to the objects you want to publish.
Summary	This panel presents a visual summary of your selections and offers the following options:	
	Output script to file	Lets you specify the name and path of a file to store the output script.

Save existing operation	Lets you save a copy the choices you made on the wizard panels. On subsequent publications, you can then playback the saved settings, optionally changing specific option settings. You specify the path and name of the file in which to save the migration operation.
Append date and timestamp to output file name and **Save output file in date specific folders**	Lets you store and name output files using time and date elements.

Click **Finish** when ready to proceed with the extraction.

Execute	Displays a real-time status of the operation and a summary of the results.

Session Information (DBArtisan - ORCL)

Oracle tracks detailed information about the resources used by each session on a database. This information is useful for analyzing resource utilization patterns between user sessions to monitor performance.

The information displayed is derived from the Oracle system tables, SYS.V_$STATNAME and SYS.V_$SESSTAT. Consult the Oracle documentation or detailed descriptions in the meanings of each session parameter.

Tip: Waits lets you analyze bottlenecks and discover what a database is waiting on.

Viewing Session Information

1 On the **Utilities** menu, select **Session Information**.

DBArtisan opens the **User Session Information** dialog.

2 Complete the User Session Information Dialog Box.

3 When you finish viewing SQL, click **Close** to return to the **User Session Information** dialog.

4 When you finish viewing the session information, click **Close**.

SQL Logfile (DBArtisan)

You can instruct DBArtisan to log all of its SQL statements to a file so that you have an audit trail of application activity. The SQL Logfile captures all the SQL sent by the DBArtisan to populate its editors or to execute your commands. You can use this logfile not only to audit application activity, but also to build SQL scripts for querying system catalogs.

Activating SQL Logging

1 On the **Logfile** menu, select **Log SQL**.

Opening the Logfile

1 On the **Logfile** menu, select **Open**. DBArtisan opens the SQL Logfile into an SQL window.

Logfile Options

You can open the Options Editor from the **Logfile** menu option and make any changes to your logging options. For details, see Logging Options.

Truncating the Logfile

If the SQL Logfile is growing too large, you can flush it directly from the main menu.

1 On the **Logfile** menu, select **Truncate**. DBArtisan deletes the contents of the SQL Logfile.

Top Sessions (DBArtisan - ORCL)

Being able to identify the top resource users in an Oracle system is a valuable tuning asset. The Top Sessions Options Dialog Boxlets you quickly pinpoint the heaviest users of I/O and memory usage. The Top Sessions utility quickly finds and ranks user sessions based on their resource usage.

Opening the Top Sessions Dialog Box

1 On the **Utilities** menu, select **Top Sessions**.

DBArtisan opens the **Top Sessions** dialog.

The table below describes the options and functionality on the **Top Sessions** dialog.

Tip: Click the **Options** button to open the Top Sessions Options Dialog Box.

Top Sessions Options Dialog Box

The table below describes the options and functionality on the **Top Sessions Options** dialog:

Option	Description
Exclude Sys and System	Select to force the query used to display top session data to be changed.
Exclude Background Process	Select to force the query used to display top session data to be changed.
Limit Top Session Display to	Lets you specify the number of sessions to display. These options will force the query used to display top session data to be changed.
No Limit	
Sort Top Sessions By	Lets you custom soft the display

User Session Information Dialog Box

The table below describes the options and functionality on the **User Session Information** dialog:

Option	Description
User	Lets you select a user.
OS User Name	Displays the operating system user name.
Session ID	Lets you select the application.
Machine Name	Displays the name of the datasource.
Status	Displays the session status.
Terminal Name	Displays the terminal name.
Command	Displays the command.
Program	Displays the program used.
Client Platform	IBM DB2 FOR WINDOW/UNIX ONLY: Lets you select the client platform.
Client Protocol	IBM DB2 FOR WINDOW/UNIX ONLY: Lets you select the client protocol.
Session Parameters	IBM DB2 FOR WINDOW/UNIX ONLY: Lets you select the session parameters.
User Session Information Dialog Box - Statistical Details tab	ORACLE ONLY: Displays statistical details.
User Session Information Dialog Box - Waits Tab	ORACLE ONLY: Displays waits.
User Session Information Dialog Box - Current SQL Tab	ORACLE ONLY: Displays the current SQL.

User Session Information Dialog Box - Statistical Details tab

The table below describes the options and functionality on the **Statistical Details** tab of the **User Session Information** dialog:

Option	Description
Statistical Details grid	Displays the session parameters and value (if any).

User Session Information Dialog Box - Waits Tab

The table below describes the options and functionality on the **Statistical Details** tab of the **User Session Information** dialog:

Option	Description
Historical Waits	If selected, displays historical waits in the waits grid.
Current Waits	If selected, displays current waits in the waits grid.
Include Idle Waits	If selected, displays idle waits in the waits grid.
Waits grid	Displays the wait information, including Session ID, Event, Seconds in Wait, Wait Time, State, and other information.

User Session Information Dialog Box - Current SQL Tab

The table below describes the options and functionality on the **Statistical Details** tab of the **User Session Information** dialog:

Option	Description
SQL box	Displays current SQL.

Project Management

The following project management-oriented functions are available:

- Project support
- Version Control

Project support

Database project management facilities help you organize, alter, and keep track of changes to database objects or SQL scripts. The project management facilities act as a repository to maintain all source code for a database project. Version control functions and build management facilities are also incorporated and help you manage and build projects. Once a project has been created, you can:

- Review a file's history.
- Return to earlier versions of a file.
- Develop concurrently.

A project generally includes SQL script files that you can maintain and create in unison with your database administration and development cycle. Projects can also contain subfolders. You can create projects manually by inserting existing SQL script files. You can also create projects automatically by reverse-engineering a database schema or an existing version control project.

When administering or developing SQL database schemas, SQL source code files are the foundation of effective change management for database objects. The DDL commands used to create database objects on SQL database servers are often in a constant state of flux as new columns and constraints are added to table schemas or stored procedure logic is changed. The ability to track and store these changes to files alleviates any disruption in the development environment.

Lengthy script files containing the DDL to compile stored procedures and triggers on a database server such as Oracle or Sybase Adaptive Server go through constant revisions similar to C program files and word processing documents. The Project Management facilities let you effectively monitor revisions for your database servers.

For more information, see:

- Create a New Project
- Working With Projects

Create a New Project

Step-by-step wizards guide you through the process of creating a new project. You have a variety of options to choose from when you create a new project. You can create new projects:

- From a database.
- From existing files.
- From a version control project.
- Without initialization.

The New Project dialog box lets you select how you want to create a new project. Using the New Project dialog box, you can name, specify a file location, and provide a description of the project.

New Project Dialog Box

The New Project dialog box lets you select how you want to create a new project. Using the New Project dialog box, you can name, specify a file location, and provide a description of the project.

The table below describes the options and functionality of the New Project dialog box:

Option	Description
Name	Lets you enter a name for the project.
Location	Lets you type or browse and locate a directory for the project. NOTE: This is set as the working directory for your project.
Description	OPTIONAL: Lets you enter a description of the project.
Initialize New Project	Lets you select how you want to create a new project. From Database - Lets you reverse engineer an existing database. From Existing Files - Lets you create a project from scratch, or use an existing project. From Version Control Project - Lets you reverse-engineer a project from existing version control system projects. This is helpful for users who have already created database projects in their version control systems. Do Not Initialize - Lets you create an empty project where you can later add files and/or database objects.

For more information, see Completing the New Project Dialog Box.

Completing the New Project Dialog Box

To complete the New Project Dialog Box, do the following:

1 On the File menu, select New, and then Project.

The New Project Dialog Box opens.

2 In Name, enter a name for the project.

3 In Location, type or browse and locate a directory for the project.

4 OPTIONAL: In Description, lets you enter a description of the project.

5 In Initialize New Project, select how you want to create a new project.

6 Click OK.

Create a New Project From a Database

A context-sensitive wizard reverse-engineers all or part of an existing database. It lets you create a project containing files based on the relevant object types. A separate file is created for each database object.

Reverse engineering is a powerful tool for analyzing, controlling, and documenting existing database objects. You can use the extracted SQL source code for archival and reference purposes.

Note: The project script file build order is automatically discovered and set by referencing the system catalog to determine dependencies.

To start a new project from a database

1 Select **File > New > Project**. The **New Project** dialog box opens.

2 Complete the New Project dialog box, ensuring you select the **From Database** option, and then click **OK**. The **Reverse Engineer Wizard** opens.

3 Use the following table as a guide to understanding and setting options in the wizard:

Panel	Tasks and Settings	Description
Connection	**Source Datasource**	Lets you select the datasource for which you want to start a project..
Catalogs (only displayed for multiple database DBMS datasources)	**Source Database**	Lets you select one or more databases for which you want to start a project..
Object Selection	**Schemas**	Opens a dialog that lets you narrow the choice of candidate objects by selecting only those associated with specified schema/owners.
	Object Types	Lets you select the object types that will be extracted. You can use the expand/collapse icons to hide and show the dependent object types for each object. As you select object types, all objects of that type (that satisfy the **Schemas** criteria) are made available for selection in the **Objects** list. Selected dependent object types are also made available in the **Objects** list. For information on setting the dependent object type automatically selected to be extracted along with each object type, see DDL Extract Options.
	Objects	Lets you select the specific objects that will be extracted. For each object selected, you can also select the specific dependent objects that are to be extracted.

Options	This panel lets you choose extraction options and view an **Example Preview** script that is updated as you choose options. It also lets you save your options choices as a template, with an option to use the saved template as the default. Click **Finish** when ready to proceed with the extraction.
Execute	Displays the status of the extraction with details on the number of objects retrieved, the number of errors detected, and the elapsed time. On completion, the schema definition is stored in the file you specified on the previous panel. Filter options (Show All, Show Errors, and Show Warnings) let you display specific types of messages generated during the extraction process. When the extraction completes, you can also use the following options: **Continue** Adds the individual files for each obect to the project. **Report** Opens a detailed report on the results of the schema extraction.

Create a New Project From Existing Files

When creating a new project, you can start by adding files of the following types:

- Scripts (.sql)
- DDL (.ddl)
- Procedures (.prc, .pro)
- ER/Studio files (.dml)
- Queries (.qry)
- Text files (.txt)
- VB Scrpt files (.vbs)
- JScript files (.js)
- Web files files (.html, .htm, .asp)

To create a new project from an existing file, do the following:

1 On the File menu, click New, and then click Project.

The New Project dialog box opens.

2 Complete the New Project dialog box and select the From Existing Files option.

3 Click OK.

The Add File(s) to Project dialog box opens.

4 Use the dialog to locate the directory containing the files, and select one or more files to add to the project.

5 When you finish, click Open.

For details on adding additional files to a project, see Add File(s) to a Project.

Create a New Project From a Version Control Project

You can reverse-engineer a project from existing version control system projects. This is helpful if you have already created database projects in a version control system.

Note: The Intersolv PVCS API does not support the creation of a project from a source code control project.

You can slsocreate a project from a version control project that contains sub-directories, while including files from those sub-directories in your project.

To create a new project from a Version Control project, do the following:

1 On the File menu, click New, and then click Project.

The New Project dialog box opens.

2 Complete the New Project dialog box and select From Version Control Project.

3 Click OK.

The Choose project from (Version Control name) dialog box opens.

4 Select the project or files you want to include, and then click OK.

The Files to be included in (project path) dialog box opens.

5 Select the project of files you want to include, and then click OK.

Create a New Project Without Initialization

You can simply create a project with any further initialization. You can subsequently use the following topics to add additional resources:

- Add files.
- Add database objects.

To create a new project without initialization, do the following:

1 On the File menu, click New, and then click Project.

The New Project dialog box opens.

2 Complete the New Project dialog box and select the Do Not Initialize option.

3 Click OK.

Working with Projects

A project is similar to a file system. Both are a collection of files that you create and maintain. Because projects are hierarchical, you can place a subfolder under another project, and nest subfolders. Once you have created a new project, use the following topics to help you maintain and modify the project.

Related Topics

New Project Dialog Box

Opening an Existing Project

Opening a Recent Project

Closing a Project

Build Project

Set Build Order

Add Database Object File(s) to Project Wizard

Execute Project Files

Add File(s) to a Project

Open a File from a Project

Subfolders

Project Properties

Confirm Delete Dialog Box

Opening an Existing Project

To open an existing project, do the following:

1 On the File menu click Open Project.

The Open Project dialog box opens.

2 In File name, type the name and location of the project or use browse to locate the project.

Note: Project files are designated with a *.epj extension.

3 Click Open.

The Project Tab containing the project opens.

For more information, see Working with Projects.

Opening a Recent Project

To open a recent project, do the following:

1 On the File menu select Recent Projects, and then select a project.

The Project Tab containing the project opens.

For more information, see Working with Projects.

Closing a Project

To close a project, do the following:

1 On the File menu, click Close Project.

2 On the Project Tab, right-click and then click Close Project.

For more information, see Working with Projects.

Build Project

The Build Project dialog box lets you:

- Generate a project build script and display it in a SQL window.
- Execute the project build immediately on build target.
- Schedule a project build.

The table below describes the Build Project dialog box:

Option	Description
Datasource	Displays the target datasource.
Database	Displays the target database.
Build subfolders	Select to include subfolders.
Generate a Project Build Script and Display it in a SQL window.	Select to generate the project build script and have it opened in a SQL window.
Execute Project Build Immediately on Build Target	Select to execute build immediately.
Schedule Project Build for a Later Time	Select to schedule the build and then specify optional options.

For more information, see Completing the Build Project Dialog Box.

Completing the Build Project Dialog Box

To complete the Build Project dialog box, do the following:

1 On the File menu click Open Project.

The Open Project dialog box opens.

2 In File name, type the name and location of the project or use browse to locate the project.

Note: Project files are designated with a *.epj extension.

3 Click Open.

The Project Tab containing the project opens.

4 On the Project Tab, right-click the target project, and then select Build.

The Build Project dialog box opens.

5 Select options.

6 Click OK.

Set Build Order

You can specify the order in which you want your project files built in the Set Build Order dialog box. If you created your project manually or from a version control project, you must specify a build order, otherwise the files are built in the order that they appear in the tree of the Project Tab.

Setting Build Order

1 On the File menu click Open Project.

The Open Project dialog box opens.

2 In File name, type the name and location of the project or use browse to locate the project.

Note: Project files are designated with a *.epj extension.

3 Click Open.

The Project Tab containing the project opens.

4 On the Project menu, click Build Order.

The Set Build Order dialog box opens.

5 Click the files you want to move and then click the Up and Down to change the order to build the files.

6 When you finish specifying the order, select OK.

Note: The next time you build the project, the new build order is used.

Add Database Object File(s) to Project Wizard

You can add database objects to an existing project using a simple wizard that can reverse-engineer an entire database or any portion of it. This lets you keep your project in sync with databases where objects are constantly being created and updated.

Completing the Add Database Object File(s) to Project Wizard

To add database objects to a project, do the following:

1 On the File menu click Open Project.

The Open Project dialog box opens.

2 In File name, type the name and location of the project or use browse to locate the project.

Note: Project files are designated with a *.epj extension.

3 Click Open.

The Project Tab containing the project opens.

4 On the Project menu, click Add Database Objects.

The Reverse Engineer Wizard opens.

5 Use the information in Create a New Project From a Database to complete the wizard.

Execute Project Files

You can directly execute project script files from the Project Tab using the File Execution Facility. You can also execute multiple scripts in parallel against different datasources. If a file has been placed under version control, you need to perform a Check Out operation to execute the file. Otherwise, the file opens in read-only mode.

To execute project files, do the following:

1 On the Project Tab, right-click the files you want to execute, and then click File Execution Facility.

The Script/File Execution Facilities dialog box opens.

Add File(s) to a Project

You can add external files to your project. This lets you manipulate and expand your project as needed.

To add files to a project from, do the following:

1 On the Project menu, click Add Files.

The Add Files to Project dialog box opens.

2 Use the dialog to locate the directory containing the files you want to add to the project and then select the individual files to be added.

3 When ready, click Open.

Open a File from a Project

You can open a file directly from the Project Tab using a number of different methods.

To open a file from a project, do the following:

1 On the Project Tab, right-click the file(s) you want to open, and then click Open.

The file(s) open in a new SQL window.

Subfolders

Subfolders help categorize your source code files. On the Project Tab of the Database Explorer, you can:

- Create subfolders.
- Delete subfolders.
- Rename subfolders.
- Sort subfolders.

For more information, see Working with Projects.

Creating a New Subfolder

To create a new subfolder:

1 On the File menu click Open Project.

The Open Project dialog box opens.

2 In File name, type the name and location of the project or use browse to locate the project.

Note: Project files are designated with a *.epj extension.

3 Click Open.

The Project Tab containing the project opens.

4 On the Project menu, click New Subfolder.

5 Type the name of the new subfolder and then press Enter.

Deleting a Subfolder

To delete a project, do the following:

1 On the File menu click Open Project.

The Open Project dialog box opens.

2 In File name, type the name and location of the project or use browse to locate the project.

Note: Project files are designated with a *.epj extension.

3 Click Open.

The Project Tab containing the project opens.

4 On the Project Tab, right-click a subfolder, and then click Delete.

The Confirm Delete dialog box opens.

5 In the Confirm Delete dialog box, select the Delete local copy to delete the local copy of the subfolder, and then click OK.

Confirm Delete Dialog Box

The table below describes the options and functionality on the Confirm Delete Dialog Box:

Option	Description
Delete local copy	Deletes the local copy of the subfolder.

For more information, see Deleting a Subfolder.

Renaming a Subfolder

To rename a subfolder, do the following:

1 On the File menu click Open Project.

The Open Project dialog box opens.

2 In File name, type the name and location of the project or use browse to locate the project.

Note: Project files are designated with a *.epj extension.

3 Click Open.

The Project Tab containing the project opens.

4 On the Project Tab, right-click a subfolder, and then click Rename.

5 Type the name of the new subfolder and then press Enter.

Sorting Subfolders

To sort a subfolder, do the following:

1 On the File menu click Open Project.

The Open Project dialog box.

2 In File name, type the name and location of the project or use browse to locate the project.

Note: Project files are designated with a *.epj extension.

3 Click Open.

The Project Tab containing the project.

4 On the Project Tab, right-click the subfolders you want to sort and then click Sort.

Project Properties

You can view properties of Projects, Subfolders, and individual files. The Project Properties, Subfolder Properties, and File Properties dialog boxes display information about the Projects, Subfolders, and files.

For more information, see Working with Projects.

Viewing Project Properties

To view project properties, do the following:

1 On the File menu click Open Project.

The Open Project dialog box opens.

2 In File name, type the name and location of the project or use browse to locate the project.

Note: Project files are designated with a *.epj extension.

3 Click Open.

The Project Tab containing the project opens.

4 On the Project Tab, right-click one or more selected projects and then click Properties.

The Project File Properties or Multiple Selection Properties dialog box opens.

5 Complete the Project File Properties dialog box or Multiple Selection Properties Dialog Box.

6 Click OK.

For more information, see:

Project Properties

Project File Properties Dialog Box

Multiple Selection Properties Dialog Box

The table below describes the options and functionality of the Project Properties dialog box:

Option	Description
Description	Lets you type or edit the common description for the selected files.
Include In Build	Sets the files to be included in the build.
Object Type	Lets you associate a database object type with the file, as appropriate.

For more information, see Viewing Project Properties.

Project File Properties Dialog Box

The table below describes the options and functionality of the Project Properties dialog box:

Option	Description
Name	Displays the name of the project.
Full Specification	Displays the project location.
Description	Displays the project description. OPTIONAL: Lets you type or edit the project description.
Associated Datasource	Lets you select a datasource for the project.
Associated Database	Lets you type the name of the database.

For more information, see Viewing Project Properties.

Viewing Subfolder Properties

To view subfolder properties, do the following:

1 On the File menu click Open Project.

The Open Project dialog box opens.

2 In File name, type the name and location of the project or use browse to locate the project.

Note: Project files are designated with a *.epj extension.

3 Click Open.

The Project Tab containing the project opens.

4 On the Project Tab, select a subfolder.

5 Right-click the subfolder, and then click Properties.

The Subfolder Properties dialog box opens.

6 Complete the Subfolder Properties dialog box.

7 Click OK.

For more information, see:

Project Properties

Subfolder Properties Dialog Box

Subfolder Properties Dialog Box

The table below describes the options and functionality of the Subfolder Properties dialog box:

Option	Description
Name	Displays the name of the subfolder. You can enter or edit the subfolder name.
Full Specification	Displays the subfolder location.
Status	Displays the subfolder status.

For more information, see Viewing Subfolder Properties.

Viewing File Properties

To view file properties, do the following:

1 Click File and then Open Project.

The Open Project dialog box opens.

2 Type the project name or select the project.

3 Click Open.

The project opens on the Projects Tab.

4 On the Project Tab, select a file.

5 Right-click the file, and then click Properties.

The File Properties dialog box opens.

6 Complete the File Properties dialog box.

7 Click OK.

For more information, see:

Project Properties

File Properties Dialog Box

File Properties Dialog Box

The table below describes the options and functionality of the File Properties dialog box:

Option	Description
Name	Displays the file name.
Full Specification	Displays the file location.
Description	Displays the file description. OPTIONAL: Type or edit the file description.
Include In Build	Sets the file to be included in the build.
Last Modified	Displays the date and time of the last modification.
Size	Displays the file size.
Status	Displays the file status.
Object Type	Displays the file object type. Select an object type if the file is unspecified.

Version Control

Version control archives files and tracks changes to files over time. With an integrated version control, you can easily track changes to database objects.

Version control addresses the following issues:

- Team Development By controlling access to a file so that only one person at a time can modify, it prevents accidental replacement or loss of another user's changes.
- Version Tracking By archiving and tracking versions of source code files, you can to retrieve them if necessary, thereby effectively creating files so that source code can be reused.
- Safety By adding database object scripts and files, it creates backups in case of loss, thereby ensuring a recovered version of source code.

When you create a project, you can immediately place the project into version control or you can Create a New Project From a Version Control Project. You can also add projects and files to version control later selecting **Project** menu, **Version Control** and **Add To Version Control...**.

Version Control Integration

To use integrated version control features, you must have the version control client software installed on the same computer, or the MSSCCI plug-in for the selected Version Control System. You must select the appropriate version control on the Version Control Tab of the Option Editor.

DBArtisan supports all Version Control Systems compatible with MSSCCI. Other Version Control Systems not compatible with MSSCCI may work under certain conditions.

> **Note:** 64 bit versions of DBArtisan can now use 32 bit MSSCCI providers by choosing to do so in the version control option page. Most providers are 32 bits only, so this allows users to continue to use the 64 bit version of our application even if the have a source control that has no 64 bit provider available.

Version Control Configuration

Information about how to configure a Version Control system is described in Version Control Options.

Using Version Control

Once you add a file or project to the version control, most functions found in the underlying version control system are available directly. Basic version control procedures include:

- Get Latest Version
- Check Out
- Check In

- Undo Check Out
- Open
- Delete
- Remove from Version Control
- Sort
- Show History
- Show Difference
- Version Control Properties

Note: Version control functionality depends on your underlying version control system. For more information on version control procedures, consult the documentation included with your version control system.

Topics

- Working with Projects in Version Control

Version Control Functionality - Open

The **Open** functionality opens the selected file(s) with the application registered for the type(s) of the selected file(s).

Version Control Functionality - Delete

The **Delete** functionality deletes the local copy of the selected item from the tree. To remove a file from version control, see Remove From Version Control.

Version Control Functionality - Sort

The **Sort** functionality sorts the tree items alphabetically.

Version Control Functionality - Get Latest Version

The Get Latest Version functionality lets you access the latest version of a file for viewing only. The Get functionality creates a local copy of the most current version of a project file in your working folder. The file is read-only, so any modifications cannot be saved.

Before working with a project, you should perform a Get on the entire project to ensure that you are working with the latest copy of the project. You should also perform a project-level, recursive Get at intervals to ensure that you have the latest version of files, in the event that they have been altered by others working on the same project.

The following table describes the option and functionality on the Get from Version Control dialog box:

Option	Description
Files to Get	Lets you select file(s) to get latest version of.
Advanced	Click to open the Advanced Get Options dialog box.

Tip: You can specify the file directory in the Version Control option of the Options Editor. For details, see Version Control Options.

Getting Latest Version of a Project

To get the latest version of project, do the following:

1. Click **File** and then **Open Project...**.

The **Open Project** dialog box opens.

2. Type the project name or select the project.
3. Click **Open**.

The project on the **Projects Tab** opens.

4. Click the project or target files.
5. On the **Project** menu, click **Version Control**.

The **Version Control** menu opens.

6. Click **Get Latest Version...**.

The **Get From Version Control** dialog box opens.

7. In the **Files to Get** box, click the project or files.
8. For advanced options, click **Advanced**.

9. Click **OK**.

The most current version of the file is written to your working directory.

Getting Latest Version of a File

To get the latest version of a file, do the following:

1. On the **Project** Tab, right-click the target file(s) and select **Get Latest Version...**.

The **Get From Version Control** dialog box opens.

2. In the **Files to Get** box, select the file(s).
3. For advanced options, click **Advanced**.
4. Click **OK**.

The most current version of the file is added to the **Project** Tab.

For more information, see Using Version Control.

Version Control Functionality - Check Out

The Check Out functionality retrieves a copy of one or more selected files and creates a writable working file copy in the working directory. You must perform a Check Out to edit any file that has been placed under version control.

You can check out a single file, multiple files at once or an entire project. A red check mark over the file icon indicates that the file has been checked out and is writable. This does not prevent other users from performing a Get or a Check Out on the file.

The following table describes the options and functionality on the Check Out File(s) dialog box:

Option	Description
Files to be Checked Out	Displays list of files that are eligible for check out. A black mark indicates that another user has the file(s) checked out.
Advanced	Click to open the Advanced Check Out Options dialog box.

Tip: You can specify the file directory in the Version Control Working Directory option of the Options Editor. For details, see Version Control Options.

Checking Out a Project

To check-out a project, do the following:

1. Click **File** and then **Open Project**.

The **Open Project** dialog box opens.

2. Type the project name or select the project.
3. Click Open.

The project opens on the **Projects** Tab.

4. Select the target project.
5. On the **Project** menu, click **Version Control**.
6. Select **Check Out**.

The **Check Out** dialog box opens.

7. In the **Files to Be Checked Out<** box select the project.
8. For advanced options, click **Advanced**.
9. Click **OK**.

The project or files are checked out from version control and writes the most current version of the file to your working directory.

Check Out a File

To check out a file, do the following:

1. On the **Project** Tab, right-click the target file(s) and select **Check Out**.

The **Check Out File(s)** dialog box opens.

2. In the **Check Out File(s)** dialog box select the files.
3. For advanced options, click **Advanced**.
4. Click **OK**.

The file(s) is checked out from version control and the most current version of the file is added to the **Project** Tab.

For more information, see Using Version Control.

Version Control Functionality - Check In

After editing your files, you must Check In the revised file in order save the changes you made to the file in a project. The Check In functionality stores the new version of the updated file in the current project. The Check In functionality is only available if you have Checked Out a file. You have the option to Check In an entire project or individual files.

Tip: You can specify the file directory in the Version Control Working Directory option of the Options Editor. For details, see Version Control Options.

The table below describes the options and functionality on the Check In dialog box:

Option	Description
Files to be checked in	Lets you specify the file(s) to check in.
Keep checked out	Adds latest version(s) of file(s) to source control but keeps the file(s) checked out.
Comment	OPTIONAL: Lets you type an optional comment.

Checking In a Project

To check-in a project, do the following:

1. Click **File** and then **Open Project**.
2. The **Open Project** dialog box opens.
3. Type the project name or select the project.
4. Click **Open**.

The project opens on the **Project** Tab.

5. On the **Project** menu, click **Version Control**, and then **Check In**.

The **Check In** dialog box opens.

6. In the **Files to Be Checked In** box click the project or files.
7. To update the version control copy but keep the project or files checked out so that you can continue working, select the **Keep Checked Out** check box.
8. **OPTIONAL:** In the **Comment** text box type a description of the changes.
9. Click **OK**.

Check In a File

To check in a file, do the following:

1. On the **Project** Tab, right-click the target file(s) and select **Check In**.

The **Check In File(s)** dialog box opens.

2. In the **Check In File(s)** dialog box select the files.
3. To update the version control copy but keep the files checked out so that you can continue working, select the **Keep Checked Out** check box.
4. **OPTIONAL:** In the **Comment** text box type a description of the changes.
5. Click **OK.**

For more information, see Using Version Control.

Version Control Functionality - Undo Check Out

If you decide that you do not want to save any revisions you have made to a checked out file, you can undo the procedure that releases the lock placed on the project or file. However, you do not have the option of deleting the local copy of the file.

Tip: You can specify the file directory in the Version Control Working Directory option of the Options Editor. For details, see Version Control Options.

The table below describes the options and functionality on the Undo Check Out dialog box.

Option	Description
Cancel the checkout for the following files	Lets you specify the files to undo checkout.
Advanced	Opens the Undo Check Out Advanced Options dialog box that lets you leave, replace, or apply the default action to your local copy.

Undoing Checkout for a Project

To undo a checkout, do the following:

1. Click **File** and then **Open Project**.

The **Open Project** dialog box opens.

2. Type the project name or select the project.
3. Click **Open**.

The project opens on the **Project** Tab.

4. Click the project.
5. On the **Project** menu, click **Version Control**.
6. Click **Undo Check Out**.

The **Undo Check Out** dialog box opens.

7. In the **Cancel the check out for the following files** box of the **Undo Check Out** dialog box, click the project.
8. Click **OK**.

   Note: If you Undo Check Out, you lose any changes you have made to the local copy of your project.

Undoing Checkout for a File

To undo checkout for a file, do the following:

1. On the **Project** Tab, right-click the target file(s) and select **Undo Checkout**.

The **Undo Checkout** dialog box opens.

2. In the **Cancel the check out for the following files** box of the **Undo Check Out** dialog box, click the project.
3. Click **OK**.

   Note: If you Undo Check Out, you lose any changes you have made to the local copy of your file(s).

For more information, see Using Version Control.

Version Control Functionality - Show History

The Show History functionality lets you view the history of version control files.

Showing History for a Project

To show history, do the following:

1. Click **File** and then **Open Project**.

The **Open Project** dialog box opens.

2. Type the project name or select the project.
3. Click **Open**.

The project opens on the **Project** Tab.

4. Click the target file.
5. On the **Project** menu, click **Version Control**, and then select **Show History**.

The **History** dialog box opens.

> **Note:** The **History** dialog box depends on your version control system.

Showing History for a File

To show history, do the following:

1. On the **Project** Tab, right-click the target file(s) and select **Show History**.

The **History** dialog box opens.

> **Note:** The **History** dialog box depends on your version control system.

For more information, see Using Version Control.

Version Control Functionality - Show Differences

The Show Differences functionality lets you view any differences between the current files in your working folder and the master files in the version control database. You cannot make changes to the files from this dialog box because it is used for display purposes only.

Viewing Project Differences

To view project differences, do the following:

1. Click **File** and then **Open Project**.

The **Open Project** dialog box opens.

2. Type the project name or select the project.
3. Click **Open**.

The project opens on the **Project** Tab.

4. Click the target file.
5. On the **Project** menu, click **Version Control**.
6. Click **Show Differences**.

The **Differences** dialog box opens.

If the file in your working directory is the same as the one in the project, a message tells you they are identical. If there are differences, the **Differences** dialog box from your version control system opens and displays the two versions of the file side-by-side, highlighting any differences.

7. In the **Differences** dialog box, you can maneuver through the files by using the **Up** and **Down** arrows.
8. To set **Diff Options**, click the **Options** button.

   Note: The **Differences** dialog box and **Options** depend on your version control system.

Showing History for a File

To show history, do the following:

1. On the **Project** Tab, right-click the target file(s) and select **Show History**.

The **Difference** dialog box opens.

2. In the **Difference** dialog box, you can maneuver through the files by using the **Up** and **Down** arrows.

> **Note:** The **Difference** dialog box and options depend on your version control system.

For more information, see Using Version Control.

Version Control Functionality - Remove from Version Control

The Remove from Version Control functionality lets you remove entire projects or files from version control.

Note: This functionality does not destroy the file permanently from the source control system or remove the local copies of the file(s).

Removing a Project from Version Control

To remove a project from Version Control, do the following:

1. Click **File and then** Open Project**.**

The **Open Project** dialog box opens.

2. Type the project name or select the project.
3. Click **Open**.

The project opens on the **Project** Tab.

4. On the **Project** menu, click **Version Control**, and then click **Remove from Version Control**.

The **Remove File(s)** dialog box opens.

5. To permanently destroy the project select the check box, click **OK**.

Removing a File from Version Control

To remove a file from Version Control, do the following:

1. On the **Project**, right-click the target file(s) and select **Remove from Version Control**.

The **Remove File(s)** dialog box opens.

2. Select the file(s) in the **Files to be Removed from Version Control** box.
3. Click **OK**.

Version Control Properties

The Version Control File Properties dialog box displays general information and check out status, links, and paths.

Viewing Version Control Properties for a Project

To view the version control properties, do the following:

1. Click **File** and then **Open Project**.

The **Open Project** dialog box opens.

2. Type the project name or select the project.
3. Click **Open**.

The project opens on the **Project** Tab.

4. Click the target file.
5. On the **Project** menu, click **Version Control**, and then **Version Control Properties**.

The **Version Control Properties** dialog box opens.

6. Review properties.
7. Click **Close**.

Viewing Version Control Properties for a File

To view the version control properties, do the following:

1. On the **Project** Tab, right-click the target file(s) and select **Version Control Properties**.

The **Version Control Properties** dialog box opens.

2. Review properties.
3. Click **Close**.

For more information, see Using Version Control.

Working with Projects in Version Control

Projects can exist independently of version control. You can add an entire project to version control at any point. When you decide to place a project under version control, a project is created on the underlying version control system that has been specified.

Once a project or file has been added to version control, the features of your version control system are available directly from the Project Tab. Any changes you make to a project or file from within DBArtisan are simultaneously changed in your version control system.

The table below describes the options and functionality on the Add to Version Control dialog box:

Option	Description
Files to be added	Lets you select files to add to Version Control.
Comment	OPTIONAL: Lets you add a comment.
Check out immediately	Select to add file and keep it checked-out.
Store only latest	Select to add the latest version.
Remove local copy	Select to add file and remove the local copy.

Completing the Add to Version Control Dialog Box

To add a project to version, do the following:

1. Click **File** and then **Open Project**.

The **Open Project** dialog box opens.

2. Type the project name or select the project.
3. Click **Open**.

The project opens on the **Project** Tab.

4. On the **Project** menu, click **Version Control**.

The **Add to Version Control** dialog box opens.

5. Click **Add to Version Control**.
6. Select options needed.
7. Click **OK** to add a project, and then click **Create**.

   Note: The project file icon dims to indicate that the project has been placed under version control.

For information on opening projects, see Working with Projects.

Tools

The **Tools** menu offers the following:

Tool	Description
Find in Files	Lets you find a phrase or character in your files.
Database Search	Lets you search for objects whose DDL contains specified strings.
Script/File Execution Facilities	Two stand-alone utilities that let you type or paste SQL statements to be executed or execute a script in a file.
Scheduling	Describes tools you can use to schedule tasks.
Visual Difference	Lets you compare two files or database objects.
Query Builder	Lets you build queries using a visual interface.
Import Data	Lets you import data from **.xls**, **.sql**, **.tab**, or **.csv** files.
SQL Profiler	Provides access to the PL/SQL Profiler.
Table/Index Size Estimator	Provides access to the **Estimate Size** object action.
Embarcadero Product Menu Commands	Correspond to installed or available Embarcadero Technologies products.
Available DBMS Utility Menu Commands	Lets you start third party utilities.
Code Workbench	Lets you set up keystroke-saving shortcuts that can be used in the SQL Editor.
Customize (Tools menu)	Lets you customize the user interface and set up shortcuts.
Options (Tools menu)	Lets you configure features and specify preferences.
Data Editor	Provides a live data editor.

Find in Files

The Find in Files tool lets you search files for a phrase or character.

To search files for specified characters:

1 On the **Tools** menu, select **Find in Files**. A **Find in Files** dialog opens.

2 Use the table below as a guide to choosing settings on this dialog box:

Option	Description
Find what	Specifies the character(s) or phrase you want to find. Use the browse arrow button next to the textbox to choose options from a pop-up list.
In files/file types	Specifies the files in which to search for the character(s) or phrase. Either enter the filename(s) in the drop-down box, or click the arrow to choose a file type.
In folder	Specifies the directory where the file(s) is located. Click the browse button to view your Windows Explorer.
Other common Search options	The **Find in Files** dialog lets you specify the following common search options: **Match whole word only**, **Match case**, **Regular Expression** (tells the application whether the specified character(s) is a regular expression), **Look in subfolders**, and **Output to Pane 2** (displays the results in another window).

3 When ready, click **Find**.

Database Search

The powerful Database Search tool lets you search for objects whose DDL contains a specified character string, across multiple databases.

Platform Availability

DB2 LUW	DB2 z/OS	ITB/FBD *	MySQL	ORCL	PSTGRS *	SQL SVR	SYB ASE	SYB IQ
✓	✓	✓	✓	✓	✓	✓	✓	

To find instances of a string on multiple databases:

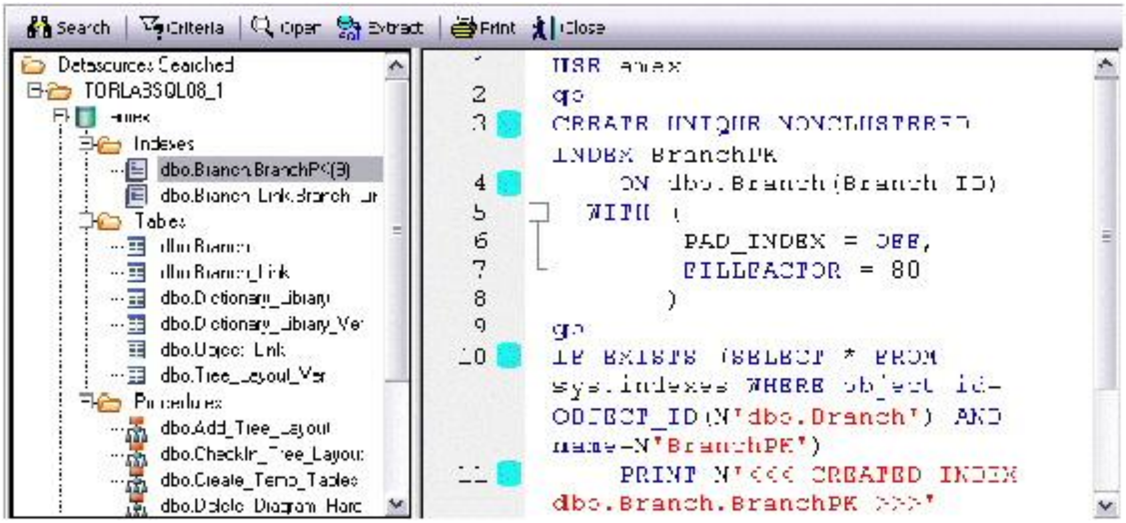
1 On the **Tools** menu, click **Database Search**. The **Database Search Wizard** opens.

2 Use the following table as a guide to understanding and modifying the settings on the panels of this wizard:

Panel	Settings and tasks
Select the datasources you want to search	Depending on the DBMS platform, you may also be able to further qualify by database, user, or schema.
Select the objects you want to search	Lets you specify the character string to search for, select a case matching setting, and specify the object types that are to be searched. Depending on the DBMS platform, searchable object types are **defaults**, **foreign keys**, **functions**, **indexes**, **materialized tables**, **package bodies**, **packages**, **primary keys**, **procedures**, **rules**, **tables**, **triggers**, **unique keys**, and **views**.

1 Click **Finish** to start the search.

A progress dialog box is displayed while the search runs. When the search completes, the results open in a **Database Search** window. The left pane lists all objects whose DDL satisfied the search. The right pane displays the DDL of a selected object. Lines containing the specified character string are flagged.



When viewing results, the following options are available from the **Database Search** window toolbar:

Button	Description
Search	Reopens the **Database Search** wizard with the most recent settings specified. This lets you further qualify settings for a new search.
Criteria	Opens a window displaying the case matching option and character string specified for the search.
Open	Opens the selected object in an object editor. For information on using object editors, see Modifying objects using editors.
Extract	Extracts the DDL for a selected object to a SQL editor window. For more information, see Using the SQL Editor.
Print	Lets you print the target object SQL.

Script/File Execution Facilities

Two stand-alone utilities establish multiple threads and database connections, letting you simultaneously execute SQL statements against multiple cross-DBMS platform datasources.

- Script Execution Facility lets you type or paste SQL statements to be executed
- File Execution Facility lets you load files containing SQL statements to be executed

Both utilities also work in conjunction with supported scheduling facilities, letting you schedule execution jobs. After completing a scheduled job, a results report lists errors, verifies script execution, and details the output of the job. Results can be automatically sent to any e-mail or network recipients.

Both utilities use a tabbed dialog box where you set the parameters and options for the script execution.

To execute SQL using one of these facilities

1 On the **Tools** menu, select either **Script Execution Facility** or **File Execution Facility**.

A tabbed dialog opens. It lets you provide the SQL to be executed and provide additional details.

2 Use the following table as a guide to selecting settings on the tab of this dialog:

Tab	Setting	Description
Script (Script Execution Facility only)	**Script** box	Lets you type or paste a script.
Files (File Execution Facility only)	**Show Full File Paths**	Select to display the full path. Deselect to display only the file name.
	File Name	Displays the file names.
	Add and **Remove**	Use these controls to add or remove files.
	View	Opens the View File dialog box.
	Up and **Down**	Use these controls to move the selected file up or down in the list.
Target	**Select the Target Datasource(s) to Execute the Script Against**	Only Connected Datasources - Displays only datasources that are currently connected in the Datasource grid. All DBMS Types - Displays all DBMS types in the Datasource grid.
	Datasource grid	Displays the target datasource(s) to execute the script/file against. Select a datasource name. If the datasource has multiple databases, type in a database in the Database box.

Output	**Graphical Output**	If selected, specifies a graphical output.
	File Output	If selected, specifies a file output. Directory - Type or browse to enter the full path and directory name in which you want to place the output file. File Type - Specifies a file type. Include column titles when saving - If selected, lets you save column titles. Open files with registered applications - If selected, opens files with registered applications.
Notify	**Job Description**	Lets you enter a job description. This description will be the subject of the notification E-mail.
	E-mail address	Lets you enter E-mail addresses. Separate each E-mail address with a semicolon (;).
	Net Send User Names	Lets you enter net send user names. Separate each name with a semicolon (;).

3 When ready, click **Execute**.

Scheduling

You can schedule routine tasks and jobs using the following utilities:

- Microsoft Task Scheduler
- ETSQLX Command Line Utility

Microsoft Task Scheduler

You can use the Microsoft Task Scheduler to schedule jobs. The Microsoft Task Scheduler is included with various Microsoft applications. If you do not have this program on your system, the first time you attempt to schedule a job, you are offered a link to the Microsoft Web site where you can download the Microsoft Task Scheduler at no cost.

Note: The ETSQLX command line utility runs a scheduled job even if the application is not running. For more information, see ETSQLX Command Line Utility.

To open the Microsoft Job Scheduler

1 On the **Tools** menu, click **Scheduler**.

2 Use the associated online Help for assistance with using this Microsoft utility.

ETSQLX Command Line Utility

The ETSQLX command line utility, is a multi threaded, cross-platform, SQL scripting engine. You can use ETSQLX in conjunction with the Microsoft Task Scheduler to schedule and automate routine jobs. ETSQLX creates batch files (with the extension.cfg) containing commands to execute automated and scheduled jobs. ETSQLX creates a directory, CFG, in which it stores the.cfg files. You can run.cfg files directly from the command line.

Note: ETSQLX supports.csv, .tab, .htm, and.html formats for result reports attachments.

Visual Difference

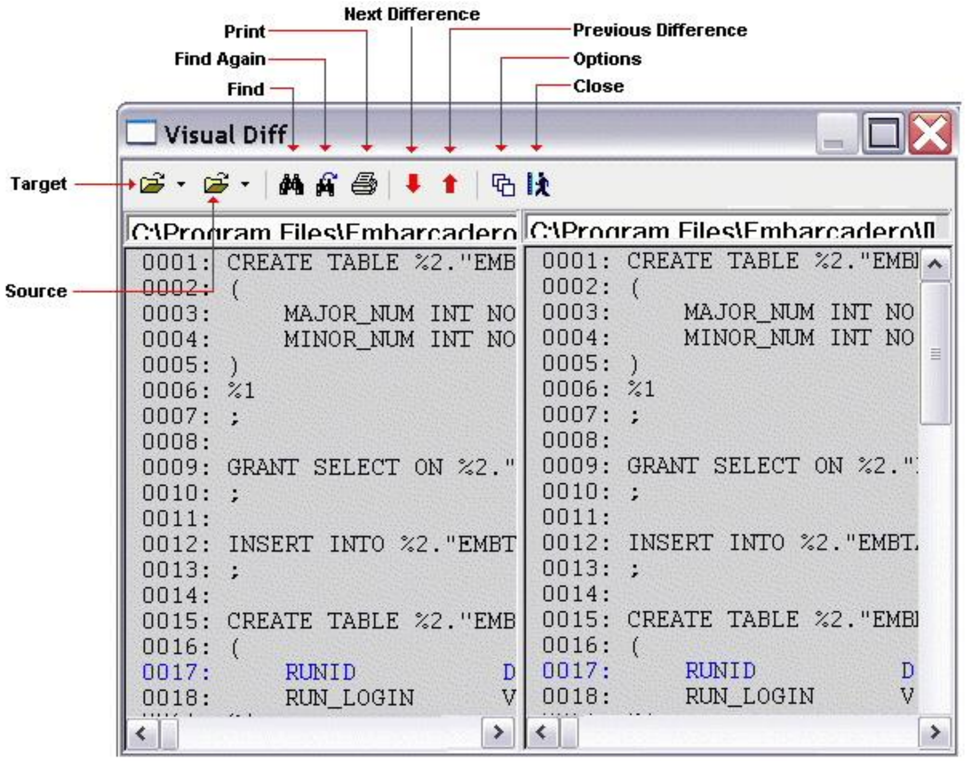
You can compare two text files or database objects (extracted DDL). Using the **Visual Diff** dialog box, you can easily synchronize and analyze database objects or files across multiple database platforms. The files are displayed side by side. The Visual Difference utility highlights any differences between two files. Viewing differences between objects and files helps you negotiate between the different phases of development as well as providing a visual aid to rapidly changing and evolving production environments.

Note: Because contents of the **Visual Diff** dialog box are read-only, you cannot modify files or objects directly from this dialog box.

To open the Visual Diff dialog box

1 On the **Tools** menu, click **Visual Diff**.

The **Visual Diff** dialog box opens.



C:\Program Files\Embarcadero

```
0001: CREATE TABLE %2."EMB
0002: (
0003:     MAJOR_NUM INT NO
0004:     MINOR_NUM INT NO
0005: )
0006: %1
0007: ;
0008:
0009: GRANT SELECT ON %2."
0010: ;
0011:
0012: INSERT INTO %2."EMBT
0013: ;
0014:
0015: CREATE TABLE %2."EMB
0016: (
0017:     RUNID           D
0018:     RUN_LOGIN       V
```

C:\Program Files\Embarcadero\

```
0001: CREATE TABLE %2."EMB
0002: (
0003:     MAJOR_NUM INT NO
0004:     MINOR_NUM INT NO
0005: )
0006: %1
0007: ;
0008:
0009: GRANT SELECT ON %2."
0010: ;
0011:
0012: INSERT INTO %2."EMBT
0013: ;
0014:
0015: CREATE TABLE %2."EMB
0016: (
0017:     RUNID           D
0018:     RUN_LOGIN       V
```

The **Visual Diff** dialog box is composed of two panes; the left pane displays your source object or file and the right pane shows your target object or file. The toolbar provides the basic visual search functions. See the following topics for details:

- Setting Visual Difference options
- Specifying the Files or Database Objects to Compare
- Navigating to the next or previous highlighted difference

  - Finding character strings in the source or target
  - Printing a Pane of the Visual Difference Dialog Box

Setting Visual Difference options

The **Visual Diff** dialog box lets you set display and comparison options to help you customize the dialog box to view differences in a comprehensive manner.

To set options for visual difference operations

1 On the **Tools** menu, click **Visual Diff**.

2 Click the **Options** icon on the toolbar.

⧉

The **Visual Diff Option**s dialog box opens.

3 Use the following table as a guide to setting options in this dialog:

Option	Description	Default
Display Line Numbers	Indicates that line numbers should be displayed.	Off
Display Hidden Characters	Indicates that hidden, nonprintable characters should be displayed.	Off
Ignore White Space	Indicates that White Space (such as spaces, carriage returns, line feeds, and tabs) should be ignored. If this option is set on, text will be considered equivalent regardless of white space. Otherwise the text will be shown as being different.	On
Ignore Hidden Characters	Indicates that nonprintable characters should be excluded.	Off
Ignore Case	Indicates that case should not be a differentiating factor.	On

4 Click **OK**.

Specifying the Files or Database Objects to Compare

You can compare two files or database objects side-by-side in the **Visual Diff** dialog box. The file or object you want to compare is called the source. The file or object you want to compare the source to is the target.

To open the source and target

1 On the **Tools** menu, click **Visual Diff**.

2 On the toolbar, click the down arrow next to the Source icon and then click **File** or **Database Object**.

The **Select the 1st File to Compare** or **Select the 1st Database Object to Compare** dialog box opens.

3 Take one of the following actions:

- Use the **Select the 1st File to Compare** dialog box to locate and select the source file
- Use the **Select the 1st Database Object to Compare** dialog box to navigate the datasource tree to locate and select the source database object.

4 Repeat those actions to select a target file or database object. A progress dialog is briefly displayed, showing processing of the differences between the source and target.

Note: The Visual Difference Utility highlights all differences between the two files.

Once you have added the source and target, you can navigate back and forward among the highlighted differences, search the source or target for character strings, or print the contents of the source or target pane. For details, see the following topics:

- Navigating to the next or previous highlighted difference
- Finding character strings in the source or target
- Printing a Pane of the Visual Difference Dialog Box

For more general information, see Visual Difference.

Navigating to the next or previous highlighted difference

You can navigate through the **Visual Diff** dialog box using the up and down arrow buttons. You can move back and forth between highlighted differences in your compared files or database objects.

To select the next or previous difference in the source and targets

1 Open the **Visual Diff** dialog box and load the source and target files or database objects. For details, see Specifying the Files or Database Objects to Compare.

2 On the toolbar, click the down arrow to go to the next difference or the up arrow to go to the previous difference.

⬇⬆

For more general information, see Visual Difference.

Finding character strings in the source or target

The **Visual Diff** dialog box lets you search for text in your files or database objects.

To search for text in the source or target pane

1 Open the **Visual Diff** dialog box and load the source and target files or database objects. For details, see Specifying the Files or Database Objects to Compare.

1 Place your cursor inside the pane you want to search.

2 Click the Find icon on the Visual Difference toolbar.

The **Find** dialog box opens.

3 In the **Find What** box, enter the search string.

4 Use the **Match**... controls specify whole word searching or case sensitivity.

5 Click **Find Next** to find the next occurrence of the search string.

After specifying the search string in a visual difference session, you can repeat the search without opening the **Find** dialog

To find the next instance of the search string

1 Click the Find Next icon on the Visual Difference toolbar.

For more general information, see Visual Difference.

Printing a Pane of the Visual Difference Dialog Box

You can print each pane of the **Visual Diff** dialog box.

To print the target or source pane

1 Open the **Visual Diff** dialog box and load the source and target files or database objects. For details, see Specifying the Files or Database Objects to Compare.

2 Position your cursor inside the pane you want to print.

3 Click the Print icon on the Visual Difference toolbar.

🖨

4 Click **OK.**

Note: You can only print one pane of the **Visual Diff** dialog box at a time.

For more general information, see Visual Difference.

SQL Profiler

The **Tools** menu provides access to the SQL Profiler, a license-specific module:

- For licensing information, see Licensing Your Product.
- For detailed information on using the profiler, see PL/SQL Profiler.

Table/Index Size Estimator

The **Tools** menu provides access to the **Estimate Size** object action, available against Oracle and Sybase ASE tables and indexes. For details, see Estimate Size.

Import Data

It is often necessary to import data into database tables from external sources. You may need to bring sample data into a database to establish test case scenarios or import data for use in the real world. Business analysts, for example, often use spreadsheets from outside sources. Putting such data into tables can help them perform detailed queries and make informed decisions based on meaningful data.

The **Import Data** tool lets you import data from a Microsoft Excel (**.xls**) file or any SQL data file format in which data values are stored in plain-text form and delimited by tab, semicolon, comma, space, or other user-specified character (such as **.sql**, **.tab**, or **.csv**).

Note: When you import a text file, the wizard may give the impression of stalling. Press ESC and you should be able to continue without further incident.

To Open the Import Data Wizard:

1 Connect to the datasource to which you want to import data and ensure that the datasource is selected. For more information, see Connecting to Datasources.

2 On the **Tools** menu, select **Import Data**. The **Import Data Wizard** opens.

3 Use the following table as a guide to providing information as you work through the panels of this wizard:

Panel	Setting	Description
Data Properties	**Specify the file to be used in this data load operation**	Type the path and name of the file that you want to imprt from or use the browse button to locate and select the file.
	Which table do you want to load data into?	If working against a Sybase or SQL Server datasource, select the **Catalog** of tables. Select the **Schema** that owns the target table. Select the particular **Table** where you want to import the data.
Delimiter	**What character delimited the columns in the data file?**	Choose among **Tab**, **Semicolon**, **Comma**, or **Space**. Alternatively, select **Other** and type the specific delimiting character.
	First Row Contains Field Names	By choosing this, if the first row contains column names, these will be displayed as the first row of the file. You can still change column mappings if you want. If you do not select this, generic **Field 1**, **Field 2**,... **Field***n* will indicate column mappings.

Column Mapping	**Note:** You must map all mandatory (NOT NULL) column fields to a column heading before you can enable the Finish button and import the data. To map columns, click the grey bar at the top of the grid and a drop-down list appears. The names in this drop-down list are columns in the table that's going to receive the data. These need to sync up to the data that's being imported. Red, or mandatory columns, turn blue in the column list when they have been assigned. A maximum of 10 rows are displayed here to make mapping columns easier. All rows will be imported when the job completes. If there are more columns in the file you are importing than in the receiving table, the extra columns will be ignored and you can create the insert statements. You cannot complete the import operation if columns are mapped more than once. The import operation will not let you generate invalid columns.	
Excel Column Mapping (.xls files only)	**Sheet**	If the file you selected has more than one sheet, identify the worksheet that has the data you want to import.
	Start/End Cell	Identify the starting and ending cells of data you want to import and click Refresh. Or, accept the default cells.

4 Click **Finish**.

Query Builder

Query Builder is a database productivity tool that lets you construct, structure, and manipulate up to five different types of queries simultaneously. It includes a separate graphical interface that opens within your current workspace. You can run Query Builder against all Embarcadero Technologies supported database platforms.

Query Builder displays the interconnections of your queries as you work. The powerful visual components of Query Builder let you see your query grow and change to fit your needs. Query Builder eases the task of drawing data from tables by automatically creating correct SQL code as you build a statement. You can use Query Builder to create and execute SELECT statements for tables and views. You can also test queries, and easily adjust your information, before you save. Query Builder does not rely on knowledge of the underlying SQL code.

You can save and reopen queries in Query Builder. Query Builder automatically checks for changes in your tables or columns between the time you save the query and the time you reopen it.

The table below describes the types of queries available in Query Builder:

Query Type	Description
Building a SELECT Statement	Create, manipulate and execute SELECT Statements for tables and views.
Building an INSERT Statement	Create and manipulate INSERT Statements for tables.
Building an UPDATE Statement	Create and manipulate UPDATE Statements for tables.
Building a DELETE Statement	Create and manipulate DELETE Statements for tables.
Building a CREATE VIEW Statement	Create and manipulate CREATE VIEW Statements for tables and views.

Note: You can execute SELECT statements directly from Query Builder. INSERT, UPDATE, DELETE, and CREATE VIEW statements must be moved to an ISQL Editor for execution.

For more information, see:

Query Builder Design

Using Query Builder

Query Builder Design

Query Builder lets you build DML statements using an intuitive, graphical interface. It offers you a powerful and flexible way to quickly create complex statements without sacrificing time manipulating SQL code. Query Builder lets you add tables or columns, create joins, and change statements within the graphic display. It also lets you have multiple sessions working at the same time.

Query Builder includes many different features to assist you in building and manipulating your query:

- Statement Properties
- Workspace Windows
- Query Builder Explorer Window
- Query Builder Toolbar
- SQL Diagram Pane
- SQL Statement Pane
- Tables and Views Shortcut Menus
- Query Builder Dialog Boxes

For more information, see Query Builder

Workspace Windows

The Workspace Windows provide a comprehensive view of your data structure and query. The table below describes the Workspace Windows:

Pane	Description
Query Builder Explorer Window	Includes two tabs that display selected object details: Tables/Views DML
SQL Diagram Pane	Displays tables or views included in the current query.
SQL Statement Pane	Displays the SQL code, and when appropriate, a Results Tab.

For more information, see Query Builder

Query Builder Explorer Window

The Query Builder Explorer is a separate tree that exposes all the tables and views in your target database. It also displays your current query structure. The Query Builder Explorer includes two tabs that display information about the selected objects:

Tables/Views Tab

The Tables/View Tab displays information about the selected tables or views. You can use the drop-down lists to change your table or view, and when appropriate, the owner. The table below describes each drop-down list on the Tables/Views Tab:

List	Description
First	Displays all databases for a target Microsoft SQL Server or Sybase ASE.
Second	Displays all valid owners.

Note: To change your current database, select the new database in the Explorer, and then open another Query Builder session. Query Builder prompts you to save the current session prior to opening a new session. For details, see Saving and Reopening Queries.

DML Tab

The DML Tab displays all the basic elements of a query statement in the SQL Statement Tree. You can access any element of the current statement display and perform SQL editing from the SQL Statement Tree.

For more information, see:

Creating a Clause Using the SQL Statement Tree

Query Builder

SQL Diagram Pane

The SQL Diagram Pane displays tables, views, and joins included in the current query. You can manipulate elements of your query, using the mouse functionality, in the SQL Diagram Pane. From the SQL Diagram Pane you can:

- Working with Tables and Views in the SQL Diagram Pane
- Joins
- Working with Columns in the SQL Diagram Pane

All changes in the SQL diagram reflect in correct SQL code in the SQL Statement Pane.

For more information, see Query Builder

SQL Statement Pane

The SQL Statement Pane displays the current query SQL code. When you run a query, Query Builder displays results to your query in the SQL Statement Pane. The SQL Statement Pane is divided into two tabs:

SQL Tab

The SQL Tab displays the query in progress. It displays each element of your query as you build it, and updates as you do edits such as selecting or deselecting columns, adding clauses, and creating joins. You can open the current statement directly into an ISQL editor or copy it to the clipboard for later use.

Results Tab

The Results Tab displays the results of your executed query in the Results grid. To edit data, use the Data editor application from Query Builder. When you begin building a new query, the tab title changes to Old Results until you execute the new query. For details, see Using Data Editor with Query Builder.

For more information, see Query Builder

Query Builder Toolbar

The Query Builder tool bar lets you access commonly used features. The table below describes Query Builder tool bar functionality:

Name	Function
Copy	Copies the current SQL statement to the clipboard.
Statement Box	Displays the type of statement currently on display in the main workspace window.
Stop Execution	Stops an executing query.
Execute	Executes the current SELECT or CREATE VIEW statement. If the button is not available, the statement is not executable.
New	Adjusts to the target node in the Query Builder Explorer window.
Edit	Displays, on the DML Tab, the ORDER BY or GROUP BY dialog boxes when target node is selected.
Delete	Deletes the target object.
Auto Layout	Resets the main workspace to the auto layout mode.
Auto Join	Finds and joins, automatically, like items by name.
Statement Check	Checks query syntax.
Edit Data	Opens Data Editor.
Close	Closes the current query.

Note: Query Builder adjusts tool availability to match the current query functionality.

For more information, see Query Builder.

Tables and Views Shortcut Menus

Query Builder includes a shortcut menu that lets you manipulate a table or view. The table below describes the table shortcut options:

Option	Description
Delete	Removes the table from the SQL Diagram Pane, and the SQL Statement.
Title Font	Specifies the table title font for this diagram.
Column Font	Specifies the column font for this diagram.
Background Color	Specifies the table background color for this diagram.
Select Star	Selects every column in the table.
Select None	Deselects every column in the table.
Bring to Front	Moves the table to the top layer of the diagram.
Properties	For more information, see Table Properties.

Note: Your selection applies to all selected tables and views.

For more information, see Query Builder

Tables and Views Keyboard Commands

Query Builder provides a number of keyboard shortcuts that let you quickly construct queries. The table below describes the keyboard commands:

Keyboard Command	Location	Description
ESCAPE	SQL Diagram Pane	Breaks off a join.
F5	Query Builder	Refreshes screen and runs Schema Change Detection. In a CREATE VIEW, this key adds the new view to the Table Tree Pane.
CTRL A	SQL Diagram Pane	Selects all tables and joins in the current diagram.
F1	Query builder and application	Obtains context sensitive Help.

For more information, see Query Builder

Query Builder Dialog Boxes

Query Builder includes a number of dialog boxes to assist you in building and customizing your query.

Dialog Box	Description
Statement Properties	Specifies general properties in an individual Query Builder session.
Table Properties	Specifies column selection and alias names for a table or view.
Column Properties	Specifies column functionality within SELECT and CREATE VIEW statements.

For more information, see Query Builder

Statement Properties

The Statement Properties dialog box lets you customize properties in an individual Query Builder session. For example, you can set options to limit the number of rows returned in a query to save execution time, or turn off the auto join function to manually control all joins for an individual query. These properties override the global options set in the Options editor for the current session. For details, see Query Builder Options.

The table below describes the options and functionality of the Statement Properties dialog box.

Interface Element	Option	Description	Default
Code Generation	Generate Use Database statement	Adds a line of SQL code indicating which database or instance is used in the statement.	Selected
	Generate owner names	Adds a line of SQL code showing the table owner name as part of the query.	Selected
	Include Row Count limits	Includes the output row limit set in the Execution settings.	Selected
Execution	Max Row Count in Results Set	Sets row count limits to build and check a query without congesting server processes when a query executes.	1000 rows
General	Show Column Data types in Query Diagram	Lets Query Builder reveal the data type in each column for tables in the SQL Diagram Pane.	Not selected
	Confirm on Item delete	Lets Query Builder open a Confirm Delete dialog box when an item is deleted. NOTE: Clearing this function can result in unexpected changes to your query diagram and statement.	Selected
	Auto populate views	Lets Query Builder automatically populate views.	Not selected

Auto Join	Require Indexes	Joins indexed columns automatically, and requires indexed columns for joins.	Selected
	Require same data type	Automatically joins columns with the same data type.	Selected
Syntax Checker	Automatic Syntax Check	Lets Query Builder check syntax every time an execute statement, refresh or copy statement begins.	Selected
	Run Automatically	Lets Query Builder automatically detect like names and data types and create joins for multiple tables.	Selected
Display	Columns Font	Lets you set the font, font style, size, and color of column fonts.	Available
	Title Font	Lets you set the font, font style, size, and color of table/view title fonts.	Available
	Table Color	Lets you set the background color of your tables in the SQL Diagram Pane.	Available

Note: If you set options in the Options editor while Query Builder is running, a warning indicates that you are about to change options or properties. For details, see Query Builder Options.

Completing the Statement Properties Dialog Box

To complete the Statement Properties dialog box, do the following:

1 On the Query Builder menu, click Statement Properties.

2 Set options.

3 Click OK.

Query Builder saves the options.

For more information, see Query Builder

Table Properties

The Tables Properties dialog box lets you set parameters for tables or views in your SQL Diagram. The table below describes the options and functionality on the Table Properties dialog box.

Option	Description
Table Alias	Creates an alias name for your table.
Show Datatypes	Shows or hides the datatype for every column in the target table.
Displayed Columns	Displays columns visible in the SQL Diagram.
Hidden Columns	Displays columns hidden in the SQL Diagram.
Hide All	Moves all non selected columns in the table to the Hidden Columns window.
Display All	Moves all columns in the table to the Displayed Columns window.
Right Arrow	Moves a target file from Displayed Columns to Hidden Columns.
Left Arrow	Moves a target file from Hidden Columns to Displayed Columns.

Completing the Table Properties Dialog Box

To complete the Table Properties dialog box, do the following:

1 Double click the target table or view title bar.

OR

Right-click target table or view, and then click Properties.

2 If you only want to hide or display columns in your table, click the arrow button on the table title bar.

3 You can also edit view properties from the Table Properties dialog box.

4 Click OK.

Query Builder saves the changes.

For more information, see Query Builder

Column Properties

The Column Properties dialog box lets you set properties for individual columns in your SELECT or CREATE VIEW statements. You can set aggregate functions and create an alias for an individual column.

The Column Properties dialog box is not available for INSERT, UPDATE or DELETE statements.The table below describes the options and functions Columns Properties dialog box:

Interface Element	Description
Tables/Views	Displays all tables and views in the SQL Diagram Pane.
Aggregate	Specifies aggregate options for the target column. AVG - An average is taken for a column with an int or numeric datatype. COUNT - Returns the number of rows which contain data for the target column. MAX - Returns the highest number in a row in the column. MIN - Returns the lowest number in a row in the column. SUM - Returns the sum of the target column in all rows which contain data. This function is only operable on int or numeric datatypes.
Alias	Displays the alias name for the target column. Lets you type the name of the alias. NOTE: Query Builder displays the results of an aggregate column without a column name unless you create an alias for that column.
Available Columns	Displays all available columns in the target table or view.
Selected Columns	Displays all selected columns in the target table or view. To create an aggregate function or alias for a different column, select target column, select an aggregate function, and then type the name of the alias.
Select All	Moves all columns in the Available Columns box to the Selected Columns box.
Clear All	Moves all columns in the Selected Columns box to the Available Columns box.
Right Arrow	Moves target column in the Available Columns box to the Selected Columns box.
Left Arrow	Moves target column in the Selected Columns box to the Available Columns box.
Select List Statement	Displays the current query.

Completing the Column Properties Dialog Box

To complete the Column Properties dialog box, do the following:

1 On the SQL Statement Tree, double-click target column.

2 Select options.

3 Click OK.

For more information, see Query Builder

Join Properties

Query Builder lets you edit joins with the Join editor. You can edit join parameters in a SELECT, UPDATE, DELETE, and CREATE VIEW Statement.

The table below describes the options and functionality on the Join dialog box.

Option	Description
From Table Column	The primary column in the join.
To Table Column	The secondary column in the join.
Select the join relation operator	Click the target join operator. If it is not equals, the operator displays on the join in the SQL Diagram Pane.
Join Type: Inner	Click to make the join an inner join. Aggregates are only available for inner joins.
Join Type: Left Outer	Click to make the join a left outer join.
Join Type: Right Outer	Click to make the join a right outer join.

Note: For IBM DB2 for Linux, Unix, and Windows and IBM DB2 for z/OS and OS/390 servers, there is an additional join object in the SQL Statement Tree. The Join On node displays join relations between columns in IBM DB2 for Linux, Unix, and Windows and IBM DB2 for z/OS and OS/390 tables and views.

Completing the Join Dialog Box

To complete the Join dialog box, do the following:

1 In the SQL Diagram Pane, right-click the target join, and then click Properties.

OR

In the SQL Diagram Pane, double-click the target join.

OR

On the SQL Statement Tree, expand the Where and And nodes, and then double-click the target join.

2 Select options.

3 Click OK.

For more information, see Query Builder

Using Query Builder

Query Builder provides a visual display of your queries as you construct them. You can run Query Builder against any registered datasource. Query Builder lets you build five separate types of queries simultaneously:

- Building a SELECT Statement
- Building an INSERT Statement
- Building an UPDATE Statement
- Building a DELETE Statement
- Building a CREATE VIEW Statement

You can execute a SELECT statement from Query Builder. To execute an INSERT, UPDATE, DELETE, and CREATE VIEW statement, copy them to an ISQL Editor. You can also copy the statements to the clipboard for later use in the ISQL Editor.

Query Builder also lets you save a statement at any time so that you can open them later for editing or execution. Saving and Reopening Queries.

You can open Query Builder with multiple tables or views with the same or different owners. If you open tables or views with different owners, Query Builder displays "All Owners" in the Owner drop-down list. You can start multiple Query Builder sessions. You can use different tables and views for each query. You can also toggle back and forth among each of the queries. For more information, see Working with Tables and Views in the SQL Diagram Pane.

You can save and reopen queries in Query Builder. Query Builder automatically checks for changes in your database or instance between the time you save the query and the time you reopen it with the Schema Change detection component.

Query Builder is integrated with Data editor so you can edit data in real time and then continue to build your query with the new information embedded in the query. For details, see Saving and Reopening Queries.

Topics

- Main Steps in Using Query Builder
- Building a Query
- Working with Tables and Views in the SQL Diagram Pane
- Working with Columns in the SQL Diagram Pane
- Selecting ALL or DISTINCT Columns
- Joins
- Auto Layout
- Auto Joins

- Creating a Clause Using the SQL Statement Tree
- Subqueries
- Syntax Checker
- Saving and Reopening Queries
- Using Data Editor with Query Builder

Main Steps in Using Query Builder

To use Query Builder, do the following:

- Selecting a Database
- Selecting a Statement
- Selecting Tables and Views
- Selecting Columns

Note: You can start Query Builder directly from a table or view which automatically selects the instance or database which contains that table or view.

1 On the Tools menu, click Query Builder.

Query Builder opens.

Selecting a Database

To create an SQL statement, first select an instance or database.

Note: You can start Query Builder directly from a table or view which automatically selects the database which contains that table or view.

If you are working with Microsoft SQL Server or Sybase ASE, Query Builder provides two drop-down lists. The first drop-down list displays all available databases for the target server. The second drop-down list displays owners.

If you are working with Oracle, IBM DB2 for z/OS and OS/390 or IBM DB2 for Linux, Unix, and Windows the first drop-down list is unavailable.

Note: You can start Query Builder directly from a table or view which automatically selects the database which contains that table or view.

1 Start Query Builder.

2 In the database drop-down list, click the target instance or database.

3 In the owners drop-down list, select the appropriate owner.

Query Builder is ready for Statement selection.

4 To select different instances or databases while Query Builder is running, on the Tables/Views Tab, in the database drop-down list, click the target instance or database.

The current query clears and a warning prompt is displayed.

5 To save the current query, click Yes.

The Save As dialog box opens.

6 To continue without saving, click No.

For more information on saving queries in Query Builder, seeSaving and Reopening Queriess.

Selecting a Statement

Query Builder lets you build SELECT, INSERT, UPDATE, DELETE, and CREATE VIEW queries simultaneously.

To select a statement, do the following:

1 On the Query Builder tool bar, click the statement drop-down list, and then click the target statement type.

OR

In the SQL Diagram Pane, right-click, and then click the target statement type.

For more information, see Using Query Builder.

Selecting Tables and Views

To build a query, open one or more tables or views in the SQL Diagram Pane. For details, see SQL Diagram Pane.

You can use different tables or views for each type of query.

Tip: For multiple tables: Press SHIFT+click for adjacent tables or CTRL+click for nonadjacent tables. You can also drag the bounding line with your pointer to select multiple tables.

To select a Table or View, do the following:

1 In the Tables/Views Tab, drag the target table or view to the Diagram Pane.

OR

In the Tables/Views Tab, click target table or view and then, on the Query Builder tool bar, click Add.

OR

In the Tables/Views Tab, right-click target table or view, and then click Add.

Query Builder displays the target table(s) and view(s) in the SQL Diagram Pane.

For more information, see Using Query Builder.

Selecting Columns

You must select at least one column to build a query. Query Builder displays columns in each table in the SQL Diagram window. By default, Query Builder exposes every column in a table. You can select the columns you want to use for your query. Query Builder orders them, in your statement, in the select order.

Query Builder lets you select columns. For details, see:

  - SQL Diagram Pane.
  - Selecting Columns in the SQL Statement Tree.

You can select an individual column or all columns. Query Builder orders them, in your statement, in the select order. You can reorder columns after you set them in your diagram or statement.

For more information, see:

Selecting Columns in the SQL Diagram Pane

Selecting Columns in the SQL Statement Tree

Selecting all Columns

Selecting Columns in the SQL Diagram Pane

To select a column in the SQL Diagram Pane, do the following:

1 Select the check box to the left of the target column name.

For more information, see Using Query Builder.

Selecting Columns in the SQL Statement Tree

Query Builder lets you select and set individual properties using the **Selected Column Properties** Dialog Box. For details, see Column Properties.

For more information, see Using Query Builder.

Selecting all Columns

Query Builder uses columns in statements based on the order of selection. When you select all columns, Query Builder displays the columns in the order they appear in the table.

Note: Query Builder lets you select all columns in single or multiple tables.

To select all columns, do the following:

1 On the Query Builder menu, click Select Star.

For more information, see Using Query Builder.

Building a Query

Query Builder lets you build SELECT, INSERT, UPDATE, SELECT, and CREATE VIEW statements, which you can run separately or simultaneously, depending on your needs.

To build a Query, do the following:

- Selecting a Database
- Selecting a Statement
- Selecting Tables and Views
- Selecting Columns

Note: You can start Query Builder directly from a table or view which automatically selects the database which contains that table or view.

Query Builder lets you build queries that include both tables and views in the SQL Diagram Pane for SELECT and CREATE VIEW statements. For the INSERT, UPDATE, and DELETE statements, use one or the other object, but you cannot use both.

Once you make your selections, you can edit, restructure, and streamline your query. Query Builder offers many options for streamlining your queries.

For more information, see:

Working with Tables and Views in the SQL Diagram Pane

Working with Columns in the SQL Diagram Pane

Joins

Creating a Clause Using the SQL Statement Tree

Selecting Columns in the SQL Statement Tree

Subqueries

Aliases

Building a SELECT Statement

Query Builder lets you construct and execute simple-to-complex SELECT statements using data from any table or view. You can also create and edit joins for SELECT statements. Query Builder can check your query and warn you if there are syntax errors with the Syntax Checker.

For related information, see Joins and Syntax Checker.

To build a SELECT statement, do the following:

1 On the Tools menu, click Query Builder.

The Query Builder opens.

2 In the statement drop-down list, click SELECT.

3 In the Table Tree Pane, select target table(s) or view(s) and move them to the SQL Diagram Pane.

4 In the target table or view, click target column(s), or click Select Star to select every column.

5 To check syntax, click Check.

6 To copy the statement, click Copy.

7 To execute the statement, click Execute.

Copying a SELECT Statement from the SQL Statement Pane

To copy any part of a statement from the SQL Statement Pane, do the following:

1 Open Query Builder, then begin a new SELECT statement.

OR

Open an existing SELECT statement.

2 In the SQL Statement Pane, select all, or the target portion of the statement.

3 On the Query Builder tool bar, click Copy.

OR

In the SQL Statement Pane, right-click, and then click Copy.

Query Builder makes the target statement portion available on the clipboard.

For more information, seeQuery Builder

Building an INSERT Statement

Query Builder lets you construct and execute simple-to-complex INSERT statements using data from any table. To execute an INSERT statement, copy it to an ISQL Editor. You can also copy the statement to the clipboard for later use in the ISQL Editor.

Query Builder also lets you save your statement at any time so that you can open it later for editing or execution. For details, see Saving and Reopening Queries.

Building an INSERT Statement

To build an INSERT Statement, do the following:

1 On the Tools menu, click Query Builder.

The Query Builder opens.

2 In the statement drop-down list, click INSERT.

3 In the Table Tree Pane, select target table, and move it to the SQL Diagram Pane.

4 In the target table, click target column(s).

Copying an INSERT Statement from the SQL Statement Pane

To copy any part of a statement from the SQL Statement Pane, do the following:

1 Open Query Builder, then begin a new INSERT statement.

OR

Open an existing INSERT statement.

2 In the SQL Statement Pane, select all, or the target portion of the statement.

3 On the Query Builder tool bar, click Copy.

OR

In the SQL Statement Pane, right-click, and then click Copy.

Query Builder makes the target statement portion available on the clipboard.

For more information, seeQuery Builder

Building an UPDATE Statement

Query Builder lets you construct and execute simple-to-complex UPDATE statement using data from any table. To execute an UPDATE statement, copy it to an ISQL Editor. You can also copy the statement to the clipboard for later use in the ISQL Editor.

Query Builder also lets you save your statement at any time so that you can open it later for editing or execution. For details, see Saving and Reopening Queries.

Building an UPDATE Statement

To build an UPDATE statement, do the following:

1 On the Tools menu, click Query Builder.

The Query Builder opens.

2 In the statement drop-down list, click UPDATE.

3 In the Table Tree Pane, select target table and move it to the SQL Diagram Pane.

4 In the target table, click target column(s).

Copying an UPDATE Statement from the SQL Statement Pane

To copy any part of a statement from the SQL Statement Pane, do the following:

1 Open Query Builder, then begin a new UPDATE statement.

OR

Open an existing UPDATE statement.

2 In the SQL Statement Pane, select all, or the target portion of the statement.

3 On the Query Builder tool bar, click Copy.

OR

In the SQL Statement Pane, right-click, and then click Copy.

Query Builder makes the target statement portion available on the clipboard.

For more information, seeQuery Builder

Building a DELETE Statement

Query Builder lets you construct DELETE statements using data from any table. Query Builder displays a Confirmation Option Message box when you create a DELETE statement. You can set the **Statement Properties** dialog box to display or hide this message when creating a DELETE statement. For details, see Statement Properties.

To execute a DELETE statement, copy it to an ISQL Editor. You can also copy the statement to the clipboard for later use in the ISQL Editor.

Query Builder also lets you save your statement at any time so that you can open it later for editing or execution. For details, see Saving and Reopening Queries.

Building a DELETE Statement

To build a DELETE statement, do the following:

1 On the Tools menu, click Query Builder.

The Query Builder opens.

2 In the statement drop-down list, click DELETE.

3 In the Table Tree Pane, select target table, and move it to the SQL Diagram Pane.

Copying a DELETE Statement from the SQL Statement Pane

To copy any part of a statement from the SQL Statement Pane, do the following:

1 Open Query Builder, then begin a new DELETE statement.

OR

Open an existing DELETE statement.

2 In the SQL Statement Pane, select all, or the target portion of the statement.

3 On the Query Builder tool bar, click Copy.

OR

In the SQL Statement Pane, right-click, and then click Copy.

Query Builder makes the target statement portion available on the clipboard.

For more information, seeQuery Builder

Building a CREATE VIEW Statement

Query Builder lets you construct and execute simple-to-complex CREATE VIEW statements using data from any table or view. You can also copy the statement to the clipboard for later use in the ISQL Editor.

Query Builder also lets you save your statement at any time so that you can open it later for editing or execution. For details, see Saving and Reopening Queries.

To build a CREATE VIEW statement, do the following:

1 On the Tools menu, click Query Builder.

The Query Builder opens.

2 In the statement drop-down list, click CREATE VIEW.

3 In the Table Tree Pane, select target table or view and move it to the SQL Diagram Pane.

Note: Query Builder supports multiple tables and views in a CREATE VIEW statement.

4 In the target table or view, click the target column(s).

5 To check syntax, click Check.

6 To copy the statement, click Copy.

7 To execute the CREATE VIEW Statement, click the SQL Statement Pane, and then press any key.

Query Builder opens the Edit SQL dialog box.

8 Click OK.

Caution: If you have used this method previously, and you selected the Please do not show me this dialog again check box, on the Edit SQL dialog box, Query Builder does not display the Edit SQL dialog box. It pastes your statement directly to the ISQL Editor.

The ISQL Editor opens.

9 In the ISQL Editor, on the line, CREATE VIEW NAME AS, replace the word NAME with a name for your view.

10 On the tool bar, click Execute.

The CREATE VIEW query executes.

11 To close the Editor, click Close.

The ISQL Editor save message opens.

12 Click No.

13 To add the view to the table tree, on the Query Builder menu, click Refresh.

Query Builder adds the view to the Table Tree Pane.

Copying a CREATE VIEW Statement from the SQL Statement Pane

To copy any part of a statement from the SQL Statement Pane, do the following:

1 Open Query Builder, then begin a new CREATE VIEW statement.

OR

Open an existing CREATE VIEW statement.

2 In the SQL Statement Pane, select all, or the target portion of the statement.

3 On the Query Builder tool bar, click Copy.

OR

In the SQL Statement Pane, right-click, and then click Copy.

Query Builder makes the target statement portion available on the clipboard.

For more information, seeQuery Builder

Working with Tables and Views in the SQL Diagram Pane

Query Builder lets you organize your tables and views in the SQL Diagram Pane. You can also customize appearance, change visual aspects, and adjust layout while continuing to manufacture a query. You can resize or customize a selected table and view, or move them to the front or back of the diagram. The key symbol indicates a column that is indexed or participates in a primary key.

- Selecting and Deselecting Tables and Views
- Moving Tables and Views
- Moving Additional Tables and Views to the SQL Diagram Pane
- Deleting a Table or View

Query Builder can automatically dictate a layout in the SQL Diagram Pane using the **Auto Layout** button. For details, see Auto Layout.

Selecting and Deselecting Tables and Views

You can select tables and views in the SQL Diagram Pane. You can make changes to more than one table or view simultaneously by selecting multiple tables or views.

To select and deselect Tables and Views, do the following:

1 To select a table, click the table title bar.

2 To select more than one table, drag the pointer to enclose all target tables with the bounding line.

Query Builder selects all target tables; none have handles.

3 To select all tables, in the SQL Diagram, right-click, and then click Select All.

4 Click the SQL Diagram workspace to deselect all tables.

For more information, see **Working with Tables and Views in the SQL Diagram Pane**.

Moving Tables and Views

Query Builder lets you move tables and views in the SQL Diagram Pane. It also moves selections and joins with the tables and views.

To move Tables and Views, do the following:

1 To move a table or view, drag the title bar to the target location.

Note: If you select more than one table or view, Query Builder moves all selected tables and views and any joins with the pointer.

For more information, see **Working with Tables and Views in the SQL Diagram Pane**.

Moving Additional Tables and Views to the SQL Diagram Pane

Query Builder sets tables and views in your statement in the order that you move them to the SQL Diagram Pane. Tables and views moved into the Diagram Pane appear first in your statement, including all joins connecting that table. To change the order of tables, move them back into the Table Tree and re-select them in the order in which you would like to join them.

Moving Additional Tables or Views

To move additional tables or views, do the following:

1 Click the target table or view and drag it to the Diagram Pane.

For multiple tables or views: Use SHIFT+click for adjacent tables or views or use CTRL+click for nonadjacent tables and views.

OR

Click the target table or view, and then on the Query Builder tool bar, click Add.

OR

Right-click the target table or view, and then click Add.

For multiple tables or views: Use SHIFT+click for adjacent tables or views or use CTRL+click for non-adjacent tables and views.

Note: Moving a table or view to the SQL Diagram Pane is not available while a query is executing.

For more information, see **Working with Tables and Views in the SQL Diagram Pane**.

Deleting a Table or View

To delete tables from the SQL Diagram Pane, do the following:

1 Right-click the target table or view, and then click Delete.

OR

In the SQL Diagram, click target table or view, and then on the Query Builder tool bar, click Delete.

OR

In the SQL Diagram, right-click the target table or view, and then click Delete.

Query Builder deletes the table from the SQL Diagram, SQL Statement, and SQL Statement Tree.

For more information, see **Working with Tables and Views in the SQL Diagram Pane**.

Working with Columns in the SQL Diagram Pane

You can customize queries by selecting and deselecting columns in the SQL Diagram Pane. You can customize columns using the **Selected Column Properties** dialog box. For details, see Column Properties.

Selecting and deselecting columns

You can select and deselect columns in the SQL Diagram. Query Builder lets you select and deselect individual columns or all columns. Your results reflect the order of selection. You can change the order of columns after you set them in your diagram or statement.

Tip: You can also select, re-order and deselect columns in the SQL Statement Tree. For more information, see Selecting Columns in the SQL Statement Tree.

Selecting Individual Columns

To select individual columns, do the following:

1 To select a column, in the SQL Diagram, select the check box to the left of the target column name.

Deselecting Individual Columns

To deselect individual columns, do the following:

1 To deselect a column, in the SQL Diagram, select the check box to the left of the target column name.

Note: When you clear the columns, Query Builder deletes the columns and any sub clauses from the SQL Statement Pane and SQL Statement Tree.

Selecting All Columns

To select all columns, do the following:

1 On the Query Builder menu, click Select Star.

Note: Query Builder uses columns in statements based on the order of selection. When you select all columns, Query Builder displays the columns as they appear in the table.

Deselecting All Columns

To deselect all columns, do the following:

1 On the Query Builder menu, click Select None.

Query Builder adds or removes selected columns from the SQL Statement Tree and the SQL Statement Pane.

Selecting ALL or DISTINCT Columns

Selecting ALL or DISTINCT columns is a way to filter data in your query. Selecting ALL columns means all rows displays results in the grid regardless of duplication in non-primary key columns. The DISTINCT column function is a query process that limits duplicate data in non-primary key columns to rows with the first iteration of any identical data. For example, if there are two identical addresses for different last names, and the column with a primary key does not participate in the query, only the row with the first instance of the address displays in the results of the query.

To select ALL or DISTINCT columns, do the following:

1 In the Statement Tree pane, right-click the ALL or DISTINCT node, click Properties, and then select the ALL or DISTINCT check box.

OR

In the Statement Tree pane, double click the ALL or DISTINCT node. Query Builder toggles to the opposite function.

Note: You can change between ALL or DISTINCT at any time prior to executing or copying a query.

Joins

Joins let you distill the information in your database to a usable form. Query Builder lets you create, manipulate, and edit work with joins without requiring knowledge of the underlying SQL code. Query Builder lets you create any type of join for SELECT and CREATE VIEW Statements. You can create self joins for UPDATE or DELETE Statements. You cannot create joins for INSERT Statements.

Query Builder includes four types of joins. The table below describes joins and their availability in Query Builder:

Join	Statement Availability	Description
Inner Joins	SELECT, CREATE VIEW, DELETE, UPDATE	Returns data from the joined tables that match the query's join criteria and set a relation between tables or views. Inner joins return results where the join condition is true.
Left Outer Joins	SELECT, CREATE VIEW	Returns all data from the primary table and data from the joined tables that match the query's join criteria and set a join relation operator from a column in a primary table or view to a column in a secondary table or view.
Right Outer Joins	SELECT, CREATE VIEW	Returns all data from the primary table and data from the joined tables that match the query's join criteria and set a join relation operator from a column in a secondary table or view to a column in a primary table or view.
Self Joins	SELECT, CREATE VIEW	Set a relation between columns in the same table.

In the Query Builder SQL Diagram Pane, you can create, edit, and delete joins. You can edit joins in the **Join** dialog box. For details, see Editing Joins.

You can set Query Builder options, in the Options editor, to automatically create joins. For details, see Query Builder Options.

Joins are the way you can filter data in relational databases. Query Builder lets you change the types of joins between tables, views and columns. It is important that you have some knowledge of the data in your tables, and the datatypes for each column. This information helps you frame a better query, and filter your data for maximum effect.

Inner Joins

Inner joins are the most common types of joins for SELECT statements. An inner join returns information from two tables where the relation between two target columns is true for both columns.

The join operand determines the relation results, for example, if the join operand is equals, then identical data, in two columns, is the only result. If the join operand is not equals, Query Builder only returns data that is different between two columns.

For example, if you have an inner join matching territory numbers between the table dbo.Managers and dbo.Clients, running the query returns all Managers and Clients with matching territory numbers:

Query Builder displays the following results from this query with an inner join

Note: Query Builder displays results of columns in the order of selection. You can reorder columns by deselecting and selecting in the SQL Diagram Pane, the Selected Columns Properties dialog box, or the SQL Statement Tree.

For more information, see**Joins**.

Left Outer Joins

Left outer joins bring back a different data set than Inner Joins. Left outer joins retrieve all the data in columns selected from the primary table, and only matching data from the joined or secondary table.

For example, in the same pair of tables, a left inner join from dbo.Managers to dbo.Clients, where the columns Current Territory and Territory are joined, displays different results.

Note: There is one additional manager who does not have a client, but because a left outer join includes all data from selected columns in the primary table, the last entry in the illustration is displayed.

For more information, see**Joins**.

Right Outer Joins

Right outer joins return opposite results from Left Outer Joins. In a right outer join, you are asking for all the information in the secondary table's column, and the join operator's matching information from the primary table.

For example, in the same set of data we used in the left outer join example, a right outer join returns all clients from dbo.Client, and only managers who match territory numbers, in the joined column.

Note: The managers are the same as the first, inner join, but a right outer join returns the additional clients without matching managers.

For more information, see**Joins**.

Self Joins

A self join is a join within a single table. Query Builder lets you return specific information from a single table using a self join.

For example, in our example table, there is a column for the number of clients and another column with the goal client total for a territory.

A self join can ascertain which managers are reaching their quota. Notice that the join relation operator in the example is greater than or equal to, which shows managers exceeding quota as well.

For more information, see**Joins**.

Adding and Deleting a Join in the SQL Diagram Pane

Query Builder lets you add and delete joins. This method adds a WHERE clause in your query. You can join different tables and or views in a SELECT or CREATE VIEW statement.

Adding a Join

To add a Join, do the following:

1 In the SQL Diagram Pane, drag the target column to the second column.

Query Builder displays both a line joining the two columns in the SQL Diagram Pane and the corresponding SQL code in the SQL Statement Pane.

Removing a Join

Query Builder lets you remove joins from your query. Query Builder automatically deletes joins from the query in the SQL Statement Pane, when you remove them from the SQL Diagram Pane.

To remove a join, do the following:

1 Click the target join, and then on the Query Builder tool bar, click Delete.

OR

Right-click the target join, and then click Delete.

Query Builder deletes the Join.

For more information, see**Joins**.

Editing Joins

Query Builder lets you edit joins with the Join editor. You can edit join parameters in a SELECT, UPDATE, DELETE, and CREATE VIEW Statement.

The table below describes the options in the Join dialog box:

Option	Description
From Table Column	The primary column in the join.
To Table Column	The secondary column in the join.
Select the join relation operator	Click the target join operator. If it is not equals, the operator displays on the join in the SQL Diagram Pane.
Join Type: Inner	Click to make the join an inner join. Aggregates are only available for inner joins.
Join Type: Left Outer	Click to make the join a left outer join.
Join Type: Right Outer	Click to make the join a right outer join.

Completing the Join Dialog Box

1 In the SQL Diagram Pane, right-click the target join, and then click Properties.

OR

In the SQL Diagram Pane, double-click the target join.

OR

On the SQL Statement Tree, expand the Where and And nodes, and then double-click the target join.

Query Builder opens the Join dialog box.

Note: For IBM DB2 for Linux, Unix, and Windows and IBM DB2 for z/OS and OS/390 servers, there is an additional join object in the SQL Statement Tree. The Join On node displays join relations between columns in IBM DB2 for Linux, Unix, and Windows and IBM DB2 for z/OS and OS/390 tables and views.

Changing a Join Color

Query Builder lets you change the color at a join in the SQL Diagram Pane. Complex statements using many tables and multiple joins can be easier to view if joins have different colors.

To change the color of a join, do the following:

1 Right-click the target join, and then click Color.

Query Builder opens the Color dialog box.

2 In the Basic colors grid, click a target color

OR

Click Define Custom Colors, then create a custom color.

Note: Query Builder lets you save custom colors for the current color. Click Add to Custom Color to have the option of using that color for your queries.

3 Click OK.

For more information, see **Joins**.

Auto Layout

The Auto Layout function displays tables and views in the SQL Diagram Pane. It makes the best use of the available area in the SQL Diagram Pane by placing your tables and views in the most efficient manner.

If the automatic join function is on, Query Builder displays all joins between columns in your diagram. Query Builder lets you run the automatic layout function any time you have tables or views in the SQL Diagram Pane. For details, see Auto Joins.

Using Auto Layout

To use Auto Layout, do the following:

1 On the Query Builder menu, click Auto Layout.

Query Builder organizes your tables in the SQL Diagram Pane.

Auto Joins

Query Builder includes an automatic join function that displays joins between selected tables and views in the SQL Diagram Pane. The Auto Join function seeks columns with the same name and data type. You can set global automatic join parameters in the Options Editor.

You can use the Statement Properties Editor to set local join parameters for the current Query Builder session without changing the global parameters.For details, see Statement Properties.

Using Auto Join

To use Auto Join, do the following:

1 On the Query Builder menu, click Auto Join.

Query Builder joins columns in the SQL Diagram Pane.

Creating a Clause Using the SQL Statement Tree

Query Builder lets you build more detailed WHERE, ORDER BY, GROUP BY, and HAVING clauses using the SQL Statement Tree. Query Builder lets you add clauses to SELECT, UPDATE, DELETE, and CREATE VIEW statements.

Note: Query Builder does not support clauses for INSERT statements.

For more information, see the following topics:

- Creating a WHERE Clause
- Deleting a WHERE Clause
- Creating an AND Clause in a WHERE Clause
- Deleting an AND Clause
- Inserting an AND or OR Clause
- Deleting an OR Clause
- Creating an ORDER BY Clause
- Changing the Sort Order in an ORDER BY Clause
- Deleting an ORDER BY Clause
- Creating a GROUP BY Clause
- Deleting a GROUP BY Clause
- Creating a HAVING Clause
- Deleting a HAVING Clause

Creating a WHERE Clause

Query Builder lets you create a WHERE clause from the SQL Statement Tree which automatically displays in your query.

Note: Any additional WHERE clauses are displayed as HAVING clauses.

The table below describes the options and functionality on the Where dialog box.

Option	Description
Operand (Left)	Lets you click the target column for the first part of your WHERE clause. NOTE: Query Builder lists every column in all tables in the SQL Diagram in the Operand lists.
Operator	Lets you select the target operator.
Operand (Right)	Lets you click the target column for the second part of your WHERE clause. Query Builder automatically writes the query language in the Statement option box.

Note: Query Builder does not display clause phrases created from the SQL Statement Tree in the SQL Diagram Pane.

Creating a WHERE Clause

To Create a WHERE clause, do the following:

1 Click the WHERE node, and then on the Query Builder tool bar, click New.

OR

Right-click the WHERE node, and then click New.

For more information, see Selecting Columns in the SQL Statement Tree.

Deleting a WHERE Clause

To delete a WHERE clause, do the following:

1 Expand the AND node, and then on the Query Builder tool bar, click Delete.

OR

Expand the AND node, right-click target column and then click Delete.

Query Builder deletes the target clause and removes it from the SQL Statement Pane.

For more information, see Selecting Columns in the SQL Statement Tree.

Creating an AND Clause in a WHERE Clause

Query Builder lets you add an AND clause from the SQL Statement Tree which automatically displays in your query.

The table below describes the options and functionality on the Where dialog box.

Option	Description
Operand (Left)	Lets you click the target column for the first part of your WHERE clause.
Operator	Lets you select the target operator.
Operand (Right)	Lets you click the target column for the second part of your WHERE clause. Query Builder automatically writes the query language in the Statement option box.
New Button	Click to clear your selections but remain in the Where dialog box. Query Builder adds another AND clause to your query.

To open the Where dialog box, do the following:

1 Click the AND node, and then on the Query Builder tool bar, click New.

OR

Expand the WHERE node, right-click the AND node, and then click New.

For more information, see Selecting Columns in the SQL Statement Tree.

Deleting an AND Clause

To delete an AND clause, do the following:

1 Expand the AND node, click target column, and then on the Query Builder tool bar, click Delete.

OR

Expand the AND node, click target column, and then on the keyboard press DELETE.

OR

Expand the AND node, right-click the target column, and then click Delete.

Query Builder deletes the target clause and removes it from the SQL Statement Pane.

For more information, see Selecting Columns in the SQL Statement Tree.

Inserting an AND or OR Clause

Query Builder lets you insert an AND or an OR WHERE clause from the SQL Statement Tree which automatically displays in your query. Query Builder lets you insert AND or OR clauses at any appropriate point in the SQL Statement Tree.

The table below describes the options and functionality on the Where dialog box.

Option	Description
Operand (Left)	Lets you click the target column for the first part of your WHERE clause.
Operator	Lets you select the target operator.
Operand (Right)	Lets you click the target column for the second part of your WHERE clause. Query Builder automatically writes the query language in the Statement option box.
New Button	Click to clear your selections but remain in the Where dialog box. Query Builder adds another AND clause to your query.

To insert an AND or OR Clause, do the following:

1 On the SQL Statement Tree, expand the WHERE node, right-click the target AND node, then click Insert, and then click And or Or.

For more information, see Selecting Columns in the SQL Statement Tree..

Deleting an OR Clause

To delete an OR clause, do the following:

1 Expand the OR node, and then on the Query Builder tool bar, click Delete.

OR

Expand the OR node, right-click the target column and then click Delete.

Query Builder deletes the target clause and removes it from the SQL Statement Pane.

For more information, see Selecting Columns in the SQL Statement Tree.

Creating an ORDER BY Clause

Query Builder lets you create an ORDER BY clause from the SQL Statement Tree which automatically displays in your query.

The table below describes the Order By Columns dialog box.

Option	Description
Available Columns	Select target column(s) and click the right arrow. Query Builder moves target column from the Available Columns list to the Order By Columns list. NOTE: Query Builder sorts query results based on the order that columns are placed in the ORDER BY clause.
Order	Lets you select the target sort order. ASC - Ascending DESC - Descending Query Builder displays the SQL language in the Order By Statement box.

To open the Order By Columns dialog box, do the following:

1 On the SQL Statement Tree, click the ORDER BY node, and then on the Query Builder tool bar, click Properties.

OR

On the SQL Statement Tree, right-click the ORDER BY node, and then click Properties.

For more information, see Selecting Columns in the SQL Statement Tree.

Changing the Sort Order in an ORDER BY Clause

To quickly change the sort order of a column in a query, do the following:

1 On the SQL Statement Tree, expand the ORDER BY node, and then double-click the target column.

OR

On the SQL Statement Tree, expand the ORDER BY node, then right-click the target column, and then click Properties.

Query Builder opens the Order dialog box.

2 Click the target sort order, and then click OK.

Query Builder appends the Order By clause for target column with the appropriate sort order in the SQL Statement Pane.

For more information, see Selecting Columns in the SQL Statement Tree.

Deleting an ORDER BY Clause

To delete an ORDER BY clause, do the following:

1 Expand the ORDER BY node, and then on the Query Builder tool bar, click Delete.

OR

Expand the ORDER BY node, right-click the target column, and then click Delete.

Query Builder deletes the target clause and removes it from the SQL Statement Pane.

For more information, see Selecting Columns in the SQL Statement Tree.

Creating a GROUP BY Clause

The table below describes the options and functionality on the Group By Columns dialog box.

Option	Description
Selected Columns	Select target column(s) and click the right arrow. Or click the Select All button. Query Builder moves target column from the Selected Columns list to the Group By Columns list. NOTE: Query Builder sorts query results based on the order that columns are placed in the ORDER BY clause.
Clear All Button	Click to move target column from the Group By Columns list to the Selected Columns list. Query Builder displays the SQL language in the Group By Statement window.

Creating a GROUP BY Clause

To create a GROUP BY clause from the SQL Statement Tree which automatically displays in your query, do the following:

1 On the SQL Statement Tree, double-click the GROUP BY node.

OR

On the SQL Statement Tree, right-click the GROUP BY node, and then click New.

Query Builder adds all the selected columns in your table(s) to the GROUP BY node in the SQL Statement Tree, and to the appropriate location in the SQL Statement Pane.

2 On the GROUP BY node, double-click any column.

OR

On the GROUP BY node, click any column, then on the Query Builder menu, click New.

OR

On the GROUP BY node, right-click any column, then click Properties.

For more information, see Selecting Columns in the SQL Statement Tree.

Deleting a GROUP BY Clause

To delete a GROUP BY clause, do the following:

1 On the SQL Statement Tree expand the GROUP BY node, and then on the Query Builder tool bar, click Delete.

OR

On the SQL Statement Tree Expand the GROUP BY node, right-click the target column, and then click Delete.

Query Builder deletes the target clause and removes it from the SQL Statement Pane.

For more information, see Selecting Columns in the SQL Statement Tree.

Creating a HAVING Clause

A HAVING clause is a type of WHERE clause. It filters additional information from your tables. Query Builder lets you create a HAVING clause from the SQL Statement Tree which automatically displays in your query. Query Builder lists every column in all tables in the SQL Diagram in the Operand lists. Query Builder displays the datatype of a column in the operand boxes.

The table below describes the options and functionality on the Having dialog box.

Option	Description
Operand (Left)	Lets you click the target column for the first part of your HAVING clause.
Operator	Lets you select the target operator.
Operand (Right)	Lets you click the target column for the second part of your HAVING clause. Query Builder automatically writes the query language in the Statement option box.
New Button	Click to clear your selections but remain in the Having dialog box. Query Builder adds another AND clause to your query.

Note: Query Builder does not display clause phrases created from the SQL Statement Tree in the SQL Diagram Pane.

To create a HAVING clause, do the following:

1 On the SQL Statement Tree, expand the HAVING node, and then expand the And node. If there is not a join listed on the And node, double-click And. If there is a join listed, use the shortcut option below.

OR

On the SQL Statement Tree, right-click the HAVING node, and then click New.

For more information, see Selecting Columns in the SQL Statement Tree.

Deleting a HAVING Clause

To delete a HAVING clause, do the following:

1 On the SQL Statement Tree expand the HAVING node, and then on the Query Builder tool bar, click Delete.

OR

On the SQL Statement Tree expand the HAVING node, right-click the target column, and then click Delete.

Query Builder deletes the target clause and removes it from the SQL Statement Pane.

For more information, see Selecting Columns in the SQL Statement Tree.

Changing Tables and Columns Location in the SQL Statement Tree

Query Builder lets you move tables and columns on the SQL Statement Tree by dragging them to new locations. You can move columns from the AND and OR nodes to an AND or OR node on the WHERE and HAVING clause nodes. Query Builder changes the query in the SQL Statement Pane to match each move. Query Builder moves tables or columns you are dragging below target table or column.

To move a table or column in the SQL Statement Tree, do the following:

1 Expand target node, then drag the target table or column to a new location.

Query Builder makes the appropriate change in the query in the SQL Statement Pane.

Note: Query Builder lets you select multiple tables or columns.

2 To move a table or column to the bottom of a node, drag it to the target node.

Query Builder displays the target table or column at the bottom of target node.

For more information, see Selecting Columns in the SQL Statement Tree.

Subqueries

Query Builder lets you build subqueries for SELECT and CREATE VIEW statements in the WHERE or HAVING clause. The table below describes the options available for a subquery in Query Builder:

Operand	Location	Description
EXISTS	Left operand	Specifies data that exists in a column.
NOT EXISTS	Left operand	Specifies data that does not exist in a column.
ANY	Right operand	Specifies data satisfying the operator parameters.
ALL	Right operand	Specifies data satisfying the operator parameters.
SELECT	Right operand	Specifies data satisfying the operator parameters.

The table below describes the options and functionality on the Where or Having dialog boxes.

Option	Description
Operand (Left)	Lets you click the target column for the first part of your clause.
Operator	Lets you select the target operator.
Operand (Right)	Lets you click the target column for the second part of your clause. Query Builder displays the working subquery in the Statement window.
Subquery	Paste or type the SUBQUERY statement.

To use the WHERE and HAVING dialog boxes to create subqueries, do the following:

1 On the SQL Statement Tree, expand the Where or Having node, and then expand the And node. If there is not a join listed on the And node, double-click And. If there is a join listed, use the shortcut option below.

OR

On the SQL Statement Tree, right-click the Where or Having node, and then click New.

For more information, see Selecting Columns in the SQL Statement Tree.

Syntax Checker

The Syntax Checker scans SQL statements for errors. You can check your syntax at any time while you are fashioning a query, or a Procedure or Function. Query Builder can automatically run a syntax check to validate your query when you are executing or copying a statement. The options for the Syntax Checker tool are set in the Options Editor. For details,see Query Builder Options.

Note: Query Builder lets you continue with your query even if there are errors detected in the syntax.

Using the Syntax Checker

The table below describes the possible syntax errors the Query Builder Syntax Checker tool displays, in order:

Error	Description
Does the query contain duplicate aliases?	Query Builder returns an error message when it detects duplicate aliases.
If the query has a HAVING clause, is there a GROUP BY clause?	Query Builder returns an error message when it detects a HAVING clause without a GROUP BY clause.
If there are aggregates, or a GROUP BY clause, are all columns in one or the other?	Query Builder returns an error message when it detects an aggregate, or a GROUP BY clause without all columns in one or the other.
Are there joins against non-indexed columns, or columns not participating in a primary key?	Query Builder returns a warning when it detects a join against a non-indexed column, or a column not participating in a primary key.
Are there joins between different datatypes?	Query Builder returns a warning when it detects a join between different datatypes.
Are there cross-products in the query?	Query Builder returns a warning when it detects a cross-product in the query.

Saving and Reopening Queries

You can save and reopen queries in Query Builder. Saving a query saves the SQL Diagram, SQL Statement, and Query Builder Explorer view. Query Builder automatically checks for changes in your database or instance between the time you save the query and the time you reopen it. Query Builder prompts you to save any time you try to close Query Builder, or on application shutdown.

Query Builder runs Schema Change detection any time you set a query to execute, refresh the data, or open a saved query.

You can open multiple saved queries simultaneously.

Saving Queries

To save a query using standard Save and Save As functions, do the following:

1 On the File menu, click Save or Save As.

The Save As dialog box opens.

2 In the File name box, type the name of the query.

Note: By default, the product appends the.qbl extension to Query Builder files. If there is more than one Query Builder session in progress when you save, the file is further appended with an integer, for example.qbl2.

Tip: You can save data in text (*.txt) and XML (*.xml) file formats.

3 Click OK.

The file is saved and the Save As dialog box closes.

Reopening Queries

You can open a query using standard Open functions. Query Builder displays the Query Builder diagram, statement and Query Builder Explorer Pane and it checks the instance or database for schema changes.

The Query Builder Schema Change Detection component checks for:

- Renamed or dropped tables referenced in the query. Renamed tables that have been renamed are considered dropped.
- Renamed or dropped columns referenced in the query. Renamed columns are considered dropped and inserted.
- Columns added or reordered in tables referenced in the query.

If Query Builder detects a change, it opens the Schema Change Detected dialog box. The dialog box displays details of changes to your schema.

Query Builder opens an ISQL Editor with the last saved versions of the SQL statement.

Using Data Editor with Query Builder

SELECT statements. You can open multiple Data Editor sessions so that you can continue to change your data until you find the best match for your query.

Caution: Data Editor is a real-time editor. Changes in your data using Data Editor are permanent.

Opening the Data Editor from Query Builder

To open the Data Editor from Query Builder, do the following:

1 On the Tools menu, click Query Builder.

Query Builder opens.

2 Select a database or instance. For details, see Selecting a Database.

3 Select a table. For details, see Selecting Tables and Views.

4 Select a column, or columns. For details, see Selecting Columns.

5 On the Query Builder menu, click Edit Data.

The Data Editor opens.

For more information on using the Data Editor, seeData Editor

Embarcadero Product Menu Commands

The **Tools** menu lists all installed Embarcadero Technologies products. This lets you toggle to or start another Embarcadero product.

Of the products listed among these menu commands, the most closely integrated with DBArtisan is Embarcadero Team Server. Team Server provides access to datasource definitions, model metadata, and alerts/notifications.streams. For details, see DB Team Server Support.

Available DBMS Utility Menu Commands

The **Tools** menu has commands that let you start installed third party utilities.

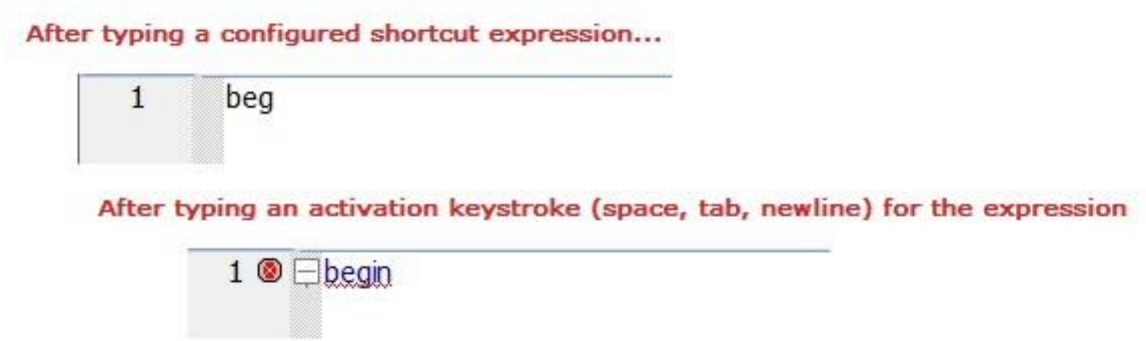
Code Workbench

The Code Workbench reduces the time needed for common day-to-day coding tasks. You can:

- Define auto replacement expressions that can be used to quickly insert commonly used blocks of SQL syntax or commands. For details, see Working with Code Workbench Auto Replace Shortcuts.
- Import and Export Code Workbench settings for client sharing purposes. For details, see Importing and Exporting Settings in Code Workbench.

Working with Code Workbench Auto Replace Shortcuts

The Code Workbench's Auto Replace feature lets you create a set of shortcuts associated with words, terms, and other character strings commonly used when scripting. Subsequently, when using the SQL Editor, you can have the full character string automatically substituted for the shortcut. For example, if you have a **beg** shortcut set up for the word **begin**, if you type **beg** in the the SQL Editor, followed by an activation keystroke (space, tab, or newline), **beg** is replaced by the full word **begin**.



After typing a configured shortcut expression...

```
beg
```

After typing an activation keystroke (space, tab, newline) for the expression

```
begin
```

The following topics describe the steps involved in working with Auto Replace shortcuts:

- Enabling Auto Replace Shortcut Substitution
- Configuring Auto Replace Shortcuts
- Using Code Workbench Auto Replace Shortcuts in the ISQL Window

Enabling Auto Replace Shortcut Substitution

Before configured Auto Replace shortcuts can be automatically substituted with the associated full terms, you must first enable the Auto Replace feature.

To enable Auto Replace

1 Select **Tools > Code Workbench** to open the **Code Workbench** dialog

2 Select **Enable Auto Replace**.

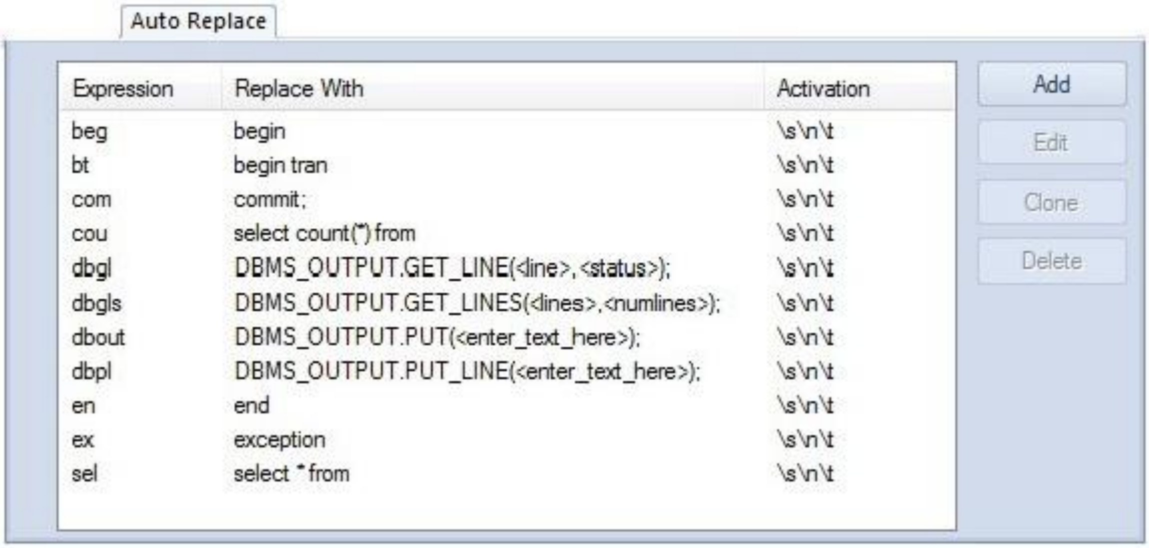
Note: If **Replace substrings** is unselected, only blank-delimited occurrences of auto replace expressions are replaced when an activation event occurs. If selected, all occurrences of auto replace expressions are replaced. For example, consider an Auto Replace expression of **BEG** that is to be replaced with **Begin**. With **Replace substrings** enabled, **BEGBEG** would be replaced with **BeginBegin** when an activation event occurs. With **Replace substrings** disabled, no replacement would occur.

Also see the following topics:

- Working with Code Workbench Auto Replace Shortcuts, for an introduction
- Configuring Auto Replace Shortcuts, for details on setting up shortcuts
- Using Code Workbench Auto Replace Shortcuts in the ISQL Window, for details on invoking Auto Replace.

Configuring Auto Replace Shortcuts

The **Code Workbench** window's **Auto Replace** tab lets you maintain your list of Auto Replace shortcuts.



Expression	Replace With	Activation
beg	begin	\s\n\t
bt	begin tran	\s\n\t
com	commit;	\s\n\t
cou	select count(*) from	\s\n\t
dbgl	DBMS_OUTPUT.GET_LINE(<line>,<status>);	\s\n\t
dbgls	DBMS_OUTPUT.GET_LINES(<lines>,<numlines>);	\s\n\t
dbout	DBMS_OUTPUT.PUT(<enter_text_here>);	\s\n\t
dbpl	DBMS_OUTPUT.PUT_LINE(<enter_text_here>);	\s\n\t
en	end	\s\n\t
ex	exception	\s\n\t
sel	select * from	\s\n\t

Each entry in the list consists of the following:

- The shortcut
- The full term that will be substituted for the shortcut
- The keystroke events that trigger substitution of the term for the short cut.

To open the Code Workbench and navigate to the Auto Replace tab

1 Select **Tools > Code Workbench**.

To add a shortcut to the list

1 On the **Auto Replace** tab, click **Add**.

2 In the **Expression** box, type a new expression.

3 In the **Activation** box, press one or more of SPACE, TAB, or NEWLINE to set those as activation keys for this shortcut.

4 In the **Replace With** box, type the replacement string.

5 Click **OK**.

The replace expression is now ready for use in the ISQL Window.

To delete a shortcut from the list

1 Select one or more shortcuts from the list, click **Delete**, and confirm when prompted.

To edit an existing shortcut

1 Select a shortcut from the list, click **Edit**, and modify the **Expression**, **Activation**, or **Replace With** boxes, as necessary.

Also see the following topics:

- Working with Code Workbench Auto Replace Shortcuts, for an introduction
- Enabling Auto Replace Shortcut Substitution, for details on enabling the feature

Using Code Workbench Auto Replace Shortcuts in the ISQL Window

When Auto Replace is enabled and shortcuts are configured, you can type a replace shortcut in the SQL Editor instead of typing the associated expression. Each time you type one of the activation keys, all shortcuts associated with the activation key are replaced with the defined replacement expression.

Also see the following topics:

- Working with Code Workbench Auto Replace Shortcuts, for an introduction
- Enabling Auto Replace Shortcut Substitution, for details on enabling the feature
- Configuring Auto Replace Shortcuts, for details on setting up shortcuts

Importing and Exporting Settings in Code Workbench

This feature helps standardize your application settings. Code Workbench settings can be exported to an **.xml** file, sent to another user, and subsequently imported by that user.

Exporting Settings

1 Select **Tools > Code Workbench**.

2 Click **Export Settings**. Code Workbench opens a **Save as** dialog box.

3 Use the dialog to specify a location and file name, and then click **Save**. Code Workbench saves your settings as an **.xml** file.

Importing Settings

1 Select **Tools > Code Workbench**.

2 Click **Import Settings**. Code Workbench opens an **Open** dialog box.

3 Use the dialog to locate and select the **.xml** file, and then click **Open**.

Code Workbench imports the settings.

For more general information, see Code Workbench.

Customize (Tools menu)

The **Customize** dialog lets you specify user interface preferences and set up shortcuts.

For detailed DBArtisan information, see Customizing the DBArtisan Display.

Options (Tools menu)

The **Tools** menu provides access to the Options Editor, used to configure features and set preferences. For details on using the editor and available options, see Specifying Application Preferences and Feature Options.

Data Editor

The Edit Data function opens the Data Editor. You can use the Data Editor to edit your tables in real-time. The Data Editor supports all editable datatypes and is an alternative way to add, edit, or delete data from your tables.

Note: You can use Data Editor within Query Builder to edit data in tables while you create SELECT statements. You can open multiple Data Editor sessions so that you can continue to change your data until you find the best match query. For details, see Query Builder.

The Data Editor includes a Data Editor Filter that lets you select the columns in your table that you want to edit. You must select at least one column to use the Data Editor. The Data Editor Filter is not available for the Query Builder. For more information, see Data Editor Filter.

Tip: You can customize Data Editor options in the Options editor. For detals, see Data Editor Options.

For more information, see:

- Data Editor Design
- Using Data Editor

Data Editor Design

The Data Editor includes the following components:

- Data Editor Edit Window
- Data Editor ISQL Window
- Data Editor Toolbar
- Notes on XML Types and Unicode Display in the Data Editor
- Data Editor Filter
- Editing Date and Time Functions
- Using Data Editor

Data Editor Edit Window

Data Editor displays all the information in the target table in the Data Editor Edit Window. You can edit data directly in this window.

For more information, see:

Data Editor ISQL Window

Data Editor Toolbar

Data Editor FilterEditing Date and Time Functions

Using Data Editor

Data Editor ISQL Window

The Data Editor ISQL Window displays the active SQL statement, which uses the data from the target table.

When appropriate, Data Editor displays a History Tab. The History Tab displays all SQL Statements created in the current session. If there is an error, Data Editor displays an Error Tab. The Error Tab details any errors in data entry encountered during execution.

For more information, see:

Data Editor Edit Window

Data Editor Toolbar

Data Editor Filter

Editing Date and Time Functions

Using Data Editor

Data Editor Toolbar

The Data Editor tool bar lets you access commonly used features.

The table below describes the function of each Data Editor tool.

Description	Function
Stop Button	Stops loading data to the Data Editor. Data Editor displays rows up to the stopping point.
List of options for the target table	Displays the editing mode for the target table.
Execute SQL button	Executes the current SQL statement for the target table.
Insert Record button	Inserts new record for the target table. New records display at the end of the table. For related information, see Default Value Handling.
Save Current Row button	Saves data in the current selected row. Data Editor prompts to save when you attempt to leave a row in Live mode. For related information, see Default Value Handling.
Remove Data button	Removes data in target row. Data Editor displays an optional prompt.
Clear SQL Text button	Clears SQL text from the SQL Statement Pane.
Undo button	Undoes the most recent operation.
Redo button	Redoes the most recent operation.
First Record button	Moves to the first record in the target table.
Last Record button	Moves to the final record in the target table.
Filter Data button	Filters table using the target cell as the filter parameter.
Refresh button	Reloads data for target table
Calendar button	Sets correct format for target date/time cell. Enables the Calendar window.
Date/Time Format Builder button	For details, see Editing Date and Time Functions.
Date/Time Format Undo button	Undoes the last date/time format display.
Date/Time Format Redo button	Redoes the last date/time format display.
Close button	Closes and exits Data Editor.

For more information, see:

Data Editor Edit Window

Data Editor ISQL Window

Data Editor Filter

Editing Date and Time Functions

Using Data Editor

Data Editor Filter

The Data Editor Filter displays the columns of a target table and the corresponding SELECT SQL Statement. You can select columns from the filter for selective data editing.

For more information, see:

Data Editor Edit Window

Data Editor ISQL Window

Data Editor Filter

Editing Date and Time Functions

Using Data Editor

Extract Data as XML

Notes on XML Types and Unicode Display in the Data Editor

When working with data in the Data editor, keep the following in mind:

- XML data types are supported for IBM DB2 for Windows, Unix, and Linux, Microsoft SQL Server, and Oracle. In the Data editor, XML data types are displayed and entered as LOB content.
- Support for display of Unicode characters is provided as follows:
- IBM DB2 for Windows, Unix, and Linux: **character**, **clob**, **varchar**, and **longvarchar** types
- SQL Server: **nchar**, **nvarchar**, **ntext**, and **nvarchar(max)** types.
- Oracle 9i, and 10g: **NCHAR**, **NVARCHAR2** and **NCLOB** for non-Unicode UTF8 Character Set Instances and **NCHAR**, **NVARCHAR2**, **CHAR**, **VARCHAR2**, **LONG**, **NCLOB** and **CLOB** for Unicode UTF8 Character Set Instances
- Sybase ASE 12.5 and 15.2: **UNICHAR**,**UNIVARCHAR** and **UNITEXT** for non-Unicode UTF8 Character Set Instances and **UNICHAR**, **UNIVARCHAR**, **UNITEXT**, **NCHAR**, **NVARCHAR**, **CHAR**, **VARCHAR** and **TEXT** for Unicode UTF8 Character Set Instances

Using Data Editor

Data Editor lets you edit data in your tables with any editable datatype without leaving the parent application. Data Editor lets you use your new data immediately.

Caution: Data Editor is a real-time editor. Changes in your data using Data Editor are permanent.

The table below describes the functions and options of the Data Editor:

Option	Description
Live	Edits data one row at a time. You must execute when you leave the row.
Batch	Edits data in multiple rows before executing.

Note: You can also use the Data Editor to edit date and time functions in a table. For details, see Editing Date and Time Functions.

Note: If you make an incorrect edit in a cell, Data Editor displays the error on the Error Tab of the ISQL Editor. Data Editor does not incorporate this error(s) in data into the table. Data Editor saves any changes in data prior to the error message.

Caution: Data Editor is a real-time editor. Changes in your data using Data Editor are permanent.

For more information, see:

Data Editor Edit Window

Data Editor ISQL Window

Data Editor Filter

Editing Date and Time Functions

Using Data Editor

Default Value Handling

Editing Date and Time Functions

The Data Editor lets you edit date and time functions in a table. Data Editor uses a calendar tool to guarantee accurate input for date and time data. You can also change the display of date and time using the Date/Time Format Builder.

For more information, see:

Data Editor Toolbar

Editing Date and Time Functions

Date/time Format Builder

The Date/Time Format Builder lets you customize your date/time display. The Data Editor uses this format to display your dates and times. You control how the Data Editor displays the dates and time by using predefined formats, or by customizing a format to fit your needs.

The Data Editor uses the default date/time format of your Operating System. If you do not make any global changes, the Date/Time Format Builder displays dates and times using the default formats of your operating system.If you make changes to dates and times in the Data Editor, the changes are committed in the format used by the database.

Note: The changes you make using the Date/Time Format Builder do not affect the way your database stores dates and times.

Editing the Date/time Display

You can edit the date/time display on a global, table, or column level. The table below describes the different ways you can edit your date/time format display:

Option	Description	Access
Global	Lets you make global changes to the Data Editor date display from the Data Editor Tab of the Options Editor. For details, see Editing date/time globally.	Options Editor
Grid	Lets you make changes to the date display of the entire Data Editor grid for that session only. For details, see Editing grid date/time.	Data Editor grid
Column	Lets you make changes to the date display of a single column in the Data Editor for that session only. For details, see Editing column date/time.	Data Editor column

Note: Date/Time formats changed on a table or column level are valid for that session only.

Editing date/time globally

You can use the Options Editor to make global changes to your date/time display in the Data editor. When you change the date/time format, using the Options Editor, the Data Editor displays all dates and times in the global manner. To change the date/time display for a particular session, see Editing grid date/time or Editing column date/time.

To edit the date and time globally, do the following:

1 On the File menu, click Options.

The Options Editor opens.

2 On the Options Editor, click the Data Editor Tab.

3 On the Data Editor Tab, click ...

The Date/Time Format Builder dialog box opens.

4 On the Date/Time Format Builder dialog box, click the Date/Time Format list, and then click the target predefined date/time format.

5 To customize the date/time format to your specifications, click Customize.

The Date/Time Format Builder dialog box opens.

6 On the Date/Time Format Builder dialog box, select the appropriate Date/Time Format Options:

Option	Description
Date/Time Format	Displays the predefined Date/Time format.
Day Format	Lets you choose the day display.
Separator	Lets you choose the display separator between the day, month, and year.
Month Format	Lets you choose the month display.
Year Format	Lets you choose the year display.
Date Order	Lets you choose the date order display.
Hour Format	Lets you choose the hour display.
Minute	Lets you choose the minute display.
Sec Format	Lets you choose the second display.
AM/PM	Lets you choose the AM/PM display.
Date/Time Order	Lets you choose the date/time order display.
Format Display	Displays the current format.
Sample	Displays a sample of the current format.

7 When you have finished selecting the Date/Time format options, click OK.

The date/time format changes are accepted the Date/Time Format Builder dialog box closes.

8 On the Options Editor, select the appropriate Default Date/Time Format options:

Option	Description
Use Calendar Control as default	If selected, the **Calendar Control** window is used. For details, see Editing Date and Time Functions.
Two-digit year system setting warning	If selected, a warning is sent when you use a two-digit year system setting.

9 Click OK.

The Default Date/Time Format changes are accepted the Options Editor closes.

Note: To use a different format for a particular session, change the date/time at the session level.

Editing grid date/time

You can change the date/time display for a particular session when working in the Data Editor.

The Data Editor does not maintain the format changes once you close your session. To make this display permanent, use the Editing Global Date/Time Format. For details, see Editing date/time globally.

To edit the grid date and time, do the following:

1 On the Datasource Navigator/Explorer, select the target table.

2 Right-click the table, and then click Edit Data.

The Data Editor opens.

3 On the Data Editor tool bar, click Date/Time Format Builder.

The Date/Time Format Builder opens.

4 On the Date/Time Format Builder, click the Date/Time Format list, and then click the target predefined date/time format.

5 To customize the date/time format to your specifications, click Customize.

The Date/Time Format Builder dialog box opens.

6 On the Date/Time Format Builder dialog box, select the appropriate Date/Time Format Options:

Option	Description
Date/Time Format	Displays the predefined Date/Time format.
Day Format	Lets you choose the day display.
Separator	Lets you choose the display separator between the day, month, and year.
Month Format	Lets you choose the month display.
Year Format	Lets you choose the year display.
Date Order	Lets you choose the date order display.
Hour Format	Lets you choose the hour display.
Minute	Lets you choose the minute display.
Sec Format	Lets you choose the second display.
AM/PM	Lets you choose the AM/PM display.
Date/Time Order	Lets you choose the date/time order display.
Format Display	Displays the current format.
Sample	Displays a sample of the current format.

7 When you have finished selecting the Date/Time format options, click OK.

The date/time format changes are accepted and the Date/Time Format Builder dialog box closes.

- To undo changes, on the Data Editor tool bar, click Undo Change.
- To redo changes, on the Data Editor tool bar, click Redo Change.

Note: Date/Time formats changed on a table level are valid for that session only.

Editing column date/time

You can change the date/time display for a particular column when working in the Data Editor.

The Data Editor does not maintain the format changes once you close your session. To change the format for the entire grid, see Editing Grid Date/Time Format. For details, see Editing grid date/time.

To make this display permanent, Editing Global Date/Time Format. For details, see Editing date/time globally.

To edit the column date and time, do the following:

1 On the Datasource Navigator/Explorer, select the target table.

2 Right-click the table, and click Edit Data.

The Data Editor opens.

3 On the Data Editor, click the column header to select the column.

4 Right-click the column and click Format.

The Date/Time Format Builder opens.

5 On the Date/Time Format Builder dialog box, click the Date/Time Format list, and then click the target predefined date/time format.

6 To customize the date/time format to your specifications, click Customize.

The Date/Time Format Builder dialog box opens.

7 On the Date/Time Format Builder dialog box, select the appropriate Date/Time Format Options:

Option	Description
Date/Time Format	Displays the predefined Date/Time format.
Day Format	Lets you choose the day display.
Separator	Lets you choose the display separator between the day, month, and year.
Month Format	Lets you choose the month display.
Year Format	Lets you choose the year display.
Date Order	Lets you choose the date order display.
Hour Format	Lets you choose the hour display.
Minute	Lets you choose the minute display.
Sec Format	Lets you choose the second display.
AM/PM	Lets you choose the AM/PM display.
Date/Time Order	Lets you choose the date/time order display.
Format Display	Displays the current format.
Sample	Displays a sample of the current format.

8 When you have finished selecting the Date/Time format options, click OK.

The date/time format changes are accepted and the Date/Time Format Builder dialog box closes.

- To undo changes, on the Data Editor tool bar, click Undo Format.
- To redo changes, on the Data Editor tool bar, click Redo Format.

Note: Date/Time formats changed on a column level are valid for that session only.

Default Value Handling

When adding new records, when the transaction committing a row occurs, if no value has been entered for a column defined as having a default value, the default value is used for that column and validation is not required. The default value is not visible in the Data Editor grid until you commit the record and use the Reload Data button to refresh the grid.

In the case of tables with all columns defined as having default values, for example, you can add multiple records by repeatedly clicking the Insert New Record and Save Current Row buttons.

Page setup

The table below describes the options and functionality on the Page Setup dialog box:

Option	Functionality
Margins	Lets you select the size of the left, right, top, and bottom margins.
Titles and Gridlines	Lets you select options.
Preview	Displays how the table will appear when printed.
Page Order	Lets you specify when to print columns and rows.
Center on Page	Lets you select how table floats on the page.

Add-on tools

The following are the available, license-specific tools.

- Embarcadero SQL Debugger - Provides detailed information about using the Embarcadero SQL Debugger. It Includes a step-by-step walk through to familiarize you with the features of the Embarcadero SQL Debugger.
- PL/SQL Profiler - Provides detailed information about using the DBArtisan PL/SQL Profiler. Includes a step-by-step walk through to familiarize you with the features of the PL/SQL Profiler.
- Capacity Analyst (DBArtisan) - Collects critical database storage, object, and performance metrics at scheduled intervals and provides visual analysis and reporting mechanisms that allow for historical trend analysis and forecasting.
- Performance Analyst (DBArtisan) - Provides a robust client-side monitoring product that presents a complete picture of a database's performance.
- Space Analyst (DBArtisan) - Provides sophisticated diagnostic capabilities to help you effectively locate and troubleshoot bottlenecks and performance inefficiencies that result from the challenges of space management.

For information on licensing, see Licensing.

Embarcadero SQL Debugger

The Embarcadero SQL Debugger lets you locate and fix bugs in the following elements:

- Procedures (IBM DB2 for Linux, Unix, and Widows, Microsoft SQL Server, Oracle, Sybase ASE)
- Functions (Oracle)
- Triggers (IBM DB2 for Linux, Unix, and Widows, Microsoft SQL Server, Oracle, Sybase ASE)

Note: The SQL Debugger lets you debug triggers by debugging the procedures that call them.

Note: For Oracle, you cannot debug packages, but you can debug the functions and procedures within packages.

Note: For Oracle, you cannot debug any objects contained in the Exclusion List. For more information, see Editing the Exclusion List.

The table below describes the sections of this chapter:

Section	Description
SQL Debugger Modules and DBMS Support	Describes the specific SQL Debugger modules and DBMS versions supported.
Embarcadero SQL Debugger Features	Provides an overview of SQL Debugger functionality.
Setting up the Embarcadero SQL Debugger	Provides details on requirements and setup tasks for each supported DBMS.
Embarcadero SQL Debugger Interface	Describes the Embarcadero SQL Debugger graphical interface that includes an editor window and four debug view windows.
Embarcadero SQL Debugger Functionality	This section describes the functionality on the SQL Debugger.
Using the Embarcadero SQL Debugger	This section describes how to run a debug session.
Tutorial Sessions	Provides walkthrough sessions for new users.

SQL Debugger Modules and DBMS Support

There are four Embarcadero SQL Debugger modules, each corresponding to a supported DBMS. Modules and DBMS support are as follows:

- **Embarcadero SQL Debugger for IBM DB2 for Linux, Unix, and Windows** - supports all versions supported by DBArtisan
- **Embarcadero SQL Debugger for Microsoft** - supports all versions supported by DBArtisan
- **Embarcadero SQL Debugger for Oracle** - supports all versions supported by DBArtisan
- **Embarcadero SQL Debugger for Sybase ASE** - supports all versions supported by DBArtisan

Each SQL Debugger version is an optional and separate add-on module.

Embarcadero SQL Debugger Features

The Embarcadero SQL Debugger lets you identify problems within your code. The Embarcadero SQL Debugger lets you:

- Interactively step through the flow of script execution.
- Examine the value of variables.
- Solve logical problems with your script design.

Note: The Debugger is available on the main menu, the Procedures window, the DDL Editor and ISQL windows.

The Embarcadero SQL Debugger offers fundamental debugging features and options to fine tune debugging. The table below describes these features:

Debugging Feature	Description
Step Into	Lets you execute each instruction step-by-step and step inside a stored object.
Step Out	Lets you stop stepping through the current object and execute the remainder of the script. This option is only active when the pointer indicates a child dependent instruction.
Step Over	Lets you execute the current instruction without stepping into any child dependents.
Insert or Remove Breakpoint	Lets you specify positions in a program where the debugger stops execution.

To set specific Debugger values on the Options Editor, see Embarcadero SQL Debugger Options.

Setting up the Embarcadero SQL Debugger

The following topics describe requirements and tasks for those DBMS that have setup requirements:

- Embarcadero SQL Debugger for Linux, Unix, and Windows Setup
- The Embarcadero SQL Debugger for Microsoft Setup
- Embarcadero SQL Debugger for Oracle Setup

In addition, you should set your debugger preferences at this time. For details, see Embarcadero SQL Debugger Options.

Embarcadero SQL Debugger for Linux, Unix, and Windows Setup

Embarcadero SQL Debugger for IBM DB2 for Linux, Unix, and Windows requires proper configuration of the server and client.

For more information, see:

- Prerequisites for Adding and Compiling Stored Procedures
- Configuring the IBM DB2 for Linux, Unix, and Windows Server for Procedures
- Prerequisites for Debugging Procedures

Prerequisites for Adding and Compiling Stored Procedures

The Embarcadero SQL Debugger for IBM DB2 for Linux, Unix, and Windows requires the following products and components.

Client

- IBM DB2 for Linux, Unix, and Windows
- DB2 Application Development Client
- DB2 Administration Client
- Communications Protocols
- Stored Procedure Builder
- Applications Development Interfaces
- System Bind Files
- DB2 Connect Server Support
- Documentation
- Base DB2 for Windows/Unix Support
- Administration and Configuration Tools

Server

- IBM DB2 for Linux, Unix, and Windows
- DB2 Enterprise Edition
- Communications Protocols
- Stored Procedure Builder
- Applications Development Interfaces

- System Bind Files
- DB2 Connect Server Support
- Documentation
- Base DB2 for Windows/Unix Support
- Administration and Configuration Tools
- Microsoft Visual Studio, Visual C++

Note: The server must have a local directory structure and file C:\program files\sqllib\function\routine\sr_cpath.bat. This file is installed with IBM DB2 and includes the C compiler options needed to compile the procedure on the server. If it is not found, install the IBM DB2 Administration and Configuration Tools option on the server.

Configuring the IBM DB2 for Linux, Unix, and Windows Server for Procedures

You can create procedures on the targeted server.

To create or run any procedure, set up the configuration environment and enable the C compiler options on the server.

To configure your server, do the following:

1 Open a DB2 Command Window, and then type:

DB2set DB2_SQLROUTINE_COMPILER_PATH="C:\program files\sqllib\function\routine\sr_cpath.bat"

DB2 sets the DB2_SQLROUTINE_COMPILER_PATH DB2 registry variable to call the required initialization script for the C compiler on the server.

To enable the C compiler options on your server:

1 Open the file C:\program files\sqllib\function\routine\sr_cpath.bat.

2 Remove the REM (remarks) prefix on the lines that match the version of Visual Studio that is installed on the server. VCV6 = version 6.0 and VCV5 = version 5.0.

Note: Only remove the REM prefix on the lines that apply to your installation of Visual Studio

3 Restart the DB2 services on the server.

Prerequisites for Debugging Procedures

To enable debugging on the server, do the following:

1 Open a DB2 Command window and type:

Db2set DB2ROUTINE_DEBUG=ON

Note: Availability of this feature depends on your licensing. For more information, see Licensing.

The Embarcadero SQL Debugger for Microsoft Setup

Embarcadero SQL Debugger for Microsoft requires proper configuration of Server Requirements and Client Requirements.

Important Note about Microsoft SQL Server 2000 Service Pack 4 (SP4)

When running Windows XP SP2, SQL Server 2000 SP4 is required on both the client and the server. To verify this is the case start MS Query Analyzer and press Help > About. The Version SQL should be 8.00.2039.

Important Notes about Microsoft SQL Server 2000 Service Pack 3 (SP3)

By default, after you install Microsoft SQL Server 2000 Service Pack 3 (SP3), you cannot use the T-SQL Debugger.

You may receive the following error message if you try to use the T-SQL Debugger:

“Server: Msg 514, Level 16, State 1, Procedure sp_sdidebug, Line 1 [Microsoft][ODBC SQL Server Driver][SQL Server]Unable to communicate with debugger on [SQL Server Name] (Error = 0x80070005). Debugging disabled for connection 53.”

Microsoft has disabled T-SQL Debugger for Application using earlier T-SQL Debugger clients for security reasons. To enable T-SQL Debugger for these Applications, a member of the sysadmins server role, such as sa must explicitly enable debugging by running the following code:

Exec sp_sdidebug 'legacy_on'

Note: You must repeat this procedure whenever you restart the server.

Server Requirements

Embarcadero SQL Debugger for Microsoft requires:

- Windows 2005
- Windows 2000
- Microsoft SQL Server version 7.0 or later

Setting Up the Server

There are three parts to setting up the server:

1Installing the Microsoft SQL Debugger Interface Subcomponent

2Configuring the Service

3Configuring the Service

Enabling SQL Debugger for Microsoft on SQL Server SP3

SQL Debugging is disabled by default in SQL Server SP3 and greater. Please refer to Microsoft Support for information regarding enabling the SQL Debugger for Microsoft on SQL Server SP3.

Installing the Microsoft SQL Debugger Interface Subcomponent

The Microsoft server must have the Development Tools, Debugger Interface subcomponent of Microsoft SQL Server. To determine if the Debugger Interface subcomponent is installed, locate the following files in the \Program Files\Common Files\Microsoft Shared\SQL Debugging directory:

- SQLDBREG.exe
- SQLDBG.dll

If these files are not in the \Program Files\Common Files\Microsoft Shared\SQL Debugging directory, install them before running the Embarcadero SQL Debugger for Microsoft.

To install the Debugger Interface subcomponent on the server after the initial installation:

1 Start Microsoft Visual Studio, Enterprise Edition Setup.

OR

Start Microsoft SQL Server Setup.

2 Select Custom Install.

Microsoft SQL Server opens the **Select Components** dialog.

3 In the Components box, select the Development Tools check box.

4 In the Sub-components box, select the Debugger Interface check box.

5 Click **Next**.

Microsoft SQL Server proceeds through the Microsoft SQL Server wizard to install the components.

For more information, see Server Requirements.

Configuring the Service

To configure the service, see the instructions for your server operating system:

- Windows 2000

Windows 2000

1 On the Windows taskbar, click the **Start** button, click **Settings**, and then click **Control Panel**.

2 Double-click **Administrative Tools**, and then click **Services**.

Windows opens the Services explorer.

3 In the right pane of the Services explorer, right click **MSSQLServer**, and then click **Properties**.

Windows opens the **Net Logon Properties** dialog.

4 Click the **Logon** tab.

5 Select the **This Account** option button.

6 In the This Account box, type (or browse to locate) the logon user account (including domain name, if necessary) of the person who will be using the Embarcadero SQL Debugger for Microsoft.

Note: This person needs admin permissions on the server.

7 In the Password and Confirm Password boxes, type the password.

8 Click **Apply**.

9 Click the **General** tab.

10 Click **Start**.

Windows starts the server and applies the changes.

Important Note about Microsoft SQL Server 2000 Service Pack 4 (SP4)

When running Windows XP SP2, SQL Server 2000 SP4 is required on both the client and the server. When running Windows XP SP2, SQL Server 2000 SP4 is required on both the client and the server. To verify this is the case start MS Query Analyzer and press Help > About. The Version SQL should be 8.00.2039.

Important Notes about Microsoft SQL Server 2000 Service Pack 3 (SP3)

By default, after you install Microsoft SQL Server 2000 Service Pack 3 (SP3), you cannot use the T-SQL Debugger.

You may receive the following error message if you try to use the T-SQL Debugger:

“Server: Msg 514, Level 16, State 1, Procedure sp_sdidebug, Line 1 [Microsoft][ODBC SQL Server Driver][SQL Server]Unable to communicate with debugger on [SQL Server Name] (Error = 0x80070005). Debugging disabled for connection 53.”

Microsoft has disabled T-SQL Debugger for Application using earlier T-SQL Debugger clients for security reasons. To enable T-SQL Debugger for these Applications, a member of the sysadmins server role, such as sa must explicitly enable debugging by running the following code:

Exec sp_sdidebug 'legacy_on'

Note: You must repeat this procedure whenever you restart the server.

For more information, see Configuring the Service.

Configuring DCOM on the Server

To configure DCOM on the server, do the following:

1 After the server restarts, on the Windows taskbar, click the **Start** button, and then click **Run**.

2 In the **Open** box, type dcomcnfg.exe.

3 Click **OK**.

Windows opens the **Distributed COM Configuration Properties** dialog.

4 Click the **Default Security** tab.

5 In the Default Access Permissions box, click **Edit Default**.

Windows opens the **Registry Value Permissions** dialog.

6 Click **Add**.

Windows opens the **Add Users and Groups** dialog.

7 In the Names box, select SYSTEM, and then click **Add**.

8 Click the **Type of Access** list, and then click **Allow Access**.

9 To let any user use the Embarcadero SQL Debugger for Microsoft, grant them remote access on the server. To grant remote access, configure their DCOM permissions on the server. In the Names box, click the target users, and then click **Add**.

Note: You can add individual users or groups.

10 Click the **Type of Access** list, and then click **Allow Access**.

11 Click **OK**.

12 Restart the server to apply the changes.

For more information, see Server Requirements.

Client Requirements

There are three categories of client requirements for the Embarcadero SQL Debugger for Microsoft:

- Operating System
- Client Connectivity
- Installing the Microsoft SQL Debugger Interface Subcomponent

Operating System

The client must be running one of the following operating systems:

- Microsoft Windows 95
- Microsoft Windows 98
- Microsoft Windows XP

For more information, see Client Requirements.

Client Connectivity

When running the Debugger on SQL Server Clients

Before you proceed, verify that the file ssdebugps.dll is registered on the client machine. This file is REQUIRED for debugging and Microsoft only installs it on the server machine and not as a part of a client-only install.

Next you must configure DCOM by following these steps:

1 At a command prompt, type dcomcnfg, and then press ENTER.

Component Services opens.

2 In Component Services, expand Component Services, expand Computers, and then expand My Computer.

3 On the toolbar, click the **Configure My Computer** button.

The **My Computer** dialog appears.

4 In the **My Computer** dialog, click the **COM Security** tab.

5 Under Access Permission, click **Edit Limits**.

The **Access Permission** dialog appears.

6 Under Group or user names, click **ANONYMOUS LOGON**.

7 Under **Permissions** for ANONYMOUS LOGON, select the **Remote Access** check box, and then click **OK**.

Finally, you need to go to the Default Properties tab of the MyComputer dialog.

1 Select None in the Default Authentication Level dropdown menu.

2 Select Impersonate in the Default Impersonation Level dropdown menu.

For more detailed information, you can refer to Microsoft's Help: :http://support.microsoft.com/default.aspx?kbid=833977#XSLTH4134121124120121120120

When running Windows XP SP2, SQL Server 2000 SP4 is required on both the client and the server.

For Microsoft SQL the client must have the Client Connectivity component of Microsoft SQL Server.

For Microsoft SQL Server

The client must have the Development Tools, Debugger Interface subcomponent of Microsoft SQL Server. To determine if the Debugger Interface subcomponent is installed, locate the following files in the \Program Files\Common Files\Microsoft Shared\SQL Debugging directory:

- SQLDBREG.exe
- SQLDBG.dll

If these files are not in the \Program Files\Common Files\Microsoft Shared\SQL Debugging directory, install them before running the Embarcadero SQL Debugger for Microsoft.

Installing the Microsoft SQL Debugger Interface on the Client

To install the Debugger Interface subcomponent on the client, do the following:

1 Start the Microsoft SQL Server Setup program.

2 Select Custom Install.

Microsoft SQL Server opens the Select Components dialog.

3 In the Components box, select the Development Tools check box.

4 In the Sub-Components box, select the Debugger Interface check box.

5 Click **Next**.

Microsoft SQL Server proceeds through the Microsoft SQL Server Wizard to install the components.

For more information, see Client Requirements.

Embarcadero SQL Debugger for Oracle Setup

The only task required in setting up the Embarcadero SQL Debugger for Oracle is Editing the Exclusion List.

Editing the Exclusion List

Upon installation, an Exclusion List is set up on your computer which includes packages that the application cannot debug. The Exclusion List is located in the main product directory. You can add or remove packages from this file by editing the Exclusion List.

To Edit the Exclusion List, do the following:

1 Open the Exclusion List, **deborcex.etd**, in a text editor, such as Microsoft Notepad or WordPad.

2 To add a package, enter the name of the package at the end of the list. Use the following format: OWNER.OBJECT_NAME.

Note: Press ENTER after each item on the list.

3 To remove a package from the Exclusion List, delete the package from the list.

Note: Embarcadero SQL Debugger for Oracle does not debug a package function or package procedure listed on the Exclusion List.

4 Save the changes to **deborcex.etd**.

Embarcadero SQL Debugger Options

You can specify debugger options from the **Debug** tab of the Options editor. For details, see Debug Options.

Embarcadero SQL Debugger Interface

The Embarcadero SQL Debugger includes an editor window and four debug view windows. When you open a debug session, the code for the object is extracted into a DDL Editor and opens four debug view windows at the bottom of the screen. The four debug view windows are optional, dockable windows designed to let you debug your script.

Tip: All Embarcadero debuggers display Performance Metrics that let you measure the execution time of each statement in the debug session.

The Embarcadero SQL Debugger includes five windows:

1DDL Editor Window

2Watch Window

3Variables Window

4Call Stack Window

5Dependency Tree Window

Working with SQL Debugger Windows

You can resize, move, dock and float the following windows:

- Watch Window
- Variables Window
- Call Stack Window
- Dependency Tree Window

1 To resize the target window, click its frame and drag it.

2 To move and dock the target window, click its grab bar and drag it.

3 To float the target window, press Shift, then click its grab bar and drag it.

DDL Editor Window

The DDL Editor displays your code in read-only format. When you start debugging, the Embarcadero SQL Debugger extracts your code into a DDL Editor. The DDL Editor uses the default syntax coloring.

For more information, see Embarcadero SQL Debugger Interface.

Note: For Oracle, LOB datatypes and REF CURSOR variables are displayed in the **Results** tab.

Watch Window

The Watch window displays the watch variables for the database object you are debugging. The Watch window also lets you specify variables you want to evaluate or modify while debugging your program.

For example, to check what happens when a variable (x) has a value of 100, you can double-click the variable in the DDL Editor, drag it into the Watch Window, and change the value to 100. When you execute the script, the Debugger uses the value x =100. This window is only visible when the SQL Debugger is active.

Note: Until you step at least once into a script, variables are not defined. Therefore, step at least once before dragging or typing a local variable in the Watch Window.

Note: You can type a fully qualified record variable into the Watch window.

Note: When you exit a debug session and reenter it, the Embarcadero SQL Debugger retains any watch variables or breakpoints you have set.

Opening and Closing the Watch Window

To open and close the Watch Window, do the following:

1 On the **Debug** menu, on the **Debug Views** sub-menu, select or clear **Watch**.

OR

Press ALT+3.

Setting a Watch Variable

To set a Watch Variable, do the following:

1 In the DDL Editor, double-click the target variable and drag it to the Watch window.

Note: Microsoft SQL Server requires that local variables begin with @. Drag the @ to the Watch Window.

2 In the Watch window, change the value of the variable.

3 On the DDL Editor, click **Debug** or **Go**.

The Embarcadero SQL Debugger executes the script using the new variable.

Removing a Watch Variable

To remove a Watch variable, do the following:

1 In the Watch window, click the target variable and press DELETE.

For more information, see Embarcadero SQL Debugger Interface.

Variables Window

The Variables window displays the local variables and their current values during script execution.

Note: You cannot edit the variables in the Variables window.

If the DDL Editor displays an external database object, and that object is a dependent of the object you are debugging, then the Variables Window automatically refreshes and displays the variables for that particular object. The Variables Window is only visible when the Debugger is active.

The Embarcadero SQL Debugger also lets you monitor your variables while debugging.

Opening and Closing the Variables Window

To open and close the Variables Window, do the following:

1 On the **Debug** menu, on the **Debug Views** sub-menu, select or clear **Variable**.

OR

Press ALT+4.

Monitoring Variables

To monitor the values of your variables while debugging, do the following:

1 In the SQL Editor, hold the pointer over the target variable.

A ScreenTip displaying the current value of that variable displays.

For more information, see Embarcadero SQL Debugger Interface.

Call Stack Window

The Call Stack window displays the stack of currently active calls. The Call Stack Window is only visible when the Debugger is active.

Opening and Closing the Call Stack Window

To open and close the Call Stack Window, do the following:

1 On the **Debug** menu, on the **Debug Views** sub-menu, select or clear **Call Stack**.

OR

Press ALT+5.

Using the Call Stack Window

To display a line of code that references the call in the DDL Editor, do the following:

1 In the Call Stack window, double-click the target line.

In the DDL Editor, a green arrow indicates the line of the referenced call.

For more information, see Embarcadero SQL Debugger Interface.

Dependency Tree Window

The Dependency Tree window displays any external database objects the script accesses. These database objects are displayed in a hierarchical tree, with the child objects as database objects accessed by the parent objects. You can use this window to display the code for a dependent database object in the DDL Editor window. This window is only visible when the Debugger is active.

Opening and Closing the Dependency Tree Window

To open and close the Dependency Tree Window, do the following:

1 On the **Debug** menu, on the **Debug Views** sub-menu, select or clear **Dependencies**.

OR

Press ALT+6.

Displaying Dependencies

To display the code for a dependent database object in the DDL Editor window, do the following:

1 In the Dependency Tree window, double-click the target object.

The SQL of the target object is displayed in the DDL Editor window.

For more information, see Embarcadero SQL Debugger Interface.

Embarcadero SQL Debugger Functionality

The Embarcadero SQL Debugger offers the following functionality:

- Input Parameters
- Step Into
- Step Out
- Step Over
- Run To Cursor
- Insert or Remove Breakpoint
- Toggle Breakpoint
- Go
- Stop
- Restart
- Break
- Close

You must be running a debugging session to use these functions. For details, see Opening a Debugging Session.

Input Parameters

Input parameters are set when you first create an object. If the object you want to debug requires input parameters, a dialog box prompts you for the input parameters when you open a debugging session.

The **Procedure Execution** dialog also lets you:

- Save input parameters as *.prm files to preserve specific input parameter configurations.
- Open *.prm files to save the effort of reentering specific input parameters.
- Reset parameters to their default setting.

The following table describes the options available in this dialog box:

Dialog box component	Description
Owner drop-down list	Displays the current procedure's owner
Procedure drop-down list	Displays the name of the current procedure.
Parameter window	Specify the required input parameters in this window. If input parameters are not required for the execution of the target procedure, a message appears in this window, stating that the procedure "has no input parameters. Press execute to run it."
Open button	Click to open an **Open** dialog, from which you can open an existing *.prm file. The saved parameters immediately populate the dialog box upon opening.
Save button	Click to save the values of your input parameters as a *.prm file. You can reopen a saved *.prm file from this dialog box at any time.
Reset button	Click to reset the parameters in the Parameter window to their default values.
Execute or Continue button	Click to execute the procedure once you have entered values for all required parameters in the Parameter window.

Note: You cannot debug a script that requires input parameters until you provide input parameters.

For more information, see Embarcadero SQL Debugger Functionality.

Step Into

Step Into lets you execute the current instruction. If the current instruction makes a call to a stored SQL object, the Embarcadero SQL Debugger steps inside the nested child object.

To use the Step Into facility, do the following:

1 On the **Debug** menu, select **Step Into**.

OR

Press F11.

The Embarcadero SQL Debugger moves the arrow to execute the current instruction.

For more information, see Embarcadero SQL Debugger Functionality.

Step Out

Step Out lets you execute the remainder of the dependent child object and resumes line-by-line, step-debugging in the parent object.

Note: Step Out is only active when the pointer indicates a child dependent instruction.

To use the Step Out facility, do the following:

1 On the **Debug** menu, select **Step Out**.

OR

Press SHIFT+F11.

The Embarcadero SQL Debugger stops stepping through the current object and executes the remainder of the script.

For more information, see Embarcadero SQL Debugger Functionality.

Step Over

Step Over lets you execute the current instruction without stepping into a nested child object if the instruction makes a call to a dependent object.

To use the Step Over, do the following:

1 On the **Debug** menu, select **Step Over**.

OR

Press F10.

The Embarcadero SQL Debugger executes the current instruction.

For more information, see Embarcadero SQL Debugger Functionality.

Run To Cursor

Run to Cursor lets you execute all instructions between the yellow arrow and the cursor.

To use the Run to Cursor facility, do the following:

1 Scroll down from the yellow arrow to the target line.

2 Click the target line.

Embarcadero SQL Debugger places the cursor on the target line.

3 On the **Debug** menu, select **Run to Cursor**.

OR

Press CTRL+F10.

The Embarcadero SQL Debugger executes all instructions between the pointer and the cursor.

For more information, see Embarcadero SQL Debugger Functionality.

Insert or Remove Breakpoint

A breakpoint is a position in a program where a debugger stops execution. When you start debugging, Embarcadero SQL Debugger opens the script in a DDL Editor. A yellow arrow pointer indicates which line the Embarcadero SQL Debugger executes next.

The Embarcadero SQL Debugger executes all lines of code between the yellow arrow and the first breakpoint. If no breakpoints are present, Embarcadero SQL Debugger debugs the entire script.

While debugging you can set one or more breakpoints in the currently executing object or in any object in the program call stack. You can Toggle Breakpoints, temporarily disable or enable breakpoints without having to add or remove breakpoints.

The Embarcadero SQL Debugger displays each enabled breakpoint as a red dot in the left margin of the DDL Editor Window, and each disabled breakpoint as a red circle.

Breakpoints are stored so that when you debug the same script on separate occasions, you can reuse the same breakpoints. You can insert a breakpoint on the line where your cursor is located, and you can remove a breakpoint on the line where your cursor is located.

Note: Script execution stops at the first breakpoint.

To insert and remove breakpoints, do the following:

1 In the DDL Editor window, click the target line of SQL.

2 On the **Debug** menu, select **Breakpoint**.

OR

Press F9.

The Embarcadero SQL Debugger inserts a new breakpoint or removes an existing breakpoint on the target line of code.

For more information, see Embarcadero SQL Debugger Functionality.

Toggle Breakpoint

After inserting a breakpoint, Toggle Breakpoint lets you enable or disable that breakpoint. Embarcadero SQL Debugger displays each enabled breakpoint as a red dot in the left margin of the DDL Editor Window, and each disabled breakpoint as a red circle.

You can toggle any breakpoint in the DDL Editor window. When you exit a debugging session and reenter it, the Embarcadero SQL Debugger retains any breakpoints you set.

To use the Toggle Breakpoint facility, do the following:

1 In the DDL Editor window, click the line of the target breakpoint.

2 On the **Debug** menu, select **Enable/Disable Breakpoint**.

OR

Press CTRL+F9.

The Embarcadero SQL Debugger toggles the breakpoint indicated by the pointer.

For more information, see Insert or Remove Breakpoint.

Go

Go lets you execute all instructions stopping only when it encounters a breakpoint or when the program is complete.

To use the Go facility, do the following:

1 On the **Debug** menu, select **Go**.

OR

Press F5.

The Embarcadero SQL Debugger executes all instructions.

For more information, see Embarcadero SQL Debugger Functionality.

Stop

Stop lets you halt the script execution and terminate the session.

To use the Stop facility, do the following:

1 On the **Debug** menu, select **Stop Debugging**.

OR

Press SHIFT+F5.

The Embarcadero SQL Debugger stops the script execution and terminates the session.

For more information, see Embarcadero SQL Debugger Functionality.

Restart

Restart lets you terminate the current debug session and open a new one. When the new session opens, you are prompted for new input parameters.

To use the Restart facility, do the following:

1 On the **Debug** menu, select **Restart**.

OR

Press CTRL+SHIFT+F5.

The Embarcadero SQL Debugger restarts the debug session.

For more information, see Embarcadero SQL Debugger Functionality.

Break

Break lets you pause the debug session.

To use the Break facility, do the following:

1 On the **Debug** menu, select **Break**.

The Embarcadero SQL Debugger suspends the debug session.

For more information, see Embarcadero SQL Debugger Functionality.

Close

Close lets you close the DDL Editor and the Embarcadero SQL Debugger.

To use the Close facility, do the following:

1 On the DDL Editor toolbar, click **Close**.

OR

In the upper right corner of the window, click **Close**.

OR

In the DDL Editor window, right-click, and then click **Close**.

The Embarcadero SQL Debugger closes the debug session.

For more information, see Embarcadero SQL Debugger Functionality

Using the Embarcadero SQL Debugger

This section offers a general overview of how to use Embarcadero SQL Debugger's full range of debugging functionality.

- Opening a Debugging Session
- Debugging an SQL Script with Embarcadero SQL Debugger

Opening a Debugging Session

When you start a debugging session, the Embarcadero SQL Debugger interface opens. For details, see Embarcadero SQL Debugger Interface.

If the target script requires input parameters, a dialog box prompts you for the necessary input parameters before displaying the target code in the SQL Editor window. When the target script is displayed in the SQL Editor window, you can begin debugging. For details, see Debugging an SQL Script with Embarcadero SQL Debugger

Note: Embarcadero SQL Debugger for IBM DB2 for Linux, Unix, and Windows, Embarcadero SQL Debugger for Sybase, and Embarcadero SQL Debugger for Microsoft only let you debug the SQL script of procedures or triggers.

To debug an object, do the following:

1 In the Navigator/Explorer, click the object node.

The objects are displayed.

- Right-click the target object, and select **Open**.

The Object Editor opens.

OR

- Right-click the target object and select **Extract**.

The DDL Editor opens, displaying the code of the target object.

2 On the toolbar, click **Debug**.

OR

In the DDL Editor window, right-click, and then click **Debug**.

If the script requests input parameters, a dialog box prompts for parameter values. If the script does not require input parameters, the script is displayed in the DDL Editor window for you to begin debugging. For details, see Debugging an SQL Script with Embarcadero SQL Debugger.

Note: You cannot use the Embarcadero SQL Debugger until it has fully initialized.

3 In the dialog box, specify the appropriate parameter values, and then click **OK** or **Continue**.

The script opens in the DDL Editor window.

Note: If the script requires Sybase ASE or Oracle types (tables, records, or Booleans) as input parameters, the Embarcadero SQL Debugger generates an anonymous block.

Debugging an SQL Script with Embarcadero SQL Debugger

After you open a debugging session and enter any required input parameters, you can begin working with your script in the Embarcadero SQL Debugger.

Debugging an SQL Script

To debug a SQL Script, do the following:

1 On the **Debug** menu, click one of the SQL Debugger options (Step Into, Step Over, and so forth) or click **Go**.

Note: You can monitor the progress of your debug session in the Variables window.

2 On the **Debug** menu, select **Breakpoint**.

OR

Press F9.

Note: When you set a breakpoint, the Call Stack window shows what was called before the breakpoint.

Note: You can use the Run To Cursor option to test the lines of code between a breakpoint and your cursor (indicated by the yellow arrow in the DDL Editor).

To check your variables:

1 In the DDL Editor, click a variable in your script and drag it to the Watch window.

2 In the Watch window, change the value of the watch variable, and then click **Go** to run your script and see the results of the new value.

To check a record in stored objects:

1 Drag the record to the Watch window.

2 In the Watch window, change the value of the record, then click **Go** to run your script and see the results of the new value.

To check the dependencies:

1 In the **Dependency Tree** window double-click the target dependent object to extract the code into a new DDL Editor.

2 Step through the script while monitoring the Dependency Tree Window.

3 When you finish debugging the script, click **Close** or **Exit**.

The T-SQL Debugger DDL Editor closes.

Note: When you exit a debug session and reenter it, the Embarcadero SQL Debugger for MSSQL retains any watch variables or breakpoints you have set.

Tutorial Sessions

The following topics provide walkthrough sessions geared to new users of the Embarcadero SQL Debugger:

- Debugging a Sample Script with Embarcadero SQL Debugger for Microsoft
- Debugging a Sample Script with Embarcadero SQL Debugger for Oracle

Debugging a Sample Script with Embarcadero SQL Debugger for Microsoft

This Getting Started section demonstrates basic debugging functionality. You will debug two procedures using the Embarcadero SQL Debugger for Microsoft.

The section topics are designed to familiarize you with basic debugging features and functionality:

- Getting Started with Embarcadero SQL Debugger for Microsoft
- Testing a Procedure
- Starting the Debugging Session
- Breakpoints
- Step Into
- Step Out
- Correcting the Script

Getting Started with Embarcadero SQL Debugger for Microsoft

This part of Debugging the Sample Script explains how to create the following two procedures to be used for debugging:

- check_modulo
- calculate_sum_with_overflow_bug

Note: The procedure calculate_sum_with_overflow_bug intentionally includes a bug which prevents it from executing successfully. You will use the Embarcadero SQL Debugger for Microsoft to identify this bug.

The Getting Started section includes:

- Creating Procedure 1
- Creating Procedure 2
- Confirming the Creation of the Procedures

Creating Procedure 1

Procedure 1, check_modulo, calculates the modulo of any two user-specified numbers. The user passes the numbers into the procedure as input parameters. The procedure returns the result as an output parameter. If the modulo equals zero, procedure execution returns the output “YES”. If the modulo is not zero, procedure execution returns the output “NO”. This procedure is nested in the second procedure, calculate_sum_with_overflow_bug.

To create this procedure, connect to a MSSQL datasource, open a new SQL editor and, in the SQL editor, type or copy and paste the following code:

```
CREATE PROCEDURE username.check_modulo
@p_dividend_in INT,
@p_divisor_in INT,
@result VARCHAR(3)OUTPUT
AS
IF @p_dividend_in % @p_divisor_in = 0
SELECT @result = 'YES'
ELSE
SELECT @result = 'NO'
go
```

Note: For the purposes of this walk-through, this procedure was created under the user name Spence. Before executing the DDL above, substitute your user name for the word “username”.

1 Connect to a Microsoft SQL Server datasource.

2 On the **Datasource** menu, select the database node, and then select the target database.

Note: For this walk-through, we recommend that you select a non-production database.

3 On the Main toolbar, click **New**.

OR

Press CTRL+N.

An SQL Editor opens in the current workspace.

4 In the SQL Editor, type the DDL for procedure check_modulo.

Note: Substitute your user name once in the DDL for this procedure.

5 On the SQL Editor toolbar, click **Execute**.

The script executes and creates Procedure 1, then opens the **SQL Editor Results** tab with the results of the script execution. If you were not able to create the procedure, check the error messages to determine the problem.

Creating Procedure 2

Procedure 2, calculate_sum_with_overflow_bug, requires two user-specified numbers as input parameters. Upon execution, the procedure calculates the sum of the all numbers divisible by five between the two user-specified numbers. This procedure calls sample procedure 1 (check_modulo) to calculate the modulo of the user-specified numbers.

Note: The procedure calculate_sum_with_overflow_bug intentionally includes a bug which prevents it from executing successfully. You will use the Embarcadero SQL Debugger for Microsoft to identify this bug.

Caution: When you input parameters, enter the smaller number in the @p_num1_in int box.

To create this procedure, connect to a MSSQL datasource, open a new SQL editor and, in the SQL editor, type or copy and paste the following code:

```
CREATE PROCEDURE username.calculate_sum_with_overflow_bug
@p_num1_in INT,
@p_num2_in INT,
@result TINYINT OUTPUT
/*INT-Integer (whole number) data from -2^31 (-2,147,483,648)
through 2^31 - 1 (2,147,483,647).
TINYINT-Integer data from 0 through 255.*/
AS
DECLARE @temp INT
DECLARE @temp_1 INT
DECLARE @v_divisor INT
DECLARE @v_condition VARCHAR(3)
SET @temp = @p_num1_in
SET @temp_1 = 0
SET @v_divisor = 5
SET @v_condition = 'NO'
WHILE 1=1
BEGIN
SELECT @temp = @temp + 1 /*Increase temp starting from p_num1*/
IF @temp = @p_num2_in /*Check if we reached p_num2*/
/*If yes, leave the LOOP*/
BREAK
/*Call Procedure 2 to check if number is divisable by 5*/
EXEC username.check_modulo @temp,@v_divisor,@result=@v_condition output
IF @v_condition = 'YES'
```

```
SELECT @temp_1 = @temp_1 + @temp

END /*WHILE LOOP*/

SELECT @result = @temp_1

RETURN

go
```

Note: For the purposes of this walk-through, this procedure was created under the user name Spence. Before executing the DDL above, substitute your user name for the word "username".

Creating Procedure 2

1 Connect to a Microsoft SQL Server datasource.

2 On the **Datasource** menu, select the database node, and then select the target database.

Note: For this walk-through, we recommend that you select a non-production database.

3 On the Main toolbar, click **New**.

OR

Press CTRL+N.

An SQL Editor opens in the current workspace.

4 In the SQL Editor, type the DDL for procedure calculate_sum_with_overflow_bug.

Note: Substitute your user name twice in the DDL for this procedure.

5 On the SQL Editor toolbar, click **Execute**.

The script executes and creates Procedure 2, then opens the **SQL Editor Results** tab with the results of the script execution. If you were not able to create the procedure, check the error messages to determine the problem.

Confirming the Creation of the Procedures

After you create Procedure 1 and Procedure 2, you can confirm their creation in the Datasource Navigator/Explorer.

Confirming the Creation of the Procedures

To confirm the creation of the procedures, do the following:

1 On the **Explorer** tab, click the Explorer list, and then click **Organize By Owner**.

The **Explorer** tab refreshes with the new display configuration.

2 On the **Explorer** tab, double-click the **Databases** node, and then double-click the target database node.

Tthe list of object owners is displayed.

3 Double-click your user name to display a list of your objects.

4 Double-click **Procedures** to display a list of procedures and confirm the creation of check_modulo and calculate_sum_with_overflow_bug.

Testing a Procedure

After you confirm the creation of the procedures, execute the procedure calculate_sum_with_overflow_bug (which includes a bug) to view its error message. This procedure requires two integer input parameters: @p_num1_in int and @p_num2_in int. For all integers between these two integers, this procedure identifies those divisible by 5, and then returns their sum.

Caution: When inputting parameters, enter the smaller number in the @p_num1_in int box.

Testing a Procedure

To test a procedure, do the following:

1 On the **Explorer** tab, right-click **calculate_sum_with_overflow_bug**, and then click **Execute**.

The Procedure Execution window opens.

2 In the Value column of the @p_num1_in row, type 1.

3 In the Value column of the @p_num2_in row, type 11.

4 Click **Execute**.

The procedure is compiled a **Results** tab opens, displaying the sum 15. There are two numbers between 1 and 11 that are divisible by 5: 5, and 10. The sum of these two numbers is 15.

5 On the **Explorer** tab, right-click **calculate_sum_with_overflow_bug**, and then click **Execute**.

The Procedure Execution window opens.

6 In the Value column of the @p_num1_in row, type 100.

7 In the Value column of the @p_num2_in row, type 121.

8 On the Procedure Execution window toolbar, click **Execute**.

The error returned states “Arithmetic overflow occurred”.

Proceed to Starting the Debugging Session.

Starting the Debugging Session

After you test the procedure, open the procedure in Embarcadero SQL Debugger for Microsoft and enter input parameters before debugging.

To start the debugging session, do the following:

1 On the **Explorer** tab, right-click the procedure, **calculate_sum_with_overflow_bug**, and then click **Debug** to start the debug session.

The DDL for the procedure is extracted into a DDL Editor and opens the **Procedure Execution** dialog.

2 In the Value column of the @p_num1_in row, type 100.

3 In the Value column of the @p_num2_in row, type 121.

4 Click **Continue**.

The dialog box closes.

The Embarcadero SQL Debugger includes the following five windows:

- DDL Editor Window
- Watch Window
- Variables Window
- Call Stack Window
- Dependency Tree Window

Proceed to Breakpoints.

Breakpoints

After you start the debugging session, insert a breakpoint into the code of the procedure calculate_sum_with_overflow_bug. Then run to the breakpoint. After you run to the breakpoint, Embarcadero SQL Debugger displays a yellow arrow on the red breakpoint icon and populates the Variables Window with values for the following variables:

Variable	Value
@temp	Current number
@p_num2_in	Second input parameter
@p_num1_in	First input parameter
@temp_1	Sum of the numbers, between the input parameters, divisible by 5
@result	Condition of the output parameter
@v_condition	Output parameter
@v_divisor	Divisor

1 In the DDL Editor, scroll to and click the following line:

EXEC username.check_modulo @temp,@v_divisor,@result=@v_condition output

Note: This line is located near the end of the procedure’s code.

2 On the **Debug** menu, select **Breakpoint**.

OR

Press F9.

A breakpoint (indicated by a dot) is inserted next to the number of the target line.

3 On the **Debug** menu, select **Go**.

OR

Press F5.

Embarcadero SQL Debugger for Microsoft displays the value of the variables before the breakpoint in the Variables Window.

Proceed to Step Into.

Step Into

After setting the breakpoint, step into the dependent procedure, check_modulo.

To use the Step Into facility, do the following:

1 On the **Debug** menu, select **Step Into**.

OR

Press F11.

The DDL for the dependent, nested procedure is extracted into the DDL Editor.

2 Step Into again.

The next part of the code executes and displays the values for the variables in the Variables Window.

The Call Stack Window displays calls to the procedures.

Proceed to Step Out.

Step Out

After you Step Into the modulo_check (nested procedure) code, step back out and return to the calculate_sum_with_overflow_bug (outside procedure) code.

To use the Step Out facility, do the following:

1 On the **Debug** menu, select **Step Out**.

OR

Press SHIFT+F11.

The DDL Editor opens, containing the code for calculate_sum_with_overflow_bug.

2 On the **Debug** menu, select **Go**.

OR

Press F5.

When the value of the variable, @temp is equal to the value of the variable, @p_num2_in, the WHILE LOOP is complete and the Embarcadero SQL Debugger for Microsoft continues to the next executable statement in the code.

3 While monitoring the value of the variables in the Variables Window, continue to click **Go** to cycle through the WHILE LOOP.

After executing the SELECT and RETURN statements, the Debugger closes a DDL Editor opens to the **Results** tab.

Proceed to Correcting the Script.

Correcting the Script

When you finished Stepping Out of the nested code and encounter the error, to fully fix the bug, do the following:

1 Locate the source of the error

2 Scroll to the line in the script displaying the error

3 Analyze the code

4 Correct the error

5 Compile the corrected script

When you first executed the procedure, the error message "Arithmetic overflow error for data type tinyint, value = 450" was displayed. According to *Microsoft SQL Server Books Online:* "This error occurs when an attempt is made to convert a float or real data type value into a data type that cannot store the result. This error prevents the operation from being completed."

The data type used in this procedure (TINYINT) stores values from 0 to 255. The sum of the four numbers between 100 and 121 that are divisible by 5 (105, 110, 115, and 120) is 450. But because the TINYINT variable @result can only accept a maximum value of 255, the error message was returned and the procedure fails.

To correct the script, do the following:

1 On the **Explorer** tab, right-click **calculate_sum_with_overflow_bug**, and then click **Extract**.

Tthe DDL for the procedure is extracted into a DDL Editor.

2 On the Edit toolbar, click **Find**.

The **Find** dialog opens.

3 In the **Find What** box, type **TINYINT**.

4 Click **Find Next**.

The first occurrence of TINYINT is selected.

5 Change the data type for @result from TINYINT to INT.

6 On the DDL Editor toolbar, click **Execute** to execute the modified script.

The script executes and the **Results** tab opens.

7 On the **Explorer** tab, right-click **calculate_sum_with_overflow_bug**, and then click **Execute**.

The **Procedure Execution** dialog opens.

8 In the Value column of the @p_num1_in row, type 100.

9 In the Value column of the @p_num2_in row, type 121.

10 Click **Execute**.

The procedure executes with the new data type and opens the **Results** tab, returning the value 450. You successfully corrected the script and debugged the procedure.

For more information, see Getting Started with Embarcadero SQL Debugger for Microsoft.

Debugging a Sample Script with Embarcadero SQL Debugger for Oracle

The installation includes a sample script intended to walk you through basic debugging functionality. The sample script creates a package that includes functions and procedures that you debug.

Note: To create the sample package, you must have CREATE privileges.

Overview

Debugging a Sample Script is divided into three sections that familiarize you with basic debugging features and functionality:

- Getting Started with Embarcadero SQL Debugger for Oracle guides you through creating the package you use in Debugging Sample Script 1 and Debugging Sample Script 2.
- Debugging Sample Script 1 guides you through debugging functionality and demonstrates the Embarcadero SQL Debugger for Oracle interface features.
- Debugging Sample Script 2 guides you through debugging functionality and error correction.

Note: For the purposes of this walk-though we have created this package under the user name DEMO_SPENCE.

Getting Started with Embarcadero SQL Debugger for Oracle

The installation includes a sample script that you execute to create a package containing functions and procedures. These functions and procedures demonstrate basic debugging features available in the Embarcadero SQL Debugger for Oracle.

Note: To create the sample package, you must have CREATE privileges.

On installation the script is placed in the \UsrScrpt subfolder of the main installation directory.

If you create the package included with the installation, you can delete it and its objects from your system when you finish working with them. The objects to delete:

- The package COUNT_TIME_INTERVAL
- The package function WEEKEND_DAYS_( )
- The package function WORKING_DAYS_( )
- The package function YEARS_ELAPSED_BETWEEN_( )
- The procedure YEARS_ELAPSED
- The procedure YEARS_ELAPSED_Y2K

Proceed to Embarcadero SQL Debugger for Oracle Overview.

Embarcadero SQL Debugger for Oracle Overview

The Getting Started section guides you through:

- Opening the sample debug script.
- Executing sample debug script.
- Changing the **Explorer** tab display.
- Confirming the creation of the package, including its functions and procedures.

Getting Started

1 On the **File** menu, select **Open**.

The **Open File(s)** dialog opens.

2 In the **Open File(s)** dialog, go to DBA900\UsrScrpt\DEBUGGER_DEMO.sql, and then click **Open**.

Note: During the installation the DEBUGGER_DEMO.sql is placed in the \UsrScrpt subfolder of the main installation folder.

The **What type of file** dialog opens.

3 On the **What type of file** dialog, click **The file includes the DDL to create a database object**, and then click **OK**.

The target script opens in an SQL Editor.

4 On the SQL Editor toolbar, click **Execute** to execute the script and create the package.

The target script executes and opens the SQL Editor **Results** tab, displaying the results of the script execution. If you were not able to create the package, check the error messages to determine the problem.

5 On the **Explorer** tab list, click **Organize by Owner**.

A list of owners is displayed.

6 On the Explorer, double-click your owner name.

A list of your schema objects is displayed.

7 Under your owner node, double-click the **Packages** node.

A COUNT_TIME_INTERVAL displayed, confirming the package’s creation.

Proceed to Debugging Sample Script 1.

Debugging Sample Script 1

Sample Script 1 demonstrates Embarcadero SQL Debugger's basic features and functionality with the function WORKING_DAYS( ), which counts the number of business days between two dates.

Debugging Sample Script 1 is divided into five parts:

- Sample Script 1 - Starting the Debug Session
- Sample Script 1 - Entering Input Parameters
- Sample Script 1 - Viewing Debug Session Results

Sample Script 1 - Starting the Debug Session

After you open and execute DEBUGGER_DEMO.sql, you can begin debugging Sample Script 1. To begin debugging the function WORKING_DAYS( ), start a debug session.

Starting the Debug Session

To start the debug session, do the following:

1 On the **Explorer** tab, under the **Packages** node, double-click the COUNT_TIME_INTERVAL node.

The COUNT_TIME_INTERVAL node opens and displays the following items:

2 Under the COUNT_TIME_INTERVAL node, double-click **Functions**.

The Functions node opens and displays the following items:

3 Under the **Functions** node, right-click WORKING_DAYS ( ), and then click **Debug** to start the debug session.

The **Function Execution** dialog opens with the current date in the boxes.

Proceed to Sample Script 1 - Entering Input Parameters.

Sample Script 1 - Entering Input Parameters

After you start a debugging session, you can enter input parameters. You cannot debug a script that requires input parameters until you input those parameters in the **Function Execution** dialog.

Input Parameters

To enter input parameters, do the following:

1 Click the **P_START_DATE DATE** box, and then click the drop-down arrow.

A calendar opens.

2 On the calendar, click the left arrow to set the month to **November 1999**.

3 Click **1**.

11/01/1999 is displayed in the Value column of P_START_DATE.

4 Click the **P_END_DATE DATE** box, and then click the drop-down arrow.

A new calendar opens.

5 On the calendar, click the left arrow to set the month to **November 1999**.

6 Click **8**.

11/08/1999 is displayed in the Value column of P_END_DATE.

7 Click **OK**.

The **Function Execution** dialog closes and then opens the following five Embarcadero SQL Debugger for Oracle interface windows:

- DDL Editor Window, which displays the SQL code for the function.
- Watch Window
- Variables Window.
- Call Stack Window
- Dependency Tree Window, which displays the dependent objects.

Proceed to Sample Script 1- Inserting Breakpoints.

Sample Script 1- Inserting Breakpoints

After you input parameters in the **Input Parameters** dialog, you can begin inserting breakpoints. In this example, you insert the breakpoints in the extracted dependent object code. After you extract this code, locate the target breakpoint lines by searching for the text DBMS_OUTPUT.

Breakpoints

To insert breakpoints, do the following:

1 In the **Dependency Tree** window, double-click the **COUNT_TIME_INTERVAL** package body.

The SQL code for the package body opens in the SQL Editor window.

2 On the **Edit** toolbar, click **Find**.

The **Find** dialog opens.

3 On the **Find** dialog, in the **Find What** box, type DBMS_OUTPUT.

4 Click **Find Next**.

In the SQL Editor, the first occurrence of DBMS_OUTPUT, on line 22 is highlighted.

5 On the SQL Editor toolbar, click **Breakpoint**.

A breakpoint is inserted next to the target line number.

6 On the **Find** dialog, click **Find Next**.

Tthe next occurrence of DBMS_OUTPUT is highlighted.

7 Click **Find Next** a third time.

The next occurrence of DBMS_OUTPUT, on line 35, is highlighted.

8 On the **Find** dialog, click **Cancel**.

The **Find** dialog closes.

9 On the **Edit** toolbar, click **Breakpoint** to insert a second breakpoint.

You should now have breakpoints set at lines 22 and 35.

Proceed to Sample Script 1- Stepping Into.

Sample Script 1- Stepping Into

After you insert breakpoints, you can step into the function code.

Step Into

To use the Step Into facility, do the following:

1 On the SQL Editor toolbar, click **Go**.

Embarcadero SQL Debugger for Oracle begins debugging and runs to the first breakpoint, placing the yellow arrow on line 22.

2 On the SQL Editor toolbar, click **Step Into**.

Embarcadero SQL Debugger for Oracle moves the yellow arrow to the next line of the code.

3 Click **Step Into** again to enter the LOOP block.

Embarcadero SQL Debugger for Oracle displays the value of the variables in the Variables window.

4 Click **Step Into** again to start moving through the LOOP block.

In the Variables window, Embarcadero SQL Debugger for Oracle updates the value of variable v_currdate from 01-NOV-1999 to 02-NOV-1999.

5 Click **Step Into** two more times.

In the Variables window, Embarcadero SQL Debugger for Oracle updates the value of v_theday from NULL to Tuesday.

Note: If you continued stepping through the LOOP block, the Embarcadero SQL Debugger for Oracle would continue to update v_currdate and v_theday until v_currdate is greater than p_end_date.

6 On the SQL Editor toolbar, click **Go**.

Embarcadero SQL Debugger runs to the next breakpoint.

7 On the SQL Editor toolbar, click **Go** once more.

Embarcadero SQL concludes the debug session and displays the Debug Session Results box.

Proceed to Sample Script 1 - Viewing Debug Session Results.

Sample Script 1 - Viewing Debug Session Results

After Stepping Into and running to the end of the code, Embarcadero SQL Debugger for Oracle displays a Debug Session Results box containing the following information:

- Variable Output
- DBMS_OUTPUT Results

Note: In this example, the Embarcadero SQL Debugger for Oracle displays a Debug Session Results box because the sample program includes DBMS_OUTPUT.

Debug Session Results

To debug session results, do the following:

1 Click **OK**.

The Debug Session Results box closes and your debug session terminates.

Proceed to Debugging Sample Script 2.

Debugging Sample Script 2

Sample Script 2 demonstrates Embarcadero SQL Debugger for Oracle's functionality when used on a function containing a bug which prevents it from executing successfully. The buggy function, WEEKEND_DAYS( ), requires input parameters and counts the number of weekend days between two dates. In this section, use Embarcadero SQL Debugger for Oracle to identify the bug, and then correct the script so that it can execute successfully.

Debugging Sample Script 2 is divided into six parts:

- Sample Script 2 - Executing the Function
- Sample Script 2 - Starting the Debug Session
- Sample Script 2 - Entering Input Parameters
- Sample Script 2- Inserting Breakpoints
- Sample Script 2- Stepping Into
- Sample Script 2 - Correcting the Script

Sample Script 2 - Executing the Function

After you open and execute DEBUGGER_DEMO.sql, you can begin debugging Sample Script 2. To begin debugging the function WEEKEND_DAYS ( ), first execute the function to discover the type of error it returns when it fails to execute.

Executing the Function

To execute the function, do the following:

1 On the **Explorer** tab, under the **Packages** node, double-click the **COUNT_TIME_INTERVAL** node.

The **COUNT_TIME_INTERVAL** node opens.

2 Double-click the **Functions** node.

The Functions node opens.

3 Click **WEEKEND_DAYS ( )**, then right-click it and click **Execute**.

The **Function Execution** dialog opens.

4 In the **Value** column of the **P_START_DATE** row, type **11/01/1999**.

5 In the **Value** column of the **P_END_DATE** row, type **11/30/1999**.

6 Click **Execute**.

The attempt to execute the function fails and returns an error indicating that the character string buffer is too small.

Proceed to Sample Script 2 - Starting the Debug Session.

Sample Script 2 - Starting the Debug Session

After you unsuccessfully execute the function WEEKEND_DAYS( ) and determine the nature of its execution error, you can start a debugging session to determine the actual cause of the error.

Starting the Debugging Session

To start the debugging session, do the following:

1 On the **Explorer** tab, under the COUNT_TIME_INTERVAL node, under the Functions node, right-click **WEEKEND_DAYS ( )**, and then click **Debug** to start the debug session.

The **Function Execution** dialog opens.

Proceed to Sample Script 2 - Entering Input Parameters.

Sample Script 2 - Entering Input Parameters

After you start the debug session, you can enter input parameters in the **Function Execution** dialog.

Entering Input Parameters

To enter input parameters, do the following:

1 At the end of the **P_START_DATE** row, click the drop-down arrow.

The calendar opens.

2 On the calendar, click Left Arrow to set the month to **November 1999**.

3 Click **1**.

11/01/1999 is displayed in the **Value** column of the **P_START_DATE** row.

4 At the end of the **P_END_DATE** row, click the drop-down arrow.

A new calendar opens.

5 On the calendar, click Left Arrow to set the month to **November 1999**.

6 Click **30**.

11/08/1999 is displayed in the **Value** column of the **P_END_DATE** row.

7 Click **Continue**.

Tthe **Function Execution** dialog gloses and then opens the following five Embarcadero SQL Debugger for Oracle interface windows:

- DDL Editor Window, which displays the SQL code for the function
- Watch Window
- Variables Window
- Call Stack Window
- Dependency Tree Window, which displays the dependent objects.

Proceed to Sample Script 2- Inserting Breakpoints.

Sample Script 2- Inserting Breakpoints

After you enter input parameters, you can begin inserting breakpoints. In this example, you insert the breakpoints in the extracted dependent object code. After you extract this code, locate the target breakpoint lines by searching for a particular line of code.

Breakpoints

To insert breakpoints, do the following:

1 In the **Dependency Tree** window, double-click the **COUNT_TIME_INTERVAL** package body.

Tthe SQL code for the package body opens in the SQL Editor.

2 On the **Edit** toolbar, click **Find**.

The **Find** dialog opens.

3 On the **Find** dialog, in the **Find What** box, type **Function weekend_days**, and then click **Find Next**.

Embarcadero SQL Debugger for Oracle highlights the first occurrence of Function weekend_days.

4 On the **Find** dialog, click **Cancel**.

The **Find** dialog closes.

5 Click line 60, the first line of executable code:

6 On the **SQL Editor** toolbar, click **Breakpoint**.

A breakpoint is inserted next to the line number.

7 Click **Go** to start debugging and run to the breakpoint.

Embarcadero SQL Debugger for Oracle places the yellow arrow on line 60 and populates the Variables window with the first set of variables in the function code.

Embarcadero SQL Debugger for Oracle also populates the Call Stack window with everything called before the breakpoint.

Proceed to Sample Script 2- Stepping Into.

Sample Script 2- Stepping Into

After you set and run to the breakpoint, you can step into the function to locate the cause of the error. To locate the cause of the error, monitor the Variables window. As you step through the code, the Variables window updates with the value of the variables.

Step Into

1 On the SQL Editor toolbar, click **Step Into**.

The yellow arrow moves to the next line of the code, line 64.

2 On the SQL Editor toolbar, click **Step Into**.

Embarcadero SQL Debugger for Oracle's Variables window updates the value of **v_currdate** to **02-NOV-1999**.

3 On the SQL Editor toolbar, click **Step Into**.

The yellow arrow moves to the next line of the code, line 66.

4 On the SQL Editor toolbar, click **Step Into**.

The yellow arrow moves to the next line of the code, line 67, and, in the Variables window, updates the value of **v_theday** to **Tuesday**.

5 On the SQL Editor toolbar, click **Step Into**.

The yellow arrow moves back to line 64 to repeat the loop.

6 On the SQL Editor toolbar, click **Step Into**.

Embarcadero SQL Debugger for Oracle's Variables window updates the value of **v_currdate** to **03-NOV-1999**.

7 On the SQL Editor toolbar, click **Step Into**.

The yellow arrow moves to the next line of the code, line 66.

8 On the SQL Editor toolbar, click **Step Into**.

The Embarcadero SQL Debugger for Oracle locates the error. The application terminates the debug session, returns an error indicating that the numeric or value character string buffer is too small, extracts the COUNT_TIME_INTERVAL package code into an SQL Editor, and returns an error indicating the line on which the code failed.

Proceed to Sample Script 2 - Correcting the Script.

Sample Script 2 - Correcting the Script

After you step through the SQL code and locate the error, you can correct the bug in Sample Script 2. When Embarcadero SQL Debugger for Oracle locates an error, it extracts the target package body into an SQL Editor. To correct this script:

- Scroll to the incorrect line in the script
- Analyze the code
- Correct the error
- Execute the corrected SQL script
- Execute the WEEKEND_DAYS ( ) function

The code in Sample Script 2 fails on line 66, returning an error when the variable **v_theday** increments from the value **Tuesday** to the value **Wednesday**. The cause of this error is found in the declarations section of the function script, where the width of the VARCHAR2 variable **v_theday** is set to **8**. Because **Wednesday** includes nine characters, the value of the variable **v_theday** fails when it attempts to place a nine-character value in an eight-character variable. To correct this error, increase the width of the variable **v_theday** to accommodate nine characters.

Correcting the Script

To correct the script, do the following:

1 On the **Explorer** tab, under the **Packages** node, under the COUNT_TIME_INTERVAL. node, right-click **Package Body**, and then click **Extract**.

The package body is extracted into an SQL Editor.

2 In the SQL Editor, scroll to line 57, the line defining the variable **v_theda**y.

3 On line 57, change the value of the width from **8** to **9**.

4 On the SQL Editor toolbar, click **Execute** to execute the script.

The scrip executes successfully.

5 On the **Explorer** tab, under the **COUNT_TIME_INTERVAL** package node, under the **Functions** node, click **WEEKEND_DAYS ( )**.

6 Right-click **WEEKEND_DAYS ( )**, and then click **Execute.**

The **Function Execution** dialog opens.

7 In the **Value** column of the **P_START_DATE** row, type **11/01/1999**.

8 In the **Value** column of the **P_END_DATE** row, type **11/30/1999**.

9 Click **Execute**.

The corrected function executes successfully.

PL/SQL Profiler

The PL/SQL Profiler module lets Oracle developers capture metrics of various PL/SQL programmable objects as they are executed in the database. Developers can use data collected in profile sessions to improve performance of PL/SQL code execution. The PL/SQL Profiler collects and stores data in database tables that let you identify and isolate performance problems and provide code coverage information. The PL/SQL Profiler lets you:

- Graphically browse PL/SQL profiling data within the Explorer Tab
- View profiling data in the right pane of the application, which is populated as you navigate the Explorer Tab
- Start and stop PL/SQL profiling sessions with a single click
- Graphically analyze time spent in each programmable object (unit)
- Graphically analyze time spent in each source code line of a unit

Requirements:

- Oracle built-in package DBMS_PROFILER
- Oracle tables:
- PLSQL_PROFILER_RUNS
- PLSQL_PROFILER_UNITS
- PLSQL_PROFILER_DATA (user's schema)

The table below describes the sections of this chapter:

Section	Description
Setting up PL/SQL Profiler	Describes the process of setting up the PL/SQL Profiler.
PL/SQL Profiler Functionality	Describes the functionality of the PL/SQL Profiler.
Using PL/SQL Profiler	Describes how to run a profile session.

Note: The PL/SQL Profiler is an optional add-on module.

Topics

- Setting up PL/SQL Profiler
- PL/SQL Profiler Explorer
- PL/SQL Profiler Functionality
- Using PL/SQL Profiler

Setting up PL/SQL Profiler

The Oracle profiling tables must be on the Oracle server before you can use the PL/SQL Profiler. The first time you open the PL/SQL Profiler, the server is checked for the profiling tables. If the profiling tables are not on the server, Oracle SQL*Plus is atarted, which installs profiling tables on the Oracle server.

To run Oracle SQL*Plus (which installs the profiling tables) your Oracle server and client must meet the following conditions:

- The Oracle server and the client have the same version of Oracle.
- The client has the Oracle\BIN directory on the path.
- The client has the Oracle file, SQLPLUS.exe in the Oracle\BIN directory.
- The following Oracle files are in the Oracle\RDBMS\ADMIN directory:
- DBMSPBP.sql
- PROFLOAD.sql
- PROFTAB.sql
- PRVTPBP.blp

Note: If the Oracle server and the client machines are running different versions of Oracle, after running SQL*Plus, the following error message is displayed: Version of package is incompatible.

PL/SQL Profiler Explorer

The PL/SQL Profiler displays profiling data in the right pane of the application, which is populated as you navigate the **Explorer** tab.

The table below describes the nodes of the PL/SQL Profiler Explorer and the corresponding information in the right pane of the application:

Node	Right pane information
PL/SQL Code Profiling	Contain all Comment, Run ID and Run Date\time data that is current stored in the Profiling tables.
Label\Comment level	Contains all Run ID and Run Date\time data for the specific Label\Comment.
Run level	Contains all Unit, Unit Name, Unit Type, Run Date\time data for the specific Run ID.

PL/SQL Profiler Functionality

The PL/SQL Profiler offers the following functionality:

- Start
- Flush
- Run Summary
- Run Detail
- Unit Summary
- Clear Profile Table
- Unit Detail
- Stop (PL/SQL Profiler)

Start

You begin a new profiling session or open a previous profiling session with the Start command.

Starting a New Profile Session

To start a new profile session, do the following:

1 If using DBArtisan, on the **Utilities** menu, select **PL/SQL Profiler > Start**.

The **PL/SQL Profiler - Start** dialog opens.

2 In the **Profile Label** box, type the name of the new profile.

Note: Each user can own one or more Profiles.

3 Click **OK**.

Profiling begins.

Starting an Existing Profile Session

1 If using DBArtisan, on the **Utilities** menu, select **PL/SQL Profiler > Start**.

The **PL/SQL Profiler - Start** dialog opens.

2 Click the **Profile Label** list, and then click the existing profile.

3 Click **OK**.

Profiling begins.

For more information, see PL/SQL Profiler Functionality.

Flush

The PL/SQL Profiler lets you move the data from the dynamic tables into analysis tables with the flush command.

The table below describes the options and functionality on the **PL/SQL Profiler - Flush** dialog:

Option	Description
Flush	Click to delete the data in a running profile.
Flush & Analyze	Click to open the **PL/SQL Profiler Run Detail** window. For details, see Run Detail.
Cancel Button	Click to abort the flush and continue the profiling session.

Note: You can only Flush a running Profile.

Flushing a Profile

To flush a profile, do the following:

1 If using DBArtisan, on the **Utilities** menu, select **PL/SQL Profiler > Flush**.

The **PL/SQL Profiler - Flush** dialog opens.

For more information, see PL/SQL Profiler Functionality.

Run Summary

The PL/SQL Profiler Run Summary window lets you to view the following information for each of your profiles:

- Run ID
- Run Date
- Total Time

Opening the Run Summary Window

To open the Run Summary Window, do the following:

1 If using DBArtisan, on the **Utilities** menu, select **PL/SQL Profiler > Run Summary**.

The **PL/SQL Profiler - Run Summary** window opens.

2 In the **PL/SQL Profiler - Run Summary** window, click the **Label** list, and then click the target profile to populate the table.

For more information, see PL/SQL Profiler Functionality.

Run Detail

The PL/SQL Profiler Run Detail window lets you to view the following information for each of your profiles:

- Run Number
- Run Date
- Run Time

The **Run Detail** tab lets you:

- View the information for all runs or you can view profile information based on the unit type or unit owner.
- View results in milliseconds, seconds and minutes.
- View graphical displays of the profiling data that let you can navigate to the specific unit within the summary portion of the window.
- Specify the number of top lines to display in the graphical portion of the interface.

Tip: Each graph is a working object. You can select data in a graph and the corresponding line of source displays in the lower pane of the interface.

Opening the Run Detail Window

To open the Run Detail Window, do the following:

1 If using DBArtisan, on the **Utilities** menu, select **PL/SQL Profiler > Run Detail**.

The **PL/SQL Profiler - Run Detail** window opens.

2 In the PL/SQL Profiler - Run Detail window:

- Click the **Label** list box, and then click the target profile.
- Click the **Run** list, and then click the target run.
- Click the **Unit Type** list, and then click the target unit type(s).
- Click the **Unit Owner** list, and then click the target unit owner(s) to populate the table.

For more information, see PL/SQL Profiler Functionality.

Unit Summary

The PL/SQL Profiler Unit Summary window lets you to view the following information for each of your profiles:

- Run ID
- Run Date
- Run Time
- Unit Time
- Percentage of Run Time

The PL/SQL Profiler Unit Summary window lets you view results in milliseconds, seconds and minutes. The Unit Summary window also displays graphs of execution statistics for the top N runs and associated units. You can use the graphical displays to navigate to the specific run within summary portion of the window.

Opening the Unit Summary Window

To open the Unit Summary Window, do the following:

1 If using DBArtisan, on the **Utilities** menu, select **PL/SQL Profiler > Unit Summary**.

The **PL/SQL Profiler - Unit Summary** window opens.

2 In the **PL/SQL Profiler - Unit Summary** window:

- Click the **Unit Owner** list, and then click the target unit owner.
- Click the **Unit Name** list, and then click the target unit name to populate the table.

For more information, see PL/SQL Profiler Functionality.

Clear Profile Table

The PL/SQL Profiler lets you delete data from the user's profile tables with the command Clear Profile Table.

Clearing a Profile Table

To clear a Profile Table, do the following:

1 If using DBArtisan, on the **Utilities** menu, select **PL/SQL Profiler > Clear Profile Table**.

The profile table is cleared.

2 If you are sure that you want to clear out the profiler tables, click **Yes**.

For more information, see PL/SQL Profiler Functionality.

Unit Detail

The PL/SQL Profiler Unit Detail window lets you to view the following information for each of your profiles:

- Average Time
- Source
- PL/SQL Script

The PL/SQL Profiler Unit Detail window lets you view results in milliseconds, seconds and minutes. The Unit Detail window also provides two calculation options for viewing unit execution time as a percentage of total execution time (total run vs unit run). Additionally, also displays graphs of execution statistics for the top N run. You can use the graphical displays to navigate to the specific line within source code portion of the window.The graphical display portion of the window contains options for viewing advanced statistics.

The Advanced View of the PL/SQL Profiler Unit Detail window lets you view the following information for each of your profiles:

- Hit Lines
- Missed Lines
- Line Number
- Calls
- Total Time
- Percentage of the Total Time
- Average Time
- Minimum Time
- Maximum Time

Opening the Unit Detail Window

To open the Unit Detail Window, do the following:

1 If using DBArtisan, on the **Utilities** menu, select **PL/SQL Profiler > Unit Detail**.

Tthe **PL/SQL Profiler - Unit Detail** window opens.

2 In the **PL/SQL Profiler - Unit Detail** window, do any of the following:

- Click the **Label** list, and then click the target profile.
- Click the **Run** list, and then click the target run.
- Click the **Unit** list, and then click the target unit to populate the table.

- Right-click, and then click **Show Only Hit Lines** to populate the table with the **Average Time** and **Source** for hit lines.
- Right-click, and then click **Show Only Missed Lines** to populate the table with the **Average Time** and **Source** for missed lines.

Opening the Unit Detail Window Advanced View

To open the **Unit Detail Window Advanced View**, do the following:

1 In the **Unit Detail** window, right-click, and then click **Advanced View** to populate the table with **Advanced View** information.

For more information, see PL/SQL Profiler Functionality.

Stop (PL/SQL Profiler)

The PL/SQL Profiler Stop command pauses the data gathering operation. Stop & Analyze populates the summary tables so that you can view the Unit Detail and Run Summary windows.

The table below describes the options and functionality on the **PL/SQL Profiler - Stop** dialog:

Option	Description
Stop	Click to stop the profiling session.
Stop & Analyze	For details, see Run Detail..
Cancel	Click to continue the profiling session.

Stopping a Profiling Session

To stop a Profiling Session, do the following:

1 If using DBArtisan, on the **Utilities** menu, select **PL/SQL Profiler > Stop**.

The **PL/SQL Profiler - Stop** dialog opens.

For more information, see PL/SQL Profiler Functionality.

Using PL/SQL Profiler

The steps in this section provide a high level overview of running a profiling session, and cover the following processes:

- Starting the Session.
- Executing the Sample Script.
- Stopping and Analyzing the Session.

Note: The first execution of a PL/SQL unit can take more time to execute because the code is loading into memory; subsequent runs take less time.

Using the PL/SQL Profiler

1 If using DBArtisan, on the **Utilities** menu, select **PL/SQL Profiler > Start**.

The **PL/SQL Profiler - Start** dialog opens.

2 In the **Profile Label** box, type the name of the new profile.

Note: Each user can own one or more Profiles.

3 Click **OK**.

Profiling begins.

4 On the Datasource Explorer, execute on one of the following PL/SQL database objects:

- Procedure
- Function
- Package Procedure
- Package Function

Profiler displays profiling data in the right pane of the application.

5 If using DBArtisan, on the **Utilities** menu, select **PL/SQL Profiler > Stop**.

The **PL/SQL Profiler - Stop** dialog opens.

The table below describes the options and functionality on the **PL/SQL Profiler - Stop** dialog:

Option	Description
Stop	Click to stop the profiling session.
Stop & Analyze	Click to open the **PL/SQL Profiler Run Detail** window. For details, see Run Detail. Click the **Label** list, and then click the target profile. Click the **Run** list, and then click the target run. Click the **Unit Type** list, and then click the target unit type(s). Click the **Unit Owner** list, and then click the target unit owner(s) to populate the table.
Cancel	Click to continue the profiling session.

7 If using DBArtisan, use the **Utilities** menu to open any of the following PL/SQL Profiler windows. For more information, see:

- Run Summary
- Unit Summary
- Unit Detail

For more information, see PL/SQL Profiler Functionality.

Topics

- Sample Profiling Session

Sample Profiling Session

The DBArtisan installations include two scripts for the sample profiling session:

- PROFILER_BUILD_DEMO.SQL
- PROFILER_DEMO.SQL

The PROFILER_BUILD_DEMO.SQL creates the objects that you profile in the walk through, and the PROFILER_DEMO.SQL is what you profile during the walk through.

Note: To create the objects in the PROFILER_BUILD_DEMO.SQL script, you need CREATE privileges.

The sample script demonstrates the following features of the PL/SQL Profiler:

- Unit Detail
- Run Detail
- Show Only Hit Lines
- Advanced View

The scripts are located in the \UsrScrpt subfolder of the main installation folder.

Overview

Sample Profiling Session is divided into six parts:

- Getting Started
- Starting the Session
- Executing the Sample Script
- Stopping the Session
- Re-running & Re-executing the Session
- Stopping & Analyzing

Proceed to Sample Profiling Session - Getting Started.

Sample Profiling Session - Getting Started

In this step of Sample Profiling Session, you create the objects that you profile in the walk through.

Overview

The Getting Started section guides you through:

- Opening PROFILER_BUILD_DEMO.SQL.
- Changing the Datasource Explorer Display.

- Confirming the Creation of the Package.

Getting Started

1 On the **File** menu, select **Open**.

The **Open Files** dialog opens.

2 In the **Open Files** dialog, navigate to the UsrScrpt subfolder of the main installation folder, press ENTER, and then double-click **PROFILER_BUILD_DEMO.SQL** to open the script in a SQL Editor window.

The **PROFILER_BUILD_DEMO.SQL** script opens in an SQL Editor window.

3 On the SQL Editor window, click **Execute**.

Tthe script executes and create the package.

4 On the Explorer window list, click **Organize by Owner**.

5 On the Explorer window, click the node of your owner name.

Your schema objects are displayed.

6 Double-click the **Packages** node to display **PF_COUNT_TIME_INTERVAL** and confirm its creation.

Note: If you were not able to create the package, check the error messages to determine the problem.

Proceed to Sample Profiling Session - Starting the Session.

Sample Profiling Session - Starting the Session

In this step of Sample Profiling Session, you start the profiling session.

Sample Profiling Session - Starting the Session

To start the session, do the following:

1 On the **File** menu, select **Open**.

The **Open Files** dialog opens.

2 In the **Open Files** dialog, type the path to the UsrScrpt directory, press ENTER, and then double-click **PROFILER_DEMO.SQL**.

The script opens in a SQL Editor window.

3 If using DBArtisan, on the **Utilities** menu, select **PL/SQL Profiler > Start**.

The **PL/SQL Profiler - Start** dialog opens.

5 In the **Profile Label** list, enter **DemoProfile**.

6 Click **OK**.

The profiling session begins.

Note: If this is the first time you start the PL/SQL Profiler, a dialog box opens.

Note: Click **Yes** to have SQL*Plus create the tables. You need to start the profiling session again (see step 3 above.)

Proceed to Sample Profiling Session - Executing the Sample Script.

Sample Profiling Session - Executing the Sample Script

In this step of Sample Profiling Session, you execute the DEMO script.

Sample Profiling Session - Executing the Sample Script

To execute the sample script, do the following:

1 On the SQL Editor window toolbar, click **Execute**.

The script executes and opens a **Results** tab.

Proceed to Sample Profiling Session - Getting Started.

Sample Profiling Session - Stopping the Session

In this step of Sample Profiling Session, you stop the profiling run.

Sample Profiling Session - Stopping the Session

To stop the session, do the following:

1 If using DBArtisan, on the **Utilities** menu, select **PL/SQL Profiler > Stop**.

The **PL/SQL Profiler - Stop** dialog opens.

2 Click **Stop**.

Proceed to Sample Profiling Session - Re-running & Re-executing the Session.

Sample Profiling Session - Re-running & Re-executing the Session

In this step of Sample Profiling Session, you run the same profile session and execute the DEMO script again.

Sample Profiling Session - Re-running & Re-executing the Session

To re-run and re-execute the session, do the following:

1 In the SQL Editor, click the **Query** tab.

1 If using DBArtisan, on the **Utilities** menu, select **PL/SQL Profiler > Start**.

The **PL/SQL Profiler - Start** dialog opens again.

2 Click the down arrow on the **Profile Label** list, and then click **DemoProfile**.

3 Click **OK**.

The profiling session begins.

4 On the SQL Editor toolbar, click **Execute**.

The script executes again and the **Results** tab opens.

Proceed to Sample Profiling Session - Stopping & Analyzing.

Sample Profiling Session - Stopping & Analyzing

In this step of Sample Profiling Session, you stop profiling and analyze the runs.

Sample Profiling Session - Stopping & Analyzing

To stop and analyze the sample profiling session, do the following:

1 If using DBArtisan, on the **Utilities** menu, select **PL/SQL Profiler > Stop**.

The **PL/SQL Profiler - Stop** dialog opens again.

2 Click **Stop & Analyze**.

The PL/SQL Profiler - Run Detail window opens.

3 Click the **Run** list, and then click **Run#x**.

Note: A number is assigned to each profiling session. These numbers increase incrementally each time you run a profiling session. x= the number that was assigned to your first run.

PL/SQL Profiler populates the grid with information on the procedure, package body and package specification.

Note: For the purposes of this walk though we have created this package under the account SCOTT.

4 Click the **Run** list again, and then click the **Run#x** for your second run.

Notice this time there is no information on the package specification. It was created in the first run.

5 Right-click, and then click **Detail**.

The **PL/SQL Profiler - Unit Detail** window opens, the grid is populated with the average time to execute each unit and the source code. Notice the time to execute SELECT object_name, in the example is 126 ms.

6 In the **PL/SQL Profiler - Unit Detail** window, click the **Run** list, and then click **Run#x** for your first run.

7 Click the **Unit** list, and then click **user name.PF_COUNT_SYSTEM_OBJECTS**.

Notice the time to execute SELECT object_name is considerably greater: in the example it is 24476 ms.

8 Right-click, and then click **Show Only Hit Lines**.

The PL/SQL Profiler shows only the lines of code that executed.

9 Right-click, and then click **Advanced View**.

The **Advanced View** window opens.

10 Continue clicking the **Run** and **Unit** lists to compare the performance of each run and each session.

This concludes the Sample Profiling Session. You can delete the objects created during the Sample Profiling Session. They are:

- Check Constraints, PLSQL_PROFILER_UNITS, PLSQL_PROFILER_DATA
- Foreign Keys, PLSQL_PROFILER_UNITS, PLSQL_PROFILER_DATA
- Package, PF_COUNT_TIME_INTERVAL
- Package functions, WEEKEND_DAYS_( ), WORKING_DAYS_( ), YEARS_ELAPSED_BETWEEN_ ()
- PL/SQL code Profiles, DemoProfile
- Primary Keys, PLSQL_PROFILER_RUNS, PLSQL_PROFILER_UNITS, PLSQL_PROFILER_DATA
- Procedure, PF_COUNT_SYSTEM_OBJECTS
- Sequence, PLSQL_PROFILER_RUNNUMBER
- Tables, PLSQL_PROFILER_RUNS, PLSQL_PROFILER_UNITS, PLSQL_PROFILER_DATA

Capacity Analyst (DBArtisan)

Welcome to the Embarcadero Capacity Analyst. The Capacity Analyst collects critical database storage, object, and performance metrics at scheduled intervals and provides visual analysis and reporting mechanisms that allow for historical trend analysis and forecasting.

Capacity Analyst lets you:

- Create and maintain Capacity Collections that serve as containers for related statistics.
- Collect statistics in Capacity Collections.
- View graphic and report-styled presentations of trend and predictive analysis scenarios.
- Maintain functions regarding the handling of stored statistics.

The table below describes the major sections of Help.

Section	Description
Getting Started with the Capacity Analyst (DBArtisan)	Provides information about installing, opening, and licensing Capacity Analyst.
Capacity Analyst Tutorial (DBArtisan)	Helps you get started with Embarcadero Capacity Analyst. Provides the foundation you need to explore the many features and benefits of Capacity Analyst.
Using Capacity Analyst (DBArtisan)	Shows how to use each component of the application. Using Capacity Analyst breaks each process and procedure into simple step-by-step instructions.

Getting Started with the Capacity Analyst (DBArtisan)

Welcome to the Embarcadero Capacity Analyst. The Capacity Analyst helps you analyze current usage trends in your key databases and lets you forecast where your databases are heading in the future.

For more information, see:

- What You Should Know Before Installing Capacity Analyst (DBArtisan)
- Capacity Analyst Repository Installation (DBArtisan)
- Uninstalling Capacity Analyst Repository (DBArtisan)
- Managing Repository Security (DBArtisan)
- Repository Information (DBArtisan)
- Licensing Capacity Analyst (DBArtisan)

What You Should Know Before Installing Capacity Analyst (DBArtisan)

When you install the Analyst, several Embarcadero-generated stored objects are pushed onto the database server:

These objects are as follows:

SQL SERVER

Procedures

eas_apply_collection_filter
eas_apply_collection_schedule
eas_apply_purge_condition
eas_apply_refresh_stats_condition
eas_create_collection
eas_delete_collection
eas_get_collection_metric_info
eas_get_metric_query
eas_get_metric_query_ex
eas_get_metric_query_ex1
eas_get_purge_condition
eas_get_purge_status
eas_get_refresh_stats_condition
eas_get_refresh_stats_status
eas_is_admin
eas_is_auto_purge_set
eas_is_collection_available
eas_is_collection_scheduled
eas_is_filter_all_set
eas_list_perf_stats
eas_list_wait_events
eas_purge_data
eas_refresh_stats
eas_resolve_collection_objects
eas_resolve_datastruct
eas_resolve_indexes
eas_resolve_perfstats
eas_resolve_tables
eas_resolve_waitevents
eas_run_collection
eas_run_datastruct
eas_run_perfstats
eas_run_waitevents
eas_set_auto_purge_status
eas_set_collection_status
eas_set_refresh_stats_status
eas_update_schedule_dates

Foreign Keys

eas_collection_fk1
eas_collection_fk2
eas_collection_object_fk1
eas_collection_object_fk2
eas_filter_fk1
eas_filter_fk2
eas_group_recfk
eas_map_fk1
eas_object_fk
eas_object_type_fk
eas_option_fk1
eas_property_fk1
eas_property_fk2
eas_schedule_fk
eas_statistic_fk1
eas_statistic_fk2
eas_statistic_fk3

Tables

eas_collection
eas_collection_object
eas_collection_sql
eas_database
eas_filter
eas_group
eas_map
eas_object
eas_object_type
eas_option
eas_option_type
eas_property
eas_schedule
eas_statistic
eas_version

Metadata

eas_column_map_insert
eas_database_insert
eas_group_insert
eas_object_type_insert
eas_option_insert
eas_option_type_insert
eas_version_insert

DB2

Foreign Keys

eas_col_obj_fk1
eas_col_obj_fk2
eas_collection_fk1
eas_collection_fk2
eas_filter_fk1
eas_filter_fk2
eas_group_recfk
eas_map_fk1
eas_object_fk
eas_object_type_fk
eas_option_fk1
eas_property_fk1
eas_property_fk2
eas_schedule_fk
eas_statistic_fk1
eas_statistic_fk2
eas_statistic_fk3

Index

eas_collection_idxs
eas_collection_object_idxs
eas_filter_idxs
eas_group_idxs
eas_map_idxs
eas_object_type_idxs
eas_option_idxs
eas_property_idxs
eas_schedule_idxs
eas_statistic_idxs

Metadata

eas_column_map_insert
eas_group_insert
eas_object_type_insert
eas_option_insert
eas_option_type_insert
eas_version_insert

Tables

eas_collection_object
eas_collection_sql
eas_database
eas_group
eas_map
eas_object_type
eas_option
eas_option_type
eas_schedule
eas_version

ORACLE

Foreign Keys

eas_collection_fk1
eas_collection_fk2
eas_collection_object_fk1
eas_collection_object_fk2
eas_collection_sql_fk1
eas_filter_fk1
eas_filter_fk2
eas_map_fk1
eas_object_fk
eas_option_fk1
eas_property_fk1
eas_property_fk2
eas_schedule_fk
eas_statistic_fk2
eas_statistic_fk3

Indexes

eas_collection_fk1
eas_collection_fk2
eas_filter_fk1
eas_filter_fk2
eas_object_fk
eas_object_uk
eas_option_fk1
eas_property_fk1
eas_property_fk2
eas_schedule_fk
eas_statistic_fk1

Metadata

eas_column_map_insert
eas_object_type_insert
eas_option_insert
eas_option_type_insert
eas_version_insert

Tables

eas_collection
eas_collection_object
eas_collection_sql
eas_database
eas_filter
eas_map
eas_object
eas_object_type
eas_option
eas_option_type
eas_property
eas_schedule
eas_statistic
eas_version

Sybase ASE

Foreign Keys

eas_collection_fk1
eas_collection_fk2
eas_collection_object_fk1
eas_collection_object_fk2
eas_filter_fk1
eas_filter_fk2
eas_group_recfk
eas_map_fk1
eas_object_fk
eas_object_type_fk
eas_option_fk1
eas_property_fk1
eas_property_fk2
eas_schedule_fk
eas_statistic_fk1
eas_statistic_fk2
eas_statistic_fk3

Indexes

eas_collection_idxs
eas_collection_object_idxs
eas_filter_idxs
eas_group_idxs
eas_map_idxs
eas_object_idxs
eas_object_type_idxs
eas_option_idxs
eas_property_idxs
eas_schedule_idxs
eas_statistic_idxs

Metadata

eas_column_map_insert
eas_database_insert
eas_group_insert
eas_object_type_insert
eas_option_insert
eas_option_type_insert
eas_version_insert

Procedures

sp_embt_ca_check_syb_version
sp_embt_ca_db_obj_space
sp_embt_ca_dbspace
sp_embt_ca_delete_collection
sp_embt_ca_get_col_metric_info
sp_embt_ca_get_metric_query_ex
sp_embt_ca_get_purge_cond
sp_embt_ca_get_purge_status
sp_embt_ca_get_refresh_cond
sp_embt_ca_get_refresh_status
sp_embt_ca_is_admin
sp_embt_ca_is_auto_purge_set
sp_embt_ca_is_col_available
sp_embt_ca_is_col_scheduled
sp_embt_ca_is_filter_all_set
sp_embt_ca_purge_data
sp_embt_ca_run_collection
sp_embt_ca_run_datastruct
sp_embt_ca_set_col_status
sp_embt_ca_upd_schedule_dates

Table

eas_collection_object
eas_collection_sql
eas_database
eas_group
eas_map
eas_object_type
eas_option
eas_option_type
eas_schedule
eas_version

Opening the Capacity Analyst (DBArtisan)

Capacity Analyst runs as a tab in the main DBArtisan window.

To open the Capacity Analyst:

1 Start DBArtisan and connect to the datasource that you want to run reports or analysis against. For details, see Connecting to Datasources.

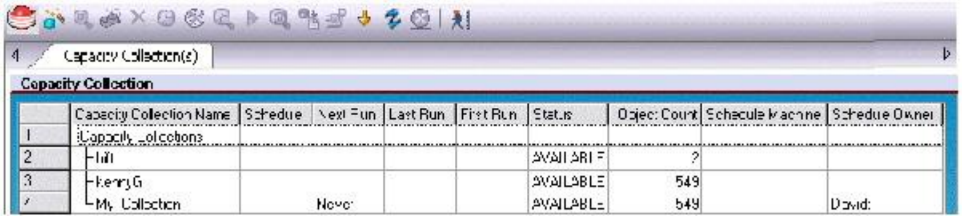
2 From the **Analyst** menu select **Capacity Analyst**.

OR

On the Analyst Series toolbar click the Capacity Analyst button.

::

Capacity Analyst opens, displaying any existing collections.



Capacity Collection

	Capacity Collection Name	Schedule	Next Run	Last Run	First Run	Status	Object Count	Schedule Machine	Schedule Owner
1	Capacity Collections								
2	[illegible]					AVAILABLE	2		
3	Kerry G					AVAILABLE	549		
4	My Collection		Never			AVAILABLE	549		David

Capacity Analyst Repository Installation (DBArtisan)

The Capacity Analyst Repository Installation Wizard lets you install the repository on each database the Capacity Analyst runs against. The repository holds all the captured metrics that you want to collect. For example, this can be the space used in an Oracle tablespace or the number of physical reads on a Microsoft SQL Server.

The wizard installs a set of objects to manage the captured data, and a security role to manage access to the repository and objects.

The Analyst Repository Installation Wizard lets you:

- Create a new repository.
- Upgrade an existing repository.
- Assign a repository owner.
- Assign object placement in the database.
- View feedback on the repository install actions.

Completing the Analyst Repository Installation Wizard

To install the repository, do the following:

1 On the DBArtisan toolbar, click the Capacity Analyst icon.

DBArtisan opens Capacity Analyst. Capacity Analyst detects that no repository is present and opens the Analyst Repository Installation Wizard.

2 Use the following table as a guide to understanding and setting options in the Wizard,

Panel	Option	Description
1	Create a new repository	Lets you create a new repository under a new schema.
	Upgrade existing repository	Lets you upgrade the version of a current repository. Updates the stored procedures, and adds any new columns to tables. Also lets you use a an existing repository that was created for a different Analyst. NOTE: Available only if the wizard detects the presence of an earlier version.
	Use current login information	IBM DB2 ONLY: Select to use the default connectivity information as the login id/password.
	Login Name	Lets you type the server administrator ID. NOTE: An administrator ID is not required after the initial repository setup.
	Password	Lets you type the server administrator password.

	Login As	ORACLE ONLY: Lets you select one of three Oracle login modes: Default - Establishes connection with no system roles. SYSDBA - Lets you perform all DBA-related functions, like startup and shutdown. NOTE: SYSDBA is required for the SYS id when you log into a 9i instance. SYSOPER - Lets you perform many DBA-related functions, like startup and shutdown.
	Using	MICROSOFT SQL SERVER: Lets you specify authentication type, either SQL Server Authentication or Windows Authentication.
2	Database	MICROSOFT SQL SERVER ONLY: Lets you select a database. SYBASE ASE ONLY: The following system databases are excluded from the list of potential targets for Repository installation: 'Replication' 'model' 'sybsecurity' 'sybsystemdb' 'sybsystemprocs' 'tempdb' 'dbccdb' 'sybdiag'
	Login	MICROSOFT SQL SERVER and SYBASE ASE ONLY: Lets you select an owner. NOTE: Should have the CREATE PROCEDURE privilege on the storage device that is selected. For Oracle, the storage for tables and storage for indexes are tablespaces. The owner must have privileges for the tablespaces.
	Owner	ORACLE ONLY: Select a new user to own the repository or click New to open the User Wizard. For details, see Users Wizard (Oracle). NOTE: Should have the CREATE PROCEDURE privilege on the storage device that is selected. For Oracle, the storage for tables and storage for indexes are tablespaces. The owner must have privileges for the tablespaces.
	Storage for tables (and its available space)	Lets you separate the storage placement of repository tables and indexes. For Oracle, the storage for tables are tablespaces. Select from the list. Or click New to open the creation wizard.
	Storage for indexes (and its available space)	Lets you separate the storage placement of repository tables and indexes. For Oracle, the storage for indexes are tablespaces. Select from the list. Or click New to open the creation wizard.
	Password	Lets you specify a password for the repository owner.
	Schema	IBM DB2 ONLY: Lets you select a schema. All the tables in the repository will be created for this schema. The default value is EMBTCA.
	Tablespace for Repository Tables	IBM DB2 ONLY: Lets you select the tablespace on which the repository tables are created.
	Tablespace for Repository Indexes	IBM DB2 ONLY: Lets you select the tablespace on which the repository tables are created.
	New	IBM DB2 ONLY: Click to open the Tablespace WIzard. FOr details, see Tablespaces Wizard (DB2 LUW).
3	Repository Login	Displays repository login
	Execution Log	Displays installation log.
	Execution Errors	Displays any errors if they occur.

Finish	Click to complete the installation. Capacity Analyst installs the necessary files and displays the installation process. When the installation finishes, it displays a finished message. Click **Finish** to open the **Capacity Analyst** tab. If the installation fails, Capacity Analyst lets you stop the repository installation. Capacity Analyst deletes the objects that it created.

3 When ready, click Install.

Capacity Analyst creates the repository. Capacity Analyst prompts you to grant other user accounts the privilege of using Capacity Analyst.

4 Select a basic security role:

- Administrator – Users granted this privilege can create, edit, remove, and view capacity collection information.
- User - Users granted this privilege can only view capacity collection information.

Tip: You can easily grant or remove users from these roles at repository installation time or afterwards. For details, see Managing Repository Security (DBArtisan).

Uninstalling Capacity Analyst Repository (DBArtisan)

The Capacity Analyst Repository Uninstall Wizard reverses the process completed by the Analyst Repository Installation Wizard.

Completing the Capacity Analyst Repository Uninstall Wizard

1 In the DBArtisan Datasource Navigator/Explorer, connect to a datasource that has repository installed.

2 On the Analyst toolbar, click the Capacity Analyst button.

3 On the Capacity Collection Editor toolbar, select Command, Repo Management, and then Uninstall Analyst Repo.

Capacity Analyst opens the Capacity Analyst Repository Uninstall Wizard.

4 Complete the wizard panels.

5 Click Finish.

Managing Repository Security (DBArtisan)

The **Manage Repository Security** dialog lets you manage repository security.

The table below describes the options and functionality on the **Manage Repository Security** dialog

Option	Description
Grant	Click to grant privileges to target user.
Revoke	Click to revoke privileges from target user.

Completing the Manage Repository Security Dialog Box

1 In the DBArtisan Datasource Navigator/Explorer, connect to a datasource that has repository installed.

2 On the Analyst toolbar, click the Capacity Analyst button.

3 On the Capacity Collection Editor toolbar, select Command, Repo Management, and then Manage Analyst Repo.

Capacity Analyst opens the **Manage Repository Security** dialog. **The Manage Repository Security** dialog lets you manage security for the Capacity Analyst.

4 To grant privileges, select the target user, and then click Grant.

Capacity Analyst opens the **Grant Privilege(s) To** dialog. The **Grant Privilege(s) To** dialog lets you select or clear the check boxes corresponding to the target privileges.

5 To revoke privileges, select the target user, and then click Revoke.

Capacity Analyst opens the **Revoke Privilege(s) From** dialog. **The Revoke Privileges (s) From** dialog lets you select or clear the check boxes corresponding to the target privileges.

6 Click OK.

Repository Information (DBArtisan)

The **Repository Information** dialog displays a tree view to present the information about the repository. The table below describes the standard nodes all datasource platforms:

Node	Description
Repository Owner	Displays the login name of the repository owner.
Object Placement	Displays the database (for SQL Server and Sybase ASE), tablespace (for Oracle), placement for tables and indexes (filegroup name - SQL Server, segment name - Sybase ASE, tablespace - Oracle and DB2).
Repository Version	Schema and logic version for the tables/indexes and procedures respectively.

Microsoft SQL Server

The **Repository Information** dialog displays the following information for Microsoft SQL Server:

- Repository owner
- Repository version
- Object placement
- Database
- Filegroup name where tables are stored
- Filegroup name where indexes are stored.
- Dates
- Date Created

Sybase ASE

The **Repository Information** dialog displays the following information for Sybase ASE:

- Repository owner
- Repository version
- Object placement
- Database
- Segment name where tables are stored
- Segment name where indexes are stored.
- Dates
- Date Created

Oracle

The **Repository Information** dialog displays the following information for Oracle:

- Repository owner
- Repository version
- Object placement
- Tablespace name where indexes and tables are stored
- Dates
- Date Created
- Date Updated

IBM DB2

The **Repository Information** dialog displays the following information for IBM DB2:

- Repository schema
- Repository owner
- Repository version
- Object placement
- Tablespace name where indexes and tables are stored
- Dates
- Date Created

Completing the Repository Information Dialog Box

1 In the DBArtisan Datasource Navigator/Explorer, connect to a datasource that has repository installed.

2 On the Analyst toolbar, click the Capacity Analyst button.

3 On the Capacity Collection Editor toolbar, select Command, Repo Management, and then Manage Analyst Repo.

Capacity Analyst opens the **Repository Information** dialog.

4 To save repository information as a report, click Save.

Capacity Analyst opens the **Save Grid** dialog.

5 In Save In, select the location for the repository file.

6 In File Name, type the file name.

7 In Save as Type, select a file type.

8 Click Save.

Capacity Analyst generates the report and returns to the **Repository Information** dialog.

9 Click OK.

Capacity Analyst closes the **Repository Information** dialog.

Licensing Capacity Analyst (DBArtisan)

Availability of this feature depends on your licensing. For more information, see Licensing.

Capacity Analyst Tutorial (DBArtisan)

Embarcadero Capacity Analyst offers intelligent insight into the past, current, and future inner-workings of a database. By collecting historical database and performance metrics at user-defined intervals, Capacity Analyst provides historical trend analysis along with smart forecasting abilities that let you plan for the future needs of key databases. Capacity Analyst runs inside the Embarcadero DBArtisan console so you can perform all your database administration and analysis in one place.

Embarcadero Capacity Analyst lets you analyze current trends in your database storage and systems performance and helps you understand when corrective action is necessary. Capacity Analyst also helps you forecast the future needs of the your database so that your key systems can continue to meet the growing needs of its users.

Proceed to About this Capacity Analyst Tutorial (DBArtisan)

Topics

- About this Capacity Analyst Tutorial (DBArtisan)
- Session 1: Creating and Scheduling a Collection (DBArtisan)
- Session 2: Using Analysis and Reporting Tools (DBArtisan)

About this Capacity Analyst Tutorial (DBArtisan)

This short tutorial helps you get started with Embarcadero Capacity Analyst. After you complete this exercise. you will have the foundation you need to explore the many features and benefits of Capacity Analyst. You will learn how to set up Capacity Analyst and become familiar with creating and scheduling statistical collections. You will also understand how to analyze database trends and perform future forecasts of your database's needs.

Proceed to Summary of the Tasks in this Tutorial

Summary of the Tasks in this Tutorial

The main categories of Capacity Analyst activities are:

- **Creating and editing collections** - a collection is basically a specified set of metrics. Capacity Analyst lets you use a wizard to create collections, specifying DBMS-specific metrics as content. You can edit and clone collections as well.
- **Executing collections** - based on counters, timers, and statistics packages available from the DBMS you are working against, collections store calculated totals. Executing a collection updates the totals in the collection. Depending on your needs, you can execute collections in an ad hoc fashion or you can schedule collections to be executed at regular intervals.
- **Analysis and reporting** - Capacity Analyst provides prepackaged reports and a visual analysis tool.

This tutorial walks you through a sample of these activities. You will create a collection with a small variety of metrics, schedule the collection to be updated each hour, and use the analysis and reporting tools.

Proceed to Choosing an Appropriate Database for the Tutorial Exercises

Choosing an Appropriate Database for the Tutorial Exercises

The full power of the analysis and reporting tools can only be realized if you work through the exercises against an active database. For best results, use an active test bed or a production system.

Rationale for a Two-part Tutorial

While a collection stores historical statistics, totals cannot reflect database activities that occurred before the collection was created. In order to view meaningful analysis and reports, enough time must pass to allow a bare minimum of database activity. With that in mind, this tutorial is presented in two parts:

- Session 1: Creating and Scheduling a Collection (DBArtisan)

- Session 2: Using Analysis and Reporting Tools (DBArtisan)

After completing the exercises in session 1, you should wait minimally four hours before starting the second session. This should allow sufficient database activity to allow for meaningful analysis.

Session 1: Creating and Scheduling a Collection (DBArtisan)

In Session 1, you will first ensure that you are properly set up to evaluate the Capacity Analyst. Subsequently, you will run the Capacity Analyst, build a small statistics collection, and ensure that it is updated on an hourly basis.

Proceed to Getting Started with Capacity Analyst (Optional)

Getting Started with Capacity Analyst (Optional)

The following topics prepare you for first use of the Capacity Analyst:

- Downloading and Installing DBArtisan
- Starting DBArtisan and Connecting to a Datasource
- Installing the Capacity Analyst Repository

When you have performed these tasks, or if you have already performed them, you can proceed to Creating and Scheduling a Capacity Collection Using Capacity Analyst.

Downloading and Installing DBArtisan

You can obtain the latest version of Capacity Analyst by downloading the DBArtisan software from the Embarcadero Web site at http://www.embarcadero.com. Capacity Analyst runs fully-contained within DBArtisan so you will need the DBArtisan product to use Capacity Analyst.

To download and install DBArtisan

1 Locate the download area of the Web site and follow the steps indicated.

2 Save the file on your computer and then double-click to launch the self-extracting file that will guide you through the installation process.

When you first install an evaluation copy of DBArtisan, you can use the tool for the duration of the trial period. After that time, a permanent license is needed.

Proceed to Starting DBArtisan and Connecting to a Datasource

Starting DBArtisan and Connecting to a Datasource

After installing DBArtisan, you must set up datasources to establish reusable connections to your database servers. For now, you can establish a single connection to the database/server you will use to evaluate Capacity Analyst. DBArtisan can detect and let you register any datasource in your network.

To connect to a database/server

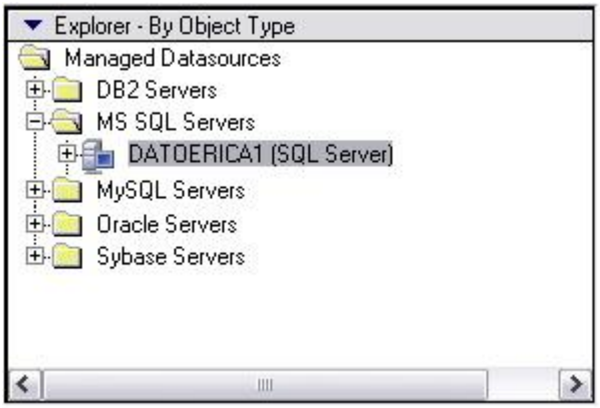
1 Start DBArtisan. For a default installation, **Start** > **Programs** > **Embarcadero DBArtisan***version number* > **DBArtisan** (where *version number* corresponds to the version of DBArtisan that you downloaded).

2 On the **Datasource** menu, select **Discover Datasources**.

The **Discover Datasources** dialog opens. After searching your network it lists all datasources to which you can connect.

3 Select the check box corresponding to the target datasource and click **Register**.

The datasource is added to the Datasource Navigator/Explorer, under the relevant DBMS folder. For example:



::

4 RIght-click the datasource and select **Connect** from the context menu.

A login dialog opens.

5 Provide valid user name and password credentials and click **OK**.

Proceed to [Installing the Capacity Analyst Repository]()

Installing the Capacity Analyst Repository

Capacity Analyst makes use of a repository to store historical statistics about your database. A repository is required for each database/server that you use with Capacity Analyst.

To install a repository on the datasource you connected to

1 From the **Analyst** menu, select **Capacity Analyst**.

OR

On the Analyst Series toolbar, click the Capacity Analyst button.

::

DBArtisan opens Capacity Analyst. Capacity Analyst detects that no repository is present and opens the Analyst Repository Installation Wizard.

Depending on the type of DBMS type of the datasource you connected to, specific options will differ. In general, the Wizard prompts for login/password credentials, as well as server, database, and storage details.

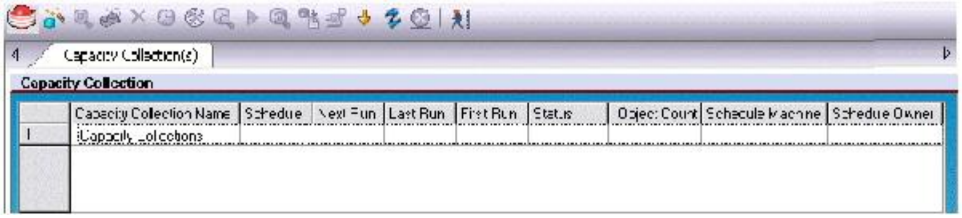
2 Provide the information that the Wizard prompts you for and when complete, click the **Install** button and provide additional information when prompted.

Capacity Analyst creates the necessary repository objects.

Proceed to Creating and Scheduling a Capacity Collection Using Capacity Analyst

Creating and Scheduling a Capacity Collection Using Capacity Analyst

When you open Capacity Analyst, Capacity Analyst displays a simple Capacity Collection Manager interface that lets you view existing collections, create new collections, and perform many other management functions.



Capacity Collection

	Capacity Collection Name	Schedule	Next Run	Last Run	First Run	Status	Object Count	Schedule Machine	Schedule Owner
1	Capacity Collections								

Proceed to Creating a New Capacity Collection Using Capacity Analyst

Creating a New Capacity Collection Using Capacity Analyst

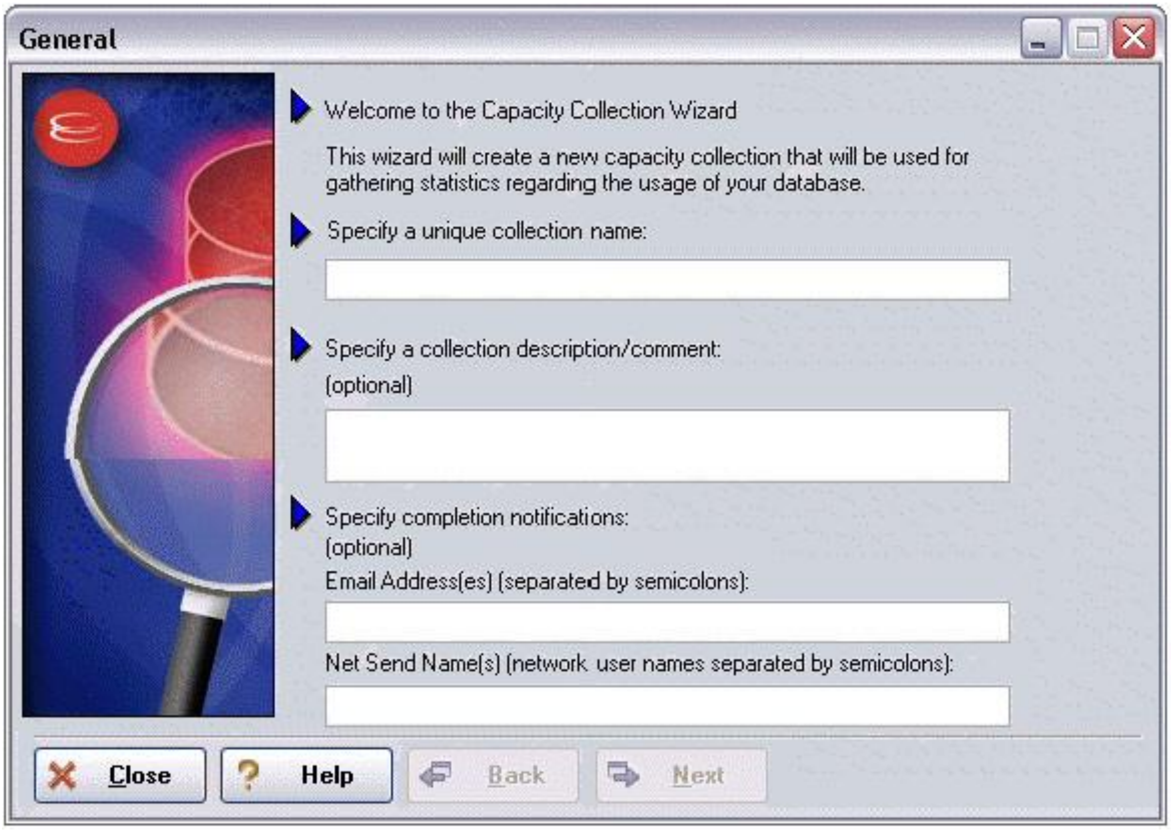
To get started collecting information for your database, you first need to create a new collection. A collection is essentially a set of related metrics or statistics that you want to report on. As you begin to make use of Capacity Analyst, you will develop a strategy for metrics you will store in each collection. For the purposes of this tutorial, you can build an arbitrary set of statistics into a collection.

You create a collection using the Capacity Collection Wizard.

To open the Capacity Collection Wizard

1 On the Capacity Analyst toolbar, click the New Collection Wizard button.

Capacity Analyst opens the Capacity Collection Wizard.



Welcome to the Capacity Collection Wizard

This wizard will create a new capacity collection that will be used for gathering statistics regarding the usage of your database.

Specify a unique collection name:

Specify a collection description/comment:
(optional)

Specify completion notifications:
(optional)
Email Address(es) (separated by semicolons):
Net Send Name(s) (network user names separated by semicolons):

::

The first panel is devoted to naming your capacity collection and specifying various collection properties like descriptions and optional job completion notifications.

To complete the Wizard

1 Provide the information in the fields provided and click the Next button to continue.

Capacity Analyst opens the **Selection Types** panel of the Capacity Collection Wizard. The second panel asks you what type of information you would like to collect for your database. While this panel prompts you to choose general categories, subsequent panels will let you provide more specific information about the choices you make here.

The specific options differ according to the type of DBMS datasource you are connected to, but in general the categories are:

- Storage structures - Depending on the DBMS you are working with, your analysis and reports can contain space and usage statistics on items such as databases and database devices, logs and backup space, partitions, tablespaces, filegroups, and files.
- DB objects - Your analysis and reports can contain space and usage statistics on tables, indexes, and for some supported DBMS types, other database objects.
- Performance-based statistics - if available for the DBMS you are connected to, this lets you use product-specific performance-based statistic packages.
- Wait events - if available for the DBMS you are connected to, this lets you work with wait-based event information provided by the DBMS.

2 For the purposes of this tutorial, select all options on the panel and then click **Next**.

The next set of panels let you select specific details. Each panel corresponds to one of the options you chose on the **Selection Types** panel.

Most panels offer categories that let you drill down by expanding the category to expose specific options. You can then select specific options and move them from the **Available** list to the **Selected** list and then click **Next** to move to the next panel.

3 Complete all panels up until the **Statistics** tab is displayed.

Tip: Since table and index statistics are supported for all DBMS, as an aid in subsequent sessions, select a variety of indexes and tables on the **DB Objects** panel.

The **Statistics** tab prompts you to update the statistics you chose. In order to use the analysis tools offered by Capacity Analyst, you must execute the collection. This performs calculations and rollups on the metrics in the collection, letting you produce up-to-date reports and charts.

4 Look over the options available on the **Statistics** panel, enable statistics update and select one or more update options, and then click **Next**.

The AutoPurge panel is displayed. It offers the option to delete older data, each time that you execute a collection.

5 Enable AutoPurge and click the **Finish** button.

Proceed to Scheduling a Collection in Capacity Analyst

Scheduling a Collection in Capacity Analyst

Immediately after your collection has been saved to the repository, Capacity Analyst prompts you if you would like to schedule the collection to run at specified intervals. It is not necessary that you schedule your collection to actually collect statistics in the repository. You can execute a collection in an ad-hoc manner at any time through the Capacity Collection Manager.

Depending on your requirements however, scheduling a collection may be the preferred route. For example, if you want to run packaged reports on a scheduled basis, scheduling collection of your collections will ensure up-to-date statistics in your scheduled reports.

For the purposes of this tutorial, you want the statistics in your collection updated every hour.

To schedule updates for your new collection

1 Select Yes when asked if you would like to schedule your new collection

The Microsoft Task Manager scheduler opens.

2 Click **Advanced** to open the **Advanced Schedule Options** dialog.

3 Select **Repeat task**, specify that the tasks is to be repeated every hour, ensure that the task is repeated until a time at least four hours from now, and click OK to close the **Advanced Schedule Options** dialog.

4 Click OK to schedule your collection and dismiss the dialog.

Tip: You can add, edit, or remove any schedule for any collection at a later time by using the management functions supplied in the Capacity Collection Manager interface.

Proceed to Reviewing Other Collection Management Functions

Reviewing Other Collection Management Functions

While you have completed all required tasks for Session 1, it is worthwhile looking at some of the options Capacity Analyst offers in creating and updating collections. Later, as you develop a strategy for setting up the collections for your enterprise, you will use additional Capacity Analyst features in managing your collections. With respect to maintaining your collections you have the following options:

Editing collections	After creating a collection, you can modify general settings and the AutoPurge option. Since a collection stores totals and rolled up statistics that can be compromised, you cannot modify your choice of statistics or the objects you generate those statistics against.
Cloning collections	Cloning lets you create a collection based on an existing collection and modify the metrics used and options specified in the original.
Deleting collections	If you no longer have use for a collection or you are going to replace it, you can delete that collection.

Similarly, when you created your collection, you selected the option to update your statistics on an hourly basis. In some cases, you may require ad hoc reports or analysis. If so, you can update your collection manually.

To manually update a collection

1 In the Capacity Collection Manager, select a collection.

2 On the Capacity Analyst toolbar, click the Execute button.

▶::

Depending on the volume of statistics you specified when creating the collection, this may take a few seconds. When complete the **Last Run** field will reflect the update.

There are other functions contained inside the Capacity Collection Manager, however those described above are likely the ones you will use the most often when managing your capacity collections. You will gain experience using these features as you learn more about Capacity Analyst.

Proceed to Session 2: Using Analysis and Reporting Tools (DBArtisan)

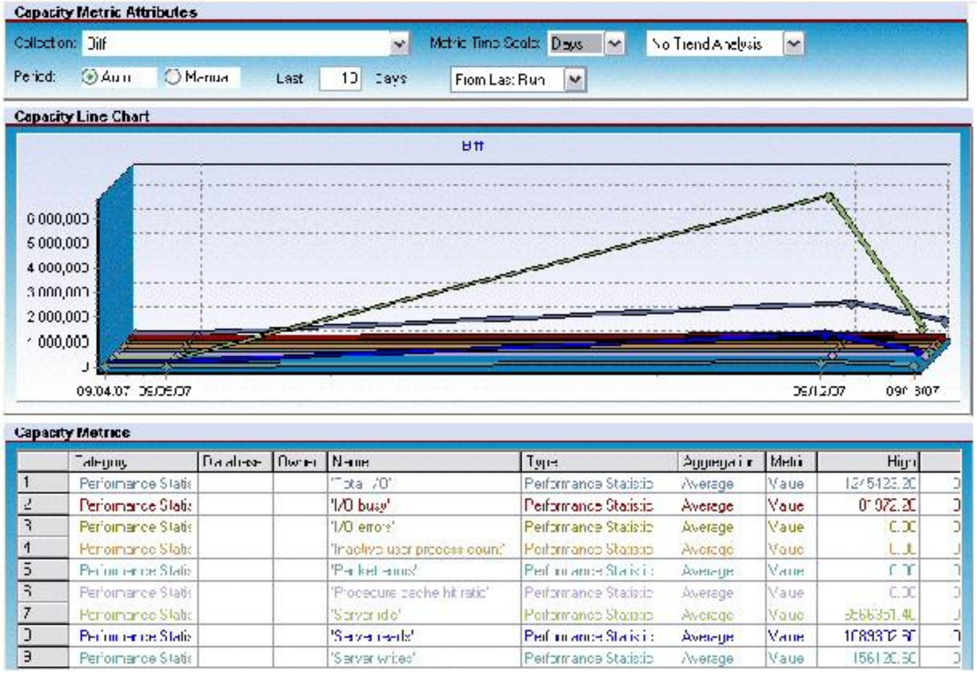
Session 2: Using Analysis and Reporting Tools (DBArtisan)

After several hourly updates of the statistics in the collection you created in Session 1, you are ready to start using the analysis tools. In Session 2, you will experiment with the Collection Analysis Viewer and generate one of the Capacity Analyst predefined reports.

Proceed to [Performing Trend Analysis Using Capacity Analyst]()

Performing Trend Analysis Using Capacity Analyst

The Collection Analysis Viewer presents selected statistics in a graphical format and lets you view historical trends and extrapolate that data into the future.



	Category	Database	Owner	Name	Type	Aggregation	Metric	High
1	Performance Statis			'Total I/O'	Performance Statistic	Average	Value	[illegible]
2	Performance Statis			'I/O busy'	Performance Statistic	Average	Value	[illegible]
3	Performance Statis			'I/O errors'	Performance Statistic	Average	Value	0.00
4	Performance Statis			'Inactive user process count'	Performance Statistic	Average	Value	0.00
5	Performance Statis			'Packet errors'	Performance Statistic	Average	Value	0.00
6	Performance Statis			'Procedure cache hit ratio'	Performance Statistic	Average	Value	0.00
7	Performance Statis			'Server idle'	Performance Statistic	Average	Value	[illegible]
8	Performance Statis			'Server reads'	Performance Statistic	Average	Value	[illegible]
9	Performance Statis			'Server writes'	Performance Statistic	Average	Value	[illegible]

To open the Collection Analysis Viewer, do the following

1 Start DBArtisan again, and connect to the datasource you created earlier. If you need help, refer to [Starting DBArtisan and Connecting to a Datasource]().

2 On the **Analyst** menu, select **Capacity Analyst**.

3 Select the Capacity Collection that you recently created.

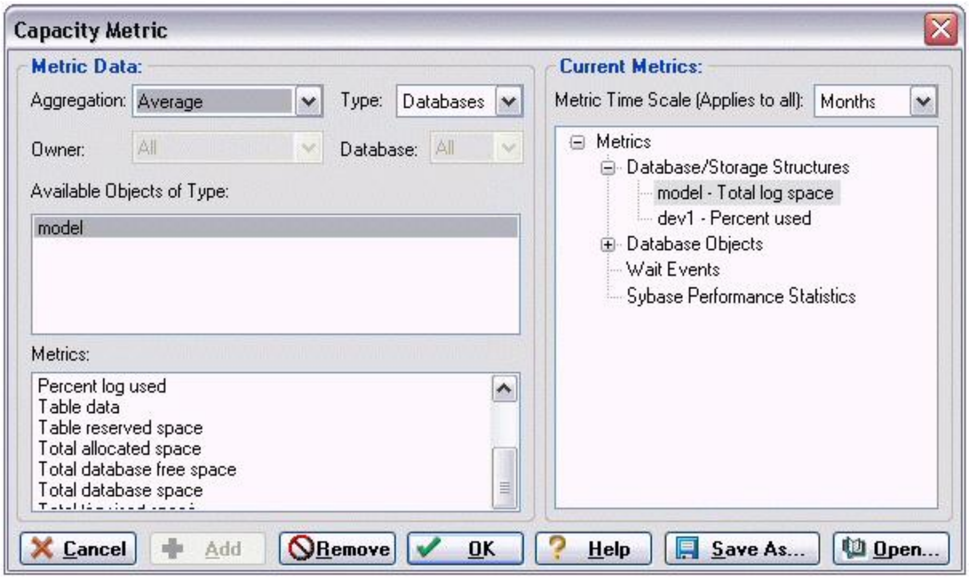
4 On the Capacity Analyst toolbar, click the Collection Analysis Viewer button.

::

Proceed to [Choosing the Charted Metrics You Want To View]()

Choosing the Charted Metrics You Want To View

The first time that you open the Collection Analysis Viewer, it prompts you select the metrics you want to work with. You specify the metrics using the displayed **Capacity Metric** dialog. For example:



Aggregation: Average
Type: Databases
Owner: All
Database: All
Available Objects of Type: model
Metrics:
Percent log used
Table data
Table reserved space
Total allocated space
Total database free space
Total database space

Metric Time Scale (Applies to all): Months
Metrics
Database/Storage Structures
model - Total log space
dev1 - Percent used
Database Objects
Wait Events
Sybase Performance Statistics

To specify the metrics that are to appear in the analysis

1 From the **Aggregation** dropdown, select an aggregation type: **Average**, **Maximum**, or **Minimum**.

2 From the **Type** dropdown, select the type of metric you want to add, such as a table or index.

Note: Certain Metric types allow you to further qualify using the **Owner** and **Database** dropdowns.

3 From the **Available Object of Type** list, choose a specific object or statistics type.

The **Metrics** list is updated with the metrics available for the **Type** and **Available Objects of Type** selections you made.

4 Select a metric from the **Metrics** list and click **Add**.

Capacity Analyst adds the metric to the **Current Metrics** list.

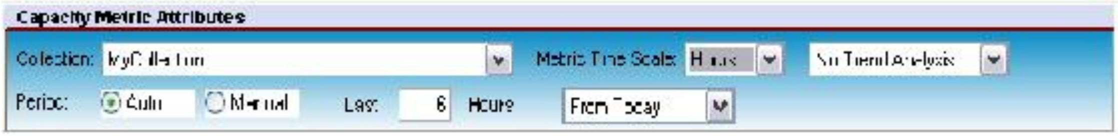
5 To add additional metrics, repeats step1 through step4. Keep in mind that the Collection Analysis VIew will calibrate its chart according to the metric with the largest value. If you are going to add multiple metrics, choose metrics that you expect will have similar ranges.

6 When finished adding metrics for the graphical portion of your analysis, click **OK**.

Proceed to Viewing Trends in Capacity Analyst

Viewing Trends in Capacity Analyst

When you finish adding metrics to the Collection Analysis Viewer, Capacity Analyst presents the data in graphical and grid form in the Collection Analysis Viewer. Your first step should be to ensure that the displayed timescale reflects the fact that you have been updating metrics for a matter of hours. You do this using the controls in the **Capacity Metric Attributes** area.



Collection: MyCollection

Metric Time Scale: Hours

No Trend Analysis

Period: ● Auto ○ Manual

Last: 6 Hours

From Today

To optimize viewing of the metrics in your collection

1 From the **Metric Time Scale** dropdown, choose **Hours**.

2 In the **Last** box, enter the number of hours since you scheduled your collection for hourly updates.

Having only statistics that cover a matter of hours, your viewing options are currently somewhat restrictive. However, during a later session you may want to experiment with the following viewing options:

- By default, the **No Trend Analysis** option is selected. This option shows historical data only. If you want statistics extrapolated into the future, you can use the **Trend to Date** or **Trend to Value** options to help you predict potential problems.
- You are currently using the **Auto** Period. It lets you quickly select an interval from the current day or the last statistics update. The **Manual** option lets you to have more granular control over the exact dates/times that Capacity Analyst uses to plot the data.
- To display the actual data values, position the pointer over the trend lines in the graph. Pop up values display the exact points on the chart.
- Each metric's statistic displays in the Analysis viewer's bottom grid with the high, low, and average value for the time period being shown for each metric.
- After your data has been displayed, you can continue to add or remove metrics from the Collection Analysis Viewer by using the Add or Remove toolbar buttons, and perform other functions using the various options available on the Collection Analysis viewer.

3 When ready to proceed to the next exercise, on the Capacity Analyst toolbar click the **Close the Active Tab** button.

::

Proceed to Using Capacity Analyst Predefined Reports

Using Capacity Analyst Predefined Reports

An alternative forecasting method is through use of predefined Analysis reports. Capacity Analyst provides reports on growth trends of devices, databases, tables, rows, and other objects, as well as reports on fragmentation details and failure prediction. Like the COllection Analysis View, the reports show historical data and can be configured to extrapolate to a future date.

To generate a report on the fastest growing database objects

1 Ensure that the collection you created is selected.

2 Click the Report button.

::

The Capacity Analyst opens the **Select Report Type** dialog.

3 To create a pre-defined trend analysis report, select Analysis Report and click **OK**.

The Capacity Analyst opens the **Capacity Analyst Report Wizard**.

4 Accept the default to create a New Analysis Report.

5 Click Next.

6 From the report drop down list, select 'N" Fastest Growing Objects. Note that there are many other reports available.

7 Accept the default **Auto Entry** time period of 1 day.

8 Accept the **Day** default rollup.

9 Specify the future date you want Capacity Analyst to use for its future 'stop' date.

10 Click Next.

11 Specify what databases/objects to include in the report.

12 Click Next.

13 Specify a Report File Name and a Report Title.

14 Click Execute.

Capacity Analyst opens the report in a new tab and stores an HTML-based version of the report in the location you specified in the **Report File Name** field.

The HTML report can be viewed in any web browser so you can build internal web sites of your forecasts for all interested parties, including project leads, and operation managers.

Using Capacity Analyst (DBArtisan)

The general categories of tasks performed using the Capacity Analyst are:

- Creating and Editing Collections (DBArtisan) - Each collection stores a set of statistics. How you set up your collections will depend on your own reporting and monitoring requirements. Capacity Analyst lets you choose the statistics in your collections and control options such as automatic purging and archiving of data.
- Updating Statistics in Collections for use in Reports and Charts (DBArtisan) - In order to produce up-to-date reports and charts, the rolled up data in collections must be updated. Capacity Analyst lets you update your collections manually or chedule regular, automated updates.
- Viewing Charts and Reports on Collections (DBArtisan) - The charts and reports available from the Capacity Analyst let you view trends and help you make forecasts regarding growth of your databases. The Collection Analysis Viewer lets you view graphical representation of the metrics in a collection as a function of time. Standard reports provide information on growth trends for devices, databases, tables, rows, and other objects, as well as fragmentation details and failure prediction.

Creating and Editing Collections (DBArtisan)

Typically, how you set up collections will be dictated by your own requirements. If your enterprise has several applications for example, each using multiple databases, you might be interested in viewing graphs and reports on an application-by-application basis, with each collection comprising statistics for several databases. At a lower level, you may be interested in viewing statistics on collections of table or indexes.

Similarly, how you set up collections can also be impacted by Capacity Analyst functionality. For example, a collection option lets you schedule statistics update at specified intervals. If your requirements include reports with different intervals, hourly as opposed to weekly for example, you could set up your collections accordingly. Or since the Collection Analysis Viewer calibrates a graph to the range of the highest valued statistic displayed, you might group statistics with similar expected values.

In short, you should consider all functionality available in Capacity Analyst before deciding on a collection setup. For details see Content and Options in a Collection and Viewing Charts and Reports on Collections (DBArtisan).

The following topics provide details on creating and editing Capacity Analyst collections:

- Creating a New Collection
- Editing a Capacity Collection
- Cloning a Collection
- Deleting a Collection

Content and Options in a Collection

The reports and analysis you work with using Capacity Analyst include such statistics as space usage and access count details on database objects, databases, files and other storage mechanisms. They can also include performance-based and wait based statistics. The specific content and options available depend on the DBMS that DBArtisan is working with. The following topics provide summaries of available options and content on a product-by-product basis:

- IBM DB2 for Windows, Unix, and Linux Capacity Collection Content and Options
- Microsoft SQL Server Capacity Collection Content and Options
- Oracle Capacity Collection Content and Options
- Sybase ASE Capacity Collection Content and Options

IBM DB2 for Windows, Unix, and Linux Capacity Collection Content and Options

When creating or editing a Capacity Analyst collection, you specify the statistics you want in that collection and specify additional notification and update options. The table below summarizes the available statistics and options on the tabs/panels of collection wizards and editors for this DBMS.

Content/Option Category	Editor Tab or Wizard panel	Description or specific DBMS feature support
Identification/Notification	**General**	Lets you provide a name and description for the collection and optionally, provide an email address or net send name for notifications.
Types of statistics collected	**Selection Types**	Lets you select the types of content and options you want to include in the collection. You provide specifics in subsequent panels/tabs.
Database/storage structure space statistics	**Storage Structures**	A DB2 capacity collection can contain usage statistics on database (partition) space, DMS tablespace space, and SMS tablespace space.
Specific database object types for which to collect statistics	**DB Objects**	A DB2 capacity collection can contain usage statistics on indexes and tables
Update and rollup options	**Statistics**	A DB2 capacity collection offers the option to update statistics with additional specific choices for tables (distribution and column options), indexes (collection of extended and sample statistics), and access options (allowing read/write access during collection).
Automatic purging of old data	**AutoPurge**	Offers the option to automatically purge data older than a specified age, each time the collection is updated.

For information on how to create and edit collections, see:

- Creating a New Collection
- Editing a Capacity Collection
- Cloning a Collection

Microsoft SQL Server Capacity Collection Content and Options

When creating or editing a Capacity Analyst collection, you specify the statistics you want in that collection and specify additional notification and update options. The table below summarizes the available statistics and options on the tabs/panels of collection wizards and editors for this DBMS.

Content/Option Category	Editor Tab or Wizard panel	Description or specific DBMS feature support
Identification/Notification	**General**	Lets you provide a name and description for the collection and optionally, provide an email address or net send name for notifications.
Types of statistics collected	**Selection Types**	.Lets you select the types of content and options you want to include in the collection. You provide specifics in subsequent panels/tabs.
Database/storage structure space statistics	**Storage Structures**	An SQL Server capacity collection can contain size statistics on database backup space and database log backup space, and usage and size statistics on database space, filegroups, and files.
Specific database object types for which to collect statistics	**DB Objects**	An SQL Server capacity collection can contain usage and size statistics on indexes and tables.
Wait-event-based statistics	**Wait Events**	An SQL Server capacity collection can contain statistics on SQL Server Wait Types
Performance-based statistics	**Performance Statistics**	Provides statistics in the following performance counter categories: Access Methods, Buffer Manager, Buffer Partition, Databases, General Statistics, Latches, Locks, Memory Manager, SQL Statistics
Update and rollup options	**Statistics**	Provides the option to update statistics with additional options to specify a scan range, specify index or columns update, and recompute statistics
Automatic purging of old data	**AutoPurge**	Offers the option to automatically purge data older than a specified age, each time the collection is updated.

For information on how to create and edit collections, see:

- Creating a New Collection
- Editing a Capacity Collection
- Cloning a Collection

Oracle Capacity Collection Content and Options

When creating or editing a Capacity Analyst collection, you specify the statistics you want in that collection and specify additional notification and update options. The table below summarizes the available statistics and options on the tabs/panels of collection wizards and editors for this DBMS.

Content/Option Category	Editor Tab or Wizard panel	Description or specific DBMS feature support
Identification/Notification	**General**	Lets you provide a name and description for the collection and optionally, provide an email address or net send name for notifications.
Types of statistics collected	**Selection Types**	Lets you select the types of content and options you want to include in the collection. You provide specifics in subsequent panels/tabs.
Database/storage structure space statistics	**Storage Structures**	All tablespace, all non-temporary tablespaces, or select specific tablespaces
Specific database object types for which to collect statistics	**DB Objects**	Lets you include statistics on clusters, indexes, index partitions, rollback segments, tables, and table partitions.
Wait-event-based statistics	**Wait Events**	Lets you include statistics based on Oracle Wait Events.
Performance-based statistics	**Global Stats**	Lets you include Global Database Statistics.
Update and rollup options	**Statistics**	Offers the option to update statistics with additional options to compute or estimate statistics.
Automatic purging of old data	**AutoPurge**	Offers the option to automatically purge data older than a specified age, each time the collection is updated.

For information on how to create and edit collections, see:

- Creating a New Collection
- Editing a Capacity Collection
- Cloning a Collection

Sybase ASE Capacity Collection Content and Options

When creating or editing a Capacity Analyst collection, you specify the statistics you want in that collection and specify additional notification and update options. The table below summarizes the available statistics and options on the tabs/panels of collection wizards and editors for this DBMS.

Content/Option Category	Editor Tab or Wizard panel	Description or specific DBMS feature support
Identification/Notification	**General**	Lets you provide a name and description for the collection and optionally, provide an email address or net send name for notifications.
Types of statistics collected	**Selection Types**	Lets you select the types of content and options you want to include in the collection. You provide specifics in subsequent panels/tabs.
Database/storage structure space statistics	**Storage Structures**	Lets you use Global Space Summary statistics for the datasource or space and usage statistics for specific database devices and databases.
Specific database object types for which to collect statistics	**DB Objects**	Lets you include statistics on indexes and tables.
Wait-event-based statistics	**Wait Events**	Wait event statistics are only available if MDI performance views have been created.
Performance-based statistics	**Performance Statistics**	Lets you include Sybase ASE Performance Statistics in a collection.
Update and rollup options	**Statistics**	Option to update statistics with additional options to update indexes or partitions
Automatic purging of old data	**AutoPurge**	Offers the option to automatically purge data older than a specified age, each time the collection is updated.

For information on how to create and edit collections, see:

- Creating a New Collection
- Editing a Capacity Collection
- Cloning a Collection

Creating a New Collection

Capacity Analyst has a WIzard that walks you through specifying the statistics and options in a collection. Before creating a new collection, you should be familiar with the material in Content and Options in a Collection.

To create a new collection

1 Open the Capacity Analyst. See Opening the Capacity Analyst (DBArtisan).

2 On the Capacity Analyst toolbar click the **New Collection Wizard** button.

::

The Capacity Collection Wizard opens.

3 Provide the information required on each panel of the Wizard. For more information on these options, see the relevant DBMS-specific topic under Content and Options in a Collection.

4 When done click **Finish**.

Capacity Analyst offers the option to schedule updates for the statistics in this collection.

5 If you decide to schedule statistics updates, use the Windows Task Scheduler to provide update details.

Capacity Analyst adds the new collection to the **Capacity Collection** list.

See the following tasks for information on tasks you can perform after creating a collection:

- Editing a Capacity Collection
- Updating Statistics in Collections for use in Reports and Charts (DBArtisan)
- Viewing Charts and Reports on Collections (DBArtisan)

Editing a Capacity Collection

Because a collection stores rolled up statistics, all gathered and calculated over a specific interval, editing statistics in a collection could corrupt the data in a collection. Therefore, you cannot modify the statistics in a collection or the devices, databases or database objects that statistics are gathered against. You can only modify the basic identifiers for the collection and the refresh and autopurge options.

To edit an existing collection

1 Open the Capacity Analyst. See Opening the Capacity Analyst (DBArtisan).

2 Select the collection to be edited and then on the Capacity Analyst toolbar click the **Open Collection** button.

🔍::

3 On the **General** tab, edit the name, description, email address, or net send name.

4 On the **Statistics** tab, enable or disable update of statistics when you save your changes.

5 On the **AutoPurge** tab, enable or disable autopurge of older data when you save your changes.

6 Click **OK**.

Cloning a Collection

The Capacity Analyst lets you create a new collection from an existing collection. This lets you build a new collection starting from properties defined in the existing collection.

To make a copy of an existing collection

1 Open the Capacity Analyst. See Opening the Capacity Analyst (DBArtisan).

2 Select the collection to be cloned and then on the Capacity Analyst toolbar click the **Clone Collection** button.

::

Capacity Analyst opens the Capacity Analyst Collection Wizard. With the exception of the collection name, all statistics and options specified for the new collection are identical to those for the existing collection.

3 Minimally, provide a name for the new collection. Complete each panel of the wizard, changing settings as required. For more information on these options, see the relevant DBMS-specific topic under Content and Options in a Collection.

4 When done click **Finish**.

Capacity Analyst offers the option to schedule updates for the statistics in this collection.

5 If you decide to schedule statistics updates, use the Windows Task Scheduler to provide update details.

Capacity Analyst adds the new collection to the **Capacity Collection** list.

See the following tasks for information on tasks you can perform after cloning a collection:

- Editing a Capacity Collection
- Updating Statistics in Collections for use in Reports and Charts (DBArtisan)
- Viewing Charts and Reports on Collections (DBArtisan)

Deleting a Collection

If you no longer use the reports or analysis associated with a collection, Capacity Analyst lets you delete the collection.

To delete a collection

1 Open the Capacity Analyst. See Opening the Capacity Analyst (DBArtisan).

2 Select a collection and then on the Capacity Analyst toolbar click the **Delete** button.

::

Capacity Analyst prompts you to verify that you want to delete the collection.

3 Click **Yes**.

Updating Statistics in Collections for use in Reports and Charts (DBArtisan)

Once you have installed and set up Capacity Analyst on a server, raw statistics are gathered and stored in the repository on that server. In order to view up-to-date charts and reports, you must execute a collection. Executing the collection calculates totals and performs all rollups for that collection in preparation for reporting.

There are two methods of updating statistics in a collection:

- Manually Updating Collections, useful prior to running an ad hoc report
- Scheduling Updates of Collections to run regularly, useful when reporting is performed on a regular basis

Manually Updating Collections

You can manually execute a statistics collection. This calculates totals and statistical indicators such as averages used for reporting for that collection.

To manually execute a collection

1 Open the Capacity Analyst. See Opening the Capacity Analyst (DBArtisan).

2 Select a collection and then on the Capacity Analyst toolbar click the **Execute** button.

▶

When complete, the **Last Run** field reflects the time that the collection was updated.

After executing a collection, you can run reports or view charts for that collection, that are up-to-date as of the time that you executed the collection. For details, see Viewing Charts and Reports on Collections (DBArtisan).

Related topics

- Creating and Editing Collections (DBArtisan)

Scheduling Updates of Collections

If you run reports or want to regularly archive statistics for intervals, you can schedule execution of statistics collections. This updates the totals, rollups, and statistical indicators such as averages, on a regular basis. You can schedule collection updates to occur regularly such as daily weekly, or monthly, or tie execution to events such as system startup, logon, or during idle time.

The following topics provide details on the scheduling tasks you can perform against a collection:

- Creating an Update Collection Schedule
- Editing an Update Collection Schedule
- Deleting an Update Collection Schedule

Creating an Update Collection Schedule

You can create an update collection schedule to have statistics in that collection updated at regular intervals or have updates triggered by events such as system startup or logon.

In setting up your schedule, keep the following points in mind:

- Some statistical collections depend on the underlying dynamics of the database objects being collected. For example, storage structure space metrics, table and index statistics, etc., are oftentimes best collected only once a day. Other statistics, such as wait events and global database metrics can be collected on a much more granular level (every hour, etc.)
- If you routinely stop and start your database at a certain time, you may want to ensure that your collection job is not scheduled to run during its downtime. Also realize that stopping and starting a database resets the counters used for wait events and global database metrics.
- If you use the Microsoft scheduler on your machine, remember to not turn your machine off during the times your collection is scheduled to run or the job will not run.

Tip: You can add, edit, or remove any schedule for any collection at a later time by using the management functions supplied in the Capacity Collection Manager interface.

To schedule updates for a collection

1 Open the Capacity Analyst. See Opening the Capacity Analyst (DBArtisan).

2 Select a Capacity Collection that currently does not have an associated execution schedule.

3 Click the Schedule button.

Capacity Analyst opens the Job Scheduler.

4 Use the Job Scheduler to specify the frequency of the report or the event that triggers the update, and other details of the schedule.

5 When finished, click **OK**.

Related topics

- Editing an Update Collection Schedule
- Viewing Charts and Reports on Collections (DBArtisan)

Editing an Update Collection Schedule

After creating an update schedule for a statistics collection, you can revise the schedule. You may wish to change the frequency of updates or the event which triggers an update, or more specific details of the update schedule.

To modify the update schedule for a collection

1 Open the Capacity Analyst. See Opening the Capacity Analyst (DBArtisan).

2 Select a Capacity Collection that currently has an associated execution schedule.

3 Click the Edit Schedule button.

::

Capacity Analyst opens the Job Scheduler.

4 Use the Job Scheduler to change the frequency of the report or the event that triggers the update, and other details of the schedule.

5 When finished, click **OK**.

Related topics

- Scheduling Updates of Collections
- For information on modifying the statistics in a collection, see Editing a Capacity Collection
- Viewing Charts and Reports on Collections (DBArtisan)

Deleting an Update Collection Schedule

If you no longer want statistics in a collection to be calculated on a regular basis or if you want to create a new, completely reworked schedule, you can delete the existing schedule for a collection.

To unschedule a Capacity Collection

1 Open the Capacity Analyst. See Opening the Capacity Analyst (DBArtisan).

2 Select a Capacity Collection.

3 On the Capacity Analyst toolbar, click the Unschedule button.

::

Capacity Analyst unschedules the Capacity Collection.

Related topics

- Scheduling Updates of Collections
- Creating an Update Collection Schedule

Purging Old Data in a Collection

Policy or practicality may dictate that the metrics in your analysis and reports not include data older than a specified date. Capacity Analyst lets you manually purge data older than a specified number of months or years.

Note: In addition to manually purging old data, when creating or editing a collection, you can set up the collection to automatically purge data older than a specified date on every update. For details, see Creating and Editing Collections (DBArtisan).

To manually purge collection data older than a specified number of months or years

1 Open the Capacity Analyst. See Opening the Capacity Analyst (DBArtisan).

2 Select a Capacity Collection.

3 On the toolbar, click the Purge button.

::

The **Perform Manual Data Purge** dialog opens.

4 Use the controls in the **Purge data older than** area to provide a specific number of months or years.

5 Click **Execute** and when prompted to confirm, click **Yes**.

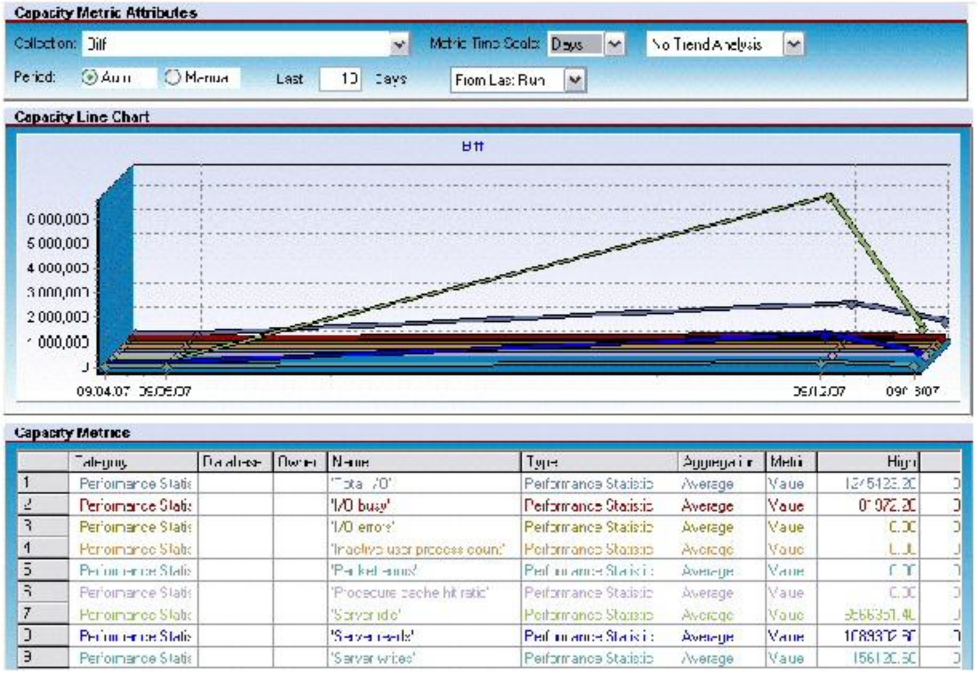
Viewing Charts and Reports on Collections (DBArtisan)

The preliminary steps in setting up Capacity Analyst are Creating and Editing Collections (DBArtisan) and Updating Statistics in Collections for use in Reports and Charts (DBArtisan). Once your collections are defined and the statistics updated, Capacity Analysts lets you view forecasting/trends charts and reports on your collections. Key options are:

- Using the Collection Analysis Viewer - lets you view charted representations of collection metrics as a function of time.
- Working with Capacity Analyst Reports - lets you work with predefined reports that focus on growth of databases, storage mechanisms, and specific database objects, as well as failure prediction and fragmentation.

Using the Collection Analysis Viewer

The Collection Analysis Viewer lets you perform ad hoc analysis of statistics in a collection. It presents the metrics in two formats: a graphical representation that shows values over time and a tabular format that includes metrics such as average, maximum, and minimum values.



Collection: Jill

Metric Time Scale: Days

No Trend Analysis

Period: ● Auto ○ Manual

Last 10 Days

From Last Run

	Category	Database	Owner	Name	Type	Aggregation	Metric	High
1	Performance Statis			'Total I/O'	Performance Statistic	Average	Value	[illegible]
2	Performance Statis			'I/O busy'	Performance Statistic	Average	Value	[illegible]
3	Performance Statis			'I/O errors'	Performance Statistic	Average	Value	0.00
4	Performance Statis			'Inactive user process count'	Performance Statistic	Average	Value	[illegible]
5	Performance Statis			'Packet errors'	Performance Statistic	Average	Value	[illegible]
6	Performance Statis			'Procedure cache hit ratio'	Performance Statistic	Average	Value	0.00
7	Performance Statis			'Server idle'	Performance Statistic	Average	Value	[illegible]
8	Performance Statis			'Server reads'	Performance Statistic	Average	Value	[illegible]
9	Performance Statis			'Server writes'	Performance Statistic	Average	Value	[illegible]

The Collection Analysis Viewer lets you:

- View historical trends for a metric and extrapolate into the future
- Compare and contrast multiple metrics
- Vary the time scale for fine-grained or coarse-grained views

The first time that you open the Collection Analysis Viewer on a particular collection, you are prompted to select the metrics you want to view for that collection. Subsequently, when using the

Collection Analysis Viewer against that collection, you can add metrics to and delete metrics from the collection.

To open the Collection Analysis Viewer on a collection

1 Open the Capacity Analyst. See Opening the Capacity Analyst (DBArtisan).

2 Select a Capacity Collection. For more information, see Creating and Editing Collections (DBArtisan).

3 On the toolbar, click the **Collection Analysis Viewer** button.

::

Note: If this is the first time you have opened this particular collection, you are prompted to specify the statistics to display. For help with this step, see Adding Metrics to the Collection Analysis Viewer for a Collection.

Capacity Analyst opens the Collection Analysis View or the selected collection.

Viewing options available when working with the Collection Analysis Viewer include:

- Using the Delete Metric button to remove a statistic from the displayed metrics for a collection
- Using the zoom buttons on the Collection Analysis Viewer toolbar to zoom in and out on the report
- Using the refresh button on the Collection Analysis Viewer toolbar to update displayed values
- Using the **Collection** dropdown to open the Collection Analysis Viewer on a different collection
- Using the **Metric Time Scale** dropdown to choose between displayed time scales of hours, days, months, or years
- Using the trending and value dropdowns to extrapolate metric values out to specified values or specific dates
- Using the **Period** controls to change the date/time range displayed
- Viewing values for metrics at specific locations on the graph by floating the cursor over specific locations on the curve

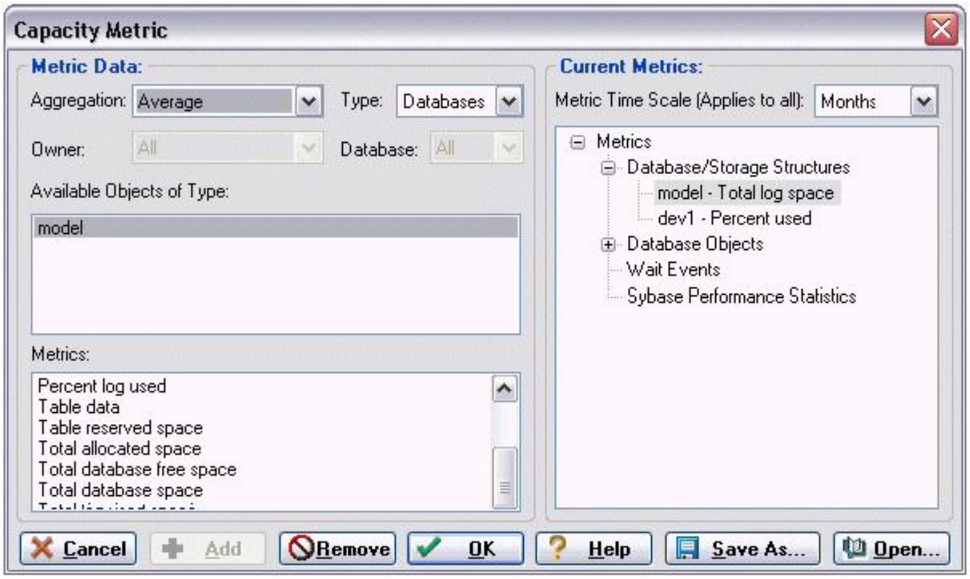
Adding Metrics to the Collection Analysis Viewer for a Collection

There are two situations in which you can add metrics viewed using the Collection Analysis Viewer:

- The first time that you open the Collection Analysis Viewer for a given collection
- When you want to manually add metrics by clicking the Collection Analysis Viewer toolbar's Add Metric button:

✚ ::

In both cases, Capacity Analyst opens a **Capacity Metric** dialog to let you add metrics.



Capacity Metric

Metric Data:

Aggregation: Average

Type: Databases

Owner: All

Database: All

Available Objects of Type:
model

Metrics:
Percent log used
Table data
Table reserved space
Total allocated space
Total database free space
Total database space

Current Metrics:

Metric Time Scale (Applies to all): Months

Metrics
  Database/Storage Structures
    model - Total log space
    dev1 - Percent used
  Database Objects
  Wait Events
  Sybase Performance Statistics

For each metric you want add to the Capacity Analyst Analysis Viewer

1 From the **Aggregation** dropdown, select an aggregation type: **Average**, **Maximum**, or **Minimum**.

2 From the **Type** dropdown, select the type of metric you want to add.

Note: Certain Metric types allow you to further qualify using the **Owner** and **Database** dropdowns.

3 From the **Available Object of Type** list, choose a specific object or statistics type.

The **Metrics** list is updated with the metrics available for the **Type** and **Available Objects of Type** selections you made.

4 Select a metric from the **Metrics** list and click **Add**.

Capacity Analyst adds the metric to the **Current Metrics** list.

5 To add additional metrics, repeats step1 through step4.

6 When finished adding metrics for the Capacity Analyst Collection Analysis Viewer, click **OK**.

Capacity Analyst updates the Collection Analyst Viewer with the new metrics.

Working with Capacity Analyst Reports

Capacity Analyst provides two report options. For details, see the following topics:

- Running a Capacity Collection Summary Report - lets you view a summary of the metrics and options currently defined for a collection.
- Running an Analysis Report - lets you view one of several predefined reports focusing on growth trends

Running a Capacity Collection Summary Report

You can produce a high-level summary of the content and options associated with a collection. It includes:

- Basic identification information such as the name, associated datasource, and owner of the collection
- A listing of the options, refresh and auto-purge for example, chosen for that collection
- A summary of the types of statistics available for use in the collection
- A detailed listing of the objects for which statistics are collected

Note: For a detailed listing of the specific statistics and options available for the DBMS you are working with, see the DBMS-specific heading under Content and Options in a Collection.

To run a Capacity Collection Summary Report

1 Open the Capacity Analyst. See Opening the Capacity Analyst (DBArtisan).

2 Select a Capacity Collection. For more information, see Creating and Editing Collections (DBArtisan).

3 On the Capacity Analyst toolbar, click the Report button.

Capacity Analyst opens the **Select Report Type** dialog.

4 Select **Capacity Collection Summary Report** and click OK.

Capacity Analyst opens the summary report in a new tab.

Running an Analysis Report

Capacity Analyst lets you run a set of predefined reports against a collection. The reports and their availability by DBMS are:

- Database growth trends (Sybase ASE, SQL Server, DB2)

- Tablespace failure prediction (Oracle)
- Tablespace growth trends (DB2, Oracle)
- Device growth trends (Sybase ASE)
- Filegroup growth trends (SQL Server)
- File growth trends (SQL Server)
- Object growth trends (Sybase ASE, SQL Server, DB2, Oracle)
- Object extent trends (Sybase ASE, SQL Server)
- Object extent failure prediction
- Database backup growth trends (SQL Server)
- ‘N’ Fastest growing databases (Sybase ASE, SQL Server, DB2)
- ‘N’ Fastest growing tablespaces (Oracle)
- ‘N’ Fastest growing objects (Sybase ASE, SQL Server, DB2, Oracle)
- Table rows trend (Sybase ASE, SQL Server, DB2, Oracle)
- Table chained/migrated rows trend (Oracle)
- Table forwarded records trend (Sybase ASE, SQL Server)
- Index level trend (Oracle)
- Object fragmentation trends (Sybase ASE, SQL Server, DB2)
- Tablespace fragmentation trend (Oracle)
- Database failure prediction (Sybase ASE, SQL Server)
- DMS tablespace failure trends (Oracle)

To run an analysis report

1 Open the Capacity Analyst. See Opening the Capacity Analyst (DBArtisan).

2 Select a Capacity Collection. For more information, see Creating and Editing Collections (DBArtisan).

3 On the Capacity Analyst toolbar, click the Report button.

::

Capacity Analyst opens the **Select Report Type** dialog.

4 Select Analysis Report and click OK.

Capacity Analyst opens the Capacity Analyst Report Wizard.

5 Use the following table as a guide to understanding the settings and options for this wizard

Panel	Tasks and settings	Description
Panel 1	**Do you wish to create a new analysis report or 'playback' a previously saved report definition?**	As the last step in generating a report, you are offered the option to save the report definition, for the purpose of running the report again. If you have already saved report definitions, you can use this panel to enable the **Previously Saved Report** option and select the existing report definition. Otherwise, to run an ad hoc report or start a new report definition, you can use the **New Analysis Report** option.
Panel 2	**What report would you like to create?**	Lets you select one of the predefined report types listed above.
	Enter the time period sample to use for the report	**Auto Entry** lets you specify a number of days, months or years while **Manual Entry** lets you provide specific start and end dates.
	Enter the time rollup to use for the report.	Lets your report display statistics rolled up by day, month, or year.
	Enter the date to use for forward predictions.	Lets you enter a future end date that the Capacity Analyst uses to forecast.
	Select options for the report.	The options offered differ by the type of report you selected and the DBMS that DBArtisan is working with.
Panel 3	**Include objects for report**.	Expand or collapse databases or owner lists to display specific object types and use the arrow buttons to move selected objects between the **Available** and **Selected** lists.
Panel 4	**Data will be presented in text/grid format.**	Lets you specify file name and path where the report will be stored and provide a title for the report.
	Save this report definition?	Lets you save the report definition, specifying the file name and path for the definition.

6 When complete, click Schedule or Execute to run the report.

When complete, the report opens in a new DBArtisan tab.

Note: When viewing the report in DBArtisan, you can use the toolbar's **Refresh** button to update values displayed.

A static, HTML version is also saved to with the file name and path you specified in the **Report File Name** field.

Performance Analyst (DBArtisan)

Performance Analyst is a robust client-side monitoring product that presents a complete picture of a database's performance. Performance Analyst is a quick, lightweight tool that lets you investigate performance bottlenecks.

The table below describes the major sections of Help.

Section	Description
Welcome to Performance Analyst (DBArtisan)	Provides information about installing, opening, and licensing Performance Analyst.
Performance Analyst Application Basics (DBArtisan)	Provides conceptual information about Performance Analyst’s product design and navigation.
Using Performance Analyst (DBArtisan)	Shows how to use each component of the application. Using Performance Analyst breaks each process and procedure into simple step-by-step instructions.
Performance Analyst for Oracle Tutorial (DBArtisan)	Helps you get started with Embarcadero Performance Analyst. Provides the foundation you need to explore the many features and benefits of Performance Analyst.
Oracle Performance Analyst Statistics (DBArtisan)	Performance Analyst includes expert help for all supported database platforms.
Performance Analyst for Microsoft SQL Server Tutorial (DBArtisan)	Helps you get started with Embarcadero Performance Analyst. Provides the foundation you need to explore the many features and benefits of Performance Analyst.
Microsoft SQL Server Performance Analyst Statistics (DBArtisan)	Performance Analyst includes expert help for all supported database platforms.
Performance Analyst for Sybase ASE Tutorial (DBArtisan)	Helps you get started with Embarcadero Performance Analyst. Provides the foundation you need to explore the many features and benefits of Performance Analyst.
Sybase ASE Performance Analyst Statistics (DBArtisan)	Performance Analyst includes expert help for all supported database platforms
Performance Analyst for IBM DB2 for Linux, Unix and Windows Tutorial (DBArtisan)	Helps you get started with Embarcadero Performance Analyst. Provides the foundation you need to explore the many features and benefits of Performance Analyst.
IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan)	Performance Analyst includes expert help for all supported database platforms

Welcome to Performance Analyst (DBArtisan)

Performance Analyst is a client-side only visual database monitor that displays current database performance data, including key performance metrics, bottleneck analysis, and user-configurable thresholds.

Running inside the Embarcadero DBArtisan console, or stand-alone in a Web browser, Performance Analyst lets you identify performance problems that threaten database availability or response times.

Supported DBMS

Embarcadero Technologies currently offers Performance Analyst for Oracle 9i - 10g. Performance Analyst does not support Oracle Clients 7.3.4, 8.1.7, and 9.2. Performance Analyst also supports Microsoft SQL Server and DB2.

For more information, see:

- What You Should Know Before Installing Performance Analyst (DBArtisan)
- Operating System Prerequisites
- Opening the Performance Analyst
- Licensing Performance Analyst

What You Should Know Before Installing Performance Analyst (DBArtisan)

When you install the Analyst, several Embarcadero-generated stored objects are pushed onto the database server:

These objects are as follows:

ORACLE PROCEDURES

Procedures

ALL_TABLESPACES
ALL_USERS
HOTTESTDBFILES
INSTANCESTARTUP
IO_DD_ACCESSPATTERNDETAIL
IO_DD_ACTIVEROLLBACKDETAILS
IO_DD_ARCHIVELOGMODE
IO_DD_CURRENTDISKSORTS
IO_DD_DAILYARCHIVELOGSUMMARY
IO_DD_DATAFILEIO
IO_DD_DBWRDETAIL
IO_DD_ESTPHYSICALIOBYTABLE
IO_DD_JOBQUEUESUMMARY
IO_DD_LGWRDETAIL
IO_DD_PROBLEMJOBS
IO_DD_REDOWASTAGE
IO_DD_REDOWASTAGE2
IO_DD_ROLLBACKIO
IO_DD_RUNNINGJOBSDETAIL
IO_DD_SESSIONROLLBACKACTIVITY
IO_DD_SESSIONSORTDETAIL
IO_DD_TABLESPACEIO
MEMORY_DD_BUFFERCACHERATIO
MEMORY_DD_BUFFERCACHEUTIL
MEMORY_DD_BUFFERPOOLHITRAT
MEMORY_DD_BUFFERPOOLMEMCONF
MEMORY_DD_BUFFERPOOLOBJASS
MEMORY_DD_CACHEDBLOCKSBYTS
MEMORY_DD_CACHEDTABLES
MEMORY_DD_DATADICTDETAILS
MEMORY_DD_LATCHDETAILS
MEMORY_DD_LATCHRATIOS
MEMORY_DD_LIBCACHEHITRATIO
MEMORY_DD_LIBCACHEPERF
MEMORY_DD_PGADETAIL
MEMORY_DD_PGAOVERVIEW
MEMORY_DD_PGAWORKAREA
MEMORY_DD_PINNEDOBJECTS
MEMORY_DD_SHAREDPOOLFRAG
MEMORY_DD_SHAREDPOOLUSAGE
MEMORY_DD_SHAREDPOOLUTIL
NOT_WAITING_SESSIONS_COUNT
OBJECT_DD_GLOBALOBJSUM
OBJECT_DD_HOTCODE
OBJECT_DD_HOTTABLES
OBJECT_DD_INDEXES
OBJECT_DD_INVALIDOBJDET
OBJECT_DD_INVALIDOBJSUM
OBJECT_DD_REORGANALYSIS
OBJECT_DD_TABLESINSAMETS
OBJECT_DD_USEROBJSUM
OBJECTBUFFERPOOLPLACEMENT
RAC_LOCKS_CONVERT
RAC_LOCKS_DLM
RAC_LOCKS_LATCH
RAC_LOCKS_PCM
RAC_MEMORY_ANALYSIS
RAC_MEMORY_DATACACHE
RAC_MEMORY_KEYRATIO
RAC_MEMORY_LIBRARYCACHE
RAC_OVERVIEW_CONFLICTS
RAC_OVERVIEW_INSTANCE
RAC_OVERVIEW_RATIOS
RAC_OVERVIEW_WAITS
RAC_OVERVIEW_WORKLOAD
RAC_PING_DATAFILE
RAC_PING_INSTANCE
RAC_PING_OBJECT
RAC_SESSIONS_BLOCKS
RAC_SESSIONS_TOP
RAC_SESSIONS_USER
RAC_SESSIONS_USERWAITS

RESPONSETIME_CURRENT_RESTIME
RESPONSETIME_CURRENTWORKLOAD
RESPONSETIME_HISTLISTEVENT
RESPONSETIME_HISTORY
RESPONSETIME_HISTORYFILE
RESPONSETIME_HISTORYOBJECT
RESPONSETIME_HISTORYSESSION
RESPONSETIME_HISTORYSQL
RESPONSETIME_HISTORYSUMMARY
RESPONSETIME_LASTHOUR
RESPONSETIME_LASTHOURDETAIL
RESPONSETIME_LISTMETRIC
RESPONSETIME_SESBOTTLENECK
RESPONSETIME_SESBOTTLENECKDTL
RESPONSETIME_SESBOTTLENECKHIST
RESPONSETIME_SESSIONTIME
RESPONSETIME_SYSBOTTLENECKHIS
RESPONSETIME_SYSTEMBOTTLENECK
RESPONSETIME_SYSTEMTIME
RESPONSETIME_WAITHISTOGRAM
RESPONSETIME_WAITHISTOLIST
RETRIEVEHOMEPAGE
RETRIEVEHOMERATIOS
RETRIEVEIOPAGE
RETRIEVEMEMORYPAGE
RETRIEVEMEMORYRATIOS
RETRIEVEOBJECTPAGE
RETRIEVEOBJECTRATIOS
RETRIEVESPACEPAGE
RETRIEVEUSERPAGE
RETRIEVEUSERRATIOS
SESSION_DD_ALLLOCKS
SESSION_DD_BLKLOCKS
SESSION_DD_CRNTWAITS
SESSION_DD_CURRENTSQL
SESSION_DD_CURRENTSQLSTAT
SESSION_DD_HISTWAITS
SESSION_DD_OPENCURSORS
SESSION_DD_OTHERSQL
SESSION_DD_OVERVIEW
SESSION_DD_OVERVIEWSTAT
SESSION_DD_TRANSACTIONS
SESSION_LAST_TEN_WAIT
SESSION_TIME_SUMMARY
SPACE_DD_ACCOBJECTGRID
SPACE_DD_DATAFILEDETAIL
SPACE_DD_DATAFILEFRAG
SPACE_DD_EXTENTANALYSIS
SPACE_DD_SPACEBYOBJECT
SPACE_DD_SPACEBYUSER
SPACE_DD_TABLESPACEFRAG
SPACE_DD_TABLESPACEGROWTH
SPACE_DD_TABLESPACEMAP
SPACE_DD_TABLESPACEOBJSUM
SPACE_DD_TOPUSERS
SPACEANALYSIS
SQLANALYSIS_HOME
SQLANALYSIS_IO
SQLANALYSIS_MEMORY
SQLANALYSIS_OBJECT
SQLANALYSIS_SPACE
SQLANALYSIS_USER
TABLEHIGHWATERMARKPROBLEMCOUNT
TOP_SYSTEM_BOTTLENECKS_10G
TOPCPUPROCESS
TOPDBFILEWAITS
TOPDISKSORTPROCESS
TOPFRAGMENTEDDATAFILES
TOPIOHOGS
TOPIOPROCESS
TOPLARGESCANPROCESS
TOPLATCHMISSES

TOPLOGFILEWAITS
TOPMEMORYHOGS
TOPMEMORYPROCESS
TOPPARSEPROCESS
TOPSESSIONBOTTLENECKS
TOPSESSIONS
TOPSESSIONS_10G
TOPSPACEHOGS
TOPSQL_DETAILS
TOPSQL_RECENT_HISTORY
TOPSQL_SQLTEXT
TOPSQL_STATS
TOPSQL_SUMMARY_DETAILS
TOPSYSTEMBOTTLENECKS
USERACTIVITYANALYSIS
USERS_DD_ALL_LOCKS
USERS_DD_BLK_LOCKS
USERS_DD_BY_TIME_WAITED
USERS_DD_BY_TOTAL_WAITS
USERS_DD_CRNTWAITS
USERS_DD_HISTWAITS
USERS_DD_HISTWAITS_10G
USERS_DD_SYSTEMWAITS_10G
VERSION
WAITING_SESSIONS_COUNT
WORSTSCHEMAFORCHAINEDROWTABLES
WORSTSCHEMAFOREXTENTPROBLEMS
WORSTSCHEMAFORINVALIDOBJECTS

SQL SERVER

Procedures

sp_embt_pa_all_database_enumerrorlog
sp_embt_pa_all_database_list
sp_embt_pa_current_sq
sp_embt_pa_database_analysis
sp_embt_pa_database_bottleneck
sp_embt_pa_database_drill_backups_db
sp_embt_pa_database_drill_backups_detail
sp_embt_pa_database_drill_backups_log
sp_embt_pa_database_drill_config_detail
sp_embt_pa_database_drill_config_server
sp_embt_pa_database_drill_error_log_content
sp_embt_pa_database_drill_error_log_detail
sp_embt_pa_database_drill_error_log_volume
sp_embt_pa_database_drill_obj_index_det
sp_embt_pa_database_drill_obj_summ
sp_embt_pa_database_drill_obj_table_det
sp_embt_pa_database_drill_overview
sp_embt_pa_database_drill_replication_agent
sp_embt_pa_database_drill_replication_thruput
sp_embt_pa_database_drill_sqlagent_alrtdet
sp_embt_pa_database_drill_sqlagent_alrtsum
sp_embt_pa_database_drill_sqlagent_jobdet
sp_embt_pa_database_drill_sqlagent_jobsum
sp_embt_pa_database_drill_sqlagent_status
sp_embt_pa_database_errorlog
sp_embt_pa_db_requiring_backup
sp_embt_pa_db_workload_analysis
sp_embt_pa_home_active_processes
sp_embt_pa_home_activity_metrics
sp_embt_pa_home_bottleneck_waits
sp_embt_pa_home_bottlenecks
sp_embt_pa_home_storage_database_count
sp_embt_pa_home_storage_totals
sp_embt_pa_instance_startup
sp_embt_pa_io_access_ratios
sp_embt_pa_io_activity_metrics
sp_embt_pa_io_bottlenecks
sp_embt_pa_io_drill_database
sp_embt_pa_io_drill_file
sp_embt_pa_io_drill_system_awe
sp_embt_pa_io_drill_system_io
sp_embt_pa_io_drill_system_physical
sp_embt_pa_io_drill_system_space
sp_embt_pa_io_drill_user_lock
sp_embt_pa_io_drill_user_obj
sp_embt_pa_io_drill_user_sql
sp_embt_pa_io_drill_user_temp
sp_embt_pa_io_get_file_stats
sp_embt_pa_io_hottest_files
sp_embt_pa_io_ratios
sp_embt_pa_io_top_processes
sp_embt_pa_memory_analysis
sp_embt_pa_memory_drill_bcache
sp_embt_pa_memory_drill_buffer
sp_embt_pa_memory_drill_buffer_page_activity
sp_embt_pa_memory_drill_log_details
sp_embt_pa_memory_drill_log_instances
sp_embt_pa_memory_drill_proc_detail
sp_embt_pa_memory_drill_proc_summary
sp_embt_pa_memory_latch_stats

sp_embt_pa_memory_ratio_mem	sp_embt_pa_memory_sql_analysis	sp_embt_pa_memory_top_latch_waits
sp_embt_pa_memory_workload_analysis	sp_embt_pa_space_bottlenecks	sp_embt_pa_space_drill_filegroups
sp_embt_pa_space_drill_files	sp_embt_pa_space_drill_object_summary	sp_embt_pa_space_drill_objects_index_det
sp_embt_pa_space_drill_objects_table_det	sp_embt_pa_space_drill_server_by_db	sp_embt_pa_space_drill_server_summary
sp_embt_pa_space_drill_vlogs	sp_embt_pa_space_free_disk_space	sp_embt_pa_space_main
sp_embt_pa_space_used_disk_space	sp_embt_pa_top_sql_database	sp_embt_pa_top_sql_details
sp_embt_pa_top_sql_home	sp_embt_pa_top_sql_query	sp_embt_pa_top_sql_refresh
sp_embt_pa_top_sql_space	sp_embt_pa_top_sql_start	sp_embt_pa_top_sql_stop
sp_embt_pa_top_sql_trace	sp_embt_pa_user_bottleneck_activities	sp_embt_pa_user_bottleneck_sessions
sp_embt_pa_user_drill_locks_by_db	sp_embt_pa_user_drill_locks_by_locktype	sp_embt_pa_user_drill_locks_detail
sp_embt_pa_user_drill_session_waits_by_time	sp_embt_pa_user_drill_session_waits_by_type	sp_embt_pa_user_drill_session_waits_detail
sp_embt_pa_user_drill_system_waits_by_time	sp_embt_pa_user_drill_system_waits_by_total_waits	sp_embt_pa_user_drill_system_waits_detail
sp_embt_pa_user_drill_top_sessions	sp_embt_pa_user_drill_top_sessions_detail	sp_embt_pa_user_ratio_login
sp_embt_pa_user_ratio_process	sp_embt_pa_user_workload	sp_embt_pa_version

Sybase ASE

Procedures

sp_embt_pa_actsql_drill
sp_embt_pa_actsql_drill_det
sp_embt_pa_current_sql
sp_embt_pa_db_backup
sp_embt_pa_db_bnecks
sp_embt_pa_db_drill_cb_filter
sp_embt_pa_db_drill_cb_idx
sp_embt_pa_db_drill_cb_tbl
sp_embt_pa_db_drill_cnfg
sp_embt_pa_db_drill_dbs
sp_embt_pa_db_drill_engines
sp_embt_pa_db_drill_obj_filter
sp_embt_pa_db_drill_obj_idx
sp_embt_pa_db_drill_obj_suspct
sp_embt_pa_db_drill_obj_tbl
sp_embt_pa_db_drill_opt
sp_embt_pa_db_drill_overview
sp_embt_pa_db_drill_sus_filter
sp_embt_pa_db_engines
sp_embt_pa_db_max_engines=
sp_embt_pa_db_metadata
sp_embt_pa_db_obj_space
sp_embt_pa_db_srv_bnecks
sp_embt_pa_db_workload
sp_embt_pa_dbspace
sp_embt_pa_home_act_metrics
sp_embt_pa_home_bottlenecks
sp_embt_pa_home_storage_tot
sp_embt_pa_instance_startup
sp_embt_pa_io_anlysis
sp_embt_pa_io_bottlenecks
sp_embt_pa_io_drill_dbs
sp_embt_pa_io_drill_device_rt
sp_embt_pa_io_drill_device_sum
sp_embt_pa_io_drill_device_typ
sp_embt_pa_io_drill_engine
sp_embt_pa_io_drill_net_eng
sp_embt_pa_io_drill_net_ses
sp_embt_pa_io_drill_object
sp_embt_pa_io_hottest_devices
sp_embt_pa_io_key_system
sp_embt_pa_io_network
sp_embt_pa_io_trans_log
sp_embt_pa_io_workload
sp_embt_pa_mem_activity
sp_embt_pa_mem_analysis
sp_embt_pa_mem_bottleneck
sp_embt_pa_mem_eff_ratios
sp_embt_pa_mem_mdcache
sp_embt_pa_mem_workload
sp_embt_pa_memory_drill_dc_act
sp_embt_pa_memory_drill_dc_cfg
sp_embt_pa_memory_drill_dc_eff
sp_embt_pa_memory_drill_dc_obj
sp_embt_pa_memory_drill_pc_act
sp_embt_pa_memory_drill_pc_det
sp_embt_pa_memory_drill_pc_use
sp_embt_pa_monitor
sp_embt_pa_space_bottlenecks
sp_embt_pa_space_db_analysis
sp_embt_pa_space_db_dev_cnt
sp_embt_pa_space_dev_analysis
sp_embt_pa_space_drill_db_sum
sp_embt_pa_space_drill_dev_det
sp_embt_pa_space_drill_dev_sum
sp_embt_pa_space_drill_indexes
sp_embt_pa_space_drill_tables
sp_embt_pa_space_indexs_filte
sp_embt_pa_space_tables_filter
sp_embt_pa_user_activity
sp_embt_pa_user_bottlenecks
sp_embt_pa_user_drill_lcks_db
sp_embt_pa_user_drill_lcks_tp
sp_embt_pa_user_drill_locks
sp_embt_pa_user_drill_seswait
sp_embt_pa_user_drill_syswait
sp_embt_pa_user_drill_trans
sp_embt_pa_user_drill_ts
sp_embt_pa_user_drill_ts_procs
sp_embt_pa_user_index_scans
sp_embt_pa_user_process
sp_embt_pa_user_transactions
sp_embt_pa_user_workload
sp_embt_pa_userdet_io

sp_embt_pa_userdet_locks
sp_embt_pa_userdet_objects
sp_embt_pa_userdet_sql
sp_embt_pa_userdet_waits
sp_embt_pa_version

Operating System Prerequisites (DBArtisan)

Performance Analyst requires prerequisites for performance monitoring Windows 2000/XP and UNIX and Linux. The connectivity requirements for the various servers are minimum requirements. The other requirements are optional.

Windows 2000/XP Prerequisites

Connectivity

Performance Analyst requires the following minimum conductivity prerequisites:

- Remote Registry Service should be running on the server if Performance Analyst is connecting to a remote machine. For local machine this does not matter.
- Adequate permissions to connect to the remote computer's registry.

Other Requirements

Performance Analyst requires the following optional prerequisites:

- Diskperf should be running on the server.

Note: You can start diskperf by executing diskperf -y from the command line.

UNIX and Linux Prerequisites

Connectivity

Performance Analyst requires the following minimum conductivity prerequisites:

- Ability to connect to the server and execute commands on the server using one of the following protocols:
- Telnet
- SSH
- Rlogin
- Make sure that the appropriate daemon is running on the server and that the userid used for connecting has the permission to connect using the appropriate protocol.

Commands

The table below describes the commands that the connection Userid needs to have access to, to collect performance stats. The default path for the Userid on UNIX should be setup so you can execute the commands without qualifying them with a path.

Command Type	Description	Commands/Files
Common Commands	Required on all UNIX and Linux servers.	ps, wc, cat, grep, vmstat, uname, uptime, who, df, netstat
Linux Commands	User should be able to access data from these files	/proc/meminfo /proc/cpuinfo /proc/partitions or iostat
AIX Commands	Required for AIX	lsdev pmcycles usr/sbin/prtconf iostat lsps pagesize
Solaris Commands	Required for Solaris	/usr/sbin/psrinfo iostat /usr/sbin/prtconf /usr/sbin/swap pagesize
HP Commands	Required for HP	/etc/swapinfo sar permission to access the following: /dev/mem /dev/kmem /var/adm/syslog/syslog.log

Opening Performance Analyst for the First Time (DBArtisan)

For Performance Analyst to work, a platform-specific package needs to be installed by the SYS account. When Performance Analyst first starts, it checks for the presence of this package and if it has not been installed, Performance Analyst prompts you to install it. Once created, any account with access to this package can monitor databases using Performance Analyst. You do not need to install any other objects on a monitored database.

Note: You must have SYS privileges to install the required packages. After the packages are installed, anyone can use them.

To open the Performance Analyst for the first time, connect to a database through DBArtisan and then select Performance Analyst from the Analyst drop-down menu.

Subsequently, you can connect to Performance Analyst by going to Start>Programs>EmbarcaderoDBArtisan>Performance Analyst.

When you open the Performance Analyst this way, you can only review one datasource at a time, as is the case when you open Performance Analyst remotely.

Note: If you are connecting through your client machine, it will save you time and give you greater flexibility to connect to Performance Analyst through the DBArtisan console. If you connect through the console, you can open multiple Performance Analyst windows for as many datasources as you like and switch between them by clicking the target tab.

You can also open Performance Analyst in stand-alone mode from your Web client. To do this, see: Opening Performance Analyst in Stand-Alone Mode from Your Web Client

Opening Performance Analyst in Stand-alone Mode from your Web Client

The Embarcadero Performance Analyst Login dialog box lets you open Performance Analyst in stand-alone mode from a Web client. This way you can check the performance of any or all datasources that are registered through DBArtisan when you are working remotely.

To connect to the Performance Analyst when you are not using the DBArtisan console, simply go to

<installation directory>:\PAnalyst\PerfAnalyst.html

When you've made the connection, a login dialog opens.

If you are connecting to the Performance Analyst from your Web client, you can only see one database at a time.

Note: Keep in mind that when you connect to Performance Analyst remotely, the Auto Connect option you may have selected when you registered the target datasource in DBArtisan is not functional. You must supply a valid user ID and password to log in to Performance Analyst successfully.

Login As: is only an option when you are logging into an Oracle database. Sybase ASE, SQL Server and DB2 log in as default. The properties for the login roles are as follows:

- Default - Establishes a connection with no system roles.
- SYSDBA - Lets you perform all DBA-related functions. SYSDBA is required for the SYS id when you log into a 9i or later instance.
- SYSOPER - Lets you perform many DBA-related functions, like startup and shutdown.

Licensing Performance Analyst (DBArtisan)

Availability of this feature depends on your licensing. For more information, see Licensing.

Performance Analyst Application Basics (DBArtisan)

Performance Analyst displays a complete picture of a database's performance. Performance Analyst is a quick, lightweight tool that lets you investigate performance bottlenecks.

For more information, see Product Design.

Topics

- Performance Analyst Product Design (DBArtisan)
- Performance Analyst Product Navigation (DBArtisan)
- Alarm Sound (DBArtisan)

Performance Analyst Product Design (DBArtisan)

The Performance Analyst product design includes Home Pages.

Home Pages for Oracle

Performance Analyst displays home pages of various performance categories with detailed performance data. The Performance Analyst product design includes the following home pages:

- Home View (DBArtisan - Oracle Performance Analyst)
- Memory View (DBArtisan - Oracle Performance Analyst)
- I/O View (DBArtisan - Oracle Performance Analyst)
- Objects View (DBArtisan - Oracle Performance Analyst)
- OS View (DBArtisan - Oracle Performance Analyst)
- RAC Detail View (DBArtisan - Oracle Performance Analyst)
- Space View (DBArtisan - Oracle Performance Analyst)
- Users View (DBArtisan - Oracle Performance Analyst)

Home Pages for Microsoft SQL Server

Performance Analyst displays home pages of various performance categories with detailed performance data. The Performance Analyst product design includes the following home pages:

- Home View (DBArtisan - SQL Server Performance Analyst)
- Database View (DBArtisan - SQL Server Performance Analyst)
- Memory View (DBArtisan - SQL Server Performance Analyst)
- I/O View (DBArtisan - SQL Server Performance Analyst)
- Space View (DBArtisan - SQL Server Performance Analyst)
- Users View (DBArtisan - SQL Server Performance Analyst)
- OS View (DBArtisan - SQL Server Performance Analyst)

Home Pages for IBM DB2 for Linux, Unix, and Windows

Performance Analyst displays home pages of various performance categories with detailed performance data. The Performance Analyst product design includes the following home pages:

- Home View (DBArtisan - DB2 LUW Performance Analyst)
- Cfg (DBArtisan - Database Configuration)
- I/O
- Instance View (DBArtisan - DB2 LUW Performance Analyst)

- Memory View (DBArtisan - DB2 LUW Performance Analyst)
- OS View (DBArtisan - DB2 LUW Performance Analyst)
- Space View (DBArtisan - DB2 LUW Performance Analyst)
- Users View (DBArtisan - DB2 LUW Performance Analyst)

Home Pages for Sybase ASE

Performance Analyst displays home pages of various performance categories with detailed performance data. The Performance Analyst product design includes the following home pages:

- Home View (DBArtisan - Sybase ASE Performance Analyst)
- Databases (DBArtisan - Sybase ASE Performance Analyst)
- I/O (DBArtisan - Sybase ASE Performance Analyst)
- Memory View (DBArtisan - Sybase ASE Performance Analyst)
- OS (DBArtisan - Sybase ASE Performance Analyst)
- Space (DBArtisan - Sybase ASE Performance Analyst)
- Users (DBArtisan - Sybase ASE Performance Analyst)

Performance Analyst Product Navigation (DBArtisan)

Navigating Performance Analyst is easily accomplished through two main mechanisms:

1 The main Performance Analyst toolbar buttons open the major performance analysis views (the toolbar buttons you see are self-explanatory) within the product as well as frequently used actions like Refresh and Performance Threshold Customizations.

2 If you right-click any area within Performance Analyst, the application displays a shortcut menu that lets you easily move between any of the main Performance Analyst home pages and drill-down views. The shortcut menu also contains context sensitive options like drilling down further into a particular statistic, copying the display to a text file, and more.

For more information, see Toolbar Icons.

Toolbar Icons

Performance Analyst has several unique icons that take you directly to some functions and statistical groups that can help you see and sort through groupings of data. The functional icons you see on the Performance Analyst toolbar are as follows:

- Top Sessions - - This takes you directly to the Top Sessions tab of the Users Detail view where you see the Top I/O, Memory, CPU, and Sessions statistics.
- Top SQL - - Opens the Top SQL detail view and lets you move among the Summary, Criteria, SQL Details, and Recent History tabs.
- Response Time (Oracle 10 g only) - - Oracle 10g collects response time statistics, which are displayed when you click the icon.
- Alarms - - Goes directly to the Alarms log.
- Threshold Maintenance - - Clicking here opens the Threshold Template Maintenance dialog. Thresholds are explained in the Using chapter.
- Refresh - - Click this to update the stats/information on whatever page you are currently scrutinizing.
- Options - - To open thePerformance Analyst options dialog, where you can change the values for Oracle, SQL Server, Sybase ASE, or DB2 options, click this cheerful checkerboard.
- Exit - - This closes Performance Analyst for the database you are reviewing. If you have multiple Performance Analysts open, clicking this does not close all Analysts.
- Maximize Window - - This icon appears on each pane of a multi-pane window and lets you expand the pane of choice to fill the screen.

- Minimize Window - - When you have seen enough of an expanded pane, you can bring it back to its original size by clicking this icon.
- Help - - Another in this series of intuitive icons, click this to open the context-sensitive Help.

Alarm Sound (DBArtisan)

When you register an alarm in Performance Analyst, it creates a periodic alarm sound. To turn off the alarm sound, do the following:

1 Click Start, Programs, Accessories, Entertainment, and then Volume Control.

Windows opens the **Volume Control** dialog.

2 Under Wave, select Mute.

Note: Hiding the Performance Analyst icon in the task bar does not mute the sound.

Using Performance Analyst (DBArtisan)

Performance Analyst allows you to organize performance metrics in the particular way you want to see them. You can drill-down into the many layers of database performance and also be able to locate performance issues in your database in a fast and effective manner. You can set alarm triggers and thresholds for multiple databases across several platforms.

For example, key ratios in Performance Analyst are calculated using delta statistics, which is superior to computing ratios with cumulative values. Delta statistics, which are values derived by using the before and after measures in each statistical refresh done in Performance Analyst, lets you see changes in performance metrics that would otherwise be invisible if cumulative statistics were used. This chapter tells you how to set the refresh intervals that best suit your needs.

So, this Using chapter explains how to reveal the layers of information you want to see. The platform-specific Expert Guides explain what you are seeing as they discuss individual statistics, the associated metrics, and troubleshooting suggestions:

- Oracle Performance Analyst Statistics (DBArtisan)
- Microsoft SQL Server Performance Analyst Statistics (DBArtisan)
- IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan)
- Sybase ASE Performance Analyst Statistics (DBArtisan)

Topics

- Performance Analyst Tools (DBArtisan)
- Alarm Log (DBArtisan)
- Changing Graph Type (DBArtisan)
- Copy (DBArtisan)
- Details (DBArtisan)
- Explain SQL (DBArtisan)
- Manage with DBArtisan (DBArtisan)
- Performance Analyst Options Dialog Box (DBArtisan)
- Response Time for Oracle 10g (DBArtisan)
- Save As... (DBArtisan)
- Thresholds (DBArtisan)

Performance Analyst Tools (DBArtisan)

Performance Analyst offers tools that help you configure what you can see when you survey target databases. It’s an easy thing to set up an alarm notification for individual statistic thresholds based on parameters you designate. And, when you work through DBArtisan, you can easily correct problems if you notice something is amiss.

Performance Analyst supports the following functionalities:

Alarm Log
Change Graph Type
Copy
Details
Explain SQL
Manage with DBArtisan
Options
Response Time (ORACLE 10g ONLY)
Save
Thresholds
Top Sessions
Top SQL

Alarm Log (DBArtisan)

The Alarm Log displays every performance violation found in your currently monitored databases. Unlike some products that only show you alarms for a particular database, Performance Analyst gives you the option to see all alarms for each database you are monitoring in one central log so you can see every problem in a single place. And, of course, you have the ability to determine alarm levels by setting the Threshold Properties.

To open an Alarm Log

- Open the Performance Analyst and then scroll down to Alarms on the drop-down menu.
- Move your mouse over the Performance Analyst toolbar and click the exclamation point (you'll see Alarms appear in a tool tip under the hand icon).

Organizing the Alarm Log

You can arrange the data in the Alarm Log in several ways:

Note: Changes you make to one Alarm Log are not enacted globally. This allows you to customize each log based on your needs database by database if you want.

- Move columns left or right to place them in an order that makes the most sense to you. Do this by dragging-and-dropping a column heading to re-place it in the target location.
- Sort column rows up and down or sideways. Right-click a column heading and select:
- Sort Ascending: For example, you can see the alarms as they move from Critical to Cleared.
- Sort Descending: For example, you can see the alarms as they move from Cleared to Critical.
- Sort Multiple Columns: Lets you group multiple columns in ascending or descending order. You cannot mix and match, however. That is, you cannot sort some columns to present data in ascending order and others to present data in descending order.
- Sort Clear: Lets you clear the relationships you created if you sorted multiple columns.

The table below describes the columns of the Alarm Log:

Column	Description
Alarm Type	Displays alarm types: Cleared, Critical, Fair and Poor.
Datasource	Datasource where the alarm was identified.
DBMS Type	Datasource DBMS type.
Date/Time	Date/time of when the alarm was observed.
Category	Displays metric category: memory, I/O, space, objects, or users.
Statistic	Metric name.
Value	Value of the statistic that caused the violation.
Explanation	Displays a text explanation of the problem. These text explanations can be very generic in nature and just plug in certain threshold metrics, etc.

Alarm Log Activities

From within the Alarms Window, right-clicking any row gives you more opportunities to get the most out of what you see. The menu selections are:

- Applicable Page - Takes you to the row-specific Category page where you can see the problem metric and what's happening with the surrounding statistics. For example, if you click a critical alarm in the Databases category, you connect immediately to the Databases home page.
- Options - Takes you to the Alarm Log Options dialog where you adjust some general alarm log properties and alarm filters.
- Clear List - When you select this, the alarm log is cleared of all entries.
- Help - Opens the Help discussion for the target element.
- Refresh - Refreshes the display to show you any updates.
- Copy - Places the data in the alarm log at that point in time on the Windows clipboard. From there you can paste it into a Word or Excel document.
- Save As - Lets you name and save the Alarm Log to the location of your choice as a Microsoft Excel worksheet. You can then manipulate the worksheet as you would any other.
- Print - Enables you to print the alarm log.

Alarm Log Options Dialog Box

The **Alarm Log Options** dialog is command central for how and where you display and manipulate previously configured alarms. Alarms themselves are configured in the Threshold Properties, but this is where you arrange the particulars of the Alarm Log your databases. Keep in mind that changes you make to this dialog box are reflected globally across all platforms.

The table below describes the columns of the **Alarm Log Options** dialog:

Option	Description
General Properties	
Do not log alarms to the alarm log	When you select this option, alarms will not be saved to an alarm log. Depending on the filters you select, you will still see alarms in the alarm window.
Log alarms to the alarm log	This option lets you save alarms to an alarm log. You can type or browse to choose a directory where you want to save the log and you can also set the maximum alarm log size in KB.
Filters	
Show alarms for only current datasource	When you select this option, you see only alarms for the current datasource in the alarm window. So, if you have multiple datasources open and you go from one to another, the alarm window will display only the alarms for the database you are observing at that very moment.
Show alarms for all currently modified datasources	This option lets you see what alarms are occurring across all databases you are connected to at the moment.
Show only active alarms	If you select this option, you'll see only those alarms that are active at that moment in time. These are alarms that you decided to include as part of a threshold template.
Show only alarms where the Explanation contains the following text	Lets you fine-tune the alarms you see depending on what you are most interested in seeing. For example, if you are particularly interested in seeing Bottleneck alarms, type Bottlenecks. NOTE: The text search for this function is case-sensitive.
Show only alarms that are of the following severity	Select to display alarms that are critical, poor, fair, or cleared.

Changing Graph Type (DBArtisan)

When you are looking at a statistic that is displayed in graph form, you can very easily change how that data is displayed. When you choose to switch from one type to another, the changes are limited to the specific graph and its data that you are manipulating. So, for example, the same page can display a bar graph and a pie chart.

The graph types available to you are:

- Line Graph
- Area Graph
- Pie Graph
- Bar Graph
- Horizontal Bar Graph

To change the graph type

- Right-click the target graph, and from the drop-down menu, select Change Graph Type.

Copy (DBArtisan)

The Copy functionality lets you copy data in a tabular grid display into the Windows clipboard. The data can then be saved and used for later comparisons as you continue to tweak your databases' performance.

To Copy data, do the following

1 Hover the pointer over the target section or metric, and then right-click and select Copy.

Performance Analyst copies data into the Windows clipboard.

2 Select an existing, or new, Word or Excel document where you want to Paste the data.

3 Save, or Save As, the document to the target location.

Details (DBArtisan)

The Details functionality lets you drill-down into the target metric data. Generally you are taken to the metric's category detail page.

To see section or metric details

- Right-click in the target section or metric and select Details from the drop-down menu.

After you are taken to your destination, right-click again to see if another level of detail is available to you.

Explain SQL (DBArtisan)

Performance Analyst lets you run Explain SQL Plans from the Top SQL window. If you are seeing problem metrics, you can take steps to correct the SQL statements.

Opening the Explain SQL/Explain Plan

1 Click the SQL button on the Performance Analyst toolbar to open the Top SQL details page.

2 Click the SQL Details tab.

You can also right-click anywhere and select Navigate to...>Top SQL>SQL Details from the drop-down menus.

3 Move down the list of SQL text in the grid and the SQL text appears in the Selected SQL Text window.

4 Press the Explain SQL button to open the Explain Plan.

Manage with DBArtisan (DBArtisan)

Manage with DBArtisan opens the DBArtisan editor for the object that has been selected for editing in a grid/tabular display of data.

By moving directly to DBArtisan from Performance Analyst, you have an opportunity to fix database problems as soon as you see them. This option is not available, however, when you open the Performance Analyst from a Web client.

Performance Analyst Options Dialog Box (DBArtisan)

The Performance Analyst Options dialog box is where you can specify various refresh intervals, threshold polling, and datasource overrides among other settings. You can set or change defaults that are both global and platform-specific.

Note: The settings on the Global and Datasource tabs apply to all datasource platforms. The SQL Server, Sybase ASE, and DB2 tab settings are platform-specific. Some options on the platform-specific tabs may look as if they are available to you, but in reality they are not. For example, if you are connected to an Oracle datasource, changing the Monitoring Refresh interval on the Sybase tab is a hollow gesture.

To open the Performance Analyst Options dialog box

- Open the Performance Analyst and then scroll down to Options on the drop-down menu.
- Right-click in the Performance Analyst window for the target datasource and select Options.
- Move your mouse over the Performance Analyst toolbar and click the multicolored square (you'll see Options appear in a tool tip under the hand icon).

The table below describes the tabs of the Performance Analyst Options dialog:

Tab	Settings Group	Option/Description
Global	Engine	**SQL metric refresh interval** - Lets you set the time, in seconds, between collect cycles for the SQL metrics reflected in Performance Analyst. SQL metrics include all non-operating system metrics. **OS metric refresh interval** - Lets you set the time, in seconds, between collect cycles for all operating system-related metrics reflected in Performance Analyst. OS metrics include all metrics shown on the OS page. **Threshold polling** - Lets you determine which metrics, if any, you want to test against your pre-set thresholds. The possible settings for the Threshold polling box are: All Metrics - Polls all metrics Current Page Only - Polls only those metrics reflected on the currently open page in Performance Analyst Off - Turns polling off. If polling is turned off, Performance Analyst does not activate Alarms. **Threshold polling interval** - Lets you set the time, in seconds, between collect cycles for threshold-related metrics. Once collected, the polled metrics are compared to the pre-set thresholds. If you select Off in the Threshold polling box, Performance Analyst does not poll these metrics, regardless of the interval. R**un queries for all tabs** - Lets you determine whether or not you want Performance Analyst to refresh all tabs according to the SQL metric refresh interval. If Run queries for all tabs is not checked, Performance Analyst runs queries only for the current tab.
	Visual	**Show alarms icon in system tray** - Lets you determine whether or not you want Performance Analyst alarms to appear in the Windows System Tray on the lower right-hand side of your screen. If you select Show alarms icon in system tray, Performance Analyst places an icon in the Windows System Tray each time an alarm is raised. **Number of Historic Values in Graphs** - Lets you specify the number of values in the graphs.

Datasource	Operating System	**OS type** - Lets you select the operating system type for the current datasource. The options are: MS Windows - Select MS Windows for all supported types of Microsoft Windows Unix or Linux - Select Unix or Linux if you operating system is any flavor of Unix **Host name** - Lets you specify the name of the host for the current datasource, if necessary. Performance Analyst automatically fills in any detectable information, including the Host name box. **Username** - Lets you select a local or network username that Performance Analyst can use to connect to the datasource. If you select the Use Datasource Username and Password check box, the Username box is unavailable. **Password** - Lets you select a local or network password that Performance Analyst can use to connect to the datasource. If you select the Use Datasource Username and Password check box, the Password box is unavailable. **Protocol** - Lets you specify a protocol for communicating with datasources on Unix-based operating systems. The options are: SSH Telnet Rlogin **Use Datasource Username and Password** - Lets you specify whether or not you want to use the datasource username and password to connect to the datasource as opposed to a network or operating system username and password **Save Username and Password** - Lets you save the username and password that appear in the Username and Password boxes.
	Overrides	**SQL metric refresh interval** - Lets you override the SQL metric refresh interval in the **Global** tab for the current datasource. The SQL metric refresh interval specifies the time, in seconds, between collect cycles for the SQL metrics reflected in Performance Analyst. SQL metrics include all non-operating system metrics. **OS metric refresh interval** - Lets you override the OS metric refresh interval in the **Global** tab for the current datasource. The OS metric refresh interval specifies the time, in seconds, between collect cycles for all operating system-related metrics reflected in Performance Analyst. OS metrics include all metrics shown on the OS page.
SQL Server	Global Options	**SQL Profiling Enabled** - Performance Analyst requires some privileges that go along with SQL Server Profiler in order to return some statistics for SQL Server databases. Check SQL Profiling Enabled to ensure that Performance Analyst can collect all the necessary metrics.
	New Datasource Defaults	**Start SQL Profiling** - Lets you set new datasources to start SQL Profiling when they are monitored by Performance Analyst. **SQL Profiling refresh interval** - Lets you set the default time for new datasources, in seconds, between collect cycles for the SQL metrics that require the SQL Server Profiler.
	Current Datasource Settings	**Start SQL Profiling** - Lets you set current datasources to start SQL Profiling when they are monitored by Performance Analyst. **SQL Profiling refresh interval** - Lets you set the default time for current datasources, in seconds, between collect cycles for the SQL metrics that require the SQL Server Profiler.

Sybase	**Global Options**	**Monitoring Table refresh interval** - Lets you specify, in seconds, the amount of time between refreshes for the Monitoring Table. The Monitoring Table is a table installed by Performance Analyst used to store various metric-related information. The default is 15 seconds.
	New Datasource Defaults	**Use noclear option in sp_sysmon** - Performance Analyst sessions are started with the `'sp_sysmon noclear'` option, ensuring that monitor values are stored in a temporary file. If monitoring is being used by another application, those results are not impacted by the Performance Analyst session zeroing out counters. NOTE: The **noclear** option is available for Sybase ASE 12.5.3 and higher and 15.0.2 and higher.
	Current Datasource Defaults	These settings are only available with a Performance Analyst session active against a Sybase ASE datasource. The settings only apply to current sessions. The **Monitoring Table refresh interval** and **Use noclear option in sp_sysmon** settings let you override the selections in the identically-named settings in the **Global Options** and New Datasource Defaults groups. **Use custom sample query** - You can use this option to build a custom sampling query and check the database response time by reviewing the Sample Query Response Time graph on the Database home page.
DB2	**Monitor Settings**	Monitor Switches - You are able to set monitor switches ON for the current session. When check marks are visible, that means they were previously set as defaults. Check or uncheck the boxes as you wish. Monitor settings are available for Buffer Pool. Statement, Lock, Table, Sort, Unit of Work. Metrics for all checked elements are gathered in the current session. SQL Statement Cache - You can collect SQL statements on each refresh, collect SQL statement only when you are on the Top SQL pages, or choose not to collect any SQL statements. Database System Monitor Heap - The database system monitor heap size shows the current value, in pages, allocated for the database system monitor data. The amount of memory required for monitoring depends on the number of monitoring applications, switch settings, and the level of database activity. If you see frequent heap overflow errors in the output message window when monitoring DB2 through Performance Analyst, adjust the setting here. Save settings as default - Gives you the option to set the monitoring settings as a default for the current DB2 database or for the current session only.

Response Time for Oracle 10g (DBArtisan)

The Response Time functionality lets you analyze response times. You can review historical data and pinpoint who was on a database and what they were doing during a specific period of time.

To see Response Time statistics

- Right-click in the target and select Navigate to...>Response Time Analysis from the drop-down menu.
- Hover your mouse over the Performance Analyst toolbar and click the magnifying glass (you'll see Response Time appear in a tool tip under the hand icon).

Save As... (DBArtisan)

The Save As option gives you an opportunity to save data in a tabular format in the location of your choice. What this means is that the current data will be saved to an existing or a new Microsoft Excel worksheet. You can then manipulate and compare data with all the functionality Microsoft Excel gives you.

To Save As

- Right-click anywhere on the metric or in the Alarm Log, for example, and choose Save As... from the drop-down menu.

Thresholds (DBArtisan)

Thresholds are the point that must be exceeded to begin producing a user-defined effect or result, or to elicit a response, such as an alarm. Performance Analyst provides default threshold templates for each database platform. The default template is applied unless you choose to apply a customized template. There are two methods to place thresholds on statistics in Performance Analyst:

- Threshold Templates - Threshold templates are logical containers used to define and deploy thresholds to datasources. You can apply many templates at a single time and you can apply a single threshold template to many datasources. To customize a template, use the Threshold Template Maintenance dialog. As you add or edit thresholds for a customized template, you make your changes in the Threshold Template Properties dialog. And finally, when you have customized a threshold template to your satisfaction, you Select Datasource(s) where you want the template to be applied.
- Datasource Thresholds - Datasource Thresholds are default or user-defined values that mark performance boundaries. You can apply Datasource Thresholds to specific statistics on a datasource. To make changes to a datasource threshold, use the Datasource Thresholds Maintenance dialog. From here, the Threshold Templates dialog gives you the opportunity to use a previously created template to override the target datasource's user thresholds.

Whenever you make changes to a threshold, whether through the Threshold Template functions or from the Datasource Thresholds, you are presented with the Threshold Properties dialog where you can specify when, and what type of alarm you want to trigger.

Additionally, you can Copy a template.

Threshold Properties Dialog Box

Thresholds are the point that must be met to begin producing a user-defined effect or result, or to elicit a response, such as an alarm. The **Threshold Properties** dialog lets you modify properties for any threshold.

To open the Threshold Properties dialog box

- Choose Threshold Maintenance... from the Performance Analyst drop-down menu to open a **Threshold Template Maintenance d**ialog box, and then click Add or Edit.
- You can also choose Datasource Thresholds... from the Performance Analyst drop-down menu to open the Datasource Thresholds Maintenance dialog, and then click Add or Edit.

The table below describes the options and functionality on the T**hreshold Properties** dialog:

Option	Description
Statistic	Lets you choose the target statistic where you want to add a threshold. NOTE: If you are adding a statistic to a template or datasource threshold, use the drop-down list to find the statistic you want to add. When you select Edit for any statistic showing in the User Thresholds window, the target statistic is displayed.
Generate Alarm If	Lets you choose or change the target alarm condition. The seven options are: greater than, greater than or equal to, is equal to, is not equal to, is not equal to, less than, and less than or equal to.
Value amount	Lets you set or change the numerical value for the threshold. You cannot leave the value text box blank when you first add the threshold.
Severity	Lets you select the target severity level: Fair, Poor, or Critical.

Threshold Templates Dialog Box

Threshold Templates are logical containers used to define and deploy thresholds to datasources. Performance Analyst offers a default template for each platform. You can copy or customize and apply templates to your heart's content. You cannot, however delete the default template.

To open the Threshold Templates dialog box

- Choose Datasource Thresholds... from the Performance Analyst drop-down menu to open the **Datasource Thresholds Maintenance** dialog, and then click Change.

The T**hreshold Templates** dialog displays the available Threshold Templates. The table below describes the options and functionality of the **Threshold Templates** dialog:

Option	Description
Name	Displays the name(s) of the available platform-specific Threshold Template(s).
Description	Displays the description(s), if any of available Threshold Template(s).
Override Datasource's User Thresholds	Lets you choose which thresholds to use on the target datasource - the thresholds in the Threshold Template or the user-defined thresholds on the target datasource. If you select the check box, Performance Analyst will override any Datasource Thresholds on the target datasource with the thresholds in the Thresholds Template.

Threshold Template Properties

Threshold Templates are logical containers used to define and deploy thresholds to datasources. The **Threshold Template Properties** dialog is where you create a custom template or where you edit an existing one.

To open the Threshold Template Properties dialog box

- Choose Threshold Maintenance... from the Performance Analyst drop-down menu to open a **Threshold Template Maintenance** dialog, and then click Add or Edit.

The table below describes the options and functionality on the **Threshold Template Properties** dialog

Option	Description
Name	Lets you name the threshold template you are creating. If you are editing a template, target template's name appears here.
Description	Lets you type or review the target template's description.
Platform	Offers a drop-down list of supported platforms. The selection box displays the target template's platform, but you can use this list to see thresholds from other platforms.
Threshold/Severity/Condition/Template	Displays each threshold contained in each template for a given platform. Here you see the severity and condition of each template as well.
Arrow	Lets you select thresholds for the Threshold Template. When you click the arrow, Performance Analyst moves selected thresholds to the Threshold Template you are creating or editing.
Add	Use this function to add statistics to the threshold template you are creating/editing. When you click this button, Performance Analyst opens the Threshold Properties dialog.
Edit	Lets you edit the selected Datasource Thresholds appearing in the Current Template window. When you click this button, Performance Analyst opens the Threshold Properties dialog.
Delete	Lets you delete the selected Datasource Thresholds. When you click this button, Performance Analyst deletes the selected thresholds.
Save	Saves any changes you made in this dialog box to create the new or modified template.

Threshold Template Maintenance Dialog Box

Threshold Templates are logical containers used to define and deploy thresholds to datasources. The **Threshold Template Maintenance** dialog lets you add or edit templates, as well as configure template maintenance options. For example, you can copy the original Embarcadero Threshold template and modify it to fit your needs. You can control which statistics are sampled for performance violations, as well as the levels of severity to report. When you are satisfied with your Template, you can save it, and then, in one action, apply it to one or more datasources, thereby immediately enforcing your performance standards.

To open the Threshold Template Maintenance dialog box

- Choose Threshold Maintenance... from the Performance Analyst drop-down menu.
- Hover your mouse over the red and green cylinder on the Performance Analyst toolbar and look for the Threshold Maintenance tool-tip.

The table below describes the options and functionality on the **Threshold Template Maintenance** dialog:

Option	Description
Platform	This is a drop-down list of the platforms. The default is the platform of the datasource you connected to most recently, but you can make another selection.
Name	Displays the name(s) of available Threshold Template(s).
Description	Displays the description(s) of available Threshold Template(s). For customized templates, the description is merely a repetition of the template name unless you wrote something when you created or edited the template.
Add	Lets you create Threshold Templates. When you click this button, Performance Analyst opens the Threshold Template Propertiesdialog.
Edit	Lets you edit the selected Threshold Templates. When you click this button, Performance Analyst opens the Threshold Template Properties dialog.
Delete	Lets you delete the selected Threshold Templates.
Copy	Lets you create copies of the selected Threshold Template. When you click this button, Performance Analyst opens the Copy Template dialog.
Apply	Lets you select datasources to apply the selected Threshold Templates. When you click this button, Performance Analyst opens the Select Datasource(s) dialog.
Select/Unselect All	Lets you make global additions to your templates. NOTE: You cannot edit or delete Default templates.

Datasource Thresholds Maintenance Dialog Box

Datasource Thresholds are default or user-defined values that mark performance boundaries. You can apply Datasource Thresholds to specific statistics on a datasource. You can also combine Datasource Thresholds to create a template.

Performance Analyst offers the **Datasource Thresholds Maintenance** dialog to let you modify Datasource Thresholds. There are two panes in the **Datasource Thresholds Maintenance** dialog

- Applied Threshold Template
- User Thresholds

To open the Datasource Thresholds Maintenance dialog box

- Choose Datasource Thresholds... from the Performance Analyst drop-down menu.
- Right-click in any open Performance Analyst window and choose Datasource Thresholds... from the drop-down list.

The table below describes the options and functionality in the Applied Threshold Template box of the **Datasource Thresholds Maintenance** dialog:

Option	Description
Name	Displays the name of the target Threshold Template.
Description	Displays any description of the Threshold Template.
Change	Lets you view and edit all platform-specific Threshold Templates. When you click this button, Performance Analyst opens the Threshold Templates dialog.
Remove	Removes all thresholds in the Applied Threshold Template grid.
Threshold/Severity/Condition/	Displays the statistic on which the threshold is placed, whether it is critical, fair, or poor, and the condition that triggers the alarm.
Block	Prevents that threshold statistic from being triggered in the template. It's almost as if that threshold never existed in the template. You can, however, reinstate the threshold trigger by highlighting and deleting it from the User Thresholds pane.
Override	Copies the threshold settings from the selected Threshold Template and applies the change to the user-defined Datasource Thresholds. Click this button if the template has the correct Severity and Threshold, but you want to modify the condition for triggering the alarm. When you create a "User Threshold," you are overriding the threshold specified by applied template or you are including an additional threshold not included in the applied template. This may be useful if you have a group of individuals all monitoring the same primary statistics (and thus using the same template) but want to be able to give individual users the ability to modify/add/block a threshold used by the template, without affecting the original template.

The table below describes the options and functionality in the User Threshold box of the **Datasource Thresholds Maintenance** dialog:

Option	Description
Threshold	Displays the statistic on which the threshold is placed.
Severity	Displays the severity for the threshold.
Condition	Displays the condition for the threshold.
Add	Lets you create Datasource Thresholds. When you click this button, Performance Analyst opens the Threshold Properties dialog.
Edit	Lets you edit the selected Datasource Thresholds. When you click this button, Performance Analyst opens the Threshold Properties dialog. If you Edit a Blocked threshold, it loses its blocked status.
Delete	Lets you delete the selected Datasource Thresholds. When you apply this to a blocked threshold, it is removed from the user thresholds.

Select Datasource(s)

When you have created or edited a threshold template, and you are ready to distribute it, you have the opportunity to apply it to one or more target datasources using the **Select Datasources** dialog.

To open the Select Datasource(s) dialog box

- Choose Threshold Maintenance... from the Performance Analyst drop-down menu, and then click Apply.

The table below describes the options and functionality on the **Select Datasources** dialog:

Option	Description
Datasources	You can apply the target threshold template to one or more datasources. You can only apply the target threshold template to the platform where it was created.
Override Datasource's User Thresholds	Lets you choose which thresholds to use on the target datasource - the thresholds in the Threshold Template or the user-defined thresholds on the target datasource. If you select the check box, Performance Analyst will override any Datasource Thresholds on the target datasource with the thresholds in the Thresholds Template.

Copy Template

The **Copy Template** dialog lets you copy an existing Threshold Template, rename it, and save it for later use.

To open the Copy Template dialog box

1 Choose Threshold Maintenance... from the Performance Analyst drop-down menu, to open the **Threshold Template Maintenance** dialog.

2 Select the template you want to replicate, and then click Copy.

The table below describes the options and functionality of the **Copy Template** dialog:

Option	Functionality
Name	Lets you type the name for the new Threshold Template. The default name is Copy of Frankenstein, or whatever it is you named your template.
Description	Lets you type the description of the new Threshold Template. The default description is simply a repetition of the template name.

Top Sessions

The Top Sessions functionality lets you drill-down into the target metric data and quickly pinpoint the heaviest users of I/O and memory. Generally you are taken to the database-specific Expert Guide when you need an explanation of what you are seeing.

To open Top Sessions

- Hover your mouse over the little head on the Performance Analyst toolbar and look for the Top Sessions tool-tip.

To open the Session Details View

- Double-click any entry in the Top Sessions grid of the Top Sessions view, and the Session Details view opens.

Top SQL

The Top SQL view displays all SQL running or executed in the current datasource. You can use the lists and boxes to reorder the display according to your specific needs.

To view Top SQL, do the following:

- On the Performance Analyst toolbar, click the SQL button.

Performance Analyst for Oracle Tutorial (DBArtisan)

Embarcadero Performance Analyst provides key insight into the complex world of database performance. Running inside the Embarcadero DBArtisan console, or stand-alone in a Web browser, Performance Analyst helps you pinpoint performance problems that threaten a database's availability or response times. The powerful analysis facilities will satisfy advanced experts, yet novices will find Performance Analyst's interface easy to use and the expert help a welcome aid to troubleshooting difficult performance issues.

About This Performance Analyst for Oracle Tutorial

This tutorial is intended to help you get started with Embarcadero Performance Analyst.

After completion of this tutorial, you will have the foundation you need to explore the many features and benefits of Performance Analyst. You will have learned how to accurately monitor your database using the most popular performance methodologies in use today. You will know how to drill down into the many layers of database performance and also be able to locate performance issues in your database in a fast and effective manner. Finally, you'll understand how to fix performance problems using the integration that exists between Performance Analyst and Embarcadero DBArtisan.

This tutorial is divided into 8 sections.

Session 1: Getting Started With Performance Analyst for Oracle (DBArtisan)

Session 2: Using Ratio-based performance analysis in Performance Analyst for Oracle (DBArtisan)

Session 3: Using Bottleneck/wait-based Performance Analysis in Performance Analyst for Microsoft (DBArtisan)

Session 4: Using Workload Performance Analysis in Performance Analyst for Oracle (DBArtisan)

Session 5: Using SQL Performance Analysis in Performance Analyst for Oracle (DBArtisan)

Session 6: Using Operating System Performance Analysis in Performance Analyst for Sybase ASE (DBArtisan)

Session 7: Identifying and Getting Help with Performance Problems Using Performance Analyst for Oracle (DBArtisan)

Session 8: Fixing Performance Problems Using Performance Analyst for Oracle (DBArtisan)

You can use this basic tutorial as a roadmap of product highlights.

After starting this tutorial, you can select Help from the menu to find many additional resources that complement and build on many of the activities shown in this brief tutorial.

Proceed to Session 1: Getting Started With Performance Analyst for Oracle (DBArtisan)

Topics

- Session 1: Getting Started With Performance Analyst for Oracle (DBArtisan)
- Session 2: Using Ratio-based performance analysis in Performance Analyst for Oracle (DBArtisan)
- Session 3: Using Bottleneck/wait-based Performance Analysis in Performance Analyst for Oracle (DBArtisan)
- Session 4: Using Workload Performance Analysis in Performance Analyst for Oracle (DBArtisan)
- Session 5: Using SQL Performance Analysis in Performance Analyst for Oracle (DBArtisan)
- Session 6: Using Operating System Performance Analysis in Performance Analyst for Sybase ASE (DBArtisan)
- Session 7: Identifying and Getting Help with Performance Problems Using Performance Analyst for Oracle (DBArtisan)
- Session 8: Fixing Performance Problems Using Performance Analyst for Oracle (DBArtisan)

Session 1: Getting Started With Performance Analyst for Oracle (DBArtisan)

Download and Install

You can obtain the latest version of Performance Analyst by downloading the DBArtisan software from the Embarcadero Web site at http://www.embarcadero.com/downloads/downloaddbartisan.asp. Performance Analyst runs fully contained within DBArtisan so you will need the DBArtisan product to use Performance Analyst (although Performance Analyst can also run in standalone fashion within a Microsoft Internet Explorer Web browser).

Click Download, and follow the steps indicated. Save the file on your computer and then double-click to launch the self-extracting file that will guide you through the installation process.

When you first install an evaluation copy of DBArtisan, you can use the tool for 14 days. After that time, a permanent license is needed.

Proceed to Overview for Performance Analyst for Oracle

Overview for Performance Analyst for Oracle

The graphic below illustrates all the main elements of the DBArtisan User Interface.

When DBArtisan is connected to an Oracle database, you can open Performance Analyst two ways:

1 On the toolbar, click the Performance Analyst icon.

2 On **Utilities** menu, select **Performance Analyst**.

DBArtisan opens Performance Analyst.

Proceed to Session 2: Using Ratio-based performance analysis in Performance Analyst for Oracle (DBArtisan)

Session 2: Using Ratio-based performance analysis in Performance Analyst for Oracle (DBArtisan)

Ratio-based analysis has been around for many years, and used to be the only technique database administrators used when they were called on to diagnose the cause of a database slowdown. Performance ratios are very good roll-up mechanisms for busy DBAs, making possible the analysis-at-a-glance approach, however ratios must be computed correctly for them to be meaningful. Key ratios in Performance Analyst are calculated using delta statistics, which is superior to computing ratios with cumulative values. Delta statistics, which are values derived by using the before and after measures in each statistical refresh done in Performance Analyst, lets you see changes in performance metrics that would otherwise be invisible if cumulative statistics were used. Performance Analyst makes it easy to see the ratios that are computed with delta statistics vs. cumulative. A triangle symbol denotes a delta-based statistic, while square symbols represent a cumulative-based measure.

Ratio-based analysis is available for all performance categories (memory, etc.) where ratios can assist in determining the performance level of a database. For example, if you would like to see memory-specific ratios, on the Performance Analyst toolbar click Memory. Performance Analyst displays the Memory home page that showcases all important memory-related ratios.

Proceed to Obtaining Detail Information for a Ratio Using Performance Analyst for Oracle

Obtaining Detail Information for a Ratio Using Performance Analyst for Oracle

A ratio gives you a global picture for a particular performance measure. However, you often need more information regarding a statistical measure so you can make an informed decision. Performance Analyst has easy-to-use drill down features that make obtaining detailed information for various ratios a breeze.

For example, to view more information for the Buffer Cache Hit Ratio statistic, do the following:

1 On either the Main Home page or the Memory Home page, locate the Buffer Cache Hit Ratio statistic.

2 Position the pointer over the statistic.

The pointer changes to a hand symbol.

3 Double-click the left mouse button.

Performance Analyst opens the **Buffer Cache** tab of the memory drill down view.

4 Or right-click the statistic, select Details or Navigate to… > Memory > Buffer Cache.

Performance Analyst opens the **Buffer Cache** tab of the memory drill down view.

Proceed to Session 3: Using Bottleneck/wait-based Performance Analysis in Performance Analyst for Oracle (DBArtisan)

Session 3: Using Bottleneck/wait-based Performance Analysis in Performance Analyst for Oracle (DBArtisan)

When an Oracle database is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things, or it can be an indicator that a database bottleneck exists. Bottleneck analysis is a valid method of measuring performance because it helps you track where a database and user sessions have been spending their time. If latch contention or heavy table-scan activity has been dragging a database’s performance down, you can use bottleneck analysis to confirm the actual root cause.

Note: For Performance Analyst to show wait times for Oracle wait events, the database configuration parameter TIMED_STATISTICS must be set to TRUE.

Performance Analyst uses two broad areas of metrics to form its bottleneck analysis. The first is the Oracle wait event interface, which displays where the database and user sessions have been bottlenecked. But, when using bottleneck analysis you can not rely only on the information contained in the wait event views that Oracle provides. For example, an object may attempt to extend into another extent of space in a tablespace and yet, be denied if no such free space exists. Such a failure will not be reflected in any wait event, but still represents a very real bottleneck to the database. For this reason, Performance Analyst also shows non-wait event based bottlenecks that represent a very real threat to a database’s availability and performance.

Performance Analyst prominently displays bottleneck analysis on every performance home page. The Main home page, displays a summary of all key bottlenecks threatening the database. You can also see specific bottlenecks for memory, I/O, space, objects, and users by navigating to each home page and viewing the Bottleneck Analysis section for that home page. For example, to see bottlenecks that specifically relate to I/O, do the following

1 On the toolbar, click the I/O button.

2 Locate the Bottleneck Analysis section at the lower-left portion of the screen

The I/O bottleneck analysis section displays the top database and log-related Oracle wait events (ranking them by wait time and not actual wait count), along with the most active database files.

Proceed to Obtaining Detail on System Bottlenecks Using Performance Analyst for Oracle

Obtaining Detail on System Bottlenecks Using Performance Analyst for Oracle

You can easily obtain more information regarding global bottlenecks that are plaguing a database by drilling down into a particular bottleneck metric. For example, if you would like to see more data regarding the current top system bottlenecks, do the following:

1 Open the Main home page.

2 Double-click the Top System Bottlenecks area.

Performance Analyst opens the System Waits view.

3 Or right-click anywhere in Performance Analyst, point to Users, and then click System Waits.

Performance Analyst opens the System Waits view.

The System Waits view displays critical details regarding all the wait events that Oracle has recorded. You can easily see what events that occurred the most often as well as the events that have accumulated the most wait time.

Proceed to Viewing Detail for Session Bottlenecks in Performance Analyst for Oracle

Viewing Detail for Session Bottlenecks in Performance Analyst for Oracle

Not only can you obtain detail information for system bottlenecks, but you can also use Performance Analyst to get drill down information regarding session-level bottlenecks. There are several layers to session-level details you can view.

To see more information regarding session-level waits, do the following:

1 Open the Main home page.

2 Double-click the Top Session Bottlenecks area.

Performance Analyst opens the Session Waits drill down view.

3 Or right-click anywhere in Performance Analyst, point to Users, and then click Session Waits.

Performance Analyst opens the Session Waits drill down view.

When you are in the Session Waits drill down view, to drill further into a session's details, do the following:

1 In the Historical Waits or Current Waits grid, select a session that appears to be experiencing critical waits.

2 Double-click the session or right-click, and then select Details.

Performance Analyst opens a drill down view that shows key statistics for the selected session, including its current work activities, wait events, and SQL statements.

Proceed to Session 4: Using Workload Performance Analysis in Performance Analyst for Oracle (DBArtisan)

Session 4: Using Workload Performance Analysis in Performance Analyst for Oracle (DBArtisan)

Key ratios give you a global perspective on database activity. Bottleneck analysis gives you insight into things that are holding up user activity and throughput. But another technique is necessary if a database professional is to really get a handle on what's occurring inside a badly performing database.

Workload analysis involves the investigation of two critical areas of a database's performance:

- Session resource consumption and activity
- SQL execution

Without looking at these two key performance categories, you can miss a lot of what could be responsible for perceived performance problems. When performance on a database takes a sudden nosedive, it is not uncommon to find one or two sessions that are causing the bulk of the workload. Finding these processes can be easily accomplished by viewing session metadata coupled with resource consumption and statistical execution statistics.

Proceed to Pinpointing Top Resource Consumers in Performance Analyst for Oracle

Pinpointing Top Resource Consumers in Performance Analyst for Oracle

Performance Analyst identifies top resource consumers in a couple of different ways. On the main home page, Performance Analyst highlights the top resource sessions across physical and logical I/O, memory usage, and CPU consumption.

The percentage used across all statistical categories are displayed so you can immediately pinpoint a session that is using all or most of a particular resource. The top resource sessions are also displayed on the memory, I/O, and other home pages as well with a listing for the top sessions for that category. For example, the top memory users appear on the memory home page.

Proceed to Getting Workload Details Using Performance Analyst for Oracle

Getting Workload Details Using Performance Analyst for Oracle

Performance Analyst offers several details on top resource sessions in addition to the summaries provided on each home page. A detailed and flexible top sessions view lets you view a summary of all the major resource sessions on the database as well as every critical detail regarding each session connected to the database.

Further drill downs are available that highlights specific details regarding each session's activities and submitted SQL. To see detailed information regarding a session identified in one of Performance Analyst's Workload Analysis sections, do the following:

1 In a Workload Analysis display, position the pointer over one of the processes.

2 Double-click the process or right-click, and then select Details.

If you want like to see all key workload analysis metadata for sessions, do the following:

1 On the toolbar, click the Top Sessions button.

2 Or right-click anywhere in Performance Analyst, point to Users, and then click Session Waits.

Proceed to Session 5: Using SQL Performance Analysis in Performance Analyst for Oracle (DBArtisan)

Session 5: Using SQL Performance Analysis in Performance Analyst for Oracle (DBArtisan)

SQL-based performance analysis is really a subset of Workload Analysis. Understanding current and historical SQL execution patterns lets you to have the second set of data points necessary to properly perform workload analysis. You may find that optimizing SQL code produces some of the best performance-enhancing boosts available for a database.

Proceed to Uncovering Resource-intensive SQL in Performance Analyst for Oracle

Uncovering Resource-intensive SQL in Performance Analyst for Oracle

Performance Analyst offers plenty of insight into SQL performance and execution patterns. Each one of the home pages depicts the most costly SQL statements as determined by a variety of different metrics for that performance category (most physical, logical I/O, etc.) For example, Performance Analyst might show that a single SQL statement is responsible for almost 60 percent of all the physical I/O on the database.

Proceed to Getting Details on Top SQL with Performance Analyst for Oracle

Getting Details on Top SQL with Performance Analyst for Oracle

Getting details on identified SQL statements is easy in Performance Analyst. For example, if you would like to see further information on a SQL statement identified on the Performance Analyst Home page, do the following:

1 In the SQL Analysis section, position the pointer over Hash Values.

2 Double-click or right-click, and then select Details.

Performance Analyst opens the Top SQL summary page that shows a breakdown of all top SQL statements along with their execution statistics

If you would like to see the full SQL text of a SQL statement, do the following

1 In the Top SQL Details section, double-click a statement.

Performance Analyst opens the **SQL Details** tab of the Top SQL view to display the entire SQL statement and the execution metrics. While on this pane, you can also perform an EXPLAIN of the SQL statement and examine its access path information.

Note: To automatically tune a problem SQL statement if you have Embarcadero SQL Tuner installed on your workstation, do the following:

2 On the **SQL Details** tab, select a problem statement, and then click the Tune SQL button.

SQL Tuner opens and you can begin a tuning session.

Proceed to Session 6: Using Operating System Performance Analysis in Performance Analyst for Sybase ASE (DBArtisan)

Session 6: Using Operating System Performance Analysis in Performance Analyst for Sybase ASE (DBArtisan)

In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database sever for resources (CPU, I/O or Memory), a slow CPU, insufficient I/O devices, and insufficient memory. Performance Analyst allows you to examine the relevant operating system metrics for Windows Platforms (NT /2000/XP/2003) as they relate to Oracle.

To view the operating system metrics of the machine hosting the Sybase ASE server you are monitoring, do the following:

1 Click the OS Performance Analyst toolbar button.

OR

2 Navigate to the OS Home Page by going to OS, and then Home.

Note: The first time you invoke the OS feature you will be prompted for an operating system login. Please specify the correct operating system, host name, username and password, and protocol.

The Operating System home page has been designed in a similar fashion as the datasource home page. On this page you can see the pertinent operating system metrics displayed in five quadrants broken down into the following categories:

- Resource Usage
- Memory Analysis
- Disk Analysis
- Workload Analysis
- Bottleneck Analysis

Proceed to Getting Details on Top SQL with Performance Analyst for Oracle

Obtaining Details on the Operating System Using Performance Analyst for Oracle

You can easily obtain more information regarding operating system problems that are degrading database performance by drilling down into a particular analysis quadrant. For example if you'd like to see more information regarding the processes running on the host machine, do the following:

1 Navigate to the Performance Analyst Operating System home page.

2 Double-click the Top Memory Process in the Workload Analysis quadrant.

Performance Analyst opens the Processes detail view.

OR

3 Right-click anywhere in Performance Analyst and select the Navigate to, OS, and then Processes.

The Processes detail tab shows all the processes on the host machine. From this view you can easily determine which processes are showing high marks in I/O or CPU, for example, by simply double clicking on the column headers and sorting the processes in ascending or descending order.

Proceed to Session 7: Identifying and Getting Help with Performance Problems Using Performance Analyst for Oracle (DBArtisan)

Session 7: Identifying and Getting Help with Performance Problems Using Performance Analyst for Oracle (DBArtisan)

You do not need to be an advanced expert in performance tuning to spot trouble in a database if you use Performance Analyst. There are a number of mechanisms in the tool that help you quickly locate areas of concern within a database.

Proceed to Using the Alarm Log

Using the Alarm Log

Performance Analyst has a central and flexible alarm log that lists every performance violation found in all monitored databases. Unlike some products that only show you alarms for a particular database, Performance Analyst lists all alarms for each database you are monitoring in one central log so you can see every problem in one place. To view the Performance Analyst Alarm log, do the following:

1 On the toolbar, click the Alarms button.

Once in the Alarm log, you can perform a number of actions including:

- Selecting an alarm and viewing the application page that shows details regarding the metric in violation.
- Filtering the types of alarms shown through the Options dialog.
- Clearing all current alarms.
- Copying or saving the Alarm log into a different format.

To select any of these options, do the following:

1 Position the pointer over any area of the Alarm Log.

2 In the shortcut menu, select an option.

Proceed to Recognizing Performance Violations

Recognizing Performance Violations

Performance Analyst makes it easy to visually pinpoint performance violations in a database. Any statistic that does not conform to your predefined performance standards is colorized in any view that it appears so you can easily see potential troubleshoots.

The colors used to highlight a particular performance violation depend on the severity of the problem. Performance Analyst uses several layers of performance standards/threshold violations with each layer being represented by a different color. You can control everything regarding threshold violations, including how thresholds are determined, their color representation in Performance Analyst’s display, and more by using the Threshold Template feature of Performance Analyst.

How can Performance Analyst inform you of performance violations in your database if you minimize the product to your desktop and are working on other things? Even if Performance Analyst isn't right in front of you, it can still let you know that a problem has been found in your database through the sister alarm icon. When Performance Analyst detects a problem in one of your databases, one of the things the product does is enable an alarm icon in your workstation's systray area that communicates the presence of alarms in your database. So even if the Performance Analyst interface isn't in sight, the product can let you know that something in your database requires attention.

Proceed to Creating Threshold Templates

Creating Threshold Templates

How does Performance Analyst know what to signal as a problem in a database? It uses predefined performance standards that you define and enforce. All of this is accomplished by using Performance Analyst's Threshold templates. A template is a set of performance metrics and their thresholds that can be reused to enforce performance standards across any number of databases. Embarcadero provides a default performance template that contains nearly all of the most-watched performance statistics so you can begin monitoring with confidence right out of the box. However, you can create your own Threshold templates so you only are alerted to performance violations that are important to you and your particular environment.

To view or create Threshold templates, do the following:

1 On the toolbar, click the Thresholds button.

Performance Analyst opens the **Threshold maintenance** dialog.

The **Threshold Template Properties** dialog lets you select the options to add or edit templates, as well as other maintenance options. For example, you can copy the original Embarcadero Threshold template and modify it to fit your own needs. You can control what statistics are sampled for performance violations as well as the levels of severity you would like reported. Once you are satisfied with your template, you can save it and then apply it to one or more datasources in one action to immediately begin enforcing your performance standards.

Proceed to Getting Help with Performance Problems

Getting Help with Performance Problems

As databases become more complex, so do the problems that plague them. Knowing what to do for a particular performance problem can be hard in some cases, but Performance Analyst helps by providing a complete set of expert help that you can use as a guide when performance problems occur. To obtain expert help for any statistic shown in Performance Analyst, do the following:

1 Position the pointer over a statistic that you need help.

2 Right-click and select Help.

The Expert Help gives you a general description of the statistic, the range of values that are considered optimal, and some suggestions on how to fix a problem situation. Expert help can also be invoked for any of the Ratio, Bottleneck, SQL, or Workload Analysis sections shown on every home page by clicking on the help icon for that section, which is located at the top right corner of the section.

Proceed to Session 8: Fixing Performance Problems Using Performance Analyst for Oracle (DBArtisan)

Session 8: Fixing Performance Problems Using Performance Analyst for Oracle (DBArtisan)

Finding performance problems is one thing, but fixing them is another. Many performance monitors point out a particular problem in a database, but then offer no way to actually correct the situation. Such is not the case with Performance Analyst. Because of its tight integration with DBArtisan, certain problems found in Performance Analyst can be visually corrected by DBArtisan.

If an Oracle tablespace is about to run out of free space, Performance Analyst can issue an alarm that a space shortage exists. Once the space problem has been identified, do the following:

1 On the Space home page toolbar, click the Space button.

2 In the Space Analyst grid, right-click a tablespace, and then select Manage with DBArtisan.

DBArtisan opens an editor where you can perform corrective actions, such as enabling Autoextend for the tablespace, resizing the tablespace’s existing datafile(s), or adding a new datafile for the tablespace.

Oracle Performance Analyst Statistics (DBArtisan)

This section includes expert help for all Oracle categories and statistics in Performance Analyst views. For detailed information on using the application, see Using Performance Analyst. This guide includes the following sections:

Home View (DBArtisan - Oracle Performance Analyst)

The Home view lets you review availability and overall performance of all monitored databases from a single window. The Home page includes the following sections:

Bottleneck Analysis Pane
Key Ratio Analysis Pane
SQL Analysis Pane
Storage Analysis Pane
Workload Analysis Pane

Key Ratio Analysis Pane

Database performance analysts typically use one of two methods for examining the performance levels of a database: Ratio-based or wait/bottleneck-based. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as roll-up mechanisms for at-a-glance performance analysis. Succinctly presented performance ratios can be quickly scanned to see if any database needs immediate attention.

While there are certainly many opinions as to which rules to follow, there are some standards to adhere to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios a you examine come from the v$sysstat performance view. The performance view maintains a count of all the occurrences (in the VALUE column) of a particular database incident (in the NAME column) since the database was brought up. For databases that are kept up for long periods of time, these values can grow quite large and impacts how a particular ratio is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.

While there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each database’s individual ‘personality’ with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following ratios are used on the Performance Analyst Home page to succinctly communicate the general overall performance levels of the monitored database:

Buffer Cache Hit Ratio
Library Cache Hit Ratio
Dictionary Cache Hit Ratio
Memory Sort Ratio
Parse/Execute Ratio
Buffer Busy Waits
Rollback Contention Ratio
Latch Miss Ratio
Parallel Query Busy Ratio
Free Shared Pool Percent

Buffer Cache Hit Ratio

Data read from memory will produce end-user response times many times faster than when that same data is read from disk. Keeping physical I/O's to an absolute minimum is one of the purposes of the Oracle buffer/data cache.

The buffer cache hit ratio is a terrific indicator of how often user requests for data are satisfied through memory vs. being physically read from disk. A number of key counters in Oracle are used to arrive at this statistic. The table below lists them:

Counter	Description
DB BLOCK GETS	Data read from memory for DML operations.
CONSISTENT GETS	Data read from rollback segments in memory.
PHYSICAL READS	Data read physically from disk.
DIRECT READS	Data read physically from disk that bypasses the buffer cache. Direct reads are filtered out of overall physical reads so an accurate cache hit ratio can be determined.

Dividing the data read from memory by data read from disk yields the cache hit ratio.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Key Ratio Analysis PaneOracle Performance Analyst Statistics (DBArtisan) > Memory View (DBArtisan - Oracle Performance Analyst) > Key Ratio Analysis pane
Metrics	To help ensure excellent performance, you want to keep your cache hit ratio in the neighborhood of 90% or higher. However, you should be aware that every database has its own 'personality' and may exhibit excellent performance with below average readings for the cache hit ratio. You should also be aware that excessive logical I/O activity can produce a very high cache hit ratio while actually degrading overall database performance. Consistently viewed low readings (60% or less) may require tuning attention on the part of the DBA. **Note:** The db_cache_size parameter value may need to be adjusted. The db_cache_size parameter is dynamic and may be altered without stopping and starting the database instance.
Correction	If a problem is found, then: *Increase the size of the db_cache_size parameter through use of the ALTER SYSTEM SET db_cache_size command value (take caution to not over-allocate; ensure enough free memory exists on server before increasing value).Monitor the new value to see if performance improves.* If using an SPFILE, save the new configuration values so Oracle will reuse them each time the database is stopped and re-started.

Library Cache Hit Ratio

Oracle's shared pool is made up of two main memory areas - the Library Cache and the Data Dictionary Cache, and offers a number of tuning possibilities.

The library cache holds commonly used SQL statements - basically database code objects. A method for improving performance in Oracle is to encourage the reuse of SQL statements so expensive parse operations may be avoided. The library cache assists this tuning effort.

The library cache hit ratio offers a key indicator in determining the performance of the shared pool. It shows how often SQL code is being reused by other database users vs. the number of times a SQL statement is broken down, parsed, and then loaded (or reloaded) into the shared pool.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Key Ratio Analysis PaneOracle Performance Analyst Statistics (DBArtisan) > Memory View (DBArtisan - Oracle Performance Analyst) > Key Ratio Analysis pane
Metrics	A high library cache hit ratio is a desirable thing. You should strive for a hit ratio between 95-100%, with 99% being a good performance benchmark for code reuse. Note that when a database is first started, the library cache hit ratio will not be at an optimal level because all code being used will be relatively new, and as such, must be parsed and placed into the shared pool. If, however, after a solid hour or two of steady database time, the library cache hit ratio has not increased to desirable levels, you should look into the possibility of increasing the shared_pool_size parameter. Other red flags that can indicate a too small shared pool include: *A wait count for the event 'latch free' of ten or greaterThe library cache wait count of two or greater.* These indicators can be tracked with Performance Analyst's Bottleneck and Wait detail views. A way of improving the library cache hit ratio is by encouraging code reuse through the implementation of bind variables. Discouraging hard coded literals in application code and instead making use of variables bound at run time aids in the reuse of SQL code that is maintained in Oracle's shared pool. Note that bind variables can have an affect on the cost-based optimizer though. A second way is to pin frequently used code objects in memory so they will always be there when needed. This can be accomplished through the use of the system supplied DBMS_SHARED_POOL package. You can use Performance Analyst to view objects in the shared pool that are always present and/or have increasing reload numbers to help identify objects that are good candidates for pinning.
Correction	If a problem is found, then: *Increase the size of the shared_pool_size parameter through use of the ALTER SYSTEM SET shared_pool_size command value (take caution to not over-allocate; ensure enough free memory exists on server before increasing value).Monitor the new value to see if performance improves.If using an SPFILE, save the new configuration values so Oracle will reuse them each time the database is stopped and re-started.* If you determine that SQL literals are causing SQL to not be reused, you can: *Change the cursor_sharing parameter to FORCE by using the ALTER SYSTEM SET cursor_sharing=FORCE command.Monitor database to see if parse activity is reduced and library cache reloads shrink.* If using an SPFILE, save the new configuration values so Oracle will reuse them each time the database is stopped and re-started. If using an Init.ora file, add the cursor_sharing=FORCE parameter to the file.

Dictionary Cache Hit Ratio

Oracle's shared pool offers an number of tuning possibilities and is made up of two main memory areas:

1 Library Cache

2 Data Dictionary Cache

The dictionary cache hit ratio offers another key indicator in determining the performance of the shared pool. It shows how often object definitions are found in memory vs. having to read them in from disk. Because Oracle references the data dictionary many times when an SQL statement is processed, it is imperative that as much of this vital reference information be kept in RAM as possible.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Key Ratio Analysis PaneOracle Performance Analyst Statistics (DBArtisan) > Memory View (DBArtisan - Oracle Performance Analyst) > Key Ratio Analysis pane
Metrics	Just as with the library cache, a high data dictionary cache hit ratio is desirable. Strive for a hit ratio between 90-100%, with 95% being a good performance benchmark. **NOTE:** When a database is first started, the data dictionary cache hit ratio is not at an optimal level because all references to object definitions are relatively new, and as such, must be placed into the shared pool. Look for hit ratios in the eighty's for new database startups. If, however, after a solid hour or two of steady database time, the data dictionary cache hit ratio has not increased to desirable levels, increase the shared_pool_size parameter. Databases supporting applications that involve large number of objects (such as an Oracle Financials installation) should have larger than normal shared pools to support the required object definitions. Although each parameter is not individually tunable (it was in Oracle6), you can see which area of the dictionary cache could be pulling the overall hit ratio down.
Correction	If a problem is found, do the following: *Increase the size of the shared_pool_size parameter through use of the ALTER SYSTEM SET shared_pool_size command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.Monitor the new value to see if performance improves.If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.*

Memory Sort Ratio

Oracle9i or later has the option of running automatic PGA memory management. Oracle has introduced an Oracle parameter called pga_aggregate_target. When this parameter is set and you are using dedicated Oracle connections, Oracle ignores all of the PGA parameters in the Oracle file, including sort_area_size, hash_area_size and sort_area_retained_size. Oracle recommends that the value of pga_aggregate_target be set to the amount of remaining memory (less a 10% overhead for other server tasks) on a server after the instance has been started.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Key Ratio Analysis PaneOracle Performance Analyst Statistics (DBArtisan) > Memory View (DBArtisan - Oracle Performance Analyst) > Key Ratio Analysis paneOracle Performance Analyst Statistics (DBArtisan) > Users View (DBArtisan - Oracle Performance Analyst) > Home > Key User Analysis Pane
Metrics	If the memory sort ratio falls below 90%, and you are on Oracle9i or later, investigate the use of pga_aggregate_target. Once the pga_aggregate_target has been set, Oracle automatically manages PGA memory allocation, based on the individual needs of each Oracle connection. Oracle9i or later allows the pga_aggregate_target parameter to be modified at the instance level with the alter system command, thereby lets you dynamically adjust the total RAM region available to Oracle9i. Oracle9i also introduced a new parameter called workarea_size_policy. When this parameter is set to automatic, all Oracle connections benefits from the shared PGA memory. When workarea_size_policy is set to manual, connections allocates memory according to the values for the sort_area_size parameter. Under the automatic mode, Oracle tries to maximize the number of work areas that are using optimal memory and uses one-pass memory for the others.
Correction	If you find a problem, do the following: *Edit the Init.ora or SPFILE file for the database.Increase the amount of sort_area_size to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value. EVERY user receives this amount for sorting.Cycle the Oracle server when possible to allow the new value to take effect.Monitor new value to see if performance improves.* In addition to increasing the amount of memory devoted to sorting, find inefficient SQL that cause needless sorts. For example, UNION ALL does not cause a sort whereas UNION does in an SQL query (to eliminate duplicate rows). DISTINCT is frequently misapplied to SQL statements and causes unnecessary sort actions. There are times you simply cannot stop sort activity. This being the case, try to keep it in memory whenever possible. However, large data warehousing systems oftentimes simply exhaust RAM sort allotments, so if disk sorts must occur, try to ensure three things: *Your user's TEMPORARY TABLESPACE assignment is not the SYSTEM tablespace, which is the default assignment.***NOTE:***For Oracle9i or later, you can specify a default tablespace other than SYSTEM for every user account that is created.The TEMPORARY TABLESPACE assigned to your users is placed on a fast disk.The TEMPORARY TABLESPACE has the tablespace parameter TEMPORARY assigned to it, which allows sort activity to be performed in a more efficient manner.*

Parse/Execute Ratio

Each time a new SQL statement is submitted to Oracle, the kernel must ‘parse’ the statement, which involves syntax checks, security checks, and object validations. The Parse/Execute Ratio shows the percentage of SQL executed that did not incur a hard parse.

Note: This statistic is available on the Home page, Memory home page, and the Users home page.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Key Ratio Analysis PaneOracle Performance Analyst Statistics (DBArtisan) > Memory View (DBArtisan - Oracle Performance Analyst) > Key Ratio Analysis paneOracle Performance Analyst Statistics (DBArtisan) > Users View (DBArtisan - Oracle Performance Analyst) > Key User Analysis Pane
Metrics	Seeing low values might indicate that users are firing SQL with many hard-coded literals instead of using bind variables within an application. High values (90% and greater) generally indicate Oracle is saving precious CPU by avoiding heavy parse tasks.
Correction	The best way to reduce unnecessary parse activity is to encourage SQL statement reuse. This can be done by promoting SQL execution through the use of stored procedures or applications where bind variables can be used. Oftentimes, literals in otherwise identical SQL statements can cause unneeded parse work for Oracle. The use of bind variables can counter that problem. If you determine that SQL literals are causing SQL to not be reused, do the following: *Change the cursor_sharing parameter to FORCE by using the ALTER SYSTEM SET cursor_sharing=FORCE command.Monitor database to see if parse activity is reduced and library cache reloads shrink.If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started. If using an Init.ora file, add the cursor_sharing=FORCE parameter to the file.*

Buffer Busy Waits

Buffer busy waits occur when a process needs to access a data block in the buffer cache, but cannot because it is being used by another process. A wait event generally happens because a buffer is being read into the buffer cache by another process or the buffer is in the buffer cache, but cannot be switched to a compatible mode immediately.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Key Ratio Analysis PaneOracle Performance Analyst Statistics (DBArtisan) > Memory View (DBArtisan - Oracle Performance Analyst) > Home > Bottleneck Analysis pane
Metrics	Buffer busy waits normally center around contention for rollback segments, too small an INITRANS setting for tables, or insufficient free lists for tables.
Correction	You can use the automatic segment management feature in Oracle9i locally-managed tablespaces to help make free list problems a thing of the past. Using an UNDO tablespace in 9i or later can help remedy any rollback contention problem. You can also obtain which objects have actually experienced buffer busy waits in Oracle9i or later by querying the sys.v_$segment_statistics. This view is not populated unless the configuration parameter statistics_level is set to TYPICAL or ALL.

Rollback Contention Ratio

Rollback segments are used by Oracle to hold data needed to rollback any changes made through inserts, updates, or deletes to various Oracle objects. They also allow Oracle to have read consistency for long running queries, are used for recovery purposes, and play a role during exports of database information. In a heavy transaction processing environment, rollback segments are accessed continuously and therefore are subject to contention problems. The Rollback Contention Ratio helps identify contention occurring on the system relating to rollbacks.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Key Ratio Analysis PaneOracle Performance Analyst Statistics (DBArtisan) > Objects View (DBArtisan - Oracle Performance Analyst) > Home > Key Ratio Analysis Pane
Metrics	Overall, if the rollback contention ratio approaches 1% or more, create more rollback segments. Also consider creating a specialized, larger, rollback segment to be used by long running transactions. Doing so alleviates dynamic rollback extensions and cuts down heavily on ORA-01555 Snapshot Too Old errors.
Correction	Begin by creating new rollback segments and altering them to be online for use. Then monitor the overall contention ratio to see if it begins to drop. If you are using Oracle9i or later, consider using an UNDO tablespace and allowing Oracle to automatically control rollback segment management.

Latch Miss Ratio

Protecting the many memory structures in Oracle's SGA are latches. They ensure that one and only one process at a time can run or modify any memory structure at the same instant. Much more restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism - so either you get it or you do not and are forced to continually retry.

The latch miss ratio defines the number of times a process obtained a willing-to-wait latch vs. missing the attempt.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Key Ratio Analysis Pane
Metrics	If the latch miss ratio exceeds 1%, you should take action to resolve the amount of latch contention.

Correction

Examine the details regarding the latch contention. Increasing the shared_pool_size can assist in latch problems also. The table below describes latches:

Latch	Description
Cache buffer chain latch	Protects paths to database block buffers in the buffer cache. High I/O loads tend to cause contention for this latch. You can alleviate contention somewhat by adding more buffers to the cache (through the db_block_buffers or db_cache_size parameter) or by adding more LRU latch chain latches with the db_block_lru_latches parameter.
Library cache latches	Protects cached SQL statements in the library cache area of the Oracle shared pool. Contention for this latch is the usual result of literals being used for SQL statements instead of bind variables.

Parallel Query Busy Ratio

Oracle's parallel query feature, when used properly, allows for terrific increases in performance for many SQL and utility operations. Parallel operations can be introduced through SQL hints, specified in an object's DDL, or used in command line arguments for utility programs (like SQL*Loader). To effectively service parallel query requests, ensure that enough query servers exist in the database instance. The Parallel Query Busy Ratio is an indicator of how busy all the servers are on the database in question.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Key Ratio Analysis Pane
Metrics	If the Parallel Query Busy Ratio approaches 80-90%, add more query servers to the database, or examining parallel requests to ensure they are being used in an efficient and necessary manner.
Correction	To fix, do the following: *Edit the Init.ora file or SPFILE for the database.Increase the amount of parallel_max_servers to a higher value.Cycle the Oracle server when possible to allow the new value to take effect.Monitor new value to see if performance improves.* You can also investigate the use of the parallel_automatic_tuning parameter.

Free Shared Pool Percent

Oracle's shared pool need not use all of the memory given to it through the shared_pool_size parameter. If the database does not have many object and code definitions to reference, then the shared pool can contain an abundance of free memory that is not being used.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Key Ratio Analysis PaneOracle Performance Analyst Statistics (DBArtisan) > Memory View (DBArtisan - Oracle Performance Analyst) > Key Ratio Analysis pane
Metrics	Under-allocating the shared pool size can have a serious impact on your database's performance, but over-allocating the shared pool can have run time ramifications as well. If you have a good chunk of memory allocated to the Oracle shared pool that is never used, it might be more of a performance enhancement to reduce the shared pool amount and instead give the memory to the buffer/data cache, or even back to the operating system itself. In terms of knowing when to reduce the shared pool, a good benchmark is continually seeing 2-3MB of free memory. On the other hand, if after an hour or so of beginning database operation, you see that virtually no free memory is left in the shared pool, or you are seeing ORA-4031 errors (that indicate definitions cannot find enough contiguous free space in the shared pool), increase the pool by 10% or more.
Correction	If you continuously see little or no free memory in the shared pool, do the following: *Increase the size of the shared_pool_size parameter through use of the ALTER SYSTEM SET shared_pool_size command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.Monitor the new value to see if performance improves.If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.* You can also use the ALTER SYSTEM FLUSHED SHARED_POOL command to remove all objects from the shared pool and start with a clean slate.

Bottleneck Analysis Pane

When an Oracle database is unning, every connected process is either busy doing work or waiting to perform work. A process that is waiting can mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. And this is where wait-based or bottleneck analysis comes into play. DBAs used this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance because it helps you track where a database has been spending its time. If latch contention or heavy table scan activity has been dragging a database's performance down, you can use bottleneck analysis to confirm the actual root cause. Once one or more wait events or other bottlenecks have been pinpointed as possible performance vampires, you can determine which sessions and objects are causing the problem.

For wait event analysis to be properly carried out, it is imperative that the timed_statistics initialization parameter be set to TRUE. By default this parameter is set to FALSE, which disallows the collection of wait times for each wait event defined to the Oracle engine. For one to really understand the impact of wait events on database performance, you need to discover what the database is or has been waiting on, and the durations of the waits. Having both allows a complete picture to be formed regarding the magnitude of wait-initiated performance degradations. Almost all Oracle experts now agree that allowing the collection of timing statistics adds little if anything to database overhead, so setting timed_statistics to TRUE should not be a worry. The Performance Analyst Home page identifies the top system and session waits that can decrease performance.

When using bottleneck analysis, you cannot rely only on the information contained in the wait event views that Oracle provides. For example, an object can attempt to extend into another extent of space in a tablespace and yet be denied if no such free space exists. Such a failure is not reflected in any wait event, but still represents a very real bottleneck to the database. In the same way that you cannot depend on only a few ratios to properly carry out ratio-based performance analysis, an administrator must include other statistical metrics in their overall bottleneck analysis framework to obtain an accurate performance risk assessment. Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system. The Performance Analyst Home page displays space-related bottlenecks as well as other sources of contention/bottlenecks that can decrease the overall performance of your system.

The following bottleneck indicators are used on the Performance Analyst Home page to succinctly communicate the general overall performance level of the monitored database:

Top System Bottlenecks	Top Session Bottlenecks	Problem Tablespaces
Problem Objects	Current Object Blocks	Enqueue Waits
Free List Waits	Storage Analysis Pane	

Top System Bottlenecks

When viewing wait statistics, there are many levels of detail that you can view. The first level is the system view, which provides a global, cumulative snapshot of all the waits that have occurred on a system. Viewing these numbers can help you determine which wait events caused the most commotion in a database so far. The Top System Bottlenecks section identifies the top waits that have occurred on the Oracle database based on the number of waits per event.

For Oracle 10g only you see a graph that provides a historical look at the last hour and allows you to see what wait classes experienced the most wait activity in terms of time waited. To get detail into which wait events have accumulated the most time, you can drill down into the System Waits view.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Bottleneck Analysis Pane
Correction	Appendix A in the Oracle Reference manual contains a listing and description of every current wait event defined in Oracle. DBAs unfamiliar with what each event represents should keep this listing close by as they examine wait-based event metrics. For example, a 'db file scattered read' event is typically indicative of table scan operations. If you see many of these events, then you can begin to see if large table scans are occurring in the database. Like the 'db file scattered read' event, each wait event has its own meaning and individual end-resolution diagnosis. After looking at system-level wait activity, you can discover which current connections are responsible for waits at the system level. Performance Analyst reports on historical and current wait events at the session level, making this investigation easy to accomplish

Top Session Bottlenecks

When viewing wait statistics, there are many levels of detail that you can view. The first level is the system view, which provides a global, cumulative snapshot of all the waits that have occurred on a system. The second level is a historical look at waits from a session level. The third is which sessions are currently experiencing waits. The Top Session Bottlenecks section identifies the top sessions that are currently waiting, based on their wait time in seconds.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Bottleneck Analysis Pane
Correction	Oracle documentation contains a listing and description of every current wait event defined in Oracle. DBAs unfamiliar with what each event represents should keep this listing close by as they examine wait-based event metrics. The most common wait viewed at the session level is an 'enqueue' wait, which typically identifies lock contention. If enqueue waits are observed, then you can check the "Current Object Blocks" count on the Performance Analyst Home page, as well as the Users View (DBArtisan - Oracle Performance Analyst) which displays locks and blocking locks detail. As with an enqueue event, each wait event has its own meaning and individual end-resolution diagnosis.

Problem Tablespaces

The Problem Tablespaces metric is a count of all tablespaces in the database whose free space percentage amount has fallen below a user-defined limit.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Bottleneck Analysis Pane
Metrics	A rule of thumb for dynamic tablespaces (those with growing objects) is to keep a minimum of 10-15% free space available for object growth.
Correction	There are two ways to prevent a tablespace from running out of available free space: *Turn AUTOEXTEND on for the underlying tablespace's datafiles. This allows them to automatically grow when free space in the datafile has been exhausted.Using the ALTER TABLESPACE … ADD DATAFILE… command, you can manually add a new datafile to a tablespace that is about to run out of available free space.*

Problem Objects

The Problem Objects statistic is a count of all objects in the database that are in danger of reaching their maximum extent limit or cannot allocate their next extent of free space because of a lack of overall or contiguous free space in their parent tablespace.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Bottleneck Analysis Pane
Metrics	Modify any object approaching their maximum extent limit or unable to allocate a new extent of space so they can continue to grow in size.
Correction	Depending on the situation, there are a number of things you can do to prevent object extent problems: *Turn AUTOEXTEND on for the underlying parent tablespace's datafiles. This allows a tablespace to automatically grow when free space in the datafile has been exhausted, and allows an object to extend even when little or no current free space is available.Using the ALTER TABLESPACE … ADD DATAFILE… command, you can manually add a new datafile to a tablespace that is about to run out of available free space.You can alter an object that is at or near their maximum extent limit so that the object has unlimited extents.You can use locally-managed tablespaces to ensure that no object ever reaches its maximum extent limit, because all objects are allowed unlimited extents.An object can be reorganized into another tablespace or reorganized in general to reduce the number of extents the object currently takes up.*

Current Object Blocks

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Although Oracle supports unlimited row-level locking, blocking lock situations do crop up. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency.

Note: This statistic is also called Sessions Blocked on the Users View (DBArtisan - Oracle Performance Analyst) and Session Blocks on the Objects View (DBArtisan - Oracle Performance Analyst)Home page.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Bottleneck Analysis Pane
Metrics	Investigate any indicator above zero immediately before the situation has a chance to mushroom.
Correction	Once discovered, a blocking lock situation can normally be quickly remedied. You can issue a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is where it gets tricky. You can look at the Users View (DBArtisan - Oracle Performance Analyst) detail and view all current blocking locks to see exactly which sessions are holding the currently restrictive locks. The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. Most DBAs who have had to face Oracle Forms applications have suffered through the dreaded SELECT … FOR UPDATE statements that place unnecessary restrictive locks on nearly every read operation, and know all too well that good coding practice is important. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

Enqueue Waits

An enqueue is an advanced locking device that allows multiple database processes to share certain resources. Enqueue waits typically occur when sessions wait to be granted a requested lock. Sometimes these locks are internal Oracle locks while other times they could be locks for rows of data. Enqueues are issued implicitly by Oracle.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Top Session BottlenecksOracle Performance Analyst Statistics (DBArtisan) > Objects View (DBArtisan - Oracle Performance Analyst) > Bottleneck Analysis Pane
Metrics	Investigate any enqueue waits that read consistently above one or more (delta statistics).
Correction	Removing contention for enqueues is almost always an application design issue. If many enqueue waits are seen, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. Examine the use of indexes to make sure all referencing foreign keys are indexes and that SQL is tuned to not tarry over rows in the database during modification operations. Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).

Free List Waits

Free lists are lists of Oracle data blocks that contain free space for an Oracle object. Every table has at least one free list. Free lists are used to locate free blocks of space when a request is made of a table for the insertion of a row. Free list contention can reduce the performance of applications when many processes are involved in the insertion of data to the same table.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Top Session BottlenecksOracle Performance Analyst Statistics (DBArtisan) > Objects View (DBArtisan - Oracle Performance Analyst) > Bottleneck Analysis Pane
Metrics	If consistent numbers for free list waits continues to appear, add additional free lists to the most dynamic database objects through the use of the STORAGE parameter. Another indicator of insufficient free lists are consistent, non-zero numbers for the buffer busy wait event.
Correction	If free list waits are observed, add more free lists to tables and indexes with high insert rates. For Oracle9i or later, objects can be created inside of locally-managed tablespaces that use the automatic segment management feature, which entirely does away with free lists.

Storage Analysis Pane

While DBAs focus on memory settings and tuning SQL, storage problems can play a major role in wrecking an otherwise well-running database.

Storage problems generally take one of two forms:

1 Hit-the-wall, which can bring things to a complete standstill.

2 Performance vampire, which slowly drains the performance of a database over time.

Storage problems have the capability to bring the curtain down on a database very quickly, as in the case of an archive log destination running out of free space. But storage problems can also silently work behind the scenes to slowly but surely rob a database of its performance. For example, you can access a hub table in a busy database when an application is first given life, but over time, if it develops a heavy migrated row problem, it can cause things to run very differently.

The Storage Analysis section of the Performance Analyst Home page reports on the total used and free space of all tablespaces in the database as well as whether the database is running in Archive Log mode. For more detailed information on database space, use the Performance Analyst Space View (DBArtisan - Oracle Performance Analyst).

Total Used Space/Total Free Space

These statistics represent the total used and free space available in all tablespaces/datafiles in the database. Although good to know, a more detailed listing by tablespace is needed to determine where any actual space shortages exist in the database.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Storage Analysis Pane
Metrics	If any one tablespace begins to approach 90% used (and the objects contained within it are dynamic and growing as opposed to static), take action to prevent any future space allocation errors.
Correction	Here are a some things you can do to prevent a tablespace from running out of available free space: *Turn AUTOEXTEND on for the underlying tablespace's datafiles. This allows them to automatically grow when free space in the datafile has been exhausted.Using the ALTER TABLESPACE … ADD DATAFILE… command, you can manually add a new datafile to a tablespace that is about to run out of available free space.*

Archive Log

Oracle can be placed into archivelog mode, which tells the database to make copies of the online redo log files for point-in-time recovery purposes. The Archive Log statistic displays whether the database is running in archivelog mode or not. This information is displayed in the Storage Analysis section of the Performance Analyst Home page. For more information, see Storage Analysis Pane.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Storage Analysis Pane

SQL Analysis Pane

Much of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times. SQL statements that use much memory can also cause a problem in a database.

Before you can identify problem SQL in your database, you have to ask the question, “What is bad SQL?” What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every Oracle professional's satisfaction. The table below lists some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

Criteria	Description
Overall Response (Elapsed) Time	The amount of time the query took to parse, execute, and fetch the data needed to satisfy the query. It should not include the network time needed to make the round trip from the requesting client workstation to the database server. This statistic is available in Oracle9i or later.
CPU Time	The amount of CPU time the query took to parse, execute, and fetch the data needed to satisfy the query.
Physical I/O	This is often used as the major statistic in terms of identifying good vs. bad SQL, this is a measure of how many disk reads the query caused to satisfy the user's request. While you certainly want to control disk I/O where possible, it is important that you not focus solely on physical I/O as the single benchmark of inefficient SQL. Make no mistake, disk access is slower than memory access and also consumes processing time making the physical to logical transition, but you need to look at the entire I/O picture of a SQL statement, which includes looking at a statements' logical I/O as well.
Logical I/O	the number of memory reads the query took to satisfy the user's request. The goal of tuning I/O for a query should be to examine both logical and physical I/O, and use appropriate mechanisms to keep both to a minimum.
Repetition	The number of times the query has been executed. A problem in this area is not as easy to spot as the others unless you know your application well. A query that takes a fraction of a second to execute can still cause a headache on your system if it is executed erroneously (for example, a query that executes in a runaway PL/SQL loop) over and over again.

There are other criteria that you can examine like sort activity or access plan statistics (that show items like Cartesian joins and the like), but more often than not, these measures are reflected in the criteria listed above. Fortunately, Oracle records all the above measures (some only in 9i), which makes tracking the SQL that has been submitted against an Oracle database much easier.

Metrics

When you begin to look for inefficient SQL in a database, there are two primary questions you want answered:

1 What HAS been the worst SQL that has historically run in my database?

2 What IS the worst SQL that is running right now in my database?

When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It can be that the query is in an inefficient PL/SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

There is the possibility that the SQL statement just is not tuned well. To determine that, you can go further into Performance Analyst's Top SQL view and, if you have Embarcadero SQL Tuner installed, you can port the SQL over to SQL Tuner to better optimize the statement.

Workload Analysis Pane

When your database experiences performance slowdowns, you should answer the following:

- Who is currently connected to the database?
- What resources are they using?
- What are they currently executing?

The Workload Analysis section of the Performance Analyst Home page provides insight into the leading resource hogs of a database as well as a count of total active and inactive processes. You can view detailed information into what each leading session is currently involved with.

The Workload Analysis section of the Home page includes:

Top Processes Active User Processes Inactive User Processes

The buffer cache hit ratio is an indicator of how often user requests for data are satisfied through memory vs. being physically read from disk. Data read from memory produces user response times many times faster than when that same data is read from disk. Keeping physical I/Os to an absolute minimum is one of the Oracle buffer cache's purposes in life.

The table below describes the key counters Performance Analyst uses to calculate the buffer cache hit ratio:

Key Counter	Description
DB BLOCK GETS	Data read from memory for DML operations.
CONSISTENT GETS	Data read from rollback segments in memory.
PHYSICAL READS	Data read physically from disk.
Direct Reads	Data read physically from disk that bypasses the buffer cache. Direct reads are filtered out of overall physical reads so an accurate cache hit ratio can be determined.

Dividing the data read from memory by data read from disk yields the cache hit ratio.

Metrics

To help ensure excellent performance, you want to keep your cache hit ratio in the neighborhood of 90% or higher. However, every database has its own ‘personality’ and can exhibit excellent performance with below average readings for the cache hit ratio. Excessive logical I/O activity can produce a very high cache hit ratio while actually degrading overall database performance.

Investigate consistent low readings of 60% or less.

Note: The db_cache_size parameter may need adjusting. The db_cache_size parameter is dynamic and can be altered without stopping and starting the database instance.

Correction

If a problem is found, do the following:

- Increase the size of the db_cache_size parameter through use of the ALTER SYSTEM SET db_cache_size command value (take caution to not over-allocate; ensure enough free memory exists on server before increasing value).
- Monitor the new value to see if performance improves.
- If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

Top Processes

When the database population as a whole experiences a system slowdown, it is not uncommon to find one or two users who are responsible for bringing the system to its knees. In the best of worlds, users have an evenly divided amount of memory usage, disk I/O, CPU utilization, and parse activity. Unfortunately, this is not usually the case. Users submit large batch jobs during peak OLTP activity, or when sessions that are firing off untuned queries on a critical system.

If you are seeing a slowdown in your database, and cannot seem to find a root cause, one thing to examine is the resource consumption of the leading sessions on a system. A handful of users can overwhelm the I/O capabilities of Oracle (through untuned queries or runaway batch jobs) or hammer the CPU or memory structures of the database and server.

Performance Analyst makes it easy to pinpoint the top sessions by showing the leading processes at the physical I/O, logical I/O, memory, and CPU usage levels.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Workload Analysis Pane
Metrics	If any one session uses more than 50% of a total resource (CPU, memory, etc.) go into the session to find out what they are currently executing.

Active User Processes

The Active User Processes statistic is the total number of active and open threads reported in the database. Active Sessions displays the number of processes actively performing work.

For more information, see Inactive User Processes.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Workload Analysis Pane

Inactive User Processes

This metric is the total number of threads logged on to the database that are idle at the current time.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Oracle Performance Analyst)** > Workload Analysis Pane
Metrics	A large number of inactive users could indicate user sessions that have mistakenly been left logged on. Because each user thread consumes a portion of memory on the Oracle server, sever any sessions not needing a connection to reduce resource usage.
Correction	Double-click this statistic to open the Top Sessions Tab of the Users Detail page. On this tab you can check sessions that have many seconds idle and/or that have been logged on for very long periods of time as indicated by the logon time column. After verifying that the session is no longer necessary, you can KILL the session. For more information, see Active User Processes.

Memory View (DBArtisan - Oracle Performance Analyst)

In addition to a Home page, the Memory category of Oracle Performance Analyst includes the following tabbed pages:

Buffer Cache Tab Buffer Pools Tab Dictionary Cache Tab

Latch Activity Tab Library Cache Tab Shared Pool Tab

PGA Tab

Home

The Memory home page includes the following sections:

Bottleneck Analysis pane Key Ratio Analysis pane SGA Analysis Pane

SQL Analysis Pane Workload Analysis Pane

Key Ratio Analysis pane

Object-related database activity can be examined using both ratio-based and wait/bottleneck-based analysis. Ratio-based analysis involves examining a number of key database ratios and statistical readings that can be used to indicate how active certain object types are. Performance ratios serve as roll-up mechanisms for busy DBAs to use for at-a-glance performance analysis.

When using ratio-based analysis, there are some standards to adhere to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that you examines come from the v$sysstat performance view. The performance view maintains a count of all the occurrences (in the VALUE column) of a particular database incident (in the NAME column) since the database was brought up. For databases that are kept up for long periods of time, these values can grow quite large and impacts how a particular ratio is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each database’s individual ‘personality’ with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead you to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following memory ratios are used on the Performance Analyst Memory home page to succinctly communicate the general overall memory performance levels of the monitored database. They duplicate statistics shown on the Home page. For detailed descriptions of those statistics, see Key Ratio Analysis Pane.

Buffer Cache Hit Ratio
Library Cache Hit Ratio
Dictionary Cache Hit Ratio
Memory Sort Ratio
Free Shared Pool Percent
Parse/Execute Ratio

Bottleneck Analysis pane

When an Oracle database is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. You can use Bottleneck Analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Memory bottlenecks can definitely cause performance degradation in an otherwise well-running database. Typically, these bottlenecks center around Oracle's buffer/data cache, library cache, and occasionally log buffer memory regions. To help you identify such problems, the following statistics are presented on the Memory home page:

Free Buffer Wait Average
Object Reloads
Redo Log Space Waits
Redo Log Space Wait Time
Top Latch Misses

The following statistic, available on this pane, duplicates a statistic available on the Key Ratio Analysis Pane of the Home View (DBArtisan - Oracle Performance Analyst) page:

- Buffer Busy Waits

Free Buffer Wait Average

There are times when a user must wait to obtain a clean buffer in the buffer/data cache. The free buffer wait average provides a metric that communicates the average number of tree buffer waits..

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis pane
Metrics	Seeing consistent non-zero numbers for the free buffer wait average might indicate an undersized data/buffer cache.
Correction	If the Object Reloads statistic, along with other shared pool metrics, indicates an undersized shared pool, you can do the following depending on the version of Oracle you are running: If a problem is found: *Increase the size of the shared_pool_size parameter through use of the ALTER SYSTEM SET shared_pool_size command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.Monitor the new value to see if performance improves.If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.* If you suspect that there are hot code objects that would benefit from being pinned into the shared pool, then you can go into shared pool details using Performance Analyst's monitors. You can easily spot the general code object types or specific objects that should be pinned in the pool from these monitors. Once identified, you can use the dbms_shared_pool package to pin the hot code objects into RAM.

Object Reloads

When a SQL statement or code object (stored procedure, trigger, etc.) is executed for the first time, Oracle loads the code definition into the shared pool. If the statement or code object is executed again, Oracle can reuse the object definition in memory, and thereby shorten execution time.

If a code object is aged out of the shared pool, and then re-requested, Oracle must load the object back into the pool. Such activity can lead to reduced response times. The Object Reloads statistic provides you with a count of how many objects have been reloaded back into the shared pool.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis pane
Metrics	Seeing consistent non-zero numbers for object reloads could indicate a too small shared pool or that help identify that there are hot code objects that should be pinned in the shared pool so they cannot be removed.
Correction	If the Object Reloads statistic, along with other shared pool metrics, indicates an undersized shared pool, you can do the following depending on the version of Oracle you are running: If a problem is found: *Increase the size of the shared_pool_size parameter through use of the ALTER SYSTEM SET shared_pool_size command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.Monitor the new value to see if performance improves.If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.* If you suspect that there are hot code objects that would benefit from being pinned into the shared pool, then you can go into shared pool details using Performance Analyst's monitors. You can easily spot the general code object types or specific objects that should be pinned in the pool from these monitors. Once identified, you can use the dbms_shared_pool package to pin the hot code objects into RAM.

Redo Log Space Waits

The Oracle database is able to manage recovery by recording all changes made to a database through the use of redo log files. Oracle writes modifications made to a database to these files that have the capability to be archived off to another medium for disaster recovery. The background process that performs these operations is Oracle's Log Writer (LGWR). There is a buffer area in Oracle's System Global Area (SGA) that is used to reduce redo log file I/O, whose size, or lack thereof, can affect performance in a busy system. Sometimes a user process must wait for space in this redo log buffer. Oracle uses the log buffer area to cache redo entries prior to writing them to disk, and if the buffer is not large enough for the redo entry load, waits can occur..

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis pane
Metrics	The two main numbers to watch are the redo log space requests and the redo log wait time. If either statistic strays too far from zero, increase the log_buffer parameter and add more memory to the redo log buffer.
Correction	To fix, do the following: *Edit the Init.ora file or SPFILE for the database.Increase the amount of log_buffer to a higher value (take caution to not over-allocate; ensure enough free memory exists on server before increasing value.Cycle the Oracle server when possible to allow the new value to take effect.Monitor new value to see if performance improves.* On some Oracle platforms, when adjusting the log_buffer parameter, make sure you make the amount a multiple of the block size. Otherwise, on database startup, Oracle returns an error stating that you have entered an invalid amount for the redo log buffer. **NOTE:** Oracle increases the log_buffer parameter if you make it smaller than its default size for a given platform.

Redo Log Space Wait Time

The Oracle database is able to manage recovery by recording all changes made to a database through the use of redo log files. Oracle writes modifications made to a database to these files that have the capability to be archived off to another medium for disaster recovery. The background process that performs these operations is Oracle's Log Writer (LGWR). There is a buffer area in Oracle's System Global Area (SGA) that is used to reduce redo log file I/O, whose size, or lack thereof, can affect performance in a busy system. Sometimes a user process must wait for space in this redo log buffer. Oracle uses the log buffer area to cache redo entries prior to writing them to disk, and if the buffer is not large enough for the redo entry load, waits can occur.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis pane
Metrics	The two main numbers to watch are the redo log space requests and the redo log wait time. If either statistic strays too far from zero, increase the log_buffer parameter and add more memory to the redo log buffer.
Correction	To fix, do the following: *Edit the Init.ora file or SPFILE for the database.Increase the amount of log_buffer to a higher value (take caution to not over-allocate; ensure enough free memory exists on server before increasing value.Cycle the Oracle server when possible to allow the new value to take effect.Monitor new value to see if performance improves.* On some Oracle platforms, when adjusting the log_buffer parameter, make sure you make the amount a multiple of the block size. Otherwise, on database startup, Oracle returns an error stating that you have entered an invalid amount for the redo log buffer. **NOTE:** Oracle increases the log_buffer parameter if you make it smaller than its default size for a given platform.

Top Latch Misses

Latches protect the many memory structures in Oracle's SGA. They ensure that one and only one process at a time can run or modify any memory structure at the same instant. More restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism.

The Top Latch Misses chart indicates the top latch waits as ranked by total misses.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis pane
Metrics	Seeing increasing numbers for the same latch miss can require further accessing session latch details, which can be accomplished inside Performance Analyst.
Correction	The remedy for latch misses depends on the latch itself. Buffer chain latches, for example, are indicative of high I/O rates in databases. Increasing the buffer/data cache can help as might adding more buffer LRU chain latches by increasing the db_block_lru_latches parameter. Library cache latch misses are oftentimes produced by non-reused SQL in the shared pool. Increasing the shared_pool_size can help, but a better remedy might be changing SQL to use bind variables or setting the cursor_sharing parameter to FORCE.

SQL Analysis Pane

Much of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times. SQL statements that use much memory can also cause a problem in a database. The SQL Analysis for memory shows what SQL statements have consumed the largest percentages of shareable, persistent, and runtime memory

Before you can identify problem SQL in your database, you have to ask the question, "What is bad SQL?" What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every Oracle professional's satisfaction. The table lists some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

Criteria	Description
Overall Response (Elapsed) Time	The time the query took to parse, execute, and fetch the data needed to satisfy the query. It should not include the network time needed to make the round trip from the requesting client workstation to the database server. This statistic is available in Oracle9i or later.
CPU Time	The CPU time the query took to parse, execute, and fetch the data needed to satisfy the query.
Physical I/O	This is often used as the major statistic in terms of identifying good vs. bad SQL, this is a measure of how many disk reads the query caused to satisfy the user's request. While you certainly want to control disk I/O where possible, it is important that you not focus solely on physical I/O as the single benchmark of inefficient SQL. Make no mistake, disk access is slower than memory access and also consumes processing time making the physical to logical transition, but you need to look at the entire I/O picture of a SQL statement, which includes looking at a statements' logical I/O as well.
Logical I/O	The number of memory reads the query took to satisfy the user's request. The goal of tuning I/O for a query should be to examine both logical and physical I/O, and use appropriate mechanisms to keep both to a minimum.
Repetition	The number of times the query has been executed. A problem in this area is not as easy to spot as the others unless you know your application well. A query that takes a fraction of a second to execute can be a headache on your system if it has executed erroneously (for example, a query that executes in a runaway PL/SQL loop) over and over again.

There are other criteria that you can examine like sort activity or access plan statistics (that show items like Cartesian joins and the like), but more often than not, these measures are reflected in the criteria listed above. Fortunately, Oracle records all the above measures (some only in 9i), which makes tracking the SQL that has been submitted against an Oracle database much easier.

Metrics

When you begin to look for inefficient SQL in a database, there are two primary questions you want answered:

1 What HAS been the worst SQL that has historically been run in my database?

2 What IS the worst SQL that is running right now in my database?

When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It could be that the query is in an inefficient PL/SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

There is the possibility that the SQL statement just is not tuned well. You can view Performance Analyst's Top SQL view and, if you have Embarcadero SQL Tuner installed, you can port the SQL over to SQL Tuner to better optimize the statement.

SGA Analysis Pane

Most DBAs know all about the Oracle System Global Area (SGA). The SGA is Oracle's memory structural area devoted to facilitating the transfer of data and information between clients and the Oracle database. The table below describes Oracle memory structures:

Memory Structure	Description
Default buffer cache	Maintains data blocks when they are read from the database. If you do not specifically place objects in another data cache, then any data requested by clients from the database is placed into this cache. The memory area is controlled by the db_cache_size parameter.
Keep buffer cache	You can assign various objects to a special cache that retains those object's requested blocks in RAM for as long as the database remains up. The keep cache's main is for often-referenced lookup tables that should be kept in memory at all times for fast access. The buffer_pool_keep parameter controls the size of this cache in Oracle8, while the db_keep_cache_size parameter handles the cache in Oracle9i or later. The keep pool is a sub-pool of the default buffer cache.
Recycle buffer cache	The opposite of the keep cache. When large table scans occur, the data that fills a memory cache will likely not be needed again and should be quickly discarded from RAM so that they do not occupy memory space and prevent needed blocks from assuming their place. Objects containing such data can be assigned to the recycle pool to ensure that such a thing does indeed occur. The buffer_pool_recycle parameter controls the size of this cache in Oracle8 and earlier, while the db_recycle_cache_size parameter handles the cache in Oracle9i or later.
Specific block size caches	You can create tablespaces whose blocksize differs from the overall database blocksize. When data is read into the SGA from these tablespaces, their data has to be placed into memory regions that can accommodate their special block size. Oracle9i or later has memory settings for 2K, 4K, 8K, 16K, and 32K caches. The configuration parameter names are in the pattern of db_nk_cache_size.
Shared pool	Holds object structure as well as code definitions, and other metadata. Setting the proper amount of memory in the shared pool assists a great deal in improving overall performance with respect to code execution and object references. The shared_pool_size parameter controls the memory region.
Large pool	You can configure an optional, specialized memory region called the large pool that holds items for shared server operations, backup and restore tasks, and other miscellaneous things. The large_pool_size parameter controls the memory region. The large pool is also used for sorting when the multi-threaded server (MTS) is implemented.
Java pool	Handles the memory for Java methods, class definitions, etc. The java_pool_size parameter controls the amount of memory for this area.
Redo log buffer	Buffers modifications that are made to the database before they are physically written to the redo log files. The log_buffer configuration parameter controls the memory area.

Oracle also maintains a fixed area in the SGA that contains a number of atomic variables, pointers, and other miscellaneous structures that reference areas of the SGA.

Workload Analysis Pane

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. Oftentimes, user connections can get out of hand with memory consumption, and extreme cases have caused headaches at both the database and operating system level (ORA-4030 errors).

If your server does not have an overabundance of memory, periodically check to see who your heavy memory users are along with the total percentage of memory each takes up. If you see one or two users who have more than 25-50% of the total memory usage, investigate the sessions further to see the activities they are performing.

Buffer Cache Tab

The Buffer Cache tab includes the following sections:

Buffer Cache Hit Ratio Buffer Cache Utilization Cached Blocks by Tablespace

Cached Tables

Cached Blocks by Tablespace

Oracle's buffer/data cache contains blocks read in from disk at the user's request. The Cached Blocks by Tablespace section displays the amount of data held currently in the cache by tablespace.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > Buffer Cache Tab

Cached Tables

One of the ways you can help data from a table stay in the buffer/data cache as long as possible is to enable the table's CACHE property. CACHE specifies that blocks retrieved for a table be placed at the most recently used end of the LRU list in the buffer cache when a full table scan is performed. This is particularly useful for small lookup tables.

The Cached Tables section displays the data for all tables that have their CACHE property enabled. The table below lists the information available in this section:

Column	Description
Owner	The owner of the table.
Table Name	The name of the table.
Table Size	The size of the table in bytes.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > Buffer Cache Tab
Metrics	CACHE is best suited for small lookup tables. Therefore, if you see a large table that has been cached, you may want to reevaluate whether it is smart to cache the contents of that table in Oracle's buffer/data cache.
Correction	If you wish to disable the CACHE property of a table, you can alter the table and set the property to false.

Buffer Cache Utilization

The Buffer Cache Utilization section shows the state and use of the buffers currently in the buffer cache along with a count of how many there are for each state. Typically, the statuses of the buffers are: being read, free, read and modified, and read and not modified.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > Buffer Cache Tab
Metrics	This grouping of statistics is quite helpful in determining if you have an overabundance of block buffers allocated. Consistently seeing large number of free buffers will clue you in to the fact that you can reduce the amount of block buffers in the cache and give memory back to the operating system. If, however, you see no free buffers within the first hour of bringing your Oracle database up, then you may consider adding more buffers to the cache.
Correction	If a problem is found, then: *Edit the Init.ora file or spfile for the database.Increase the amount of db_cache_size to a higher value if free buffers are not found (take caution to not over-allocate; ensure enough free memory exists on server before increasing value). Reduce the number of an overabundance of FREE buffers are present.Monitor new value to see if more numbers of free buffers show up for the FREE status.*

Dictionary Cache Tab

Oracle's data dictionary is a component of the shared pool that contains system elements necessary for the parsing of SQL statements, security resolution, object definitions, and more.

The overall data dictionary cache hit ratio provides a key indicator in determining the performance of the shared pool, and shows how often object definitions are found in memory vs. having to read them in from disk. Because Oracle references the data dictionary many times when an SQL statement is processed, it is imperative that as much as possible of this vital reference information be kept in RAM.

The Dictionary Cache tab shows the individual elements of the data dictionary along with their associated hit ratios. In versions 6.x of Oracle, these individual elements could be tuned, but in versions 7.x and later, the only main method for tuning them involves the adjustment of the entire shared pool setting. Although not tunable from an individual parameter level, each displayed element gives insight into which area of the data dictionary is either adding to or detracting from overall performance.

The table below describes the information available on this tab:

Column	Description
Parameter	The name of the individual data dictionary element.
Usage	The number of cache entries that contain valid data.
Gets	The number of requests for this element.
Get Misses	The number of requests resulting in a cache miss.
Hit Ratio	The ratio of cache hits versus misses of total requests. The maximum is 100%.
Scans	The number of scan requests.
Scan Misses	The number of times that a scan failed to find the needed data in the cache.
Scan Completes	The number of times the list was scanned completely.
Modifications	The number of insert, update, and delete actions.
Flushes	The number of disk flushes.

Metrics

An overall high data dictionary cache hit ratio is desirable, as are high hit ratios in each individual parameter. You should strive for a hit ratio between 90-100%, with 95% being a good performance benchmark.

Note: When a database is first started, the data dictionary cache hit ratio is not at an optimal level because all references to object definitions are relatively new, and as such, must be placed into the shared pool. Look for hit ratios between 80-90% for new database startups. If, however, after an hour or two of steady database time, the data dictionary cache hit ratio and individual hit ratios, have not increased to desirable levels, you should look into the possibility of increasing the shared_pool_size parameter.

Note: Databases supporting applications that involve large number of objects (such as an Oracle Financials installation) should have larger than normal shared pools to support the required object definitions.

Buffer Pools Tab

The Buffer Pools tab includes the following sections:

Buffer Pool Hit Ratio Buffer Pool Memory Configuration Buffer Pool Object Assignments

Buffer Pool Hit Ratio

Because data objects can be referenced with different usage patterns, Oracle8 and later offers the option to intelligently place objects into one of three buffer caches. The table below describes the types of buffer caches:

Cache	Description
KEEP	Designed to minimize misses in the buffer cache. Small objects that are frequently referenced are candidates for the KEEP buffer pool.
RECYCLE	Designed to avoid having large numbers of infrequently accessed blocks of data crowd out objects that need to be referenced in RAM, the RECYCLE pool is good for large objects that are scanned from beginning to end without the need for keeping all their data in RAM.
DEFAULT	The traditional cache for all other data objects.

Note: Unless you specify the KEEP or RECYCLE buffer cache, Oracle automatically places objects into the DEFAULT buffer cache.

The Buffer Pool Hit Ratio section the hit ratios for each of the Oracle8 and later buffer caches so you can easily see how often the objects placed into the various caches are being referenced in memory. Examining how often data is satisfied from memory vs. disk will help you determine if the caches are large enough and if they are being used in an optimal manner. The table below describes the information available in this section:

Column	Description
Buffer pool name	The name of the Oracle buffer pool.
Buffer pool hit ratio	The overall hit ratio for the particular cache.

Note: The DEFAULT buffer cache will only be shown for those installations not using the specialized caches available in Oracle8 and later.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > Buffer Pools Tab
Metrics	The KEEP buffer pool should maintain a hit ratio as close to 100% as possible. However, the buffer pool hit ratio is not near 100% until the database has been up and running in a typical state for a short time. A poor hit ratio for the RECYCLE buffer pool may not be a bad thing since there is little chance of reusing a block stored in the buffer pool before it is aged out. **NOTE:** If you place objects into the KEEP buffer pool, you should periodically reexamine their object sizes to ensure that they are not growing to a physical state that will jeopardize the performance of the KEEP pool.

Buffer Pool Object Assignments

Because data objects can be referenced with different types of usage patterns, Oracle8 and later offers the option to intelligently place objects into one of three buffer caches. The table below describes these three buffer caches:

Cache	Description
KEEP	Designed to minimize misses in the buffer cache. Small objects that are frequently referenced are candidates for the KEEP buffer pool.
RECYCLE	Designed to avoid having large numbers of infrequently accessed blocks of data crowd out objects that need to be referenced in RAM, the RECYCLE pool is good for large objects that are scanned from beginning to end without the need for keeping all their data in RAM.
DEFAULT	The traditional cache for all other data objects.

Objects not specified to go into either the KEEP or RECYCLE caches will automatically be placed into the DEFAULT buffer cache.

The Buffer Pool Object Assignments section provides a summary with respect to which types of objects have been assigned to the various buffer caches. The table below describes the information available in this section:

Column	Description
Object Type	The type of object (table, index, etc.).
Buffer Pool	The buffer cache name (KEEP, etc.).
Total Objects Assigned	The number of objects that have been assigned to the particular cache.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > Buffer Pools Tab
Metrics	When looking at overall database I/O activity, you should keep an eye out for objects that may lend themselves to being placed into a particular buffer cache. Consider using a KEEP cache for relatively small, frequently accessed tables that require fast response times. Large tables with random I/O activity and are scanned from beginning to end a lot are good candidates for a RECYCLE cache.

Buffer Pool Memory Configuration

Because data objects can be referenced with different types of usage patterns, Oracle8 and later offers the option to intelligently place objects into one of three buffer caches. The table below describes these three buffer caches:

Cache	Description
KEEP	Designed to minimize misses in the buffer cache. Small objects that are frequently referenced are candidates for the KEEP buffer pool.
RECYCLE	Designed to avoid having large numbers of infrequently accessed blocks of data crowd out objects that need to be referenced in RAM, the RECYCLE pool is good for large objects that are scanned from beginning to end without the need for keeping all their data in RAM.
DEFAULT	The traditional cache for all other data objects.

Objects not specified to go into either the KEEP or RECYCLE caches will automatically be placed into the DEFAULT buffer cache.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > Buffer Pools Tab
Metrics	When looking at overall database I/O activity, you should keep an eye out for objects that may lend themselves to being placed into a particular buffer cache. Consider using a KEEP cache for relatively small, frequently accessed tables that require fast response times. Large tables with random I/O activity and are scanned from beginning to end a lot are good candidates for a RECYCLE cache.
Correction	Objects can be placed into different buffer pools at object creation time (using the STORAGE...BUFFER_POOL option) or existing objects can be set into a different pool with the ALTER command. Note that tables, partitions, and indexes may be placed into the different caches. If you just want to use the DEFAULT buffer pool and not enable any special caches, you can still encourage Oracle to keep certain objects in the buffer cache as long as possible using the CACHE parameter. For example, issuing the command ALTER TABLE … CACHE specifies that the blocks retrieved for this table be placed at the most recently used end of the LRU list in the DEFAULT buffer cache when a full table scan is performed. The CACHE hint may also be used in SQL statements to cache a table, but used in this form, the blocks will only be cached until the next time the database is shut down. Once the database comes back up, the CACHE hint would have to be issued in a SQL statement again to cache the needed blocks of data.

Latch Activity Tab

The Latch Activity tab includes the following sections:

Immediate Latch Hit Ratio Latch Detail Latch Hit Ratio

Latch Sleep Ratio

Latch Hit Ratio

Protecting the many memory structures in Oracle's SGA are latches. They ensure that one and only one process at a time can run or modify any memory structure at the same instant. Much more restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism so either you get it or you do not and are forced to continually retry. The latch hit ratio defines the number of times a process obtained a willing-to-wait.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > Latch Activity Tab
Metrics	If the latch hit ratio falls below 99%, then action should be taken to resolve the amount of latch contention occurring.

Correction

The details regarding the latch contention should be examined. Increasing the shared_pool_size can assist in latch problems also. There are a few specialized cases of latch contention that can be rectified pretty easily. The table below describes them:

Contention	Description
Cache buffer chain latch	This latch is responsible for protecting paths to database block buffers in the buffer cache. Very high I/O loads tend to cause contention for this latch. You can alleviate contention somewhat by adding more buffers to the cache (through the db_block_buffers/db_cache_size parameter) or by adding more LRU latch chain latches with the db_block_lru_latches parameter.
Library cache latches	Protects cached SQL statements in the library cache area of the Oracle shared pool. Contention for this latch is the usual result of literals being used for SQL statements instead of bind variables.

Immediate Latch Hit Ratio

Protecting the many memory structures in Oracle's SGA are latches. They ensure that one and only one process at a time can run or modify any memory structure at the same instant. Much more restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism so either you get it or you do not and are forced to continually retry. The latch immediate miss ratio defines the number of times a process obtained a not-willing-to-wait latch.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > Latch Activity Tab
Metrics	If the value falls below 99%, then action should be taken to resolve the amount of latch contention occurring.

Correction

The details regarding the latch contention should be examined. Increasing the shared_pool_size can assist in latch problems also. There are a few specialized cases of latch contention that can be rectified pretty easily. The table below describes them:

Contention	Description
Cache buffer chain latch	This latch is responsible for protecting paths to database block buffers in the buffer cache. Very high I/O loads tend to cause contention for this latch. You can alleviate contention somewhat by adding more buffers to the cache (through the db_block_buffers/db_cache_size parameter) or by adding more LRU latch chain latches with the db_block_lru_latches parameter.
Library cache latches	Protects cached SQL statements in the library cache area of the Oracle shared pool. Contention for this latch is the usual result of literals being used for SQL statements instead of bind variables.

Latch Sleep Ratio

Protecting the many memory structures in Oracle's SGA are latches. They ensure that one and only one process at a time can run or modify any memory structure at the same instant. Much more restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism so either you get it or you do not and are forced to continually retry.

A sleep indicates that a latch could not be obtained for a process, and that the attempt will be retried. A low ratio indicates many processes that had to sleep multiple times before obtaining a requested latch.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > Latch Activity Tab
Metrics	Sleeps should be kept as low as possible. If the overall sleep ratio falls below 99%, then action should be taken to resolve the amount of latch contention that is occurring.

Correction

The details regarding the latch contention should be examined. Increasing the shared_pool_size can assist in latch problems also. There are a few specialized cases of latch contention that can be rectified pretty easily. The table below describes them:

Contention	Description
Cache buffer chain latch	This latch is responsible for protecting paths to database block buffers in the buffer cache. Very high I/O loads tend to cause contention for this latch. You can alleviate contention somewhat by adding more buffers to the cache (through the db_block_buffers/db_cache_size parameter) or by adding more LRU latch chain latches with the db_block_lru_latches parameter.
Library cache latches	Protects cached SQL statements in the library cache area of the Oracle shared pool. Contention for this latch is the usual result of literals being used for SQL statements instead of bind variables.

Latch Detail

Protecting the many memory structures in Oracle's SGA are latches. They ensure that one and only one process at a time can run or modify any memory structure at the same instant. Much more restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism so either you get it or you do not and are forced to continually retry.

The Latch Detail section presents a detailed view of latch activity. The table below lists the information available:

Column	Description
Latch Name	The name of the latch.
Gets	The number of times the latch was requested by a process.
Misses	The number of failed attempts to acquire the latch on the first attempt.
Immediate Gets	The number of nowait requests for a latch.
Immediate Misses	The number of failed nowait attempts to acquire the latch on the first attempt.
Sleeps	The number of requests that paused while waiting for a latch.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > Latch Activity Tab

Metrics

Common indicators of latch contention are a latch miss ratio (which records willing-to-wait mode latch requests) and latch immediate miss ratio (which records no-wait mode latch requests). These statistics reflect the overall health of how often latch requests were made and satisfied without waiting. If either of these exceed 1%, then latch contention may be causing system slowdowns.

There are a few latch contention situations that you can recognize and get to the bottom of quickly. The table below describes them:

Contention Situation	Description
Cache buffer chain latch	This latch is responsible for protecting paths to database block buffers in the buffer cache. Very high I/O loads tend to cause contention for this latch. You can alleviate contention somewhat by adding more buffers to the cache (through the db_block_buffers parameter) or by adding more LRU latch chain latches with the db_block_lru_latches parameter.
Library cache latches	This latch protects cached SQL statements in the library cache area of the Oracle shared pool. Contention for this latch is the usual result of literals being used for SQL statements instead of bind variables.

Library Cache Tab

The Library Cache tab includes the following sections:

Library Cache Hit Ratio Library Cache Performance Pinned Objects

Library Cache Performance

The library cache holds commonly used SQL statements - basically database code objects. A method for improving performance in Oracle is to encourage the reuse of SQL statements so expensive parse operations may be avoided. The library cache assists this tuning effort.

The Library Performance sectioned tab provides insight into how efficiently the library cache is operating. The table below describes the information available in this section:

Column	Description
Namespace	The region of the library cache.
Gets	The number of times a lock was requested for objects in the particular namespace.
Get Hit Ratio	The percentage of times (with 100% being the maximum) that the object was found in the cache.
Pins	The number of times a pin was requested for objects of this namespace.
Pin Hit Ratio	The percentage of times (with 100% being the maximum) that pin requests were successful.
Reloads	The number of times a piece of the object had to be brought back in from disk to the cache, most likely because it was flushed from the shared pool.
Invalidations	The number of times objects in this namespace were marked invalid because a dependent object was modified.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > Library Cache Tab
Metrics	**NOTE:** An overall high library cache hit ratio is a desirable thing. You should strive for a hit ratio between 95-100%, with 99% being a good performance benchmark for code reuse. When a database is first started, the library cache hit ratio, along with the individual region hit ratios, will not be at an optimal level because all code being used will be relatively new, and as such, must be parsed and placed into the shared pool. If, however, after a solid hour or two of steady database time, the library cache hit ratio has not increased to desirable levels, you should look into the possibility of increasing the shared_pool_size parameter. To keep important code objects from being aged out of the library cache, you can use the DBMS_SHARED_POOL package to pin frequently used code objects in memory so they will always be there when needed.

Pinned Objects

To keep important code objects from being aged out of the library cache, you can use the DBMS_SHARED_POOL package to pin frequently used code objects in memory so they will always be there when needed. The Pinned Objects section displays all objects in the library cache that have pinned. The table below lists the information available in this section:

Column	Description
Owner	The user account that owns the object.
Name	The name of the object.
Type	The type of the object: INDEX, TABLE, CLUSTER, VIEW, SET, SYNONYM, SEQUENCE, PROCEDURE, FUNCTION, PACKAGE, PACKAGE BODY, TRIGGER, CLASS, OBJECT, USER, DBLINK.
Shareable Memory	The amount of memory consumed by the object in the shared pool.
Loads	The number of times the object has been loaded into the cache. Note that this count also increases when an object has been invalidated.
Executions	The number of times the object has been executed by a session thread.
Locks	The number of users actively locking the object.
Pins	The number of user actively pinning the object.
Pinned?	Indicates (YES or NO) if this object has been pinned in memory with the DBMS_SHARED_POOL package.

Shared Pool Tab

The main components of the shared pool memory are the library and dictionary caches. Many of the caches in the shared pool increase or decrease in size automatically as necessary. The Shared Pool tab of the Memory Detail view presents the following sections:

Shared Space Pool Utilization - a graphical representation of the amount of used and free memory.

Shared Pool Fragmentation - displays the following statistics:

- Bucket: A work area defined by its optimal memory requirement.
- Free Space: The amount of free space in the bucket.
- Free Chunks: The number of free chunks that are available to be allocated from the shared pool.
- **Average Size:** The average size of a chunk.
- **Largest:** The size of the largest chunk.

Shared Pool Usage - The Shared Pool Usage section shows the following statistics:

- **Shared Pool Component Name**
- **Amount Used (bytes)**
- **Percent Used**

PGA Tab

The PGA tab displays statistical measures and techniques that let you manage a session's Program Global Area.

Note: This information is available for Oracle9i and later.

The PGA tab includes the following sections:

PGA Overview PGA Workarea Summary PGA Session Detail

For more information, see **Memory View (DBArtisan - Oracle Performance Analyst)**.

PGA Overview

Oracle's Program Global Area (PGA) is a private memory region containing data and control information for every connected process. Examples of PGA information are the runtime area of a cursor and memory sort space. If shared servers are used, then part of the runtime area can be located in the SGA.

Intensive queries cause a big portion of the PGA to be dedicated to work areas supporting memory intensive operators, such as the following:

- Sort-based operators (for example, ORDER BY, GROUP BY, ROLLUP, window functions)
- Hash-join
- Bitmap merge
- Bitmap create
- Write buffers used by bulk load operations

A sort operator uses a sort area to perform the in-memory sort of a set of rows. Similarly, a hash-join operator uses a hash area to build a hash table.

Note: This information is available for Oracle9i and later.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > PGA Tab
Metrics	The size of a PGA area can be controlled and tuned. Larger work areas can significantly improve performance of session work, with the trade off being higher memory consumption. The size of a session's PGA should be big enough so that it can accommodate the input data and auxiliary memory structures allocated by its requested SQL actions. This is known as the optimal size of a PGA work area. When the size of a PGA work area is smaller than optimal, the response time increases, because an extra pass is performed over part of the input data. This is known as the one-pass size of the PGA work area. Under the one-pass threshold, when the size of a work area is smaller than its input data size, multiple passes over the input data are needed. This could drastically increase the response time of the session. This is known as the multi-pass size of the work area. The goal is to have most work areas running with an optimal size (more than 90% or more for OLTP systems), while a smaller amount are running with a one-pass size (less than 10%). Multi-pass execution should be completely avoided. Prior to Oracle9i, the PGA was controlled using the SORT_AREA_SIZE, HASH_AREA_SIZE, BITMAP_MERGE_AREA_SIZE and CREATE_BITMAP_AREA_SIZE parameters. Setting these parameters is difficult, because the maximum PGA size is ideally selected based on data input size and total number of work areas active in the system. These two factors vary from one work area to another and from one point in time to another.

Correction

With Oracle9i, you can simplify and improve the way PGA memory is allocated, by enabling automatic PGA memory management. In this mode, Oracle dynamically adjusts the size of the portion of the PGA memory dedicated to work areas, based on an overall PGA memory target explicitly set by the DBA. To enable automatic PGA memory management, you have to set the initialization parameter PGA_AGGREGATE_TARGET.

The table below describes two metrics you should to watch:

Metric	Description
Over allocation count	Over-allocating PGA memory can happen if the value of PGA_AGGREGATE_TARGET is too small to accommodate the PGA other component in the previous equation plus the minimum memory required to execute the work area workload. When this happens, Oracle cannot honor the initialization parameter PGA_AGGREGATE_TARGET, and extra PGA memory needs to be allocated. If over-allocation occurs, you should increase the value of PGA_AGGREGATE_TARGET.
Cache hit percentage ORACLE 9.2 AND LATER	This metric is computed by Oracle to reflect the performance of the PGA memory component. A value of 100% means that all work areas executed by the system since instance start-up have used an optimal amount of PGA memory. When a work area cannot run optimally, one or more extra passes is performed over the input data. This reduces the cache hit percentage in proportion to the size of the input data and the number of extra passes performed.

PGA Workarea Summary

Oracle's Program Global Area (PGA) is a private memory region containing data and control information for every connected process. Examples of PGA information are the runtime area of a cursor and memory sort space. If shared servers are used, then part of the runtime area can be located in the SGA.

Intensive queries cause a big portion of the PGA to be dedicated to work areas supporting memory intensive operators, such as the following:

- Sort-based operators (for example, ORDER BY, GROUP BY, ROLLUP, window functions)
- Hash-join
- Bitmap merge
- Bitmap create
- Write buffers used by bulk load operations

A sort operator uses a sort area to perform the in-memory sort of a set of rows. Similarly, a hash-join operator uses a hash area to build a hash table.

Note: This information is available for Oracle9i and later.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > PGA Tab
Metrics	The size of a PGA area can be controlled and tuned. Larger work areas can significantly improve performance of session work, with the trade off being higher memory consumption. The size of a session's PGA should be big enough so that it can accommodate the input data and auxiliary memory structures allocated by its requested SQL actions. This is known as the optimal size of a PGA work area. When the size of a PGA work area is smaller than optimal, the response time increases, because an extra pass is performed over part of the input data. This is known as the one-pass size of the PGA work area. Under the one-pass threshold, when the size of a work area is smaller than its input data size, multiple passes over the input data are needed. This could drastically increase the response time of the session. This is known as the multi-pass size of the work area. The goal is to have most work areas running with an optimal size (more than 90% or more for OLTP systems), while a smaller amount are running with a one-pass size (less than 10%). Multi-pass execution should be completely avoided. Prior to Oracle9i, the PGA was controlled using the SORT_AREA_SIZE, HASH_AREA_SIZE, BITMAP_MERGE_AREA_SIZE and CREATE_BITMAP_AREA_SIZE parameters. Setting these parameters is difficult, because the maximum PGA size is ideally selected based on data input size and total number of work areas active in the system. These two factors vary a lot from one work area to another and from one point in time to another.
Correction	With Oracle9i, you can simplify and improve the way PGA memory is allocated, by enabling automatic PGA memory management. In this mode, Oracle dynamically adjusts the size of the portion of the PGA memory dedicated to work areas, based on an overall PGA memory target explicitly set by the DBA. To enable automatic PGA memory management, you have to set the initialization parameter PGA_AGGREGATE_TARGET. If multi-pass percentages are consistently non-zero, you should increase the size of PGA_AGGREGATE_TARGET.

PGA Session Detail

Oracle's Program Global Area (PGA) is a private memory region containing data and control information for every connected process. Examples of PGA information are the runtime area of a cursor and memory sort space. If shared servers are used, then part of the runtime area can be located in the SGA.

Intensive queries cause a big portion of the PGA to be dedicated to work areas supporting memory intensive operators, such as the following:

- Sort-based operators (for example, ORDER BY, GROUP BY, ROLLUP, window functions)
- Hash-join
- Bitmap merge
- Bitmap create
- Write buffers used by bulk load operations

A sort operator uses a sort area to perform the in-memory sort of a set of rows. Similarly, a hash-join operator uses a hash area to build a hash table.

The PGA Session Detail section displays PGA usage for all connected sessions. The table below describes the information in this section:

Data	Description
SID	The unique identified given the session by Oracle.
Username	The user account a session is using.
O/S ID	The operating system ID of the process.
Logon Time	The time the session logged into Oracle.
Machine Name	The name of the machine running the process.
PGA Alloc Memory	The amount of allocated PGA memory for the process.
PGA Used Memory	The amount of freeable or usable memory for the process.
PGA Max Memory	The maximum amount of PGA memory used by the process.

Note: This information is available for Oracle9i and later.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Oracle Performance Analyst)** > PGA Tab

I/O View (DBArtisan - Oracle Performance Analyst)

In addition to a Home page, the I/O category of Oracle Performance Analyst includes the following tabbed pages:

Access Pattern Detail Tab
Datafile I/O Tab
Job Queue Activity Tab
DBWR/LGWR Tab
Rollback I/O Tab
Sort Activity Tab
Table I/O Tab

Home

The I/O Home page includes the following sections:

Key Systems Analysis Pane
Bottleneck Analysis Pane
I/O Access Patterns - I/O
SQL Analysis Pane
Workload Analysis Pane

Key Systems Analysis Pane

When complaints begin to surface about your database's performance, oftentimes the root cause can be traced to one or more issues with I/O. The thing to keep in mind when you begin to monitor the I/O of your database is that you are actually reviewing the success of your physical design model.

All the physical storage characteristics and placements, the table and index designs, and the speed with which it all works are on display when I/O is monitored. Because a database's main index of performance is measured by how fast I/O needs are satisfied, it is your responsibility to quickly interrogate Oracle to determine if a reported database slowdown is I/O related.

Physical I/O

Physical I/O consists of Oracle going to disk to gather or write data. The database writer (DBWR) and log writer (LGWR) processes typically perform all the I/O work in the database. You can also use other processes like the checkpoint and archive processes (CKPT and ARCH). Embarcadero Performance Analyst shows three key indicators of physical I/O. The table below describes these indicators:

Indicator	Description
Physical Reads	Physical Reads is the total number of physical reads performed on all datafiles since the last refresh.
Physical Writes	Physical Writes is the total number of times the DBWR process has performed writes to various database datafiles since the last refresh
Redo Writes	Oracle's redo logs are hotbeds of I/O activity in databases that store heavily modified data. Redo log files are used to perform database recovery in the event of a system crash. Redo logs are written to in a cyclical fashion - each log file is filled up before Oracle moves on to the next file. The redo writes statistic reflects the total number of redo writes by the LGWR process since the last refresh

Metrics

The table below describes metrics for Physical I/O statistics:

Statistic	Metrics
Physical Reads	Large numbers of physical reads could reflect a too small data/buffer cache. The Buffer Cache Hit Ratio is a better indicator of overall logical vs. physical I/O.
Physical Writes	Wait events related to I/O activity are good indicators of physical I/O problems. These events include db file parallel write and db file single write.

Correction

Doing the following can negate large numbers of continuous physical reads:

- Increasing the size of the data/buffer cache.
- Pinning often-referenced objects in memory by using the KEEP buffer pool.
- Placing heavily scanned objects in larger blocksize tablespaces (16-32KB). For Oracle9i or later.
- Tune SQL statements for better efficiency.

Logical I/O

Logical I/O refers to data access performed in memory. The database writer (DBWR) and log writer (LGWR) processes typically perform all the I/O work in the database. You can also use other processes like the checkpoint and archive processes (CKPT and ARCH). Embarcadero Performance Analyst shows three key indicators of logical I/O. The table below describes these indicators:

Indicator	Description
Logical Reads	Logical Reads is the total number of db block gets and consistent gets (data read from memory) since the last refresh.
Logical Changes	Logical Changes is the total number of changes that were made to all blocks in the SGA that were part of an update or delete operation. These changes generate redo log entries and are permanent if the transaction is committed. The number of logical changes is an approximate indication of total database work.
Consistent Reads	Consistent Reads is the total number of times a consistent read was requested for a database block. Such a read is performed from Oracle's rollback segments.

Metrics

Regarding raw logical I/O counts, no hard-core metrics exist. However, because physical I/O takes longer to complete than logical (memory) I/O, you should minimize physical read operations when possible. The Buffer Cache Hit Ratio is a better indicator of overall logical vs. physical I/O.

Correction

While logical I/O is still up to 1,400 times faster than physical disk access, it would be wise to investigate the top logical I/O process using Performance Analyst and see what SQL it is executing. If one process on the system is consuming between 25-50% of the overall amount, their SQL might require tuning.

Bottleneck Analysis Pane

When an Oracle database is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting can mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. You can use Bottleneck Analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance because it helps you track where a database has been spending its time. If heavy table scan activity has been dragging a database's performance down, you can use bottleneck analysis to confirm the actual root cause. Once one or more wait events or other bottlenecks have been pinpointed as possible performance vampires, you can discover a fair amount of detail about which sessions and objects are causing the problem.

For wait event analysis to be properly carried out, it is imperative that the timed_statistics initialization parameter be set to TRUE. By default this parameter is set to FALSE, which disallows the collection of wait times for each wait event defined to the Oracle engine. For one to really understand the impact of wait events on database performance, you need to not only discover what the database is or has been waiting on, but the durations of the waits. Having both allows a complete picture to be formed regarding the magnitude of wait-initiated performance degradations. Almost all Oracle experts now agree that allowing the collection of timing statistics adds little if anything to database overhead, so setting timed_statistics to TRUE should not be a worry. The Performance Analyst I/O home page identifies the top database and log file waits that can cause decreased performance.

The Bottleneck Analysis section includes:

Hottest Database Files Top Database File Waits Top Log File Waits

Hottest Database Files

The Hottest Database Files section lists the most active database files as indicated by physical I/O activity.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Understanding the storage-level hot spots of a database is beneficial for a couple of reasons. First, you can get a feel for overworked physical disks by viewing I/O statistics at the tablespace and datafile levels. If a particular disk or set of disks is under too much strain, you can relocate the tablespaces to other less-used devices or create new tablespaces on different disks and move hot objects to them (assuming, of course, you have extra disks available). Second, if you have used standard DBA practice and placed indexes in their own tablespace, you can view the I/O statistics for that tablespace and see if the indexes are actually being used.
Correction	Here are some areas to consider when viewing Hottest Database Files: *Seeing much activity in the SYSTEM tablespace and datafiles can indicate recursive calls (space management, etc.). The use of locally-managed tablespaces can help with space management as it relates to data dictionary references.Temporary tablespaces (devoted to sort activity) showing higher volumes of physical I/O could indicate a problem with excessive disk sorts.You might want to quickly review all the physical I/O for each drive/file system and get a feel for the overworked disks on your server. If you have under utilized disk drives (with their own controllers), consider relocating some tablespaces that exhibit high I/O characteristics to those drives.*

Top Database File Waits

When viewing wait statistics, there are many levels of detail. The first level is the system view, which provides a global, cumulative snapshot of all the waits that have occurred on a system. The Top Database File Waits section filters all system wait events to present waits that are associated with database file activity. It ranks the top events by the count of total waits.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane
Correction	Oracle documentation contains a listing and description of every current wait event defined in Oracle. DBAs unfamiliar with what each event represents should keep this listing close by as they examine wait-based event metrics. The most common database file wait is a db file scattered read, which indicates table scan activity. Small table scans are normally not a problem as Oracle can access data from small tables by quickly caching and scanning them. Scans of large tables can be confirmed by examining the I/O access pattern metrics on the Performance Analyst I/O home page. Information is also available on which sessions are causing the large table scans. Finally, one thing to watch is not only the actual waits, but also the wait time in seconds. If no wait time is observed with each wait type, then performance is likely not endangered by the event.

Top Log File Waits

When viewing wait statistics, there are many levels of detail. The first level is the system view, which provides a global, cumulative snapshot of all the waits that have occurred on a system. This section filters all system wait events to present waits that are associated with log file activity. It ranks the top events by the count of total waits.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane
Correction	Oracle documentation contains a listing and description of every current wait event defined in Oracle. DBAs unfamiliar with what each event represents should keep this listing close by as they examine wait-based event metrics. The most common database file wait is a log file parallel write, which indicates how quickly Oracle can flush the log buffer. Excessive wait times associated with this event count indicate a bottleneck at the log buffer level. Finally, one thing to watch is not only the actual waits, but also the wait time in seconds. If no wait time is observed with each wait type, then performance is likely not endangered by the event.

SQL Analysis Pane

Much of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times. SQL statements that use much memory can also cause a problem in a database.

Before you can identify problem SQL in your database, you have to ask the question, “What is bad SQL?” What criteria do you use when you begin the hunt for problem SQL in your critical systems? Even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every Oracle professional's satisfaction. The SQL Analysis for I/O shows what SQL statements have consumed the largest percentages of physical and logical I/O, sort activity, and rows processed. The table below lists some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

Criteria	Description
Overall Response (Elapsed) Time	The amount of time the query took to parse, execute, and fetch the data needed to satisfy the query. It should not include the network time needed to make the round trip from the requesting client workstation to the database server. This statistic is available in Oracle9i or later.
CPU Time	The amount of CPU time the query took to parse, execute, and fetch the data needed to satisfy the query.
Physical I/O	This is often used as the major statistic in terms of identifying good vs. bad SQL, this is a measure of how many disk reads the query caused to satisfy the user's request. While you certainly want to control disk I/O where possible, it is important that you not focus solely on physical I/O as the single benchmark of inefficient SQL. Make no mistake, disk access is slower than memory access and also consumes processing time making the physical to logical transition, but you need to look at the entire I/O picture of a SQL statement, which includes looking at a statements' logical I/O as well.
Logical I/O	This is a measure of how many memory reads the query took to satisfy the user's request. The goal of tuning I/O for a query should be to examine both logical and physical I/O, and use appropriate mechanisms to keep both to a minimum.
Repetition	This is a measure of how often the query has been executed. A problem in this area is not as easy to spot as the others unless you know your application well. A query that takes a fraction of a second to execute may still be causing a headache on your system if it has executed erroneously (for example, a query that executes in a runaway PL/SQL loop) over and over again.

There are other criteria that you can examine like sort activity or access plan statistics (that show items like Cartesian joins and the like), but more often than not, these measures are reflected in the criteria listed above. Fortunately, Oracle records all the above measures (some only in 9i), which makes tracking the SQL that has been submitted against an Oracle database much easier.

Metrics When you begin to look for inefficient SQL in a database, there are two primary questions you want answered: *What HAS been the worst SQL that has historically been run in my database?What IS the worst SQL that is running right now in my database?* When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It could be that the query is in an inefficient PL/SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

There is the possibility that the SQL statement just is not tuned well. To determine that, you can go further into Performance Analyst's Top SQL view and, if you have Embarcadero SQL Tuner installed, you can port the SQL over to SQL Tuner to better optimize the statement.

I/O Access Patterns - I/O

There are many different routes that Oracle can take to access various objects contained in a database. The I/O Access Patterns section summarizes the methods Oracle is using to satisfy end-user requests for data.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - Oracle Performance Analyst)** > Home > SQL Analysis Pane
Metrics	Long table scans are typically an activity to avoid as they have the capability to cause needless physical and logical I/O as well as flood the buffer cache with seldom referenced blocks of data. You can discover which sessions have been causing the most large table scans. The table fetch continued row statistic is indicative of chained/migrated row I/O. Such activity is not desired because chained/migrated row access can cause twice the I/O needed to access a table. Oracle must pick do two or more I/Os to read a chained/migrated row in a table. You can discover which sessions have been accessing tables with chained/migrated rows.

Workload Analysis Pane

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. The Top I/O Hogs table shows the top sessions that have caused the most physical I/O on the system (both reads and writes).

Metrics	If you see one or two users who have caused more than 25-50% of the total memory usage, go further into the sessions to see the activities they are performing.

I/O Tablespace Tab

Physical I/O consists of Oracle going to disk to gather or write data. Logical I/O refers to data access performed in memory. The database writer (DBWR) and log writer (LGWR) processes typically perform all the I/O work in the database. Other processes like the checkpoint and archive processes (CKPT and ARCH) may also be used.

The Tablespace I/O tab displays details concerning the physical I/O activity at the tablespace level. The table below lists the information available on this tab:

Column	Description
Tablespace Name	The name of the tablespace.
Physical Reads	The cumulative number of physical reads.
Physical Writes	The cumulative number of physical writes.
Physical Block Reads	The cumulative number of physical block reads.
Physical Block Writes	The cumulative number of physical block writes.
Physical Read Time	The time spent reading from the tablespace (in hundredths of seconds).
Physical Write Time	The time spent writing to the tablespace (in hundredths of seconds).

Metrics Generally, you want to see much more logical I/O activity than physical I/O, at least with respect to reads, although this in and of itself is a guarantee of good I/O performance. Seeing logical and physical reads keeping pace with one another is a sure sign that the Oracle SGA's buffer cache is sized too small or that needless, large table scans are occurring, which cause blocks of data to be continually read in and flushed out of the buffer cache. Other telltale signs of trouble brewing are large amounts of activity visible in user's TEMP tablespaces. The normal interpretation of such a thing is that a large number of disk sorts are taking place (perhaps because the Init.ora/spfile parameter SORT_AREA_SIZE may be set too small).

Datafile I/O Tab

Physical I/O consists of Oracle going to disk to gather or write data. Logical I/O refers to data access performed in memory. The database writer (DBWR) and log writer (LGWR) processes typically perform all the I/O work in the database against the physical datafile used to hold information The checkpoint and archive processes (CKPT and ARCH), also perform I/O work in the database.

The Datafile I/O tab displays details concerning the physical I/O activity at the datafile level. The table below lists the information available on this tab:

Column	Description
DataFile Name	The name of the datafile.
Tablespace Name	The name of the tablespace.
Physical Reads	The cumulative number of physical reads.
Physical Writes	The cumulative number of physical writes.
Block Reads	The cumulative number of physical block reads.
Block Writes	The cumulative number of physical block writes.
Read Time	The time spent reading from the tablespace (in hundredths of seconds).
Write Time	The time spent writing to the tablespace (in hundredths of seconds).

Metrics Generally, you will want to see much more logical I/O activity than physical I/O, at least with respect to reads, although this in and of itself is a guarantee of good I/O performance. Seeing logical and physical reads keeping pace with one another is a sure sign that the Oracle SGA's buffer cache is sized too small or that needless, large table scans are occurring, which cause blocks of data to be continually read in and flushed out of the buffer cache. Other telltale signs of trouble brewing are large amounts of activity visible in user's TEMP tablespaces. The normal interpretation of such a thing is that a large number of disk sorts are taking place (perhaps because the Init.ora/spfile parameter SORT_AREA_SIZE may be set too small). This view also allows you to see the ‘hot’ files in terms of physical I/O activity. Seeing too much activity on one drive/file system may indicate a need for better file striping.

Rollback I/O Tab

The Rollback I/O tab includes the following sections:

[Active Rollback Details](#) [Rollback I/O](#) [Session Rollback Activity](#)

For more information, see [Home](#).

Rollback I/O

To undo changes made to the Oracle database from within a transaction, Oracle writes data to individual rollback segments. Oracle also uses these segments to maintain read consistency for multiple users of data that is being modified. Because Oracle reads from and writes data to rollback segments, they can become very hot areas for I/O.

The Rollback I/O section presents everything necessary to view and troubleshoot rollback problems.The table below lists the information available in this section:

Column	Description
Name	The name of the rollback segment.
Size	The size of the rollback segment in KBs.
Shrinks	The number of times the rollback segment has decreased in size.
Extents	The number of times he rollback segment has increased in size.
Gets	The number of header gets (the segment has been used).
Waits	The number of header waits.
Writes	The number of bytes written to the rollback segment.
Active Trans	Indicates whether the rollback segment is active (non zero) or not (zero value).
Status	Indicates the status of the rollback segment, with the two main results being OFFLINE (a segment is offline and unavailable for transactions) and ONLINE (a rollback segment is online and available for transactional use).
High Water Mark	The largest size that the rollback segment has ever grown to.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - Oracle Performance Analyst)** > Rollback I/O Tab
Metrics	To properly tune rollback I/O, you must first make sure that you have enough segments to accommodate the workload of the database. Constantly seeing a count of active rollback segments equal to or near the number of rollbacks defined for the database is an indicator that you should create more. An overall rollback contention ratio of 1% or greater is an indicator of too few rollbacks. Seeing wait counts greater than zero for each rollback segment is further evidence that you should create more rollback segments. Oracle9i provides the UNDO tablespace to automatically generate and eliminate the 'correct' number of rollback segments for a system given a certain workload. After ensuring that enough rollback segments exist in the database, you should then turn your attention to the question of sizing. Dynamic rollback extension can take a toll on performance when rollback segments are consistently enlarged to accommodate heavy transaction loads. Seeing rollback segments undergoing numerous extends and shrinks (as Oracle returns a segment back to its OPTIMAL setting), as well as rollback segments having current or high-water mark sizes greater than their OPTIMAL setting usually is a good indicator that they should be permanently enlarged. Again, Oracle9i's automatic undo management can assist in this process.

Active Rollback Details

Rollback or undo segments are used to guarantee transactional integrity. When a transaction has not been committed or rolled back, a rollback segment will be in use with live transactional data. The Active Rollback Details section displays information concerning active rollback segments. The table below lists the information available in this section:

Column	Description
SID	The system ID of the session using the rollback segment.
Username	The account/schema name of the session.
Rollback	The rollback segment name containing the session's transaction.
Start Time	The time the transaction was initiated.
Blocks Used	The number of rollback blocks used by the transaction.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - Oracle Performance Analyst)** > Rollback I/O Tab
Metrics	Seeing transactions with dated start times may indicate transactions that are the source of lock contention.

Session Rollback Activity

For all connected sessions, Oracle maintains statistics regarding each session's counts of rollbacks and commits. The Session Rollback Activity section displays a transactional summary/history that helps identify sessions with excessive rollback activity. The table below lists the information available in this section:

Column	Description
SID	The system ID of the session.
Username	The account/schema name of the session.
Machine Name	The machine name where the session logged on.
Logon Time	The time the session logged into Oracle.
User Rollbacks	The number of times the user manually issued the ROLLBACK statement or an error occurred during the user's transaction.
Transactions Rolled Back	The number of transactions successfully rolled back.
User Commits	The number of times a user transaction was committed.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - Oracle Performance Analyst)** > Rollback I/O Tab
Metrics	Seeing high numbers of transactions rolled back might indicate an application error or other transactional problem.

DBWR/LGWR Tab

The DBWR/LGWR tab includes the following sections:

Daily Archive Log Summary (Last 7 Days)	Database Writer Detail	Log Writer Detail
Redo Wastage	Table I/O Tab	

Database Writer Detail

The database writer process (DBWR) handles the flow of information from Oracle's physical datafiles to and from the various memory structures in the system global area (SGA). On platforms that support it, you can configure and use multiple DBWR processes. The log writer (LGWR) process manages the information contained in Oracle's online redo log files and redo log buffer area. The table below lists the information available in this section:

Column	Description
Statistic	The specific metric for the database writer.
Amount	The value for the statistic.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - Oracle Performance Analyst)** > DBWR/LGWR Tab
Metrics	Of all the statistics presented for the DBWR process, the summed dirty queue length statistic deserves attention. Non-zero values typically indicate buffers left in the write queue after a write request and may indicate that the DBWR process is falling behind.

Log Writer Detail

The database writer process (DBWR) handles the flow of information from Oracle's physical datafiles to and from the various memory structures in the system global area (SGA). On platforms that support it, you can configure and use multiple DBWR processes. The log writer (LGWR) process manages the information contained in Oracle's online redo log files and redo log buffer area. The table below lists the information available in this section:

Column	Description
Statistic	The specific metric for the log writer.
Amount	The value for the statistic.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - Oracle Performance Analyst)** > DBWR/LGWR Tab
Metrics	For the LGWR process, non-zero values seen for the redo log space requests and redo log space wait time statistics could be a cause for concern. Redo log space requests reflect the number of times a user process waited for space in the redo log buffer, while the redo log space wait time presents the total time waited in milliseconds. Both could indicate the presence of contention in the redo log buffer. Possible remedies include increasing the log_buffer size in the SGA.

Daily Archive Log Summary (Last 7 Days)

To allow for point-in-time recovery, Oracle writes copies of redo log information to disk. When a database is running in archive log mode, a DBA can (with proper backup techniques in place) recovery nicely from a database error and roll forward to almost any point in time needed, as long as the proper archive logs are in place.

The I/O needed to write these archive logs is handled by Oracle's ARCH process. The Daily Archive Log Summary shows the number of archive logs written each day for the past seven days.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - Oracle Performance Analyst)** > DBWR/LGWR Tab
Metrics	Heavy redo log activity can lead to many archive files being written out to disk. Batch jobs have the potential to move very fast - sometimes so fast that the online redo logs wrap back around before they have a chance to be archived. Messages of this nature will oftentimes show up in the Oracle alert log. If this is the case, you many want to think about increasing the size of the online redo log files, or increase the number of redo logs in general. Seeing archive files written more than one per hour or half-hour may indicate a too small redo size (or above average data modification load). If archive log protection is a concern (in other words, you do not want to lose an archive file that may be needed for recovery) and you are using Oracle8, you can now take advantage of the feature to write archive files to more than one destination on disk. It also allows multiple ARCH processes to now be invoked. Investigate the use of the Init.ora parameters log_archive_dest_n and log_archive_max_processes. Always remember one thing with respect to archive files and running Oracle in archive log mode: running out of archive file space on the server is a sure fire way to halt all activity in a database. Make sure you have plenty of free space available on your archive drives. Also, implement a purge procedure for older archives in conjunction with a good backup routine.

Redo Wastage

Oracle's redo logs are hotbeds of I/O activity in databases that store heavily modified data. Redo log files are used to perform database recovery in the event of a system crash. Redo logs are written to in a cyclical fashion - each log file is filled up before Oracle moves on to the next file. The Redo Wastage section shows how many bytes were wasted by redo log buffers being written before they were full.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - Oracle Performance Analyst)** > DBWR/LGWR Tab
Metrics	Viewing a high percentage of wasted bytes to overall redo bytes can help you identify if redo wastage is a problem on your system.
Correction	Sometimes heavy redo wastage occurs when the log_checkpoint_interval parameter is set too high. If you think this is the case for your system, then *Edit the Init.ora file for the database.Change the amount of log_checkpoint_interval to a lower value.Cycle the Oracle server when possible to allow the new value to take effect.Monitor new value to see if performance improves.*

Table I/O Tab

I/O activity at the table level has historically been difficult to get with Oracle. Oracle9i and later contain better dictionary elements to help get a clearer picture of table I/O, but earlier versions of Oracle allow an estimation only. The table below lists the information available in the Estimated Physical I/O by table on this tab:

Column	Description
Owner Name	The owner of the table
Table Name	The name of the table or table partition.
Estimated Disk Reads	The estimated number of physical and physical direct reads for the table.

Metrics Viewing high numbers of disk reads for a table could indicate excessive table scan activity. Such a table may benefit from better indexing or from being placed into a larger blocksize tablespace (Oracle9i and later).

Sort Activity Tab

The Sort Activity tab includes the following sections:

Current Disk Sorts Session Sort Detail

Session Sort Detail

Excessive sort activity can degrade a user's or overall database performance. When a sort operation occurs, Oracle attempts to perform the sort in a memory space, assigned by the DBA, which exists at the operating system level. If the sort is too large to be contained within this space, it will continue the sort on disk - specifically, in the user's assigned temporary tablespace.

The Session Sort Detail section displays historical sort statistics for every logged on session. The table below lists the information available in this section:

Column	Description
SID	The unique identified given the session by Oracle.
Username	The user account a session is using.
Machine Name	The machine name where the session logged on.
Logon Time	The time the session logged into Oracle.
Disk Sorts	The cumulative number of disk sorts for the session.
Memory Sorts	The cumulative number of memory sorts for the session.
Row Sorted	The cumulative number of rows sorted by the session.
Disk/Memory Sort Ratio	The percentage of times the session had to perform a disk sort.

Location Oracle Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - Oracle Performance Analyst)** > Sort Activity Tab

Metrics Techniques to include in your overall performance strategy are those that relate to minimizing the amount of sort activity overall and specifically sort activity that takes place on disk. A good place to start is by understanding things that cause sorts in the first place. A list of sort-related commands and SQL-related options include CREATE INDEX, ALTER INDEX … REBUILD, DISTINCT, ORDER BY, GROUP BY, UNION, INTERSECT, MINUS, IN, and NOT IN, as well as certain unindexed joins and certain correlated subqueries All of these SQL commands have the potential to create a sort. As a DBA, you probably will not know which queries will have their sorts performed entirely in memory and which ones will be forced to go to disk. There are times you simply cannot stop disk sort activity (such as in data warehousing environments). That being the case, you should ensure the following are true in your database: *Your users' TEMPORARY TABLESPACE assignment is not the SYSTEM tablespace, which is the default assignment.The TEMPORARY TABLESPACE assigned to your users is placed on a fast disk.The TEMPORARY TABLESPACE has the tablespace parameter TEMPORARY assigned to it, which allows sort activity to be performed in a more efficient manner.* If your overall memory sort ratio falls below 90%, you may want to increase the parameters devoted to memory sorts - sort_area_size and sort_area_retained_size. Keep in mind that Individual users might have the ability to alter their own sessions and increase their sort_area_size assignments! As a DBA, you may want to restrict users that have the ALTER SESSION privilege. In addition to increasing the amount of memory devoted to sorting, you should also hunt down inefficient SQL that cause needless sorts. For example, UNION ALL does not cause a sort whereas UNION does in an SQL query (to eliminate duplicate rows). DISTINCT oftentimes is coded inappropriately (especially by folks transferring from Microsoft Access, which used to use DISTINCT for nearly every SELECT query).

Current Disk Sorts

Excessive sort activity – especially disk sorts - can degrade a user's or overall database performance. When a sort operation occurs, Oracle attempts to perform the sort in a memory space, assigned by the DBA, which exists at the operating system level. If the sort is too large to be contained within this space, it will continue the sort on disk - specifically, in the user's assigned TEMPORARY TABLESPACE.

The Current Disk Sorts section displays sort statistics for every session currently performing a disk sort. The table below lists the information available in this section:

Column	Description
SQL Text	The SQL statement causing the disk sort.
SID	The unique identified given the session by Oracle.
Username	The user account a session is using.
Machine Name	The machine name where the session logged on.
Tablespace	The tablespace containing the sort segments used by the sort.
Extents	The number of extents being used for the sort.
Blocks	The number of blocks being used for the sort.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - Oracle Performance Analyst)** > Sort Activity Tab
Metrics	Techniques to include in your overall performance strategy are those that relate to minimizing the amount of sort activity overall and specifically sort activity that takes place on disk. A good place to start is by understanding things that cause sorts in the first place. A list of sort-related commands and SQL-related options include CREATE INDEX, ALTER INDEX … REBUILD, DISTINCT, ORDER BY, GROUP BY, UNION, INTERSECT, MINUS, IN, and NOT IN, as well as certain unindexed joins and certain correlated subqueries All of these SQL commands have the potential to create a sort. As a DBA, you probably will not know which queries will have their sorts performed entirely in memory and which ones will be forced to go to disk. There are times you simply cannot stop disk sort activity (such as in data warehousing environments). That being the case, you should ensure the following are true in your database: *Your users' TEMPORARY TABLESPACE assignment is not the SYSTEM tablespace, which is the default assignment.The TEMPORARY TABLESPACE assigned to your users is placed on a fast disk.The TEMPORARY TABLESPACE has the tablespace parameter TEMPORARY assigned to it, which allows sort activity to be performed in a more efficient manner.* If your overall memory sort ratio falls below 90%, you may want to increase the parameters devoted to memory sorts - sort_area_size and sort_area_retained_size. Keep in mind that Individual users might have the ability to alter their own sessions and increase their sort_area_size assignments! As a DBA, you may want to restrict users that have the ALTER SESSION privilege. In addition to increasing the amount of memory devoted to sorting, you should also hunt down inefficient SQL that cause needless sorts. For example, UNION ALL does not cause a sort whereas UNION does in an SQL query (to eliminate duplicate rows). DISTINCT oftentimes is coded inappropriately (especially by folks transferring from Microsoft Access, which used to use DISTINCT for nearly every SELECT query).

Job Queue Activity Tab

The Job Queue Activity tab includes the following sections:

Job Queue Summary Problem Jobs Running Jobs Detail

Job Queue Summary

Oracle provides a way for Oracle developers to submit and run jobs through the job queue feature of the Oracle database. The table below lists the information available on this section:

Column	Description
Total Jobs	The number of jobs defined to the database.
Broken	The number of jobs broken (disabled).
Failures	The number of job failures.
Running	The number of jobs currently running.

Problem Jobs

Oracle provides a way for Oracle developers to submit and run jobs through the job queue feature of the Oracle database. Sometimes a job may fail or become disabled (broken). The table below lists the information available in this section:

Column	Description
Job ID	The job number for the job.
Login User	The user account used to run the job.
Last Date Successfully Run	The date/time of when the job last successfully ran.
Broken	Indicates if the job is disabled or not.
Failures	The number of failures the job has experienced.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - Oracle Performance Analyst)** > Job Queue Activity Tab

Running Jobs Detail

Oracle provides a way for Oracle developers to submit and run jobs through the job queue feature of the Oracle database. The Running Jobs Detail section provides information about currently running jobs. The table below lists the information available in this section:

Column	Description
SID	The unique identified given the session by Oracle.
Job ID	The job number for the job.
Start Date/Time	Indicates when the job started.
Submitted by	Indicates the session that submitted the job.
Run As	Indicates which account is used to run the job.
Parse As	Indicates which account is used to parse the job.
Next Run Date/Time	Indicates when the job will next run.
Job Contents	Indicates the actual command used by the job.

Location Oracle Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - Oracle Performance Analyst)** > Job Queue Activity Tab

Access Pattern Detail Tab

When a session submits a set of SQL or a transaction, there are many different combinations of methods the Oracle kernel will use to satisfy the query or transaction. The Access Pattern Detail tab displays all currently connected sessions, their demographic information, and a count of the various different methods that Oracle uses to handle incoming requests. The table below lists the information available on this tab:

Column	Description
SID	The unique identified given the session by Oracle.
Username	The user account a session is using.
Machine Name	The machine name where the session logged on.
Logon Time	The time the session logged into Oracle.
Large Table Scans	The cumulative count of ‘large’ table scans with ‘large’ normally being defined as any table over five blocks.
Small Table Scans	The cumulative count of ‘small’ table scans with ‘small’ normally being defined as any table under five blocks.
Table Scan ROWID Ranges	The cumulative count of table scans where the table in question has the CACHE property enabled.
Table Scan Cache Partitions	The cumulative count for partitions that have the CACHE property enabled.
Table Scan Direct Reads	The cumulative count of table scans performed with Oracle bypassing the buffer cache (data read only once - straight from disk and not from both disk and cache).
Chained Row Accesses	The cumulative count of the number of times a chained/migrated row was accessed.
Index by ROWID	The cumulative count of the number of rows fetched from a table by ROWID (normally index access).
Index Scans	The cumulative count of fast full index scans.
Index Scan ROWID Ranges	The cumulative count of fast full index scans with ROWID endpoints specified.
Index Scan Direct Read	The cumulative count of fast full index scans using direct read (data read only once - straight from disk and not from both disk and cache).

Space View (DBArtisan - Oracle Performance Analyst)

In addition to a Home page, the Space category of Oracle Performance Analyst includes the following tabbed pages:

Extent Analysis Tab	Object Space Tab	Fragmentation Tab
Tablespace Detail Tab	Tablespace Growth Tab	User Space Tab

Home

The Space Home page includes the following sections:

Bottleneck Analysis Pane	Redo Logs	Space Analysis Pane
SQL Analysis Pane	User Space Analysis Pane	

Space Analysis Pane

The Oracle tablespace is the logical container for the physical storage needs of a database. The Space Analysis section displays details for all the tablespaces for a particular database, including their total, used, free space, and percent free, as well as if each can automatically extend to support incoming space requests.

Metrics

If any tablespace's free space percent amount goes below 10%, and at least one tablespace's datafiles does not have AUTOEXTEND enabled (or the datafile has reached its extend limit), you should take action to ensure that the tablespace does not run out of available free space.

Correction

There are two things you can do to ensure that a tablespace does not run out of available free space:

1 First, you should look into the use of Oracle's AUTOEXTEND feature. AUTOEXTEND lets you give an Oracle tablespace the ability to auto-grow when it has exhausted the free space contained within. You can let a tablespace grow in an unlimited fashion or put constraints on it to stop at a certain point. You can also dictate how much more free space the tablespace gets each time it needs more space than is available. However, AUTOEXTEND enabled for a tablespace does not mean that you cannot run out of space. Remember you still have the physical server limitations to contend with. Make sure you (or your sysadmin) keep a careful eye on the server drives that house your Oracle database files for available free space.

2 If the free space on a server drive nears its limit, disable AUTOEXTEND for the datafile(s) that are on the target drive, and use the ALTER TABLESPACE … ADD DATAFILE command to place a new datafile for the tablespace on another drive that has more free space to offer.

Tip: AUTOEXTEND is not a replacement for proactive space planning. When extra space is needed by the database, and AUTOEXTEND is activated by Oracle, performance slows as Oracle allocates more space for the tablespace. Avoiding AUTOEXTEND aids performance, albeit in a small way.

Bottleneck Analysis Pane

When using bottleneck analysis to troubleshoot a database, you cannot rely only on the information contained in the wait event views that Oracle provides. For example, an object may attempt to extend into another extent of space in a tablespace and yet be denied if no such free space exists. Such a failure is not reflected in any wait event, but still represents a very real bottleneck to the database. In the same way that you cannot depend on only a few ratios to properly carry out ratio-based performance analysis, an administrator must include other statistical metrics in their overall bottleneck analysis framework to obtain an accurate performance risk assessment. Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system. The Performance Analyst Space home page displays space-related bottlenecks that can decrease overall performance of your system.

The following bottleneck indicators are used on the Performance Analyst Space home page to succinctly communicate the general overall space situation of the monitored database:

Autoextend Datafiles at Extend Limit

Offline Tablespaces

Objects at Maximum Extents

Top Fragmented Datafiles

Objects with No Room to Expand

Autoextend Datafiles at Extend Limit

You can specify that one or more tablespace datafiles can automatically grow by enabling the datafile’s AUTOEXTEND property. you can also specify that the datafile not be permitted to grow beyond a certain limit, and once that limit is reached, the datafile and/or the tablespace may be permanently out of space.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Space View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	You should not see any positive indicator that any datafile has reached its extend limit. While it might be okay from the standpoint that a datafile cannot simply grow anymore (if the tablespace has other datafiles that can grow or that no more space is added to the tablespace), disable AUTOEXTEND on the datafile.
Correction	If a datafile has reached its extend limit, you can take a number of actions: *Eliminate the extend limit for the datafile either by increasing the limit size or allowing to grow in an unlimited fashion.If the datafile is on a disk or file system that has no free space left, you can add another datafile to the tablespace to allow for space expansion and alter the problem datafile to have AUTOEXTEND disabled.If the datafile is on a disk or file system that has no free space left, the datafile can be physically relocated to another drive or file system with more free space.*

Objects at Maximum Extents

When an object in Oracle (table, index, table partition, etc.) needs to expand, Oracle automatically allocates another extent of space to accommodate the incoming data. However, a table or index can reach its maximum extent limit, which is the maximum number of extents for an object.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Space View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis PaneOracle Performance Analyst Statistics (DBArtisan) > Objects View (DBArtisan - Oracle Performance Analyst) > Home > Bottleneck Analysis Pane
Metrics	Unless an object is static in nature, and is not expected to grow in size, no object should ever get to its maximum extent limit.
Correction	If a maximum extent limit is reached for an object, you can take a number of actions: *The object can be reorganized into a locally-managed tablespace where no maximum extent limit is applied to any object.The object can be altered to have unlimited extents.The object can be reorganized into one or a few extents.*

The good news concerning locally-managed tablespaces is that unlimited extents are the rule, so if you choose to use these storage structures in your database, you will not have to worry about an object ever reaching a maximum extent limit. Some DBAs have expressed concerns over whether an object having hundreds or thousands of extents experiences performance problems when full table scans or similar operations are performed against them. While most Oracle experts agree that such I/O degradation might have been experienced in earlier Oracle versions, most feel that such is not the case any longer, especially when locally-managed tablespaces are used.

Objects with No Room to Expand

When an object in Oracle (table, index, table partition, etc.) needs to expand, Oracle automatically allocates another extent of space to accommodate the incoming data. However, an object may be prohibited from expanding for a number of reasons:

- The tablespace does not have enough free space to accommodate the extent request, and cannot automatically extend.

- The tablespace has enough total free space, but the space is not contiguous in nature (and the tablespace cannot automatically extend) and therefore the object cannot extend.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Space View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis PaneOracle Performance Analyst Statistics (DBArtisan) > Objects View (DBArtisan - Oracle Performance Analyst) > Home > Bottleneck Analysis Pane
Metrics	Unless an object is static in nature, and is not expected to grow in size, no object should ever be prohibited from acquiring its next extent of space.
Correction	If an object cannot extend, you can take a number of actions including: *One or more of the tablespace's datafiles can have its AUTOEXTEND property enabled, which lets the object to grow even though there is no space currently available in the tablespace.The object can be reorganized into a locally-managed tablespace all extents are uniform in nature.The object can be altered to have a reduced next extent size.*

Offline Tablespaces

A tablespace ONLINE status indicates that the tablespace is in normal read/write mode. Tablespaces marked read only displays a status of READ ONLY, while tablespaces engaged in backup mode indicates such in their status. An OFFLINE tablespace during normal processing hours is a cause for concern.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Space View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	No tablespace should have a status of OFFLINE during normal business hours, unless you are performing maintenance on it.
Correction	If a tablespace has a status of OFFLINE, you can make it ONLINE using the ALTER TABLESPACE command.

Top Fragmented Datafiles

This section shows the tablespaces datafiles suffering from the highest levels of free space fragmentation in the database. Tablespaces are made up of object segments and space extents. Extents are either allocated to object segments or are free. When a tablespace is initially populated, all objects are neatly packed together in the front of the tablespace and all remaining free space is in one free chunk at the end. As objects grow (or extend) they are given new extents of space in the tablespace/datafile. If they are dropped, pockets of free space begin to appear throughout the tablespace. These pockets of space take one of two forms. The table below describes these forms:

Free Space	Description
Honeycombs	Pockets of free space that are adjacent to one another.
Bubbles	Pockets of free space that are trapped between object extents in the tablespace.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Space View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	If you see a datafile that has many chunks of free space, determine if the tablespace is experiencing honeycomb or bubble fragmentation. You can handle honeycomb fragmentation quite easily, whereas bubble fragmentation is more difficult to solve.

Correction

You can eliminate honeycomb fragmentation with the ALTER TABLESPACE…COALESCE command. Issuing this command combines all pockets of adjacent free space into one extent. Each database maintenance plan should include a job that coalesces all the free honeycombs in a tablespace into one free chunk. Although Oracle is supposed to perform this operation automatically through the SMON process, it requires you to have the PCTINCREASE parameter of the tablespace set to a nonzero value. Having PCTINCREASE set to a value greater than zero encourages tablespace fragmentation through disparately sized extents. Plus, using SMON in this way is not efficient or entirely reliable.

You can temporarily solve bubble fragmentation by performing a total tablespace reorganization. A better long-term solution for Oracle databases 8.1.5 or later is to convert tablespaces over to locally managed tablespaces. With locally managed tablespaces you either specify the initial extent size and let Oracle automatically size all other extents, or specify a uniform extent size for everything. Problems caused by fragmentation then become a thing of the past.

To help stave off fragmentation problems:

- Set PCTINCREASE to zero for all tablespaces and objects to promote same-sized extents.
- Specify equal-sized allotments for your INITIAL and NEXT object storage parameters.
- Group objects with like growth and storage needs together in their own tablespaces.
- Considering converting any dictionary-managed tablespace to locally-managed.

You should also avoid fragmentation in the SYSTEM tablespaces. The best ways to do this include:

- Ensure no user has a DEFAULT or TEMPORARY tablespace assignment of SYSTEM.
- Ensure no user has a quota set for SYSTEM.
- Ensure no user has been granted the UNLIMITED TABLESPACE privilege.

SQL Analysis Pane

Much of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times.

The SQL Analysis section of the Space home page shows a count of all SQL statements that have executed INSERT, UPDATE, DELETE, and TRUNCATE commands. Seeing these statistics should help you determine how much space-related SQL activity is occurring on your system.

Metrics The statistics for this section can be used in many different ways. For example, large numbers of inserts could indicate a growing database and that you should keep a watchful eye on free space. Heavy update numbers could mean trouble for objects containing many indexes (indexes typically undergo splits and fragmentation when large volumes of updates are thrown their way). Many deletes might signal potential table high water mark problems.

Redo Logs

Oracle's redo logs are hotbeds of I/O activity in databases that store heavily modified data. Redo log files are used to perform database recovery in the event of a system crash. Redo logs are written to in a cyclical fashion - each log file is filled up before Oracle moves on to the next file. The redo bytes statistic reflects the total amount of redo generated in bytes since the last refresh. The redo wastage bytes indicate how many bytes were wasted by redo log buffers being written before they were full.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Space View (DBArtisan - Oracle Performance Analyst)** > Home > SQL Analysis Pane
Metrics	Viewing a high percentage of wasted bytes to overall redo bytes can help you identify if redo wastage is a problem on your system.
Correction	Sometimes heavy redo wastage occurs when the log_checkpoint_interval parameter is set too high. To fix this: *Edit the Init.ora file for the database.Change the amount of log_checkpoint_interval to a lower value.Cycle the Oracle server when possible to allow the new value to take effect.Monitor new value to see if performance improves.* For Oracle9i and later, you can dynamically alter the log_checkpoint_interval parameter without stopping/starting the database.

User Space Analysis Pane

The Top Space Hogs section details the schema in the database that own the most space. In databases where object owners can be many, it is oftentimes a good idea to take a quick look at which schema are leading the way in terms of space usage. This is not restricted to production environments only, but can be extended to dynamic development systems as well, especially if there are many developers in a database that have their own sets of objects. Developers sometimes have a bad habit of creating objects (or copies of objects) that they leave in a database even though they are not being used.

Correction If a user owns many objects, or copies of objects, you can reclaim space in your database by contacting the user to see if those objects can be dropped.

Tablespace Detail Tab

The Tablespace Detail tab includes the following sections:

Datafile Detail Tablespace Object Summary

Tablespace Object Summary

Tablespaces are the logical containers for database objects inside Oracle. The Tablespace Object Summary section provides an overview of how much space is consumed in each tablespace by object type. The table below lists the information available in this section:

Column	Description
Tablespace Name	The name of the tablespace.
Status	Indicates the status of the tablespace.
Autoextend	Indicates whether one or more of the tablespace’s datafiles has the ability to automatically grow.
Total Space (MB)	The total allocated space for the tablespace.
Table (MB)	The amount of used table and table partition space, in MB.
Index (MB)	The amount of used index and index partition space, in MB.
Rollback (MB)	The amount of used rollback and undo segment space, in MB.
Cluster (MB)	The amount of used cluster space, in MB.
LOB (MB)	The amount of used large object space, in MB.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Space View (DBArtisan - Oracle Performance Analyst)** > Tablespace Detail Tab

Datafile Detail

Tablespaces are the logical containers for database objects inside Oracle, and datafiles are the actual physical files that make up a tablespace. The Datafile Detail section provides an overview of the size of each datafile and other demographic information. The table below lists the information available in this section:

Column	Description
File Name	The name of the datafile.
Status	Indicates the status of the tablespace.
Autoextend	Indicates whether the datafile has the ability to automatically grow.
Tablespace Name	The name of the tablespace.
Size (MB)	The total allocated space for the datafile.
Max Size (MB)	The maximum size the datafile can automatically grow to.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Space View (DBArtisan - Oracle Performance Analyst)** > Tablespace Detail Tab

Object Space Tab

The Object Space tab displays a summary of space and extent information for all or selected objects in the database. The table below lists the information available on this tab:

Column	Description
Owner	The owner of the object.
Object Name	The name of the object.
Object Type	The type of object (table, index, etc.).
Space (KB)	The size of the object in KB.
Extents	The number of extents used by the object.
Max Extents	The maximum number of extents allowed for the object.
Initial Extent (KB)	The initial extent size for the object.
Next Extent (KB)	The next extent size for the object.
Tablespace	The tablespace where the object resides.

Metrics

One thing to keep an eye on are objects that are close to their maximum extent limit. Such objects can be altered to have more or unlimited extents if need be. The use of locally-managed tablespace eliminates the threat of an object ever encountering a maximum extent limit. For more information, see the Home page under **Space View (DBArtisan - Oracle Performance Analyst)**.

Tablespace Growth Tab

Growing tablespaces do not have to spell problems for a DBA. Of course, up front planning is the key to sizing tablespaces correctly. Unfortunately, DBAs may not have all the information they need at tablespace creation time (or they may have the wrong information), so a tablespace can approach the end of its free space from time to time. The DBA can allow a tablespace to grow automatically (AUTOEXTEND) to prevent an out-of-space condition. Enabling AUTOEXTEND for a tablespace is quite reassuring for a DBA, but it introduces a new concept for the tablespace: monitoring data file growth. You should monitor your tablespaces that have AUTOEXTEND set so you can get an idea of the growth that is occurring in your database. Monitoring them lets you perform some mini-capacity planning and helps you get a jump-start on server upgrades.

The Tablespace Growth tab displays growth statistics for the database currently being monitored. The table below lists the information available on this tab:

Column	Description
Tablespace Name	The name of the tablespace.
Datafile Name	Thee name of the tablespace's datafile.
Autoextend	Indicates whether the tablespace has one or more datafiles that have the AUTOEXTEND feature set (allowing the datafile to automatically grow).
Created Size	The starting size for the physical datafile.
Current Size	The current size of the physical datafile.
Growth	The percentage of growth experienced for the datafile (if any).

Metrics

Using AUTOEXTEND can be very important for tablespaces devoted to disk sorting. Large data warehouses often must endure large disk sort operations. Having AUTOEXTEND enabled for tablespaces used for temporary segments (sort activity) helps large sort queries to complete when they might otherwise fail due to running out of sort space.

Just having AUTOEXTEND enabled for a tablespace does not mean you cannot run out of space. Remember, you still have to contend with the physical server limitations. Make sure you (or your sysadmin) keep a careful eye on the server drives that house your Oracle database files for available free space. If the free space on a server drive nears its limit, disable AUTOEXTEND for the datafile(s) that are on that drive, and use the traditional ALTER TABLESPACE…ADD DATAFILE command to place a new datafile for the tablespace on another drive that has more free space to offer.

Fragmentation Tab

Tablespaces are made up of object segments and space extents. Extents are either allocated to object segments or are free. When a tablespace is initially populated, all objects are neatly packed together in the front of the tablespace and all remaining free space are in one free chunk at the end. Unfortunately, this is not how things continue to be in a tablespace. As objects grow (or extend) they are given new extents of space in the tablespace. If you drop objects, pockets of free space begins to appear throughout the tablespace. These pockets of space are either honeycombs or bubbles. Honeycombs are not so difficult to deal with, but bubbles are another story.

The Fragmentation tab displays two different views of tablespace fragmentation. The first grid displays fragmentation at the tablespace level and the second grid displays fragmentation at the datafile level. The table below describes the information available in the Tablespace Fragmentation grid:

Column	Description
Tablespace Name	The tablespace name.
Free Space (MB)	The total amount of free space in MB for the tablespace.
Free Chunks	The total number of free chunks in the tablespace.
Largest Chunk (MB)	The largest free chunk (in MB) for the tablespace.

The table below describes the information available in the Datafile Fragmentation grid on the Fragmentation tab:

Column	Description
Datafile	The name of the datafile.
Autoextend	Indicates whether the datafile can automatically grow in size.
Tablespace	The tablespace name.
Free Chunks	The number of free chunks in the datafile.
Largest Chunk	The largest free chunk (in MB) for the datafile.

Metrics

To spot and correct fragmentation in your tablespaces, you should periodically monitor the fragmentation levels of your tablespaces at a global level. Doing so helps you quickly spot tablespaces that are experiencing fragmentation issues. Seeing a tablespace with only one chunk of free space is a sign that a tablespace is not having fragmentation problems. Seeing a tablespace with a couple of free chunks may not be a big deal either, because the tablespace could be made up of more than one datafile. Each datafile has its own chunk or chunks of free space.

If you see a tablespace that has many chunks of free space, the next thing to do is drill down into it and find out if the tablespace is experiencing honeycomb or bubble fragmentation. Honeycomb fragmentation occurs when pockets of free space exist that are adjacent to one another. Bubbles are pockets of free space that are trapped between object segment extents.

You can eliminate honeycomb fragmentation with the ALTER TABLESPACE...COALESCE command. This command combines all pockets of adjacent free space into one extent. It is important to do this because when an object is newly introduced to a tablespace (or an existing object needs to extend), and a contiguous chunk of free space does not exist to accommodate an object's INITIAL or NEXT size allocation, Oracle must manually coalesce all available honeycombs to try and form a large enough free chunk. This is a performance hit. If possible, you should try to minimize performance hits.

Bubble fragmentation is a more serious matter and is normally only corrected through tablespace or database reorganization. The standard technique is to perform an export of all objects in the tablespace, drop the tablespace, and then import all the objects back into the tablespace. However, this technique may just treat the symptom and not the cause of bubble fragmentation. The real issue is to address the reuse of space extents within a tablespace so that bubble fragmentation does not occur in the first place.

Oracle8 and later offers the concept of locally-managed tablespaces, which can all but eliminate tablespace fragmentation. It totally does away with the storage parameters of MINEXTENTS, MAXEXENTS, PCTINCREASE, and NEXT. With locally managed tablespaces you either specify the initial extent size and let Oracle automatically size all other extents, or specify a uniform extent size for everything. Problems caused by fragmentation then become a thing of the past.

Correction

What can you do manually to help stave off fragmentation problems? A few suggestions include:

- Set PCTINCREASE to zero for all tablespaces and objects to promote same-sized extents.
- Specify equal-sized allotments for your INITIAL and NEXT object storage parameters.
- Group objects with like growth and storage needs together in their own tablespaces.

Of all your tablespaces, you want to avoid fragmentation problems in your SYSTEM tablespace the most as this is the major hotbed tablespace for Oracle activities. The easiest way to avoid this is to not allow any user (even the default DBA ID's SYS and SYSTEM) to have access to it. There are three ways to do this:

1 Ensure no user has a DEFAULT or TEMPORARY tablespace assignment of SYSTEM.

2 Ensure no user has a quota set for SYSTEM.

3 Ensure no user has been granted the UNLIMITED TABLESPACE privilege.

Extent Analysis Tab

Object fragmentation results when objects consume multiple extents of space in a tablespace rather than a single block of space. Although performance problems with respect to object fragmentation are not as severe as they were in previous versions of Oracle, some DBAs believe that response-time penalties can still be chalked up to this situation. When multiple extents exist for an object, the amount of time it takes Oracle to scan it can be longer than if the object was made up of only one extent. This typically holds true when extents are scattered on different parts of the physical server disk. In addition, a performance hit is taken each time an object must extend into another extent of space.

The Extent Analysis tab displays objects whose extent count has exceeded a user-suggested numerical limit. The table below lists the information available on this tab if any objects are found:

Column	Description
Owner	The user account that owns the object.
Object Name	The name of the object.
Object Type	The type of object (TABLE, INDEX, etc.).
Next Extent	The next extent amount of space the object will consume.
Total Extents	The number of extents the object is currently consuming.
Max Extents	The MAXEXTENTS limit imposed on the object by the DBA.
Largest Free Chunk (KB)	The size of the largest chunk.
Tablespace Name	The name of the tablespace.

Metrics

When is object fragmentation a problem for an object? Opinions on this vary widely. As a general rule of thumb, if an object is exhibiting response time degradations, and no other factor can be attributed to the slowdown, examine the object to see how badly fragmented it is. If extent numbers are seen exceeding 100 or more, than you might want to think about reorganizing the object or using locally managed tablespaces. This rule of thumb does not apply to objects assigned to locally-managed tablespaces as extent fragmentation is expected and not a performance issue.

Correction

The best preventative for this problem is specifying the correct allocation of space for the object in the first place, but what can you do if you have objects with high numbers of extents? For tables and indexes you have three options:

1 Use Oracle's export/import utility to export, drop, and import the fragmented objects back into the database with the export parameter COMPRESS=Y. This will bring the objects back into the database with one large extent. Make sure that large enough chunks of available free space exist to accept the object back in, or you may experience space allocation errors.

2 With Oracle8, you can use the ALTER TABLE … MOVE command to reorganize a table back into one extent in a very safe and efficient manner.

3 Use ALTER INDEX … REBUILD to reorganize indexes that have moved into multiple extents.

Another thing to keep an eye on is the next extent size of an object vs. the largest free chunk of space in the tablespace. Seeing objects with larger next extents than free space may signal a problem unless the tablespace can automatically extend to accommodate the object’s next extent.

User Space Tab

The User Space tab includes the following sections:

Top Space User Accounts User Space Detail

Top Space User Accounts

It is vital that a DBA know how the various accounts assigned to a database consume space. This is true in either production or development databases, because even in development databases, space can become an issue that causes major headaches. If development accounts are creating and cloning many objects without removing them, the result can be a lot of wasted space.

The Top Space User Accounts statistic shows how much total allocated space the top twenty accounts possess in the database.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Space View (DBArtisan - Oracle Performance Analyst)** > User Space Tab
Metrics	Seeing owner account with larger than expected data volumes could warrant further investigation. In addition, seeing accounts that have objects in more tablespaces than they have quota's for should also cause you to examine their account more closely to ensure the correct space privileges are in place.

User Space Detail

The User Space Detail section allows you to drill-down and see the specifics of the Top Space User accounts.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Space View (DBArtisan - Oracle Performance Analyst)** > User Space Tab

Objects View (DBArtisan - Oracle Performance Analyst)

In addition to a Home page, the Objects category of Oracle Performance Analyst includes the following tabbed pages:

Hot Objects Tab Invalid Objects Tab Objects Summary Tab

Tables/Indexes in Same Tablespace Tab

Home

The Objects Home page includes the following sections:

Bottleneck Analysis Pane Key Ratio Analysis Pane SQL Analysis Pane

Object/Buffer Pool Placement Pane User Object Analysis Pane

Key Ratio Analysis Pane

Object-related database activity can be examined using both ratio-based and wait/bottleneck-based analysis. Ratio-based analysis involves examining a number of key database ratios and statistical readings that can be used to indicate how active certain object types are. Performance ratios serve as roll-up mechanisms for busy DBAs to use for at-a-glance performance analysis.

When using ratio-based analysis, there are some standards to adhere to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that you examine come from the v$sysstat performance view. The performance view maintains a count of all the occurrences (in the VALUE column) of a particular database incident (in the NAME column) since the database was brought up. For databases that are kept up for long periods of time, these values can grow quite large and impacts how a particular ratio is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each database's individual 'personality' with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead you to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following ratios are used on the Performance Analyst Objects home page to succinctly communicate the general session-based performance levels of the monitored database:

Active Rollback Ratio Chained Row Fetch Ratio

The following satistic, available on this pane, duplicates a statistic available on the Key Ratio Analysis Pane of the Home View (DBArtisan - Oracle Performance Analyst) page:

- Rollback Contention Ratio

Active Rollback Ratio

To undo changes made to the Oracle database from within a transaction, Oracle writes data to individual rollback segments. You can also use these to maintain read consistency for multiple users of modified data. Because Rollback Segments read and write data, they can become very hot areas for I/O. This statistic is a ratio that displays the percentage of rollback segments currently involved in work.

Seeing consistent ratios of 80% or higher could indicate the need for more rollback segments on the database.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Objects View (DBArtisan - Oracle Performance Analyst)** > Home > Key Ratio Analysis Pane
Correction	Consider the use of UNDO management, which is where Oracle itself automatically manages rollback segments in special UNDO tablespaces.

Chained Row Fetch Ratio

In normal circumstances, a row of data should fit completely inside one Oracle block. Sometimes, however, this is not the case and the table suddenly contains chained or migrated rows (rows that span more than one data block).

Chaining occurs when a row is initially too large to fit inside one block. Two or more blocks are used by Oracle to hold the row. Migration deals with rows that have grown so much that they can no longer be contained within their original block. When this occurs, Oracle relocates the row out of its original block into another block, but leaves a pointer behind to indicate the relocation.

Both chaining and migration force Oracle to perform more than one I/O to retrieve data that could normally be obtained with a single I/O operation, with the end result being degraded performance.

The chain row fetch ratio statistic represents the percentage of rows fetched that were of either the chained or migrated row type.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Objects View (DBArtisan - Oracle Performance Analyst)** > Home > Key Ratio Analysis Pane
Metrics	If the amount of chained rows in your tables exceeds 25-30%, you should take steps to eliminate the problem. Further, if the amount of chained rows accessed in your system vs. total rows accessed approaches 20-30%, you can start the process of eliminating the chained and migrated rows.

Correction

You can locate tables that contain chained rows. Once found, there are a couple of ways to reorganize tables to remove the chaining/migration problem. However, the best way to deal with chained and migrated rows is to prevent them from occurring. The table below describes two methods:

Method	Description
Use a large block size	Because chaining and migrations occur when the Oracle block is too small to hold the rows in question, make sure you are using a large enough block size for each database you create. An 8KB block size or higher is normally recommended to help stave off chained and migrated rows. If you are using Oracle9i or later, you can create special tablespaces that have larger block sizes (16-32KB) than the overall database block size and place any table that is a candidate for chained/migrated rows into them.
Use proper values of PCTFREE for tables	The necessary amount of percent free in a table helps prevent row migrations from occurring. If you have a database that houses rows with the potential to grow substantially over their initially inserted size, provide a liberal amount of PCTFREE for each of your tables.

Bottleneck Analysis Pane

When using bottleneck analysis to troubleshoot a database, you cannot rely only on the information contained in the wait event views that Oracle provides. For example, an object attempts to extend into another extent of space in a tablespace and yet be denied if no such free space exists. Such a failure is not reflected in any wait event, but still represents a very real bottleneck to the database. In the same way that you cannot depend on only a few ratios to properly carry out ratio-based performance analysis, an administrator must include other statistical metrics in their overall bottleneck analysis framework to obtain an accurate performance risk assessment. Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system. The Performance Analyst Objects home page displays object-related bottlenecks decreases the overall performance of your system.

The following bottleneck indicators are used on the Performance Analyst Objects home page to communicate the general overall conditions of objects in the monitored database:

Chained Tables
Locked Objects
Session Blocks
Table High Water Mark Problems
Invalid/ Objects
Tables/Indexes in Same Tablespace

The following statistics, available on this pane, duplicate statistics available on the Top Session Bottlenecks of the Home View (DBArtisan - Oracle Performance Analyst) page:

Enqueue Waits
Objects with No Room to Expand
Free List Waits
Objects at Maximum Extents

Chained Tables

A row of data should fit completely inside one Oracle block. Sometimes, a table suddenly contains chained or migrated rows, which are rows that span more than one data block.

Chaining occurs when a row is initially too large to fit inside one block. Two or more blocks are used by Oracle to hold the row. Migration deals with rows that have grown so much that they can no longer be contained within their original block. Oracle relocates the row out of its original block into another block, but leaves a pointer behind to indicate the relocation.

Both chaining and migration force Oracle to perform more than one I/O to retrieve data that could normally be obtained with a single I/O operation, with the end result being degraded performance.

The Chained Tables statistic identifies the number of tables in the database that contain chained or migrated rows.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Objects View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Seeing many chained/migrated tables could indicate a database whose block size is undersized. However, even one chained/migrated row table can be a problem if that table is an often-referenced or scanned table.

Correction

You can locate the actual tables that contain chained/migrated rows. Once found, there are a couple of ways to reorganize tables to remove the chaining/migration problem. However, the best way to deal with chained and migrated rows is to prevent them from occurring. The table below describes two methods:

Method	Description
Use a large block size	Because chaining and migrations occur when the Oracle block is too small to hold the rows in question, make sure you are using a large enough block size for each database you create. An 8KB block size or higher is normally recommended to help stave off chained and migrated rows. If you are using Oracle9i or later, you can create special tablespaces that have larger block sizes (16-32KB) than the overall database block size and place any table that is a candidate for chained/migrated rows into them.
Use proper values of PCTFREE for tables	The necessary amount of percent free in a table helps prevent row migrations from occurring. If you have a database that houses rows with the potential to grow substantially over their initially inserted size, provide a liberal amount of PCTFREE for each of your tables.

For more information, see Bottleneck Analysis Pane.

Session Blocks

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Although Oracle supports unlimited row-level locking, blocking lock situations do crop up. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency.

Note: This statistic is also called Current Object Blocks on the Home View (DBArtisan - Oracle Performance Analyst) page, and Sessions Blocked on the Users View (DBArtisan - Oracle Performance Analyst)Home page.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Objects View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Investigate any indicator above zero immediately before the situation has a chance to mushroom.
Correction	Once discovered, a blocking lock situation can normally be quickly remedied. You can issue a KILL against the offending process, which eliminates the user's hold on the objects they were accessing. Other user processes then almost always complete in an instant. Discovering the blocked lock situation is eased by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is more difficult. You can look at the Users View (DBArtisan - Oracle Performance Analyst) detail and view all current blocking locks to see exactly which sessions are holding the currently restrictive locks. The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. Most DBAs who have had to face Oracle Forms applications have suffered through the dreaded SELECT … FOR UPDATE statements that place unnecessary restrictive locks on nearly every read operation, and know all too well that good coding practice is important. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

Invalid/ Objects

Objects like procedures, packages, functions, triggers, and views can become invalidated for a variety of reasons. The main cause is generally a dependent object that has been altered or removed from the system. However, other objects, like indexes, can become invalid also due to scenarios like SQL*Loader problems. If an object that has become invalid is still referenced (through an application or SQL query tool), a variety of problems can result. Sometimes Oracle reports a clear error stating the problem, while other times seemingly odd behavior is exhibited by the database.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Objects View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	There is no reason to have invalid objects in a production database. If your production databases have invalid objects that are no longer needed, promptly remove them from each system. Any needed objects that are indicating an invalid status should quickly be fixed before access problems develop. It is very normal for development databases to have invalid objects because developers create, modify, and compile objects all the time. The only invalid object that really should not be present in either a development or production database is an invalid index.
Correction	If code objects have become invalidated, you can issue an ALTER … COMPILE command to see if they compile properly and become valid once again. If they do not, then check the USER_ERRORS view for any resulting error messages. Indexes can be validated once more by using the ALTER INDEX … REBUILD command.

Locked Objects

Locked Objects is a count of all objects on the system that currently have some form of lock against them.

Note: Drilling down into the count of locked objects displays detail on each object that is locked, along with the user process holding the lock and the type of lock held.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Objects View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane

Table High Water Mark Problems

One form of object fragmentation revolves around what is called a table's high water mark. Tables that are the victim of much insert-delete activity can develop performance problems due to wasted space being present. A table's high-water mark equals the last block in the table that was used to hold data. The problem in Oracle is that this high-water mark is not reset by DELETE activity, so it is possible for a table to have absolutely no data in it but contain a high-water mark that is many blocks high. When such a table is scanned, Oracle will read up to the high-water mark even if no rows exist in the table at all. This can make for some unnecessarily large scan times. For a real world example, see Bottleneck Analysis Pane.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Objects View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Seeing non-zero numbers for this metric could indicate that table scans are taking longer than they should.
Correction	Performance Analyst does not currently provide diagnostics that identify the actual tables that contain artificially inflated high water marks. Embarcadero Space Analyst provides all the levels of detail regarding table high water marks plus a reorganization wizard that can reset the high water mark and thereby improve performance of table scans against tables suffering from this problem.

Tables/Indexes in Same Tablespace

One common Oracle practice is to physically separate tables and their child indexes onto distinct physical drives or file systems. Doing so reduces the potential for I/O contention when each object is accessed. The Tables/Indexes in Same Tablespace statistic displays a count of all tables whose index(es) reside in the same tablespace as they do.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Objects View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	For development databases where performance is not an issue, seeing high counts of this metric is not an issue. Even on production systems, high counts of this metric may not be a problem, or cannot be avoided, if the server is configured with one large RAID5 disk configuration, each distinct drive or file system does not have its own controller, or there simply are not enough disks on the server to perform adequate object striping.
Correction	If the server is configured to allow for intelligent object striping, and you see a large count of tables and indexes that are housed in the same tablespace, you can: *Create a separate tablespace for indexes on a drive different than that used for tables, and perform an ALTER INDEX … REBUILD operation on all indexes to move them to the new tablespace. Certain versions of Oracle even allow the rebuild operation to occur online (using the ONLINE keyword of the REBUILD command).Use the export/import utility to move indexes to a new tablespace.Use the ALTER TABLE … MOVE command to move tables to a new tablespace. Index-organized tables can be moved online.*

SQL Analysis Pane

Certain objects in an Oracle database are accessed more than others. These objects can become a source of contention given certain conditions. The SQL Analysis section of the Objects home page identifies the most accessed tables on the database.

Metrics	DML activity against tables can cause contention for space management objects like free lists. Oracle9i or later provides automatic segment management, which can remove problems with free lists and the like.

Object/Buffer Pool Placement Pane

Because data objects can be referenced with different types of usage patterns, Oracle8 offers the option to intelligently place objects into one of three buffer caches. The table below describes the buffer caches:

Buffer Cache	Description
KEEP	Designed to minimize misses in the buffer cache. Small objects that are frequently referenced are candidates for the KEEP buffer pool.
RECYCLE	Designed to avoid having large numbers of infrequently accessed blocks of data crowd out objects that need to be referenced in RAM, the RECYCLE pool is good for large objects that are scanned from beginning to end without the need for keeping all their data in RAM.
DEFAULT	The traditional cache for all other data objects.

Note: Unless you specify the KEEP or RECYCLE buffer cache, Oracle automatically places objects into the DEFAULT buffer cache.

The Object/Buffer Pool Placement section identifies how many objects are currently assigned to the various caches.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Objects View (DBArtisan - Oracle Performance Analyst)** > Home > SQL Analysis Pane
Metrics	When looking at overall database I/O activity, you should keep an eye out for objects that you can place into a particular buffer cache. Consider using a KEEP cache for relatively small, frequently accessed tables that require fast response times. Large tables with random I/O activity and are scanned from beginning to end are good candidates for a RECYCLE cache.
Correction	Objects can be placed into different buffer pools at object creation time (using the STORAGE... BUFFER_POOL option in Oracle8 or later) or existing objects can be set into a different pool with the ALTER command. **NOTE:** Tables, partitions, and indexes can be placed into the different caches. If you just want to use the DEFAULT buffer pool and not enable any special caches, you can still encourage Oracle to keep certain objects in the cache as long as possible using the CACHE parameter. For example, issuing the command ALTER TABLE…CACHE specifies that the blocks retrieved for a table be placed at the most recently used end of the LRU list in the DEFAULT buffer cache when a full table scan is performed. You can also use the CACHE hint in SQL statements to cache a table, but used in this form, the blocks are only cached until the next time the database is shut down.

User Object Analysis Pane

All objects are created inside of user accounts called schema. The User Object Analysis section of the Performance Analyst Objects home page shows schema that are ranked the worst in the several categories. The table below lists the categories:

Worst Schema for	Description
Chained Row Tables	The schema containing the most tables with chained rows is identified.
Invalid Objects	The schema containing the most invalid objects is identified.
Extent Problems	The schema containing the most objects with maximum extent limit or extent space deficit problems is identified.

Correction	To determine the degree and severity of each type of problem, go into detailed views that present specific information for each type of object problem.

Hot Objects Tab

The Hot Objects tab includes the following sections:

- Hot Code
- Hot Tables

Hot Tables

Certain objects in an Oracle database are accessed more than others. These objects can become a source of contention given certain conditions. The Hot Tables section identifies tables that are being frequently accessed through various SQL statements. The table below describes the information available in this section:

Column	Description
Table Owner	The owner of the table.
Table Name	The name of the table.
Issued Command	The SQL statement command issued against the table.
Executions	The number of SQL executions the object has experienced.
Disk Reads	The number of estimated disk reads from the object.
Buffer Gets	The number of estimated logical I/Os from the object.
Rows Processed	The number of estimated rows processed from the object.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Objects View (DBArtisan - Oracle Performance Analyst)** > Hot Objects Tab
Metrics	DML activity against tables can cause contention for space management objects like free lists. Oracle9i and later provides automatic segment management, which can remove problems with free lists and the like.

Hot Code

Certain objects in an Oracle database are accessed more than others. Data objects can become a source of contention given certain conditions, while code objects rarely cause contention issues. The Hot Code section identifies code objects (procedure, functions, etc.) that are being frequently accessed through various SQL statements. The table below describes the information available in this section:

Column	Description
Object Owner	The owner of the object.
Object Name	The name of the objects.
Object Type	The type of object (package, etc.).
Executions	The number of estimated executions for the object.
Loads	The number of times the object was loaded into the shared pool.
Locks	The number of locks the object has experienced.
Pins	The number of times the object was pinned in the shared pool.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Objects View (DBArtisan - Oracle Performance Analyst)** > Hot Objects Tab

Metrics Often referenced code objects should be pinned in the shared pool using the Oracle DBMS_SHARED_POOL package. This can also be done through DBArtisan's pinned code utility. Objects with many executions and loads should be considered candidates for pinning.

Invalid Objects Tab

The Invalid Objects tab includes the following sections:

Invalid Objects Detail Invalid Objects Summary

Invalid Objects Summary

Objects like procedures, packages, functions, triggers, and views can become invalidated for a variety of reasons, with the main cause being a dependent object that has been altered or removed from the system. However, other objects, like indexes, can become invalid also due to scenarios like SQL*Loader problems. If an object that has become invalid is still referenced (through an application or SQL query tool), a variety of problems can result. Sometimes Oracle will report a clear error stating the problem, while other times seemingly quirky behavior will be exhibited by the database. In any event, as a DBA you should be on the lookout for objects in your database that have suddenly become invalid.

The Invalid Objects Summary section displays a count of the various objects in the database that are invalid.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Objects View (DBArtisan - Oracle Performance Analyst)** > Invalid Objects Tab

Metrics There is no reason to have invalid objects in a production database. If your production databases have invalid/unusable objects that are no longer needed, you should promptly remove them from each system. Any needed objects that are indicating an invalid status should quickly be fixed before access problems develop. Correcting invalid objects like procedures and views often involves performing an ALTER…COMPILE operation. If the object status does not return to VALID, then further examination is warranted. It is very normal for development databases to have invalid objects because developers will no doubt be creating, modifying, and compiling objects all the time. The only invalid object that really should not be present in either a development or production database is an invalid index.

Invalid Objects Detail

Objects like procedures, packages, functions, triggers, and views can become invalidated for a variety of reasons, with the main cause being a dependent object that has been altered or removed from the system. However, other objects, like indexes, can become invalid also due to scenarios like SQL*Loader problems. If an object that has become invalid is still referenced (through an application or SQL query tool), a variety of problems can result. Sometimes Oracle will report a clear error stating the problem, while other times seemingly quirky behavior will be exhibited by the database. In any event, as a DBA you should be on the lookout for objects in your database that have suddenly become invalid.

The Invalid Objects Detail section displays information concerning invalid objects. The table below describes the information available in this section:

Column	Description
User Account	The user account that owns the objects.
Object Name	The name of the invalid object.
Object Type	The type of object (procedure, view, etc.).
Created On	The timestamp of when the objects was created.
Last DDL Date	The last structural modification date for the object.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Objects View (DBArtisan - Oracle Performance Analyst)** > Invalid Objects Tab
Metrics	There is no reason to have invalid objects in a production database. If your production databases have invalid objects that are no longer needed, you should promptly remove them from each system. Any needed objects that are indicating an invalid status should quickly be fixed before access problems develop. Correcting invalid objects like procedures and views often involves performing an ALTER…COMPILE operation. If the object status does not return to VALID, then further examination is warranted. It is very normal for development databases to have invalid objects because developers will no doubt be creating, modifying, and compiling objects all the time. The only invalid object that really should not be present in either a development or production database is an invalid index.

Objects Summary Tab

The Objects Summary tab includes the following sections:

Section	Description
Global Summary	This section displays a count of all user objects in the database (minus SYS and SYSTEM objects).
User Summary	This section displays a count across user accounts of all the major object types in the database. NOTE SYS and SYSTEM accounts are not included.

Tables/Indexes in Same Tablespace Tab

To help avoid physical I/O contention, it is wise to separate a table and its indexes onto separate physical drives. This is normally accomplished by putting tables into one tablespace (whose datafiles exist on one drive) and their indexes into a different tablespace (whose datafiles exist on another drive).

The Tables/Indexes in Same Tablespace tab identifies indexes whose tables reside in the same tablespace as they do. The table below describes the information available in this section:

Column	Description
Table Owner	The owner of the table.
Table Name	The name of the table.
Index Name	The name of the index that resides in the same tablespace as the table.
Tablespace Name	The name of the tablespace.

Metrics If tables and indexes are found in the same tablespace, then either the table or index should be moved to a different tablespace if the server is set up in such a way that different drives and controllers are available that make the relocation worthwhile. Indexes are traditionally easier to move than tables. They can be relocated through many reorganization tools (like Embarcadero Space Analyst) or through the manual use of the ALTER INDEX … REBUILD command.

OS View (DBArtisan - Oracle Performance Analyst)

In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database sever for resources (CPU, I/O, or Memory), a slow CPU, insufficient or slow I/O devices, and insufficient memory. The OS Statistics page of Performance Analyst lets you examine operating system metrics for the following platforms:

- AIX
- HP-UX

Note: To view processor info and swap disk info on an HP-UX box, you need to login as ROOT in the OS login.

- Linux
- Solaris
- Unix
- Windows XP and 2000

Note: The statistics available on the OS page depend on the operating system platform.

Tip: If you magnify a graph to change back to the original size or minimize the size, close the OS Page and then reopen it again. Performance Analyst displays the graph in its original form and size.

In addition to a Home page, the OS category of Oracle Performance Analyst includes the following tabbed pages:

CPU Tab	I/O Tab	Memory Tab
Network Tab	Processes Tab	Location

Home

The OS Home page includes the following sections:

Bottleneck Analysis Pane	Disk Analysis Pane	Key Resource Usage Pane
Memory Analysis Pane	Workload Analysis Pane	

Key Resource Usage Pane

The following ratios are used on the Performance Analyst OS home page to communicate the general overall performance levels of the operating system:

Disk Time	Load Average	Processor Time
Paged Memory Used (Windows)	Swap Memory Used (AIX, HP-UX, Linux, Solaris, Unix)	

Tip: To open the CPU tab, right-click Processor Time. To open the I/O tab, right-click Disk Time. To open the Memory tab, right-click Paged Memory Used.

Disk Time

The Disk Time statistic is the percentage of elapsed time that the selected disk drive/device was busy servicing read or write requests.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	You should avoid consistently seeing values for this statistic greater then 90%.
Correction	Add more disk drives and partition the files among all of the drives.

Load Average

The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	High load averages usually mean that the system is being used heavily and response time is correspondingly slow.

Processor Time

The Processor Time statistic indicates the percentage of time the processor is working. This counter is a primary indicator of processor activity.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	If your computer seems to be running sluggishly, this statistic could be displaying a high percentage.
Correction	Upgrade to a processor with a larger L2 cache, a faster processor, or install an additional processor.

Paged Memory Used

The Paged Memory Used statistic is the ratio of Commit Memory Bytes to the Commit Limit. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk. The commit limit is determined by the size of the paging file. As the paging file increases, so does the commit limit.

Note: This statistic is available for the Windows platform.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	This value displays the current percentage value only and not an average. If the percentage of paged memory used is above 90%, you may be running out of memory.
Correction	Increase the size of page file.

Swap Memory Used

The Swap Memory Used statistic is the percentage of swap space currently in use.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	If the percentage of swap memory used is above 90%, you may be running out of memory.
Correction	Increase the size of your swap files.

Bottleneck Analysis Pane

The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

Average Disk Queue Length

Processor Queue Length

Available Physical Memory

Interrupts/Sec

Processor Queue Length

Network Output Queue Length/Network Queue Length

Processor Speed

Tip: To open the I/O tab, right-click any Details menu item. To open the CPU tab, right-click the Item Processor Queues Length Details menu item. To open the Network tab, right-click Network Output Queues Length.

Note: The statistics available in this section depend on the platform of operating system.

Average Disk Queue Length

The Average Disk Queue Length statistic is the average number of both read and write requests that were queued for the selected disk during the sample interval.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	This metric is useful in identifying I/O related bottlenecks. If the disk queue lengths for certain disks are consistently much higher than others, you may need to redistribute the load among available disks. If the disk queues lengths for all disks are consistently large, and you see a high amount of I/O activity, your disks may be inefficient.
Correction	Some things you can do if you have problems with this statistic include: *Redistribute the data on the disk with the large average disk queue to other disks.Upgrade to faster disk(s).*

Interrupts/Sec

Interrupts/Sec is the average rate, in incidents per second, at which the processor received and serviced hardware interrupts. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended. The system clock typically interrupts the processor every 10 milliseconds, creating a background of interrupt activity. This statistic shows the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis PaneOracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > CPU Tab
Metrics	A high value indicates possible excessive hardware interrupts; justification is dependent on device activity.

Network Output Queue Length/Network Queue Length

The Network Output Queue Length statistic is the number of threads in the processor queue.

Note: The name of this statistic depends on the platform of the operating system.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.
Correction	A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Page Faults/Sec

The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis PaneOracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Memory Tab
Metrics	This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
Correction	If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Processor Queue Length

The Processor Queue Length statistic is the number of threads in the processor queue.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis PaneOracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > CPU Tab
Metrics	Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.
Correction	A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Processor Speed

The Processor Speed statistic displays the speed of the active processor in MHz. The speed is approximate.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane

Processor

The Processor Statistic displays the type of processor currently in use, for example, GenuineIntel.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane

Memory Analysis Pane

The following metrics are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

Available Paged Memory (Windows)	Available Physical Memory	Available Swap Memory (AIX, HP-UX, Linux, Solaris, Unix)
Total Paged Memory/Total Swap Memory	Total Physical Memory	

Tip: To open the Memory tab, right-click any Details menu item.

Note: The statistics available in this section depend on the platform of operating system.

Available Paged Memory

The Available Paged Memory statistic shows the amount of virtual memory available for the processes.

Note: This statistic is available for the Windows platform.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Memory Analysis Pane
Metrics	If the available virtual memory is less than 10% of the total virtual memory, your system may run out of memory.
Correction	Increase the size of page file.

Available Physical Memory

The Available Physical Memory statistic represents the amount of RAM available to all processes.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Memory Analysis Pane
Metrics	This counter displays the last observed value only and not an average. Use this value with the Total physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.
Correction	Some things you can do if you have problems with this statistic include: *Check the running processes to see if there are any memory leaks.Stop any services that are not required.Install additional RAM.*

Available Swap Memory

The Available Swap Memory statistic represents the amount of virtual memory available for the processes.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Memory Analysis Pane
Metrics	If the available Available Swap Memory is less than 10% of the total Swap Memory, your system may run out of memory.
Correction	Increase the size of swap files.

Total Paged Memory/Total Swap Memory

The Total Paged Memory statistic shows the maximum amount of virtual memory available to all processes.

Note: The name of this statistic depends on the platform of the operating system.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Memory Analysis Pane
Metrics	It is recommended that this be 1½ - 3 times the amount of RAM on the system.

Total Physical Memory

The Total Physical Memory statistic shows the amount of physical memory installed on your computer.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Memory Analysis Pane
Metrics	This is an informational metric and displays the total amount installed on the machine. Use this value with the available physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Disk Analysis Pane

The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

Free Disk Space Total Disk Space Used Disk Space

Tip: To open the Space tab, right-click any Details menu item.

Note: The statistics available in this section depend on the platform of operating system.

Free Disk Space

The Free Disk Space statistic shows the unallocated space, in megabytes on all logical disk drives.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Disk Analysis Pane
Correction	There are many ways to ensure a database does not encounter a space problem due to physical space limitations: *If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.*

Used Disk Space

The Used Disk Space statistic shows the amount of allocated space, in megabytes on all logical disk drives.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Disk Analysis Pane
Correction	There are many ways to ensure a database does not encounter a space problem due to physical space limitations: *If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.*

Total Disk Space

Total Disk Space displays the total allocated and unallocated space, in megabytes on all logical disk drives.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Disk Analysis Pane
Correction	There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations, here are two: *If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.*

Workload Analysis Pane

The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

Number of Logins	Number of Processes	Top CPU Process
Top I/O Process	Top Memory Process	

Tip: To open the Processes tab, right-click any Details menu item.

Note: The statistics available in this section depend on the platform of operating system.

Number of Logins

This statistic displays the total number of logins on the server.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Workload Analysis Pane

Number of Processes

This statistic displays the total number of processes on the server.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Workload Analysis Pane

Top Memory Process

Top Memory Process shows the current process that is consuming the most amount of memory. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Workload Analysis Pane
Metrics	If you are running out of memory on the system, this is a quick way to identify the top memory user. If the displayed process is using a significant portion of the total memory, it could be causing the memory issues.

Top CPU Process

Top CPU Process shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Workload Analysis Pane
Metrics	If the amount of CPU time used by this process is close to 100% and the CPU usage is very high, this process may be the bottleneck on the server.
Correction	Investigate the process further to see if it is in an inconsistent state. Also, look at minimum requirements for CPU speed for the process. You may need to upgrade your CPU.

Top I/O Process

The Top I/O Process statistic shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Home > Workload Analysis Pane

CPU Tab

The CPU tab of the OS Detail includes the following sections:

Context Switches/Sec CPU Events CPU Utilization

Load Averages

The following statistics available on this tab, are duplicates of statistics available on the Bottleneck Analysis Pane of the **OS View (DBArtisan - Oracle Performance Analyst)**Home page:

Interrupts/Sec Processor Queue Length

Note: The sections available on this tab depend on the platform of operating system.

CPU Utilization

The CPU Utilization section includes the following information:

% Privileged Time % User Time

% Privileged Time

The % Privileged Time statistic is the percentage of elapsed time that the process threads spent executing code in privileged mode.

Note: For Windows systems, when a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. These kernel commands, are considered privileged to keep the low-level commands executing and prevent a system freeze. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > CPU Tab
Metrics	The ideal range should be 0-40% (less than 40% indicates excessive system activity).
Correction	If your CPU consistently runs at less than 40% you may need to upgrade your system to include a faster processor(s).

% User Time

The % User Time statistic is the percentage of elapsed time the processor spends in the user mode. User mode is a restricted processing mode designed for applications, environment subsystems, and integral subsystems. The alternative, privileged mode, is designed for operating system components and allows direct access to hardware and all memory. The operating system switches application threads to privileged mode to access operating system services. This counter displays the average busy time as a percentage of the sample time.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > CPU Tab
Metrics	If the Privileged Time is high in conjunction with Physical Disk Reads, consider upgrading the disk I/O subsystem.

CPU Events

The CPU Events section includes the following information:

Context Switches/Sec Load Averages System Calls/Sec

Note: The statistics available in this section depend on the platform of operating system.

Context Switches/Sec

The Context Switches/Sec section shows the combined rate at which all processors on the computer are switched from one thread to another. Context switches occur when a running thread voluntarily relinquishes the processor, is preempted by a higher priority ready thread, or switches between user-mode and privileged (kernel) mode to use an Executive or subsystem service.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > CPU Tab
Metrics	The ideal range should be between 0-10,000. GA number greater then 10,000 may indicate too many threads contending for resources.

System Calls/Sec

System calls are functions which require the services of the operating system. Usually they involve access to some data that users must not be permitted to corrupt or even change.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > CPU Tab
Correction	Try to find a good ratio between the amount of time the operating system runs (fulfilling system calls and doing process scheduling), and the amount of time the application runs. Your goal should be running 60% to 75% of the time in application mode, and 25% to 40% of the time in operating system mode. If you find that the system is spending 50% of its time in each mode, then you should investigate to determine what is wrong.

Load Averages

The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > CPU Tab
Metrics	High load averages usually mean that the system is being used heavily and the response time is correspondingly slow.

Processes Tab

The Processes tab of the OS Detail page succinctly communicates the general overall performance levels of processes. The columns available in this table depend on the platform of operating system. The table below describes the information available in the table on this tab:

Column	Description
Process	The name of the process.
User	The user of the process.
ID	The ID Process is the unique identifier of this process. ID Process numbers are reused, so they only identify a process for the lifetime of that process.
CPU	The CPU is the percentage of elapsed time that all of process threads used the processor to execution instructions.
User Mode	The User Mode is the percentage of elapsed time that the process threads spent executing code in user mode.
Memory WINDOWS ONLY	Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Memory (MB)	Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Memory	Memory is the percentage of the memory used of the total memory.
Active Memory	Active Memory is the amount of committed virtual memory, in bytes for this process. Active memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive. This counter displays the last observed value only; it is not an average.
I/O Data	The rate at which the process is reading and writing bytes in I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.
Elapsed Time	The total elapsed time, in seconds, that this process has been running.
Thread Count	The number of threads currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.
Handle Count	The total number of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in this process.
Priority	The current base priority of this process. Threads within a process can raise and lower their own base priority relative to the process' base priority.
Creating Proc ID	The Creating Process ID value is the Process ID of the process that created the process. The creating process may have terminated, so this value may no longer identify a running process.
Page Faults/Sec	Page Faults/Sec is the rate at which page faults by the threads executing in this process are occurring. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This may not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with whom the page is shared.

Page File	Page File is the current number of kilobytes that this process has used in the paging file(s). Paging files are used to store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory.
Private	Private is the current size, in kilobytes, of memory that this process has allocated that cannot be shared with other processes.

I/O Tab

The sections available on this tab depend on the platform of operating system. The I/O tab includes the following sections:

Average Disk Queue/Total Disk Queue	Bytes per I/O Operation	Disk I/O Time
Disk Transfers/Sec	I/O Details	Transfer Rate

I/O Details

The table below describes the information available in this section:

Column	Description
Disk	The disk number assignment.
Reading (KB/s)	The amount of bytes read from the device.
Writing (KB/s)	The amount of bytes written to the device.
Disk Read Time	Disk Read Time is the percentage of elapsed time that the selected disk drive was busy servicing read requests.
Disk Write Time	Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.
Disk Time	Disk Time is the percentage of elapsed time that the selected disk was busy servicing requests.
Avg. Read Queue	Avg. Disk Read Queue Length is the average number of read requests that were queued for the selected disk during the sample interval.
Avg. Write Queue	Avg. Disk Write Queue Length is the average number of write requests that were queued for the selected disk during the sample interval.
Disk Reads/Sec	Disk Reads/Sec is the rate of read operations on the disk.
Disk Writes/Sec	Disk Writes/Sec is the rate of write operations on the disk.

Note: The columns available in this table depend on the platform of operating system.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > I/O Tab

Transfer Rate

The Transfer Rate section succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

Statistic	Description	Metrics
Reading	The average number of bytes transferred from the disk during read operations.	It is measured in number of KBs per second.
Writing	The average number of bytes transferred to the disk during write operations.	It is measured in number of KBs per second.
Transferred	The amount of time required to transfer the data between the device's cache and the actual device media.	N/A

Note: The statistics available in this section depend on the platform of operating system.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > I/O Tab

Bytes per I/O Operation

The Bytes per I/O Operation section of the OS Detail succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

Statistic	Description	Metrics
Reading	The average number of bytes transferred from the disk during read operations.	It is measured in number of bytes per second.
Writing	The average number of bytes transferred to the disk during write operations.	It is measured in number of bytes per second.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > I/O Tab

Average Disk Queue/Total Disk Queue

The Average Disk Queue section of the OS Detail page succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

Statistic	Description	Information
Average Read Queue	Average Read Queue Length is the average number of read requests that were queued for a particular disk.	Sustained loads on any of the disk queue lengths could represent problems within your system.
Average Write Queue	Average Write Queue is the average number of write requests that were queued for a particular disk.	Sustained loads on any of the disk queue lengths could represent problems within your system. Contact your Windows System Administrator for assistance in troubleshooting these problems.
Average Queue	Average time in ms. to service I/O requests which for disks includes seek, rotational latency and data transfer time).	Sustained loads on any of the disk queue lengths could represent problems within your system. Contact your Unix System Administrator for assistance in troubleshooting these problems.

Note: The statistics available in this section depend on the platform of operating system.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > I/O Tab
Correction	Contact your Windows System Administrator for assistance in troubleshooting these problems.

Disk I/O Time

The Disk I/O Time section of the OS Detail page succinctly communicates the general overall performance levels of the disk's I/O. The table below describes the statistics in this section:

Statistic	Description	Metrics
% Disk Read Time	% Disk Read Time is the percentage of time during the sample interval that the disk is busy servicing read requests.	Use this in conjunction with the % Disk Write Time to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.
% Disk Write Time	% Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.	Use this in conjunction with the % Disk Read Time to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.
% Disk Time	% Disk Time is the percentage of time during the sample interval that the disk is busy servicing requests.	Use this statistic to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.

Note: The statistics available in this section depend on the platform of operating system.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > I/O Tab

Disk Transfers/Sec

The Disk Transfers/Sec section of the OS Detail page succinctly communicates the general overall performance levels of the disk transfers. The table below describes the statistics in this section

Statistic	Description
Reads/Sec	The the rate of read operations on the disk.
Writes/Sec	The rate of write operations on the disk.
Transfers/Sec	The rate of transfer operations per second.

Note: The statistics available in this section depend on the platform of operating system.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > I/O Tab

Memory Tab

The Memory tab of the OS Detail page includes the following sections:

Buffer Size	Cache Efficiency	Cache Size
Page Replacements	Paging Activity	Page Replacements

The following statistic, available on this tab, duplicates astatistic available on the Bottleneck Analysis Pane of the **OS View (DBArtisan - Oracle Performance Analyst)**Home page:

- Page Faults/Sec

Note: The sections available on this tab depend on the platform of operating system.

Buffer Size

The Buffer Size statistic represents the amount of cache used by the kernel in kilobytes.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Memory Tab

Cache Efficiency

The Cache Efficiency section of the Memory tab succinctly communicates the general overall performance levels of the server's memory. The following statistics are available in this section:

Copy Read Hits % Data Map Hits % MDL Read Hits %

Pin Read Hits %

Copy Read Hits %

The Copy Read Hits % statistic is the percentage of cache copy read requests that hit the cache and does not require a disk read to provide access to the page in the cache.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Memory Tab
Metrics	When the page is pinned in the memory, the page's physical address in the file system cache will not be altered. A copy read is a file read operation where a page in the cache is copied to the application's buffer. Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.
Correction	Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate

Data Map Hits %

The Data Map Hits % statistic is the percentage of data maps in the file system cache that could be resolved without having to retrieve a page from the disk.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Memory Tab
Metrics	Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.
Correction	Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

MDL Read Hits %

The MDL Read Hits % statistic is the percentage of Memory Descriptor List Read requests to the file system cache that hit the cache and does not require disk access to provide memory access to the pages in the cache.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Memory Tab
Metrics	Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.
Correction	Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Pin Read Hits %

The Pin Read Hits % statistic is the percentage of pin read requests that hit the file system cache and does not require a disk read in order to provide access to the page in the file system cache.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Memory Tab
Metrics	Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.
Correction	Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Cache Size

The Cache Size statistic represents the amount of cache used by the kernel in kilobytes.

Memory Available

The Memory Available section of the OS Detail page succinctly communicates the general overall performance levels of the server's memory. The statistics available in this section depend on the platform of operating system. The table below describes these statistics:

Statistic	Description	Metrics	Troubleshooting
Physical Memory	The Physical Memory statistic is the amount of committed virtual memory, in megabytes. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk.	This counter displays the last observed value only and not an average.	N/A
Physical	The Physical statistic is the total physical memory available.	N/A	N/A
Paged Memory	The Paged Memory statistic is the amount of committed virtual memory, in bytes. Used Memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive.	This counter displays the last observed value only; it is not an average.	N/A
Swap Memory	The Swap Memory statistic is the free, unreserved swap space.	N/A	Increase the size of your swap file or add additional files if you consistently see a shortage of free, unreserved swap space.
Swap	The Swap statistic is the total swap memory being used.	N/A	N/A
Total Physical	The Total Physical statistic is the total physical memory available.	N/A	N/A
Total Swap	The Total Swap statistic is the total swap memory available.	N/A	N/A
Swap-ins	The number of memory pages paged in from swapspace.	N/A	If the system is consistently low on free memory (over a 30 second average), the memory scheduler will start to swap out processes. Increase the size of your swap file or add additional files.
Swap-outs	The number of memory pages paged in from swapspace.	N/A	If the system is consistently low on free memory (over a 30 second average), the memory scheduler will start to swap out processes. Increase the size of your swap file or add additional files.
Buffers	The Buffers statistic represents the amount of buffers used by the kernel in kilobytes.	N/A	N/A

Page Faults/Sec	The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.	This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.	If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Memory Tab

Paging Activity

The Paging Activity section includes the following statistics:

Blocks Input/Sec	Blocks Output/Sec	Paged In
Paged Out	Pages Input/Sec	Pages Output/Sec

Note: The statistics available in this section depend on the platform of operating system.

Blocks Input/Sec

The Blocks Input/sec statistic is the number of blocks sent to a block device per second.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Memory Tab
Correction	Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck. *Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms*.

Blocks Output/Sec

The Blocks Output/sec statistic is the number of blocks received from a block device per second).

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Memory Tab
Correction	Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck. *Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms*.

Paged In

The number of disk block paged into core from disk which occurs when the server is experiencing a shortage of RAM. Any nonzero value indicates excessive swap activity. In any virtual memory environment, it is not uncommon to see RAM pages moved out to a swap disk. This is a special disk area in UNIX that's reserved for holding memory pages so that the processor is capable of addressing RAM in excess of its full capability. While page-out of operations are a normal part of any server's operations, page-in operations indicate that the real amount of available RAM has been exceeded and that additional RAM pages are required on the server.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Memory Tab
Correction	High values indicate the possible need to increase the amount of RAM on the system.

Paged Out

The number of disk block paged out of core from disk. In any virtual memory environment, it is not uncommon to see RAM pages moved out to a swap disk. This is a special disk area in UNIX that's reserved for holding memory pages so that the processor is capable of addressing RAM in excess of its full capability. While page-out of operations are a normal part of any server's operations, page-in operations indicate that the real amount of available RAM has been exceeded and that additional RAM pages are required on the server.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Memory Tab

Pages Input/Sec

The Pages Input/Sec statistic is the number of pages read from disk to resolve hard page faults. Hard page faults occur when a process requires code or data that is not in its working set or elsewhere in physical memory, and must be retrieved from disk.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Memory Tab
Metrics	This value was designed as a primary indicator of the kinds of faults that cause system-wide delays. It includes pages retrieved to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. This counter counts numbers of pages, and can be compared to other counts of pages, such as Memory: Page Faults/sec, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
Correction	Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck. *Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms*.

Pages Output/Sec

The Pages Output/Sec statistic is the number of pages written to disk to free up space in physical memory. Pages are written back to disk only if they are changed in physical memory. A high rate of pages output might indicate a memory shortage.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Memory Tab
Metrics	Windows NT writes more pages back to disk to free up space when low in physical memory. This counter counts numbers of pages, and can be compared to other counts of pages, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
Correction	Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck. *Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms*.

Page Replacements

The following statistics are available in this section:

Memory Freed (Pages/sec) Clock Algorithm Scans (Pages/sec)

Memory Freed (Pages/sec)

Freed or destroyed (Kb/s).

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Memory Tab

Clock Algorithm Scans (Pages/sec)

The VMM uses a technique known as the clock algorithm to select pages to be replaced. This technique takes advantage of a referenced bit for each page as an indication of what pages have been recently used (referenced). When the page-stealer routine is called, it cycles through the PFT, examining each page's referenced bit.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Memory Tab

Space Tab

The Space tab of the OS Detail page includes the following sections:

- Device Summary
- Device Detail

Disk Space Free

The Disk Space Free metric displays the amount of free disk space in megabytes.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Space Tab

Disk Space Detail

The Disk Space Detail section of the Space tab succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

Statistic	Description
Partition	The drive letter of the disk.
Local Filesystem	The name of the file system.
Type	The type of file system.
Total Space	Total size of the disk/device's capacity expressed in MBs.
Used Space	Amount of MBs currently allocated on the particular disk/device.
Free Space	Amount of MBs currently unallocated and free on the particular disk/device.
Capacity	The percentage of space used on the device.
Mounted On	The mount point of the device.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Space Tab

Device Summary

The Device Summary section of the Space tab displays a bar chart for all devices. The Device Summary section succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

Statistic	Description
Used	Amount of MBs currently allocated on the particular disk/device.
Free	Amount of MBs currently unallocated and free on the particular disk/device.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Space Tab

Device Detail

The Device Detail section of the Space tab succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

Statistic	Description
Partition	The drive letter of the disk.
Local Filesystem	The name of the file system.
Type	The type of file system.
Total Space	Total size of the disk/device's capacity expressed in MBs.
Used Space	Amount of MBs currently allocated on the particular disk/device.
Free Space	Amount of MBs currently unallocated and free on the particular disk/device.
Capacity	The percentage of space used on the device.
Mounted On	The mount point of the device.

Note: The statistics available in this section depend on the platform of operating system.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Space Tab

Network Tab

The Network tab of the OS Detail page succinctly communicates the general overall performance levels of the server's networking. The Network tab of the OS Detail page includes the following sections:

Network Details	Packet Collisions	Packet Discards
Packet Errors	Transmission Rate	Transmission Rate (Bytes)
Transmission Queue Length		

Note: The sections available on this tab depend on the platform of operating system.

Network Details

The statistics available in this section depend on the platform of operating system. The table below describes the information available in this section:

Column	Description
Network Interface	The name of network interface.
INET Address/Address	The IP address assigned to the network interface.
Pkts Sent/Sec	The number of packets sent per second.
Pkts Received/Sec	The number of packets received per second.
Sent (KB/Sec)	The number of bytes sent per second.
Received (KB/Sec)	The number of bytes received per second.
Out Pkts Discarded	The number of outbound packets discarded.
In Pkts Discarded	The number of inbound packets discarded.
Out Pkt Errors	The number of outbound packet errors.
In Pkt Errors	The number of inbound packet errors.
Queue Length	The queue length.
Collisions	The number of collisions.
Packets Discarded	The number of packets discarded.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Network Tab

Transmission Rate

The Transmission Rate section of the Network tab succinctly communicates the packet transmission rate. The following statistics are available in this section:

Packets Received/Sec Packets Sent/Sec

Packets Sent/Sec

The Packets Sent/Sec statistic is the number of packets sent over each network adapter per second.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Network Tab

Packets Received/Sec

The Packets Received/Sec statistic is the number of packets received over each network adapter per second.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Network Tab

Transmission Rate (Bytes)

The Transmission Rate (Bytes) section of the Network tab succinctly communicates the packet transmission rate. The following statistics are available in this section:

Received (KB/Sec) Sent (KB/Sec)

Sent (KB/Sec)

The Sent (KB/Sec) statistic is the rate at which bytes are sent over each network adapter including framing characters.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Network Tab

Received (KB/Sec)

The Received (KB/Sec) statistic is the rate at which bytes are received over each network adapter, including framing characters.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Network Tab

Transmission Queue Length

The number of pending outgoing packets on either the software transmit queue or the hardware transmit queue

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Network Tab

Packet Collisions

A collision occurs when two devices on the network sense that the network is idle and end up trying to send data at the same time. Some collisions are normal on a repeated network, but excess collisions can cause serious performance problems.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Network Tab

Correction Contact your network administrator if there is high rate of persistent Packet Collisions.

Packet Discards

The Packet Discards section of the Network tab succinctly communicates the packet discard rate. The following statistics are available in this section:

Inbound Packets Discarded Outbound Packets Discarded Packet Discards

Outbound Packets Discarded

The Outbound Packets Discarded statistic is the number of outbound packets that were discarded even though no errors had been detected to prevent from being transmitted.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Network Tab

Correction A possible reason for discarding such a packet could be to free up buffer space.

Inbound Packets Discarded

Received Packets Discarded is the number of received packets that were discarded even though no errors had been detected to prevent from being transmitted.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Network Tab

Correction

A possible reason for discarding such a packet could be to free up buffer space.

Packet Discards

The Packet Discards statistic represents the number of network packets discarded.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Network Tab

Correction Contact your network administrator if there is high rate of persistent packet discards.

Packet Errors

The Packet Errors section of the Network tab succinctly communicates the packet error rate. The following statistics are available in this section:

Inbound Packet Errors Outbound Packet Errorss

Outbound Packet Errors

The Outbound Packet Errors statistic is the outbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Network Tab

Inbound Packet Errors

The Packets Received Errors statistic is the received packets that contained errors preventing them from being deliverable to a higher-layer protocol.

Location Oracle Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - Oracle Performance Analyst)** > Network Tab

Top SQL View (DBArtisan - Oracle Performance Analyst)

One or two bad queries can cause a lot of trouble for the remaining sessions in a database, and so can one or two bad queries. It is important to find them before they get into a production system, but sometimes a few sneak through.

By applying custom filters and performance-related thresholds, the Top SQL view locates inefficient SQL. By applying filters to certain I/O and statistical counters, you hope to isolate queries that far exceed their nearest competitors in the same area (like disk reads). When you find them, you should reduce the number of sorts a query performs. Or, for a query that returns only a few records, you should try to minimize the number of rows a query processes.

When you begin to look for inefficient SQL in a database, there are two primary questions you need to answer:

1 What has been the worst SQL that's historically been run in my database?

2 What is the worst SQL that's running right now in my database?

When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It may be that the query is in an inefficient Transact SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

The Top SQL view displays requested SQLOracle datasources. The following tabs are available on the Top SQL page:

Summary Criteria SQL Details

Recent History (Oracle 10g only)

Summary

The Summary tab of the Top SQL page includes the following sections:

Top SQL Statements Top SQL Details

Top SQL Statements

A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times.

Before you can identify problem SQL in your database, you have to ask the question of what 'bad SQL' is. What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every Oracle professional's satisfaction. The table below describes some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

Criteria	Description
Overall Response (Elapsed) Time	This is how much time the query took to parse, execute, and fetch the data needed to satisfy the query. It should not include the network time needed to make the round trip from the requesting client workstation to the database server. This statistic is available in Oracle9i and later.
CPU Time	This is how much CPU time the query took to parse, execute, and fetch the data needed to satisfy the query.
Physical I/O	Often used as the major statistic in terms of identifying good vs. bad SQL, this is a measure of how many disk reads the query caused to satisfy the user's request. While you certainly want to control disk I/O where possible, it is important that you not focus solely on physical I/O as the single benchmark of inefficient SQL. Make no mistake, disk access is slower than memory access and also consumes processing time making the physical to logical transition, but you need to look at the entire I/O picture of a SQL statement, which includes looking at a statements' logical I/O as well.
Logical I/O	This number of memory reads the query took to satisfy the user's request. The goal of tuning I/O for a query should be to examine both logical and physical I/O, and use appropriate mechanisms to keep both to a minimum.
Repetition	This is a measure of how often the query has been executed. A problem in this area is not as easy to spot as the others unless you know your application well. A query that takes a fraction of a second to execute may still be causing a headache on your system if it is executed erroneously (for example, a query that executes in a runaway PL/SQL loop) over and over.

Top SQL Details

A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times.

Before you can identify problem SQL in your database, you have to ask the question of what 'bad SQL' is. What criteria do you use when you begin the hunt for problem SQL in your critical systems? Even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every Oracle professional's satisfaction. The Top SQL Details section displays the top SQL statements across various performance categories. The table below describes the information available in this section:

Information	Description
Top For	Indicates the performance category for the SQL statement.
Usage	Displays the percent usage for this SQL statement in this performance category.
Hash value	A unique identifier for the SQL statement assigned by Oracle.
SQL text	A snippet of the actual SQL statement.
User name	The username of the account that originally parsed the SQL.
Disk reads	The number of disk reads for the statement.
Buffer gets	The number of logical reads for the statement.
Parse calls	The number of parse calls for the statement.
Sorts	The number of sorts caused by the statement.
Executions	The number of times that statement has been executed.
Rows processed	The number of rows processed by the statement.
First load time	The date/time of when the statement was first loaded into the shared pool.
Sharable memory	The amount of sharable memory used by the statement.
Persistent memory	The amount of persistent memory used by the statement.
Runtime memory	The amount of runtime memory used by the statement.
Address	The address of the SQL statement.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Top SQL View (DBArtisan - Oracle Performance Analyst)** > Summary

Criteria

The Top SQL view displays all SQL running or executed in the current datasource. You can use the lists and boxes to reorder the display according to your specific needs. The Criteria tab of the Top SQL page includes the following sections that you can mix and match to give you the information you want to find.

- Demographic Criteria
- Statistical Criteria
- Sort Criteria

SQL Details

The SQL Details tab displays the following sections:

Top SQL Results Selected SQL Text

Top SQL Results

A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times.

Before you can identify problem SQL in your database, you have to ask the question of what 'bad SQL' is. What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every Oracle professional's satisfaction. The Top SQL results grid displays the top SQL statements across various performance categories. The table below describes the information available in this section

Information	Description
SQL Text	A snippet of the full SQL statement. Clicking the statement will cause the full statement to be presented in the Selected SQL Text grid.
User name	The username of the account that originally parsed the SQL.
Disk reads per execution	The average number of disk reads caused by the statement each time it is executed.
Buffer gets per execution	The average number of logical reads caused by the statement each time it is executed.
Buffer gets	The number of logical reads for the statement.
Disk reads	The number of disk reads for the statement.
Parse calls	The number of parse calls for the statement.
Sorts	The number of sorts caused by the statement.
Executions	The number of times that statement has been executed.
Rows processed	The number of rows processed by the statement.
Hit Ratio	The cache hit ratio for the statement.
First load time	The date/time of when the statement was first loaded into the shared pool.
Sharable memory	The amount of sharable memory used by the statement.
Persistent memory	The amount of persistent memory used by the statement.
Runtime memory	The amount of runtime memory used by the statement.
Elapsed Time per Exec	For Oracle9i and later, this is the average elapsed response time for the statement.
CPU Time per Exe	For Oracle9i and later, this is the average CPU time for the statement.
CPU Time	For Oracle9i and later, this is the total CPU time for the statement.
Elapsed Time	For Oracle9i and later, this is the total elapsed time for the statement over all executions.
Hash value	A unique identifier for the SQL statement assigned by Oracle.
Address	The address of the SQL statement.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Top SQL View (DBArtisan - Oracle Performance Analyst)** > SQL Details

Selected SQL Text

The Selected SQL Text window allows you to select any of the Top SQL result rows and get the complete SQL text. From here you can open an Explain Plan or Tune the SQL.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Top SQL View (DBArtisan - Oracle Performance Analyst)** > SQL Details

Recent History (Oracle 10g only)

The Recent History tab of the Top SQL view displays:

SQL Activity (last 24 hours) SQL Details

SQL Activity (last 24 hours)

The SQL Activity graph shows you the resource usage of SQL statements over the last 24-hour period. You can see cumulative CPU, elapsed time, and wait time for all SQL statements that were executed during the last 24 hours.

You can also select a specific time period in the graph by drawing a box around part of the graph with your mouse. The graph will change to match your selection.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Top SQL View (DBArtisan - Oracle Performance Analyst)** > Recent History (Oracle 10g only)

SQL Details

The SQL Details section of the Recent History view displays the SQL statements that correspond to the time period shown in the SQL Activity graph. The following columns are displayed:

SQL ID: The unique identifier for the SQL statement

CPU Time: The total number of CPU seconds used by the statement during the specified time period

CPU %: The percentage of CPU used by the statement during the specified time period

Elapsed Time: The total elapsed time for the statement (in seconds) during the specified time period

Elapsed %: The percentage of elapsed time for the statement during the specified time period

Wait Time: The total wait time for the statement (in seconds) during the specified time period

Wait %: The percentage of wait time for the statement during the specified time period

SQL Time: The total SQL execution time for the statement (in seconds) during the specified time period

PL/SQL Time: The total PL/SQL execution for the statement (in seconds) during the specified time period

Java Time: The total java execution time for the statement (in seconds) during the specified time period

Executions: The total number of executions for the statement (in seconds) during the specified time period

Disk Reads: The total number of disk reads for the statement (in seconds) during the specified time period

Buffer Gets: The total number of buffer gets for the statement (in seconds) during the specified time period

Sorts: The total number of sorts for the statement (in seconds) during the specified time period

Parse Calls: The total number of parse calls for the statement (in seconds) during the specified time period

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Top SQL View (DBArtisan - Oracle Performance Analyst)** > Recent History (Oracle 10g only)
Metrics	Potential SQL tuning targets will be those statements that have unusually large amounts of CPU time, elapsed time, wait time, or number of executions.

Users View (DBArtisan - Oracle Performance Analyst)

In addition to a Home page, the Users category of Oracle Performance Analyst includes the following tabbed pages:

Locks/Blocking Locks Tab Session Waits Tab System Waits Tab

Top Sessions Tab

Home

The Users home page includes the following sections:

Bottleneck Analysis Pane Key User Analysis Pane Bottleneck Analysis Pane

Transaction Analysis Pane Workload Analysis Pane

Key User Analysis Pane

User database activity can be examined using both ratio-based and wait/bottleneck-based analysis. Ratio-based analysis involves examining a number of key database ratios and statistical readings that can be used to indicate how efficiently users are accessing the database. Performance ratios serve as roll-up mechanisms for busy DBAs to use for at-a-glance performance analysis.

When using ratio-based analysis, there are some standards to adhere to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that you examine come from the v$sysstat performance view. The performance view maintains a count of all the occurrences (in the VALUE column) of a particular database incident (in the NAME column) since the database was brought up. For databases that are kept up for long periods of time, these values can grow quite large and impacts how a particular ratio is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each database’s individual ‘personality’ with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead you to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following ratios are used on the Performance Analyst Users home page to succinctly communicate the general session-based performance levels of the monitored database:

Active Sessions CPU/Parse Ratio Inactive Sessions

Open Cursors

The following statistics, available on this pane are duplicates of statistics available on the Key Ratio Analysis Pane of the Home View (DBArtisan - Oracle Performance Analyst) page:

Memory Sort Ratio Parse/Execute Ratio

Active Sessions

The Active Sessions statistic is the total number of active and open threads currently reported in the database as well as the number of processes actively performing work.

Note: For related information, see Active User Processes.

For more information, see Inactive Sessions.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > Home > Key User Analysis Pane

CPU/Parse Ratio

CPU resources are obviously required to accomplish parse activities. The CPU/Parse Ratio determines the percentage of CPU spent on user-initiated parse tasks.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > Home > Key User Analysis Pane
Metrics	Low numbers (0-5%) are desired with the CPU/parse ratio. Larger values can indicate a problem with excessive parse activity.
Correction	If you suspect an excessive parsing problem on your database, you can look into increasing the shared pool size (in hopes that SQL statements are not aged out and therefore reused) or investigate the usage of bind variable in SQL statements. If an undersized shared pool is suspected: *Increase the size of the shared_pool_size parameter through use of the ALTER SYSTEM SET shared_pool_size command value (take caution to not over-allocate; ensure enough free memory exists on server before increasing value).Monitor the new value to see if performance improves.If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.* If you determine that SQL literals are causing SQL to not be reused, you can: *Change the cursor_sharing parameter to FORCE by using the ALTER SYSTEM SET cursor_sharing=FORCE command.Monitor database to see if parse activity is reduced and library cache reloads shrink* If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started. If using an Init.ora file, add the cursor_sharing=FORCE parameter to the file.

Inactive Sessions

The Inactive Sessions statistic is the total number of threads logged on to the database that are currently idle.

Note: For related information, see Inactive User Processes.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > Home > Key User Analysis Pane
Metrics	A large number of inactive users could indicate user sessions that have mistakenly been left logged on. Because each user thread consumes a portion of memory on the Oracle server, to reduce resource usage, you should sever any session that does not need a connection.
Correction	Double-click this statistic to open the Top Sessions Tab of the Users Detail page. On this tab you can check sessions that have many seconds idle and/or that have been logged on for very long periods of time as indicated by the logon time column. After verifying that the session is no longer necessary, you can KILL the session. For more information, see Active Sessions.

Open Cursors

Open Cursors is the total number of all SQL open cursors that exist on the system. In some cases, Oracle cached cursors that have been open by PL/SQL procedures can be kept open for certain lengths of time, even though the actual activity has ceased.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > Home > Key User Analysis PaneOracle Performance Analyst Statistics (DBArtisan) > Session Details View (DBArtisan - Oracle Performance Analyst) > Current Work Tab
Metrics	You should monitor sessions to make sure that they do not approach the Open Cursor limit (specified in the Init.ora file). The parameter, open_cursors, limits how many open cursors (context areas) a session can have open at one time.
Correction	If the total number of open cursors approaches the open_cursors limit then: *Change the open_cursors parameter to FORCE by using the ALTER SYSTEM SET open_cursors=< new value > command.*

Bottleneck Analysis Pane

User database activity can be examined using both ratio-based and wait/bottleneck-based analysis. When connected to an Oracle database, every process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. You can use the Bottleneck Analysis section to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance because it helps you track where a database has been spending its time. If user latch contention or heavy table scan activity has been dragging a database’s performance down, you can use bottleneck analysis to confirm the actual root cause. Once one or more wait events or other bottlenecks have been pinpointed as possible performance vampires, you can oftentimes discover a fair amount of detail about which sessions and objects are causing the problem.

For wait event analysis to be properly carried out, it is imperative that the timed_statistics initialization parameter be set to TRUE. By default this parameter is set to FALSE, which disallows the collection of wait times for each wait event defined to the Oracle engine. For one to really understand the impact of wait events on database performance, you need to not only discover what the database is or has been waiting on, but the durations of the waits. Having both allows a complete picture to be formed regarding the magnitude of wait-initiated performance degradations. Almost all Oracle experts now agree that allowing the collection of timing statistics adds little if anything to database overhead, so setting timed_statistics to TRUE should not be a worry. The Performance Analyst Users home page identifies the current top session waits as well as the top session-related bottlenecks that have seen the most waits on the system.

When using bottleneck analysis, you cannot rely only on the information contained in the wait event views that Oracle provides. Other user-related bottlenecks include current disk sort activity. Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system.

The following bottleneck indicators are used on the Performance Analyst Users home page to succinctly communicate the general overall performance level of the monitored database:

Sessions Blocked	Sessions in Disk Sorts	Sessions with Enqueue Waits
Sessions Waiting	Top Bottleneck Events	Top Session Bottlenecks

Sessions Blocked

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Although Oracle supports unlimited row-level locking, blocking lock situations do crop up. User processes holding exclusive locks and not releasing them via a proper COMMIT generally cause most blocks.

Note: This statistic is also called Current Object Blocks on the Home View (DBArtisan - Oracle Performance Analyst) page and Session Blocks on the Objects View (DBArtisan - Oracle Performance Analyst)Bottleneck Analysis Pane.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	You should immediately investigate any indicator above zero for a blocking lock statistic before the situation has a chance to grow out of control.
Correction	Once discovered, a blocking lock situation can normally be quickly remedied. You can issue a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then nearly almost always complete in an instant. Embarcadero Performance Analyst makes it easier to discover the blocked lock situation, but the trick is to prevent the blocking lock situation in the first place. You can look at the **Users View (DBArtisan - Oracle Performance Analyst)** detail and view all current blocking locks to see exactly which sessions are holding the currently restrictive locks. The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. Most DBAs who have had to face Oracle Forms applications have suffered through the dreaded SELECT … FOR UPDATE statements that place unnecessary restrictive locks on nearly every read operation, and know all too well that good coding practice is important. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

Sessions in Disk Sorts

Oracle has the option of running automatic PGA memory management. Oracle has introduced a new Oracle parameter called pga_aggregate_target. When this parameter is set and you are using dedicated Oracle connections, Oracle ignores all of the PGA parameters in the Oracle file, including sort_area_size, hash_area_size and sort_area_retained_size. Oracle recommends that the value of pga_aggregate_target be set to the amount of remaining memory (less a 10% overhead for other server tasks) on a server after the instance has been started.

The sessions in disk sorts statistic displays a count of all sessions currently performing a disk sort.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Consistently seeing non-zero numbers for this statistic (as well as low values observed for the memory sort ratio) can indicate excessive disk sort activity. Investigate the use of pga_aggregate_target. Once the pga_aggregate_target has been set, Oracle automatically manages PGA memory allocation, based upon the individual needs of each Oracle connection. Oracle9i or later allows the pga_aggregate_target parameter to be modified at the instance level with the alter system command, thereby lets you dynamically adjust the total RAM region available to Oracle9i. Oracle also has a parameter called workarea_size_policy. When this parameter is set to automatic, all Oracle connections benefit from the shared PGA memory. When workarea_size_policy is set to manual, connections allocate memory according to the values for the sort_area_size parameter. Under the automatic mode, Oracle tries to maximize the number of work areas that are using optimal memory and uses one-pass memory for the others.
Correction	If you find a problem, do the following: *Edit the Init.ora or SPFILE file for the database.Increase the amount of sort_area_size to a higher value (take care not to not over-allocate; ensure enough free memory exists on server before increasing value). Realize that EVERY user receives this amount for sorting).Cycle the Oracle server when possible to allow the new value to take effect.Monitor new value to see if performance improves.* In addition to increasing the amount of memory devoted to sorting, you should also locate inefficient SQL that causes needless sorts. For example, in an SQL query (to eliminate duplicate rows) UNION ALL does not cause a sort whereas UNION does. People frequently code DISTINCT inappropriately (especially people transferring from Microsoft Access, which uses DISTINCT for most SELECT queries). There are times you simply cannot stop sort activity. When this happens, you should try to keep it in memory whenever possible. However, large data warehousing systems frequently exhaust RAM sort allotments, so if disk sorts must occur, ensure three things: *Your user's TEMPORARY TABLESPACE assignment is not the SYSTEM tablespace, which is the default assignment. In Oracle9i or later, you can specify a default tablespace other than SYSTEM for every user account that is created.The TEMPORARY TABLESPACE assigned to your users is placed on a fast disk.* The TEMPORARY TABLESPACE has the tablespace parameter TEMPORARY assigned to it, which allows sort activity to be performed in a more efficient manner.

Sessions with Enqueue Waits

An enqueue is an advanced locking device that allows multiple database processes to share certain resources. Enqueue waits typically occur when sessions wait to be granted a requested lock. Sometimes these locks are internal Oracle locks while other times they could be locks for rows of data. Note that enqueues are issued implicitly by Oracle.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Any enqueue waits that read consistently above 1 or more (delta statistics) should be investigated.
Correction	Removing contention for enqueues is almost always an application design issue. If many enqueue waits are seen, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. You should examine the use of indexes to make sure all referencing foreign keys are indexes and that SQL is tuned to not tarry over rows in the database during modification operations. Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).

Sessions Waiting

User connections that are waiting on a system generally occur for two reasons:

1 A process waits because a requested resource is not available.

2 A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not usually a concern. However the two wait causes mentioned above are the ones worth your time and investigation. The sessions waiting statistic is a count of all sessions that are currently waiting for one reason or another.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	To determine the actual wait causes currently experienced by user connections, you should drill down from the global count of users waiting, into the actual system and user wait details of a database.
Correction	If you find a problem, drill down into wait details to determine whether the waits are resource-related.

Top Bottleneck Events

When viewing wait statistics, there are many levels of detail. The first level is the system view, which provides a global, cumulative snapshot of all the waits that have occurred on a system. The second level is the session view, which shows details on which events connected sessions have experienced. Viewing these numbers can help you determine which session-related wait events have caused the most commotion in a database thus far. The Top Bottleneck Events section identifies the top waits that have occurred on the Oracle database based on the amount of time waited per event.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane

Correction

Appendix A in the Oracle Reference manual contains a listing and description of every current wait event defined in Oracle. DBAs unfamiliar with what each event represents should keep this listing close by as they examine wait-based event metrics. For example, a 'db file scattered read' event is typically indicative of table scan operations. If many of these events are observed, you should begin to see if large table scans are occurring in the database. Like the 'db file scattered read' event, each wait event has its own meaning and individual end-resolution diagSQL Analysis - Users

Much of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times. SQL statements that use much memory can also cause a problem in a database.

Before you can identify problem SQL in your database, you have to ask the question, "What is bad SQL?" What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every Oracle professional's satisfaction. The SQL Analysis for Users shows what SQL statements have consumed the largest percentages of executions, parse calls, CPU time, and elapsed time. The table below lists some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

Criteria	Description
Overall Response (Elapsed) Time	The amount of time the query took to parse, execute, and fetch the data needed to satisfy the query. It should not include the network time needed to make the round trip from the requesting client workstation to the database server. This statistic is available in Oracle9i or later.
CPU Time	The amount of CPU time the query took to parse, execute, and fetch the data needed to satisfy the query.
Physical I/O	This is often used as the major statistic in terms of identifying good vs. bad SQL, this is a measure of how many disk reads the query caused to satisfy the user's request. While you certainly want to control disk I/O where possible, it is important that you not focus solely on physical I/O as the single benchmark of inefficient SQL. Make no mistake, disk access is slower than memory access and also consumes processing time making the physical to logical transition, but you need to look at the entire I/O picture of a SQL statement, which includes looking at a statements' logical I/O as well.
Logical I/O	The number of memory reads the query took to satisfy the user's request. The goal of tuning I/O for a query should be to examine both logical and physical I/O, and use appropriate mechanisms to keep both to a minimum.
Repetition	The number of times the query has been executed. A problem in this area is not as easy to spot as the others unless you know your application well. A query that takes a fraction of a second to execute may still be causing a headache on your system if it has executed erroneously (for example, a query that executes in a runaway PL/SQL loop) over and over again.

There are other criteria that you can examine like sort activity or access plan statistics (that show items like Cartesian joins and the like), but more often than not, these measures are reflected in the criteria listed above.

Fortunately, Oracle records all the above measures (some only in 9i), which makes tracking the SQL that has been submitted against an Oracle database much easier.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	When you begin to look for inefficient SQL in a database, there are two primary questions you want answered: *What HAS been the worst SQL that has historically run in my database?What IS the worst SQL that is running right now in my database?*
Correction	When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It may be that the query is in an inefficient PL/SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint. There is the possibility that the SQL statement just is not tuned well. To determine that, you can go further into Performance Analyst's Top SQL view and, if you have Embarcadero SQL Tuner installed, you can port the SQL over to SQL Tuner to better optimize the statement.

Transaction Analysis Pane

On a transaction-oriented system, user transactions perform COMMITs to make permanent any data additions, changes, or deletions to the database. Or they ROLLBACK any modifications to undo any work that has encountered an error or other transaction-halting situation. The transaction analysis breakdown shows current commits and rollbacks and the percentage that each activity accounts for on the database.

Metrics	Large volumes of rollbacks can indicate a transaction breakdown on the system, that may need investigation at the application level.

Workload Analysis Pane

When the database population as a whole experiences a system slowdown, it is not uncommon to find one or two users who are responsible for bringing the system to its knees. In the best of worlds, users have a pretty evenly divided amount of memory usage, disk I/O, CPU utilization, and parse activity. However, users submit large batch jobs during peak OLTP activity, or when sessions that are fire off untuned queries on a critical system.

If you are seeing a slowdown in your database, and cannot seem to find a root cause, examine the resource consumption of the leading sessions on a system. A handful of users can overwhelm the I/O capabilities of Oracle (through untuned queries or runaway batch jobs) or hammer the CPU or memory structures of the database and server.

Performance Analyst makes it easy to pinpoint the top sessions by showing the leading processes at the physical I/O, logical I/O, memory, disk sort, table scan, and CPU usage levels.

Metrics If any one session uses more than 50% of a total resource (CPU, memory, etc.) go into the session to find out what they are currently executing.

Top Sessions Tab

The Top Sessions tab includes the following sections:

Top CPU Process Top I/O Process Top Memory Process

Top Sessions

Top I/O Process

This section identifies the Oracle process that currently has caused the most I/O usage on the database.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > Top Sessions Tab

Metrics To obtain more details on the top I/O process, locate the SID in the Top Sessions grid and drill down to obtain more granular information.

Top Memory Process

The Top Memory Process section identifies the Oracle process that currently is using the highest percentage of memory in the database.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > Top Sessions Tab

Metrics To obtain more details on the top memory process, locate the SID in the Top Sessions grid and drill down to obtain more granular information.

Top CPU Process

The Top CPU Process section identifies the Oracle process that currently has the highest percentage of CPU usage in the database.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > Top Sessions Tab

Metrics To obtain more details on the top CPU process, locate the SID in the Top Sessions grid and drill down to obtain more granular information.

Top Sessions

When a system experiences heavy activity, sometimes you will find that all the user connections are contributing somewhat equally to the overall load. More often than not, however, one or two user connections are causing most of the activity. It may be that a large batch load or other typical process is running that is perfectly okay for your system. Or it may be a runaway process or other rogue connection that needs to be tracked down and possibly eliminated.

The Top Sessions section displays information regarding all key metrics for all current sessions on a database. The table below describes the information available in this section:

Column	Description
SID	The system ID of the process.
User Name	The username of the process.
O/S ID	The operating system ID of the process.
Machine Name	The workstation where the process originated.
Logon Time	The date/time the process logged on to Oracle.
Tot Physical I/O	A summation of all physical reads and physical writes for the process (includes direct).
Tot Logical I/O	A summation of all logical I/O (buffer gets, consistent gets, etc.) for the process.
Hit Ratio	Shows how often object definitions are found in memory vs. having to read them in from disk.
Total Memory Usage	The total amount of memory used.
Parses	The total number of parses the process has caused.
Total CPU	The total CPU usage for the process.
Parse CPU	The amount of process CPU caused by parse activity.
Recursive CPU	The amount of process CPU caused by recursive calls.
Other CPU	Total CPU usage minus parse and recursive activity. This can be a negative number if the Oracle kernel is not properly reporting total CPU usage.
Disk Sorts	The total number of disk sorts caused by the process.
Memory Sorts	The total number of memory sorts caused by the process.
Rows Sorted	The total number of rows sorted by the process.
Commits	The total number of commits executed by the process.
Rollbacks	The total number of rollbacks executed by the process.
Executions	The total number of executions.
Physical Reads	The total number of physical reads produced by the process.
DB Block Gets	The total number of block gets produced by the process.
Consistent Gets	The total number of consistent gets produced by the process.
Consistent Changes	The total number of consistent changes produced by the process.
Last Wait	Event Defines the last wait event experienced by the session

Wait Class	The wait class of the last wait event
Wait time	The sessions last wait time
P1text	Text that describes the first wait parameter
P2text	Text that describes the second wait parameter
P2 r	The second wait parameter
P2Raw	The data for the second parameter
P3text	Text that describes the third wait parameter
P3	The third wait parameter
P3Raw	The data for the third parameter

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > Top Sessions Tab
Metrics	To obtain more details on any process, drill down to obtain more granular information.

System Waits Tab

The System Waits tab includes the following sections:

System Waits Wait Percent by Time Waited Wait Percent by Total Waits

System Waits

Waits on a system generally occur for three reasons:

1 A process waits because it has no work to do.

2 A process waits because a requested resource is not available.

3 A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention. The system Waits section presents all the various system waits that have occurred on the system since startup. The table below lists the information available in this section:

Column	Description
Event	The name of the wait event.
Class	Wait classes contain a collection of related wait events (like User I/O, System I/O, etc.)
Total Waits	The total number of waits for the event.
Percent of Total	The percentage of all waits that this event makes up.
Time Waited (Secs)	The total amount of time waited for the event, in seconds.
Percent Time Waited	The percentage of time that this event makes up.
Total Timeouts	The total number of timeouts for the event.
Percent Total Timeouts	The percentage of timeouts that this event makes up.
Average Waits (Secs)	The average amount of time waited for the event, in seconds.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > System Waits Tab
Metrics	Which waits should be a cause for concern and which waits can be ignored (for the most part)? Common wait events that can generally be ignored include **SQL*Net more data from client**, **SQL*Net message from client**, **client message**, **rdbms ipc message**, **pipe get**, **pmon timer**, **smon time**r, and **Null event**.

Correction

The table below describes wait events that deserve attention:

Wait Event	Description
Enqueue waits	If many enqueue waits are seen, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. You should examine the use of indexes to make sure all referencing foreign keys are indexes and that SQL is tuned to not tarry over rows in the database during modification operations. Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).
Buffer busy waits	Buffer busy waits normally center around contention for rollback segments, too small an INITRANS setting for tables, or insufficient free lists for tables. The remedy for each situation would be increasing the number of rollback segments, or altering tables to have larger settings for INITRANS to allow for more transactions per data block, and more free lists.
db file scattered read waits	An interesting wait event is the db file scattered read event. This event is indicative of table scans occurring on the system. Large numbers of them may indicate heavy scan activity and the need to revisit your indexing/physical design.
Latch free waits	Latch free waits indicate contention for latches.

The SQL*Net message to client can help a DBA diagnose wait activity outside the database.

The key statistics for waits is not the number of waits for each event, but rather the wait time and average wait time. These indicate if the event is truly causing a significant bottleneck in the database. To view the wait times for sessions and overall system wait events, you must set the timed_statistics parameter to TRUE for your Oracle databases. You can either set this parameter in your Init.ora file or alter the system dynamically with the ALTER SYSTEM SET TIMED_STATISTICS=TRUE command.

Wait Percent by Total Waits

Waits on a system generally occur for three reasons:

- A process waits because it has no work to do.
- A process waits because a requested resource is not available.
- A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention. The Wait Percent by Total Waits section displays the percentage of waits for all top non-idle wait events.

Location [Oracle Performance Analyst Statistics (DBArtisan)](#) > **Users View (DBArtisan - Oracle Performance Analyst)** > [System Waits Tab](#)

Metrics Which waits should be a cause for concern and which waits can be ignored (for the most part)? Common wait events that can generally be ignored include **SQL*Net more data from client**, **SQL*Net message from client**, **client message**, **rdbms ipc message**, **pipe get**, **pmon timer**, **smon time**r, and **Null event**.

Correction

The table below describes wait events that deserve attention:

Wait Event	Description
Enqueue waits	If many enqueue waits are seen, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. You should examine the use of indexes to make sure all referencing foreign keys are indexes and that SQL is tuned to not tarry over rows in the database during modification operations. Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).
Buffer busy waits	Buffer busy waits normally center around contention for rollback segments, too small an INITRANS setting for tables, or insufficient free lists for tables. The remedy for each situation would be increasing the number of rollback segments, or altering tables to have larger settings for INITRANS to allow for more transactions per data block, and more free lists.
db file scattered read waits	An interesting wait event is the db file scattered read event. This event is indicative of table scans occurring on the system. Large numbers of them may indicate heavy scan activity and the need to revisit your indexing/physical design.
Latch free waits	Latch free waits indicate contention for latches.

The SQL*Net message to client can help a DBA diagnose wait activity outside the database.

Wait Percent by Time Waited

Waits on a system generally occur for three reasons:

1 A process waits because it has no work to do.

2 A process waits because a requested resource is not available.

3 A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention. The Wait Percent by Time Waited section displays the percentage of waits for all top non-idle wait events.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > System Waits Tab
Metrics	The key statistics for waits is not the number of waits for each event, but rather the wait time and average wait time. These indicate if the event is truly causing a significant bottleneck in the database. Which waits should be a cause for concern and which waits can be ignored (for the most part)? Common wait events that can generally be ignored include **SQL*Net more data from client**, **SQL*Net message from client**, **client message**, **rdbms ipc message**, **pipe get**, **pmon timer**, **smon time**r, and **Null event**.

Correction

The table below describes wait events that deserve attention:

Wait Event	Description
Enqueue waits	If many enqueue waits are seen, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. You should examine the use of indexes to make sure all referencing foreign keys are indexes and that SQL is tuned to not tarry over rows in the database during modification operations. Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).
Buffer busy waits	Buffer busy waits normally center around contention for rollback segments, too small an INITRANS setting for tables, or insufficient free lists for tables. The remedy for each situation would be increasing the number of rollback segments, or altering tables to have larger settings for INITRANS to allow for more transactions per data block, and more free lists.
db file scattered read waits	An interesting wait event is the db file scattered read event. This event is indicative of table scans occurring on the system. Large numbers of them may indicate heavy scan activity and the need to revisit your indexing/physical design.
Latch free waits	Latch free waits indicate contention for latches.

The SQL*Net message to client can help a DBA diagnose wait activity outside the database.

To view the wait times for sessions and overall system wait events, you must set the timed_statistics parameter to TRUE for your Oracle databases. You can either set this parameter in your Init.ora file or alter the system dynamically with the ALTER SYSTEM SET TIMED_STATISTICS=TRUE command.

Session Waits Tab

The Session Waits tab includes the following sections:

Current Waits Historical Waits

Historical Waits

Waits on a system generally occur for three reasons:

1 A process waits because it has no work to do.

2 A process waits because a requested resource is not available.

3 A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention. The Historical Waits section presents all the various waits that have occurred for all currently connected oracle sessions. The table below lists the information available in this section:

Column	Description
Event	The name of the wait event.
Wait Class	Wait classes contain a collection of related wait events (like User I/O, System I/O, etc.)
Total Waits	The number of waits for the event.
Percent of Total Waits	The percentage of all waits that this event makes up.
Time Waited (Secs)	The total amount of time waited for the event, in seconds.
Average Wait (Secs)	The average amount of time waited for the event, in seconds.
Max Wait (Secs)	The maximum amount of time waited for the event, in seconds.
Total Timeouts	The number of timeouts for the event.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > Session Waits TabOracle Performance Analyst Statistics (DBArtisan) > Session Details View (DBArtisan - Oracle Performance Analyst) > Waits Tab
Metrics	Which waits should be a cause for concern and which waits can be ignored (for the most part)? Common wait events that can generally be ignored include **SQL*Net more data from client**, **SQL*Net message from client**, **client message**, **rdbms ipc message**, **pipe get**, **pmon timer**, **smon time**r, and **Null event**.

Correction

The table below describes wait events that deserve attention:

Wait Event	Description
Enqueue waits	If many enqueue waits are seen, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. You should examine the use of indexes to make sure all referencing foreign keys are indexes and that SQL is tuned to not tarry over rows in the database during modification operations. Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).
Buffer busy waits	Buffer busy waits normally center around contention for rollback segments, too small an INITRANS setting for tables, or insufficient free lists for tables. The remedy for each situation would be increasing the number of rollback segments, or altering tables to have larger settings for INITRANS to allow for more transactions per data block, and more free lists.
db file scattered read waits	An interesting wait event is the db file scattered read event. This event is indicative of table scans occurring on the system. Large numbers of them may indicate heavy scan activity and the need to revisit your indexing/physical design.
Latch free waits	Latch free waits indicate contention for latches.

The SQL*Net message to client can help a DBA diagnose wait activity outside the database.

The key statistics for waits is not the number of waits for each event, but rather the wait time and average wait time. These indicate if the event is truly causing a significant bottleneck in the database. To view the wait times for sessions and overall system wait events, you must set the timed_statistics parameter to TRUE for your Oracle databases. You can either set this parameter in your Init.ora file or alter the system dynamically with the ALTER SYSTEM SET TIMED_STATISTICS=TRUE command.

Current Waits

Waits on a system generally occur for three reasons:

- A process waits because it has no work to do.
- A process waits because a requested resource is not available.
- A process waits for Oracle to perform a prerequisite task for its given operation.

Session contention is merely a subset of contention that is viewable at the global database level. Oftentimes, it takes analysis at the session level to pinpoint the exact source of contention that is occurring globally. So you need to become accustomed to viewing contention statistics and waits at the user process level.

When monitoring waits with respect to user sessions, there are two areas of interest:

1 What HAS the user session been waiting on?

2 What IS the user session waiting on?

Oracle records both sets of wait statistics for you. In reviewing previous waits for a session, you can see which types of things have caused the session to be bottlenecked. The Current Waits section displays information to help you troubleshoot session contention. The table below lists the information available in this section:

Column	Description
Event	The name of the wait event.
Wait (Secs)	The number of seconds in the current wait.
Last Wait Time	A non-zero value is the session's last wait time. A zero value means the session is currently waiting. A -2 value indicates the platform cannot support a fast timing mechanism or TIMED_STATISTICS is not set for the system.
State	Indicates the possible values. They includes the following: WAITING - the session is waiting. WAITED UNKNOWN TIME - duration of last wait is not known. WAITED SHORT TIME - last wait was less than 1/100th of a second. WAITED KNOWN TIME - the wait is equal to the time of the last wait.
P1 - P3	Any additional wait parameters.
P1TEXT - P3TEXT	Any descriptions of additional parameter.
P1RAW - P3RAW	Any additional wait parameters.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > Session Waits TabOracle Performance Analyst Statistics (DBArtisan) > Session Details View (DBArtisan - Oracle Performance Analyst) > Waits Tab
Metrics	Which waits should be a cause for concern and which waits can be ignored (for the most part)? Common wait events that can generally be ignored include **SQL*Net more data from client**, **SQL*Net message from client**, **client message**, **rdbms ipc message**, **pipe get**, **pmon timer**, **smon time**r, and **Null event**.

Correction

The table below describes wait events that deserve attention:

Wait Event	Description
Enqueue waits	If many enqueue waits are seen, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. You should examine the use of indexes to make sure all referencing foreign keys are indexes and that SQL is tuned to not tarry over rows in the database during modification operations. Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).
Buffer busy waits	Buffer busy waits normally center around contention for rollback segments, too small an INITRANS setting for tables, or insufficient free lists for tables. The remedy for each situation would be increasing the number of rollback segments, or altering tables to have larger settings for INITRANS to allow for more transactions per data block, and more free lists.
db file scattered read waits	An interesting wait event is the db file scattered read event. This event is indicative of table scans occurring on the system. Large numbers of them may indicate heavy scan activity and the need to revisit your indexing/physical design.
Latch free waits	Latch free waits indicate contention for latches.

The SQL*Net message to client can help a DBA diagnose wait activity outside the database.

The key statistics for waits is not the number of waits for each event, but rather the wait time and average wait time. These indicate if the event is truly causing a significant bottleneck in the database. To view the wait times for sessions and overall system wait events, you must set the timed_statistics parameter to TRUE for your Oracle databases. You can either set this parameter in your Init.ora file or alter the system dynamically with the ALTER SYSTEM SET TIMED_STATISTICS=TRUE command.

Locks/Blocking Locks Tab

The Locks/Blocking Locks tab includes the following sections:

All Locks Blocking Locks

All Locks

To modify database information or structures, a user session must obtain a lock on the object to perform its task. Ibesides user locks, Oracle itself will issue lock requests to carry out its internal duties. The All Locks section displays information about all locks currently on a system. The table below lists the information available in this section:

Column	Description
Machine	The workstation of the Oracle process.
Serial #	The serial number of the session holding the lock.
Request	The lock requests that Oracle issues to carry out its internal duties.
Table	The table name of the locked table.
Mode	The lock mode (EXCLUSIVE, SHARE, etc.).
Blocking?	Indicates if the lock is a blocking lock.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > Locks/Blocking Locks TabOracle Performance Analyst Statistics (DBArtisan) > Session Details View (DBArtisan - Oracle Performance Analyst) > Locks Tab

Metrics Locks that are held for unusually long periods of time may be candidates for further investigation. The application logic may be inefficient or perhaps the program is not issuing frequent enough COMMITs.

Blocking Locks

Without a doubt, blocking lock situations can give the appearance of a frozen database almost more than anything else (except, perhaps, for a stuck archiver process). A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Although Oracle supports unlimited row-level locking, blocking lock situations do crop up - sometimes frequently.

The Blocking Locks section contains information relating to user accounts that are currently blocked and the sessions that are blocking them. The table below describes the information available in this section:

Column	Description
Blocked User	The user account of the session waiting for the lock.
Blocking User	The user account of the session holding the offending lock.
Waiting SID	The session identifier of the session waiting for the lock.
Holding SID	The session identifier of the session holding the offending lock.
Type	The type of lock (TRANSACTION, DML, etc.).
Mode	The lock mode (EXCLUSIVE, SHARE, etc.).
Request	The type of lock being requested by the session.
Row	The rowid of the row being held.
ID 1	Lock identifier #1 (depends on type).
ID 2	Lock identifier #2 (depends on type).
Time Waited (Secs)	The amount of time, in seconds, the blocked user has been waiting.
Locked Object	The name of the object being locked.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - Oracle Performance Analyst)** > Locks/Blocking Locks TabOracle Performance Analyst Statistics (DBArtisan) > Session Details View (DBArtisan - Oracle Performance Analyst) > Locks Tab

Metrics Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is where it gets tricky. The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. Most DBAs who have had to face Oracle Forms applications have suffered through the dreaded SELECT … FOR UPDATE statements that place unnecessary restrictive locks on nearly every read operation, and know all too well that good coding practice is important. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do. Data warehouses (whose data is mostly read) can benefit from tablespaces set in READ ONLY mode. READ ONLY signals to the other database that exclusive locks need not be used for the data contained within the tablespace. This is especially helpful in Oracle Parallel Server environments and drastically reduces ping activity.

RAC Detail View (DBArtisan - Oracle Performance Analyst)

The Performance Analyst for Oracle has been upgraded to support for the monitoring of Oracle Real Application Clustering (RAC) environments (Oracle 9i and higher).

Oracle RAC allows a single physical Oracle database to be accessed by simultaneous instances of Oracle running across several CPUs. All statistics you see are listed by instance ID.

The RAC Detail view is divided into the following tabs:

Overview Tab	Memory	Ping Activity
Locks	Sessions	

Overview

The Overview tab is of the RAC Detail view divided into the following sections:

Key Ratio Analysis Pane	Instance Analysis Pane	Bottleneck Analysis Pane
Workload Analysis Pane		

Key Ratio Analysis Pane

Database performance analysts typically use one of two methods for examining the performance levels of a database: ratio-based or wait/bottleneck-based analysis. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as very good roll-up mechanisms for busy DBAs to use for at-a-glance performance analysis. Many DBAs have large database farms to contend with and can't spend time checking detailed wait-based analysis outputs for each and every database they oversee. Succinctly presented performance ratios can assist in such situations by giving DBAs a few solid indicators that can be quickly scanned to see if any database needs immediate attention.

While there are many opinions as to what rules to follow, there are some standards that should always be adhered to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that a DBA will examine come from the `v$sysstat` performance view. This view maintains a count of all the occurrences (in the VALUE column) of a particular database incident (in the NAME column) since the database was brought up. For databases that are kept up for long periods of time, these values can grow quite large and impact how a particular ratio that a DBA may be looking at is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each database's individual "personality"with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead the DBA to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following ratios are used on the Performance Analyst RAC Overview page to succinctly communicate the general overall performance levels of the monitored database:

- Cross Instance Cache Flush ratio
- Ping Ratio
- Local Buffer Access Percent
- Local Read Percent
- Global Buffer Busy Wait Percent

Instance Analysis Pane

As a DBA managing an Oracle RAC, you will oftentimes want a quick performance snapshot of the efficiency and activity across all your Oracle RAC nodes. The Instance Analysis section provides a fast picture of which nodes of your Oracle RAC appear overworked and those that are underutilized.

The following statistics are provided:

ID: The Oracle RAC instance ID

·Name: The instance name

Sessions: The total number of sessions logged onto a RAC node

Physical I/O: A summation of all physical reads and physical writes for the RAC node

Logical I/O: A summation of all logical I/O (buffer gets, consistent gets, etc.) for the RAC node

Hit Ratio: An indicator of how often user requests for data are satisfied.

Memory Usage: The session's total memory usage against all session memory usage on the database.

Parses: The total number of parses experienced by the RAC node

Total CPU: The total CPU usage for the RAC node

Parse CPU: The amount of process CPU for the RAC node

Recursive CPU: The amount of process CPU caused by recursive calls on the RAC node

Other CPU: Total CPU usage minus parse and recursive activity. This can be a negative number if the Oracle kernel is not properly reporting total CPU usage

Disk Sorts: The total number of disk sorts for the RAC node

Memory Sorts: The total number of memory sorts for the RAC node

Rows Sorts: The total number of rows sorted on the RAC node

Commits: The total number of commits executed for the RAC node

Rollbacks: The total number of rollbacks executed for the RAC node

Executions: The total number of calls made on the RAC node

Physical Reads: The total number of physical reads produced on the RAC node

DB Block Gets: The total number of block gets produced on the RAC node

Consistent Gets: The total number of consistent gets produced on the RAC node

Consistent Changes: The total number of consistent changes produced on the RAC node

Bottleneck Analysis Pane

When an Oracle database is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting can mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. And this is where wait-based or bottleneck analysis comes into play. DBAs used this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance because it helps a DBA track where a database has been spending its time. If latch contention or heavy table scan activity has been dragging a database’s performance down, a DBA can use bottleneck analysis to confirm the actual root cause. Once one or more wait events or other bottlenecks have been pinpointed as possible performance vampires, the DBA can drill down and oftentimes discover a fair amount of detail about what sessions and objects are causing the problem.

For wait event analysis to be properly carried out, it is imperative that the timed_statistics initialization parameter be set to TRUE. By default this parameter is set to FALSE, which disallows the collection of wait times for each wait event defined to the Oracle engine. For one to really understand the impact of wait events on database performance, the DBA not only needs to discover what the database is or has been waiting on, but the durations of the waits. Having both measures in hand allows a complete picture to be formed regarding the magnitude of wait-initiated performance degradations. Almost all Oracle experts now agree that allowing the collection of timing statistics adds little if anything to database overhead, so setting timed_statistics to TRUE should not be a worry. The Performance Analyst home page identifies the top system and session waits that are currently a cause of lessened performance.

When using bottleneck analysis, you cannot rely only on information contained in the wait event views that Oracle provides. For example, an object may attempt to extend into another extent of space in a tablespace and yet be denied if no such free space exists. Such a failure will not be reflected in any wait event, but still represents a very real bottleneck to the database. In the same way that you cannot depend on only a few ratios to properly carry out ratio-based performance analysis, an administrator must include other statistical metrics in their overall bottleneck analysis framework to obtain an accurate performance risk assessment. Performance Analyst helps identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system.

For an Oracle RAC, Performance Analyst highlights the top RAC-related waits as well as conflicts that may be slowing down response times. Here you see statistics for Top Rac-Related Waits and Conflicts. Within each section, you find:

Top Rac-Related Waits

When viewing wait statistics, there are several levels of detail that a DBA can drill down into. The first level is the system view, which provides a global, cumulative snapshot of all the waits that have occurred on a system. Viewing these numbers can help a DBA determine what wait events have caused the most commotion in a database thus far. The Top RAC-related bottlenecks section display identifies the top waits that have occurred on the node that Performance Analyst is currently connected to.

Appendix A in the Oracle Reference manual contains a listing and description of every current wait event defined in Oracle. DBAs unfamiliar with what each event represents should keep this listing close by as they examine wait-based event metrics. For example, a “db file scattered read” event is typically indicative of table scan operations. If many of these events are observed, then the DBA should begin to see if large table scans are occurring in the database. Like the “db file scattered read” event, each wait event has its own meaning and individual end-resolution diagnosis.

After looking at system-level wait activity, a DBA can drill down further to discover what current connections may be responsible for any reported waits that are being observed at the system level. Performance Analyst reports on historical and current wait events at the session level, making this investigation easy to accomplish.

Conflicts

There are a number of conflicts that can occur over an Oracle RAC that can reduce response times. The Conflicts section highlights some of the more common conflicts and provides a count of how many times each has occurred.

The conflicts listed include:

Library Cache: How many library cache invalidations have occurred on the node that Performance Analyst is connected to.

Data Dictionary Cache: How many data dictionary cache invalidations have occurred on the node that Performance Analyst is connected to.

Forced Writes (Oracle 9.2 and above): Number of times a block was written to cache because an instance had used the block, but another instance had requested the lock on the block in a conflicting mode.

Pings: How many cross instance writes occurred

Cache Timeouts: The number of global cache convert timeouts that have occurred.

Workload Analysis Pane

When your phone starts ringing with complaints of performance slowdowns, one of the first things you should get a handle on is:

- Who is currently connected to the database

- What resources are they using
- What are they currently executing

The Workload Activity Analysis section of the Performance Analyst home page provides insight into the leading resource hogs across all instances of an Oracle RAC. Drill down's are available so you can easily get detailed information into what each leading session is currently involved with.

If you are seeing a slowdown in your database, and can't seem to find a root cause, one thing to examine is the resource consumption of the leading sessions on a system. Often you will find one or a handful of users making life miserable for everyone else. They accomplish this by overwhelming the I/O capabilities of Oracle (through untuned queries or runaway batch jobs) or hammering the CPU or memory structures of the database and server.

The Workload Analysis section presents:

- Top Physical I/O Process
- Top Logical I/O Process
- Top Memory Process
- Top CPU Process

For each of these, you see the SID, Instance ID, Username, %Used.

If any one session appears to be using more than 50% of a total resource (CPU, memory, etc.), then you should drill down into the session to find out what it is currently executing.

Memory

The Memory tab of the RAC Detail view is divided into the following sections:

Key Ratio Analysis Pane Memory Analysis Pane Library Cache Analysis Pane

Data Dictionary Cache Analysis Pane

Key Ratio Analysis Pane

Database performance analysts typically use one of two methods for examining the performance levels of a database: Ratio-based or wait/bottleneck-based analysis. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as very good roll-up mechanisms for busy DBAs to use for at-a-glance performance analysis. Many DBAs have large database farms to contend with and can't spend time checking detailed wait-based analysis outputs for each and every database they oversee. Succinctly presented performance ratios can assist in such situations by giving DBAs a few solid indicators that can be quickly scanned to see if any database needs immediate attention.

While there aremany opinions as to what rules to follow, there are some standards that should always be adhered to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that a DBA examines come from the `v$sysstat` performance view. This view maintains a count of all the occurrences (in the VALUE column) of a particular database incident (in the NAME column) since the database was brought up. For databases that are kept up for long periods of time, these values can grow quite large and will impact how a particular ratio that a DBA may be looking at is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each database's individual "personality" with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead the DBA to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following ratios are used in the Memory Key Ratio Analysis section to succinctly communicate the general overall performance levels of the monitored database:

- Buffer Cache Hit Ratio: How often requested information is found in memory versus on disk.
- Library Cache Hit Ratio: Shows how often SQL code is being reused by other database users versus the number of times a SQL statement is broken down, parsed, and then loaded (or reloaded) into the shared pool.
- Latch Miss Ratio: Defines the number of times a process obtained a willing-to-wait latch versus missing the attempt.

Memory Analysis Pane

The Memory Analysis section displays the memory configurations for each node currently up in the RAC. The following information is shown:

Name: The name of the instance.

DB Buffers (MB): The default memory cache that maintains data blocks when they are read from the database. If the DBA doesn't specifically place objects in another data cache (which will be covered next), then any data requested by clients from the database will be placed into this cache. This memory area is controlled by the b_cache_size parameter.

Log Buffer Size (MB): This area buffers modifications that are made to the database before they are physically written to the redo log files. The log_buffer configuration parameter controls this memory area.

Shared Pool Size (MB): This familiar area holds object structure as well as code definitions, and other metadata. Setting the proper amount of memory in the shared pool assists a great deal in improving overall performance with respect to code execution.

DB Caches (MB): Beginning in Oracle9i, a DBA can create tablespaces whose blocksize differs from the overall database blocksize. When data is read into the SGA from these tablespaces, their data has to be placed into memory regions that can accommodate their special block size. Oracle9i and above has memory settings for 2K, 4K, 8K, 16K, and 32K caches. The configuration parameter names are in the pattern of db_nk_cache_size. For each RAC node, you see the following statistics:

- Buffer Cache Hit Ratio

  - Library Cache Hit Ratio
  - Latch Miss Ratio

Library Cache Analysis Pane

The Library Cache Analysis section provides a drill down view into invalidations that have occurred across the cluster. The following information is shown:

ID: The instance ID.

Name: The instance name.

Namespace: The area of the library cache.

DLM Lock Requests: The number of get requests for instance locks.

DLM Pin Requests: The number of pin requests for instance locks.

DLM Pin Releases: The number of release requests for pin instance locks.

DLM Invalidation Requests: The number of get requests for invalidation instance locks.

DLM Invalidations: The number of invalidation pings received from other RAC instances.

Data Dictionary Cache Analysis Pane

The Data Dictionary Cache Analysis section provides a drill-down view into invalidations and conflicts that have occurred across the cluster. The following information is shown:

ID: The instance ID.

Name: The instance name.

Parameter: The specific object area of the dictionary cache.

DLM Requests: The number of lock manager requests.

DLM Conflicts: The number of lock manager conflicts encountered.

DLM Releases: The number of lock manager releases.

Ping Activity

Pinging is the process where one Oracle Instance requests another to write a set of blocks from its SGA to disk so it can obtain it in exclusive mode. To move a data block from one instance's SGA to another is a slow process. The Ping Activity tab of the RAC Detail view displays several levels of specificity, starting from general to specific. The Ping Activity tab is divided into the following sections:

Ping by Instance - shows how often pinging has occurred across all the monitored instances.

Ping by Datafile - shows the number of blocks pinged from the shared set of RAC datafiles.

Ping by Object - provides a drill down view into the amount of specific ping activity across the shared objects and datafiles of a RAC:

- Owner
- Object Name
- Partition Name
- Block Number
- Segment Type
- Kind
- Forced Reads
- Forced Writes
- Datafile

Locks

The Locks tab of the RAC Detail view is divided into the following sections:

Lock Convert Summary - The Lock Convert Summary view provides efficiency statistics regarding the speed at which the RAC acquires and converts locks across the instances. You see the average get and convert times for each RAC node.

DLM Lock Activity - The DLM (Distributed Lock Manager) Lock Activity view provides detailed information on the number of lock conversions that have occurred across the RAC.

PCM Lock Summary - The PCM (Parallel Cache Management) Lock Summary provides a summary count of PCM locks that are releasing, acquiring, or are invalid.

Latch Detail - Protecting the memory structures in Oracle's SGA are latches. Latches ensure that one and only one process at a time can run or modify any memory structure at the same instant. Much more restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism so either you get it or you don't and are forced to continually retry.

The latch detail tab of the contention detail view presents a detailed view of latch activity. Information presented includes:

- Instance Name: The name of the instance
- Name: The name of the latch
- Gets: The total number of times the latch was requested by a process
- Misses: The total number of failed attempts to acquire the latch on the first attempt
- Immediate Gets: The total number of no-wait requests for a latch
- Immediate Misses: The total number of failed no-wait attempts to acquire the latch on the first attempt

- Immediate Sleeps: The total number of requests that "paused" while waiting for a latch

Common indicators of latch contention are a latch miss ratio (which records willing-to-wait mode latch requests) and a latch immediate miss ratio (which records no-wait mode latch requests). These statistics reflect the overall health of how often latch requests were made and satisfied without waiting.

Here are a few latch contention situations that you should recognize and get to the bottom of quickly:

- Cache buffer chain latch: This latch is responsible for protecting paths to database block buffers in the buffer cache. Very high I/O loads tend to cause contention for this latch. You can alleviate contention somewhat by adding more buffers to the cache (through the db_block_buffers parameter) or by adding more LRU latch chain latches with the db_block_lru_latches parameter.
- Library cache latches: Protects cached SQL statements in the library cache area of the Oracle shared pool. Contention for this latch is the usual result of literals being used for SQL statements instead of bind variables.

Sessions

The Sessions tab of the RAC Detail view is divided into the following sections:

User Session by Instance - The User Session by Instance section provides a graphical count of the number of connected sessions for each node in the RAC.

Blocks by Instance - The Blocks by Instance section provides a graphical count of the number of blocked sessions for each node in the RAC.

User Waits by Instance - The User Waits by Instance section provides a graphical count of the number of historical session waits (non-idle) for each node in the RAC.

Top Sessions - When a system experiences heavy activity, sometimes you'll find that all the user connections are contributing somewhat equally to the overall load. Often, however, one or two user connections are causing most of the activity. It may be that a large batch load or other typical process is running that is perfectly OK for your system. Or it may be a runaway process or other rogue connection that needs to be tracked down and possibly eliminated.

The top sessions control displays information regarding all key metrics for all current sessions on a database. Information presented includes:

- SID: The system ID of the process
- Instance: The instance the session is connected to
- Username: The username of the process
- O/S ID: The operating system ID of the process
- Machine Name: The workstation where the process originated
- Logon Time: The date/time the process logged on to Oracle

- Tot Physical I/O: A summation of all physical reads and physical writes for the process (includes direct)
- Tot Physical I/O: A summation of all logical I/O (buffer gets, consistent gets, etc.) for the process
- Parses: The total number of parses the process has caused
- Total CPU: The total CPU usage for the process
- Parse CPU: The amount of process CPU caused by parse activity
- Recursive CPU: The amount of process CPU caused by recursive calls
- Other CPU: Total CPU usage minus parse and recursive activity. This can be a negative number if the Oracle kernel is not properly reporting total CPU usage
- Disk sorts: The total number of disk sorts caused by the process
- Memory sorts: The total number of memory sorts caused by the process
- Rows sorted: The total number of rows sorted by the process
- Commits: The total number of commits executed by the process
- Rollbacks: The total number of rollbacks executed by the process
- Physical reads: The total number of physical reads produced by the process
- DB block gets: The total number of block gets produced by the process
- Consistent gets: The total number of consistent gets produced by the process
- Consistent changes: The total number of consistent changes produced by the process

Response Time Analysis View (DBArtisan - Oracle 10G Only)

The Response Time Analysis View displays statistics that are a part of the Oracle 10g wait enhancements. By analyzing response times, you can review historical data and pinpoint who was on a database and what they were doing during a specific period of time. This allows you to drill deep into the root causes, if any of response time problems, up to 24 hours after the fact.

The main Response Time Analysis View displays the following tabbed pages:

Response Time	Active Time	System Bottlenecks
Session Bottlenecks	Wait Histogram	Historical Session Analysis

Response Time

The Response Time tab of the Response Time Analysis View displays the following statistics:

Current Workload	Current Response Times	Last Hour
Last Hour Detail		

Current Workload

The Current Workload section provides a breakdown of active and idle time for the Oracle database engine in terms of percentage used.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Response Time Analysis View (DBArtisan - Oracle 10G Only)** > Response Time
Metrics	Seeing consistently high percentages for the Database CPU Time (greater than 75-90%) should indicate a possibly overworked server.
Correction	If you see high numbers for the Database CPU Time, then begin looking into these statistics: *SQL Statements with high elapsed times and CPU times (use the Top SQL view to find these).Sessions with high wait times (especially in the User I/O wait class).*

Current Response Times

The Current Response Times section provides a quick view into the current average response time that a transaction experiences on the system along with the current average SQL service time. Both statistics are expressed in seconds.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Response Time Analysis View (DBArtisan - Oracle 10G Only)** > Response Time

Metrics Defining 'good' and 'bad' metrics for these two measurements can only be accomplished by viewing these statistics over time and determining what 'normal' is for your system. You can get help by checking the Last Hour section, which details the minimum, maximum, and average values for these statistics as well as other key metrics that provide insight into how well your database is running.

Last Hour

The Last Hour section provides a window into various key performance metrics. Each metric's minimum, maximum, and average value is displayed.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Response Time Analysis View (DBArtisan - Oracle 10G Only)** > Response Time

Metrics Key metrics to keep an eye on include:**CPU usage per Txn**, **Database CPU Time Ratio**, **Response Time Per Txn**, and **SQL Service Response Time**.

Correction If you see high numbers for the Database CPU Time, then begin looking into these statistics: *SQL Statements with high elapsed times and CPU times (use the Top SQL view to find these).Sessions with high wait times (especially in the User I/O wait class).*

Last Hour Detail

The Last Hour Detail section provides a graphical look back over the last hour of a selected performance metric. Oracle10g and higher automatically maintains a running history of selected metrics so you can observe trends and patterns in performance.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Response Time Analysis View (DBArtisan - Oracle 10G Only)** > Response Time

Active Time

The Active Time tab of the Response Time Analysis detail view displays the following statistics:

System Time Summary System Time Detail Session Time Detail

System Time Summary

The System Time Summary section allows you to see where, in broad terms, the database engine has been spending the largest percentages of its time.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Response Time Analysis View (DBArtisan - Oracle 10G Only)** > Active Time

Metrics

The metrics shown here include (note, this table is largely from Oracle 10g documentation):

Metric	Description
DB Time	You see the amount of elapsed time (in microseconds) spent performing Database user-level calls. Instance background processes such as PMON are not included.
DB CPU	Amount of CPU time (in microseconds) spent on database user-level calls. Instance background processes such as PMON are not included.
background cpu time	Amount of CPU time (in microseconds) taken up by database background processes.
sequence load elapsed time	Amount of elapsed time spent getting the next sequence number from the data dictionary. If a sequence is cached, then this is the amount of time spent replenishing the cache when it runs out. No time is charged when a sequence number is found in the cache. For non-cached sequences, some time is charged for every nextval call.
parse time elapsed	Amount of elapsed time spent parsing SQL statements. Both soft and hard parse times are included.
hard parse elapsed time	Amount of elapsed time spent hard parsing SQL statements.
sql execute elapsed time	Amount of elapsed time SQL statements are executing. NOTE: For select statements this also includes the amount of time spent performing fetches of query results.
connection management call elapsed time	Amount of elapsed time spent performing session connect and disconnect calls.
failed parse elapsed time	Amount of time spent performing SQL parses that ultimately fail with some parse error.
hard parse (sharing criteria) elapsed time	Amount of elapsed time spent performing SQL hard parses when the hard parse resulted from not being able to share an existing cursor in the SQL cache.
hard parse (bind mismatch) elapsed time	Amount of elapsed time spent performing SQL hard parses when the hard parse resulted from bind type or bind size mismatch with an existing cursor in the SQL cache.
PL/SQL execution elapsed time	Amount of elapsed time spent running the PL/SQL interpreter. This does not include time spent recursively executing/parsing SQL statements or time spent recursively executing the Java VM.
PL/SQL compilation elapsed time	Amount of elapsed time spent running the PL/SQL compiler.
inbound PL/SQL rpc elapsed time	Time inbound PL/SQL remote procedure calls have spent executing. It includes all time spent recursively executing SQL and JAVA, and therefore is not easily related to "PL/SQL execution elapsed time".

Java execution elapsed time	Amount of elapsed time spent running the Java VM. This does not include time spent recursively executing/parsing SQL statements or time spent recursively executing PL/SQL.

System Time Detail

The System Time Detail section allows you to see where the database engine has been spending its time. Both raw time and percentage of total metrics are included.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Response Time Analysis View (DBArtisan - Oracle 10G Only)** > Active Time

Metric

The metrics shown here include (note, this table is largely from Oracle 10g documentation):

Metric	Description
DB Time	Amount of elapsed time (in microseconds) spent performing Database user-level calls. This does not include the time spent on instance background processes such as PMON.
DB CPU	Amount of CPU time (in microseconds) spent on database user-level calls. This does not include the CPU time spent on instance background processes such as PMON.
background cpu time	Amount of CPU time (in microseconds) consumed by database background processes.
sequence load elapsed time	Amount of elapsed time spent getting the next sequence number from the data dictionary. If a sequence is cached, then this is the amount of time spent replenishing the cache when it runs out. No time is charged when a sequence number is found in the cache. For non-cached sequences, some time will be charged for every nextval call.
parse time elapsed	Amount of elapsed time spent parsing SQL statements. It includes both soft and hard parse time.
hard parse elapsed time	Amount of elapsed time spent hard parsing SQL statements.
sql execute elapsed time	Amount of elapsed time SQL statements are executing. Note that for select statements this also includes the amount of time spent performing fetches of query results.
connection management call elapsed time	Amount of elapsed time spent performing session connect and disconnect calls.
failed parse elapsed time	Amount of time spent performing SQL parses which ultimately fail with some parse error.
hard parse (sharing criteria) elapsed time	Amount of elapsed time spent performing SQL hard parses when the hard parse resulted from not being able to share an existing cursor in the SQL cache.
hard parse (bind mismatch) elapsed time	Amount of elapsed time spent performing SQL hard parses when the hard parse resulted from bind type or bind size mismatch with an existing cursor in the SQL cache.
PL/SQL execution elapsed time	Amount of elapsed time spent running the PL/SQL interpreter. This does not include time spent recursively executing/parsing SQL statements or time spent recursively executing the Java VM.
PL/SQL compilation elapsed time	Amount of elapsed time spent running the PL/SQL compiler.
inbound PL/SQL rpc elapsed time	Time inbound PL/SQL remote procedure calls have spent executing. It includes all time spent recursively executing SQL and JAVA, and therefore is not easily related to "PL/SQL execution elapsed time".

Java execution elapsed time	Amount of elapsed time spent running the Java VM. This does not include time spent recursively executing/parsing SQL statements or time spent recursively executing PL/SQL.

Session Time Detail

This section allows you to see where sessions that are currently logged on are spending their time.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Response Time Analysis View (DBArtisan - Oracle 10G Only)** > Active Time

Metrics

The metrics shown here include (note, this table is largely from Oracle 10g documentation):

Metric	Description
DB Time	Amount of elapsed time (in microseconds) spent performing Database user-level calls. This does not include the time spent on instance background processes such as PMON.
DB CPU	Amount of CPU time (in microseconds) spent on database user-level calls. This does not include the CPU time spent on instance background processes such as PMON.
background cpu time	Amount of CPU time (in microseconds) consumed by database background processes.
sequence load elapsed time	Amount of elapsed time spent getting the next sequence number from the data dictionary. If a sequence is cached, then this is the amount of time spent replenishing the cache when it runs out. No time is charged when a sequence number is found in the cache. For non-cached sequences, some time will be charged for every nextval call.
parse time elapsed	Amount of elapsed time spent parsing SQL statements. It includes both soft and hard parse time.
hard parse elapsed time	Amount of elapsed time spent hard parsing SQL statements.
sql execute elapsed time	Amount of elapsed time SQL statements are executing. Note that for select statements this also includes the amount of time spent performing fetches of query results.
connection management call elapsed time	Amount of elapsed time spent performing session connect and disconnect calls.
failed parse elapsed time	Amount of time spent performing SQL parses which ultimately fail with some parse error.
hard parse (sharing criteria) elapsed time	Amount of elapsed time spent performing SQL hard parses when the hard parse resulted from not being able to share an existing cursor in the SQL cache.
hard parse (bind mismatch) elapsed time	Amount of elapsed time spent performing SQL hard parses when the hard parse resulted from bind type or bind size mismatch with an existing cursor in the SQL cache.
PL/SQL execution elapsed time	Amount of elapsed time spent running the PL/SQL interpreter. This does not include time spent recursively executing/parsing SQL statements or time spent recursively executing the Java VM.
PL/SQL compilation elapsed time	Amount of elapsed time spent running the PL/SQL compiler.
inbound PL/SQL rpc elapsed time	Time inbound PL/SQL remote procedure calls have spent executing. It includes all time spent recursively executing SQL and JAVA, and therefore is not easily related to "PL/SQL execution elapsed time".

Java execution elapsed time	Amount of elapsed time spent running the Java VM. This does not include time spent recursively executing/parsing SQL statements or time spent recursively executing PL/SQL.

System Bottlenecks

The System Bottlenecks tab of the Response Time Analyst detail view displays the following information:

Bottleneck Summary Bottleneck Detail Bottleneck History (One Hour)

Bottleneck Summary

Waits on a system generally occur for three reasons:

1 A process waits because it has no work to do.

2 A process waits because a requested resource is not available.

3 A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention. The Bottleneck Summary section presents the broad categories of waits (called classes) so you can see if, for example, User I/O is causing high wait times. The percentage of each wait class is shown in the pie chart.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Response Time Analysis View (DBArtisan - Oracle 10G Only)** > System Bottlenecks
Metrics	Details can be viewed for the wait classes by drilling down into the System Waits view.

Bottleneck Detail

Waits on a system generally occur for one of three reasons:

1 A process waits because it has no work to do.

2 A process waits because a requested resource is not available.

3 A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention. The Bottleneck Detail section presents the broad categories of waits (called classes) so you can see if, for example, User I/O is causing high wait times. The wait class, total waits, percentage of total waits, wait time (in seconds) and percentage of wait time is shown. Note that all these metrics are cumulative since the instance was started.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Response Time Analysis View (DBArtisan - Oracle 10G Only)** > System Bottlenecks

Metrics

Details can be viewed for the wait classes by drilling down into the System Waits view.

Bottleneck History (One Hour)

Waits on a system generally occur for three reasons:

1 A process waits because it has no work to do.

2 A process waits because a requested resource is not available.

3 A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention.

The Bottleneck History section allows you to view wait counts, wait time, or a count of user sessions waiting over the past hour for all the specific wait classes (User I/O, System I/O, etc.)

Location Oracle Performance Analyst Statistics (DBArtisan) > **Response Time Analysis View (DBArtisan - Oracle 10G Only)** > System Bottlenecks

Session Bottlenecks

The Session Bottlenecks tab of the Response Time Analysis detail view displays the following information:

Session Bottleneck Summary Session Bottleneck Detail Session Wait History (One Hour)

Session Bottleneck Summary

Waits on a system generally occur for three reasons:

1 A process waits because it has no work to do.

2 A process waits because a requested resource is not available.

3 A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention.

The Session Bottleneck Summary section shows where currently logged on sessions have been spending the largest percentages of their time (over broad wait classes).

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Response Time Analysis View (DBArtisan - Oracle 10G Only)** > Session Bottlenecks
Metrics	To view the actual wait events that comprise the highest percentages of time viewed in each wait class, drill down into the Session Waits view.

Session Bottleneck Detail

Waits on a system generally occur for three reasons:

1 A process waits because it has no work to do.

2 A process waits because a requested resource is not available.

3 A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not normally a problem, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention.

The Session Bottleneck Detail section shows where currently logged on sessions have been spending the largest percentages of their time (over broad wait classes). Each session is shown along with its SID, user name, wait class, total waits, time waited (in seconds), and percent of total time.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Response Time Analysis View (DBArtisan - Oracle 10G Only)** > Session Bottlenecks
Metrics	To view the actual wait events that comprise the highest wait times/percentages of time viewed in each wait class, drill down into the Session Waits view.

Session Wait History (One Hour)

The Session Wait History allows you to look back over the last hour to see when sessions where waiting, how many waits occurred, and the amount of wait time associated during a specific period of time. Analyzing such data over time will give you an idea of how often waits are occurring and their overall impact on system performance.

Location	Oracle Performance Analyst Statistics (DBArtisan) > **Response Time Analysis View (DBArtisan - Oracle 10G Only)** > Session Bottlenecks

Wait Histogram

The Wait Histogram view allows you to view a histogram of wait times and wait counts for a variety of wait events. Using the wait histogram view, you can see, for example, that a db file scattered read event had 20 waits that took an average of 1 second and 10 waits that took an average of 2 seconds, and so on. This allows you to get a better idea of the actual impact of various wait events than viewing broad percentages or averages.

Historical Session Analysis

The Historical Session Analysis tab of the Response Time Analysis detail view shows data for:

- Historical Session Analysis
- Historical Session Analysis Detail - All Waits

Historical Session Analysis

Oracle10g and above allow you to review various historical performance metrics for your database. The Historical Analysis graph allows you to input a time period and review wait times to see when your server experienced high volumes of wait activity. You can choose to look at global wait classes or specific waits events for wait classes.

You can also select a specific time period in the graph by drawing a box around part of the graph with your mouse. The graph will change to match your selection.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Response Time Analysis View (DBArtisan - Oracle 10G Only)** > Historical Session Analysis

Historical Session Analysis Detail - All Waits

Oracle10g and above allows you to review various historical performance metrics for your database. The Historical Analysis Detail section allows you to review specific details regarding times when your database server experienced unacceptable wait times. You can review historical response time data for sessions, SQL statements, files, and objects.

This section allows you, for example, to see what wait events long running SQL statements experienced and what objects caused excessive wait times.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Response Time Analysis View (DBArtisan - Oracle 10G Only)** > Historical Session Analysis

Session Details View (DBArtisan - Oracle Performance Analyst)

The following tabbed pages are available on the Session Details page:

Current Work Tab	Locks Tab	Overview Tab
SQL Tab	Waits Tab	

Overview Tab

Several sections are available on the Overview tab of the Session Details page. The table below describes the sections:

Section	Description
Memory Usage	The Memory Usage section defines the session's total memory usage against all session memory usage on the database.
I/O Activity	The I/O section defines the session's total I/O consumption against all I/O activity on the database. NOTE Bugs in some versions of the Oracle data dictionary cause negative numbers to display if the database has experienced large amounts of I/O activity.
CPU Activity	The CPU Activity section defines the session's total CPU consumption against all CPU usage on the database. NOTE Bugs in some versions of the Oracle data dictionary cause the total CPU usage for a session to not be calculated correctly.
Statistical Detail	The Statistical Detail section details every performance metric available for the session. NOTE Each measure continue to increment as long as the session stays connected to Oracle.

Current Work Tab

The Current Work tab of the Session Details page includes the following sections:

- Open Transactions

The following statistic, available on this tab, duplicates a statistic available on the Key User Analysis Pane

- Open Cursors

Open Transactions

This Open Transactions section displays details regarding any open transaction that the session currently has. The table below describes the information available in this section:

Column	Description
SQL Text	The SQL that contains the transaction.
Start Time	The time the statement was issued.
Logical I/O	The amount of memory I/O the transaction has used.
Physical I/O	The amount of disk I/O the transaction has used.
Gets	A subset of logical I/O.
Changes	Any data modifications made by the transaction.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Session Details View (DBArtisan - Oracle Performance Analyst)** > Current Work Tab

Metrics

Seeing a long running, persistent transaction might indicate a session caught in a loop or other coding snafu. Watching a transaction sit idle could clue you into a lock contention problem that exists.

Waits Tab

The Waits tab of the Session Details page includes the following sections:

Time Summary (Oracle 10g only) Last Ten Waits (Oracle 10g only)

The following statistics, available on this tab, are duplicates of statistics available on the System Waits Tab of the Users View (DBArtisan - Oracle Performance Analyst) page:

Historical Waits Current Waits

Time Summary (Oracle 10g only)

The Time Summary section shows you where the session has been spending its time. Using the Time Summary section, you can easily determine if a session has been bottlenecked by SQL calls, PL/SQL work, or other activities.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Session Details View (DBArtisan - Oracle Performance Analyst)** > Waits Tab

Last Ten Waits (Oracle 10g only)

The Last Ten Waits section shows you, in order, the last 10 wait events experienced by the session, along with other important metrics like the wait time, and so forth.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Session Details View (DBArtisan - Oracle Performance Analyst)** > Waits Tab

Locks Tab

The following statistics, available on this tab, are duplicates of statistics availale on the Locks/Blocking Locks Tab of the Users View (DBArtisan - Oracle Performance Analyst) page:

All Locks Blocking Locks

SQL Tab

The SQL tab of the Session Details page includes the following sections:

Current SQL Current SQL Statistics Other SQL

Current SQL

The Current SQL section displays the most current SQL issued by the session.

Note: You can run an EXPLAIN PLAN on the SQL or (if you have Embarcadero SQL Tuner installed) tune the SQL for better performance.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Session Details View (DBArtisan - Oracle Performance Analyst)** > SQL Tab

Current SQL Statistics

The Current SQL Statistics section displays the performance execution metrics for the current SQL statement in graphical format. Physical and logical I/O is displayed along with rows processed and number of executions.

Location Oracle Performance Analyst Statistics (DBArtisan) > **Session Details View (DBArtisan - Oracle Performance Analyst)** > SQL Tab

Other SQL

The Other SQL section displays other SQL that has been parsed by the Oracle username. The table below describes the information available in this section:

Column	Description
SQL Text	A snippet of the SQL code issued.
Disk Reads	The number of total disk reads for the statement.
Buffer Gets	The number of total buffer gets for the statement.
Rows Processed	The number of total rows processed for the statement.
Executions	The number of times the statement has been executed.
Sorts	The number of sorts for the statement.

Note: Bugs in some versions of Oracle will cause negative numbers to display for various execution metrics.

Performance Analyst for Microsoft SQL Server Tutorial (DBArtisan)

Embarcadero Performance Analyst provides key insight into the complex world of database performance. Running inside the Embarcadero DBArtisan console, or standalone in a Web browser, Performance Analyst helps you pinpoint performance problems that threaten a database's availability or response times. The powerful analysis facilities will satisfy advanced experts, yet novices will find Performance Analyst's interface easy to use and the expert help a welcome aid to troubleshooting difficult performance issues.

Proceed to About This Performance Analyst for Microsoft Tutorial (DBArtisan)

Topics

About This Performance Analyst for Microsoft Tutorial (DBArtisan)

This tutorial is intended to help you get started with Embarcadero Performance Analyst.

After completing this tutorial, you will have the foundation you need to explore the many features and benefits of Performance Analyst. You will have learned how to accurately monitor your database using the most popular performance methodologies in use today. You will know how to drill down into the many layers of database performance and also be able to locate performance issues in your database in a fast and effective manner. Finally, you'll understand how to fix performance problems using the integration that exists between Performance Analyst and Embarcadero DBArtisan.

This tutorial is divided into 8 sections.

Session 1: Getting Started with Performance Analyst for Microsoft (DBArtisan)

Session 2: Using Ratio-based Performance Analysis in Performance Analyst for Microsoft (DBArtisan)

Session 3: Using Bottleneck/wait-based Performance Analysis in Performance Analyst for Microsoft (DBArtisan)

Session 4: Using Workload Performance Analysis in Performance Analyst for Microsoft (DBArtisan)

Session 5: Using SQL Performance Analysis in Performance Analyst for Microsoft (DBArtisan)

Session 6: Using Operating System Performance Analysis in Performance Analyst for Microsoft (DBArtisan)

Session 7: Identifying and Getting Help With Performance Problems Using Performance Analyst for Microsoft (DBArtisan)

Session 8: Fixing Identified Problems Using Performance Analyst for Microsoft (DBArtisan)

You can use this basic tutorial as a roadmap of product highlights.

After starting this tutorial, you can select Help from the menu to find many additional resources that complement and build on many of the activities shown in this brief guide.

For more information, see [DBATutorial.htm#AdditionalEvaluationResources Additional Evaluation Resources].

Session 1: Getting Started with Performance Analyst for Microsoft (DBArtisan)

Download and Install

You can obtain the latest version of Performance Analyst by downloading the DBArtisan software from the Embarcadero Web site at http://www.embarcadero.com/downloads/downloaddbartisan.asp. Performance Analyst runs fully contained within DBArtisan so you will need the DBArtisan product to use Performance Analyst (although Performance Analyst can also run in standalone fashion within a Microsoft Internet Explorer Web browser).

Click Download, and follow the steps indicated. Save the file on your computer and then double-click to launch the self-extracting file that will guide you through the installation process.

When you first install an evaluation copy of DBArtisan, you can use the tool for 14 days. After that time, a permanent license is needed.

Proceed to Overview of Performance Analyst for Microsoft

Overview of Performance Analyst for Microsoft

When DBArtisan is connected to a Microsoft SQL Server database, you can open Performance Analyst two ways:

1 On **Utilities** menu, select **Performance Analyst**.

For Performance Analyst to work, several procedures need to be installed by the SA account. When Performance Analyst is first invoked, it will check for the presence of this package and if it has not been installed, the product will prompt you to install it. Once created, any account with access to these procedures may monitor databases using Performance Analyst. No other objects need to be installed on a monitored database.

Proceed to Performance Analyst for Microsoft Product Navigation

Performance Analyst for Microsoft Product Navigation

Navigating through Performance Analyst is easily accomplished through two main mechanisms:

1 The main Performance Analyst toolbar contains buttons that invoke the major performance analysis views within the product as well as frequently used actions like Refresh and Performance Threshold Customizations.

2 Positioning the mouse pointer on any area within Performance Analyst and clicking the right mouse button will bring up a navigation menu that allows you to easily move between any of the main Performance Analyst home pages and drill down views. The right mouse menu also contains context sensitive options like drilling down further into a particular statistic, copying the display to a text file, and more.

Proceed to Session 2: Using Ratio-based Performance Analysis in Performance Analyst for Microsoft (DBArtisan)

Session 2: Using Ratio-based Performance Analysis in Performance Analyst for Microsoft (DBArtisan)

Ratio-based analysis has been around for many years, and used to be the only technique database administrators used when they were called on to diagnose the cause of a database slowdown. Performance ratios are very good roll-up mechanisms for busy DBAs, making possible the analysis-at-a-glance approach, however ratios must be computed correctly for them to be meaningful. Key ratios in Performance Analyst are calculated using delta statistics, which is superior to computing ratios with cumulative values. Delta statistics, which are values derived by using the before and after measures in each statistical refresh done in Performance Analyst, lets you see changes in performance metrics that would otherwise be invisible if cumulative statistics were used. Performance Analyst makes it easy to see the ratios that are computed with delta statistics vs. cumulative. A triangle symbol denotes a delta-based statistic, while square symbols represent a cumulative-based measure.

Ratio-based analysis is available for all performance categories (memory, etc.) where ratios can assist in determining the performance level of a database. For example, if you would like to see memory-specific ratios, on the Performance Analyst toolbar click Memory. Performance Analyst displays the Memory home page that showcases all important memory-related ratios.

Proceed to Obtaining Detail Information for a Ratio using Performance Analyst for Microsoft

Obtaining Detail Information for a Ratio using Performance Analyst for Microsoft

A ratio gives you a global picture for a particular performance measure. However, you often need more information regarding a statistical measure so you can make an informed decision. Performance Analyst has easy-to-use drill down features that make obtaining detailed information for various ratios a breeze.

For example, to view more information for the Buffer Cache Hit Ratio statistic, do the following:

1 On either the Main Home page or the Memory Home page, locate the Buffer Cache Hit Ratio statistic.

2 Position the pointer over the statistic.

The pointer changes to a hand symbol.

3 Double-click the left mouse button.

Performance Analyst opens the **Buffer Cache** tab of the memory drill down view.

OR

4 Right-click the statistic, select Details.

OR

5 Select Navigate to, Memory, Buffer Cache.

Performance Analyst opens the **Buffer Cache** tab of the memory drill down view.

Proceed to Session 3: Using Bottleneck/wait-based Performance Analysis in Performance Analyst for Microsoft (DBArtisan)

Session 3: Using Bottleneck/wait-based Performance Analysis in Performance Analyst for Microsoft (DBArtisan)

When a Microsoft SQL Server database is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things, or it can be an indicator that a database bottleneck exists. Bottleneck analysis is a valid method of measuring performance because it helps you track where a database and user sessions have been spending their time. If latch contention or heavy table-scan activity has been dragging a database's performance down, you can use bottleneck analysis to confirm the actual root cause.

Performance Analyst uses two broad areas of metrics to form its bottleneck analysis. The first is the Microsoft SQL Server wait event interface, which displays where the database and user sessions have been bottlenecked. But, when using bottleneck analysis you can not rely only on the information contained in the wait event views that Microsoft SQL Server provides. For example, an object may attempt to extend into another extent of space in a tablespace and yet, be denied if no such free space exists. Such a failure will not be reflected in any wait event, but still represents a very real bottleneck to the database. For this reason, Performance Analyst also shows non-wait event based bottlenecks that represent a very real threat to a database's availability and performance.

Performance Analyst prominently displays bottleneck analysis on every performance home page. The Main home page, displays a summary of all key bottlenecks threatening the database. You can also see specific bottlenecks for memory, I/O, space, objects, and users by navigating to each home page and viewing the Bottleneck Analysis section for that home page. For example, to see bottlenecks that specifically relate to I/O, do the following

1 On the toolbar, click the I/O button.

2 Locate the Bottleneck Analysis section at the lower-left portion of the screen.

The I/O bottleneck analysis section displays the Top I/O Wait Events by wait count and wait time, along with the Hottest Database and Log Files.

Proceed to Obtaining Detail on System Bottlenecks Using Performance Analyst for Microsoft

Obtaining Detail on System Bottlenecks Using Performance Analyst for Microsoft

You can easily obtain more information regarding global bottlenecks that are plaguing a database by drilling down into a particular bottleneck metric. For example, if you would like to see more data regarding the current top system bottlenecks, do the following:

1 Open the Main home page.

2 Double-click the Top System Bottlenecks area.

Performance Analyst opens the System Waits view.

3 Or right-click anywhere in Performance Analyst, point to Users, and then click System Waits.

Performance Analyst opens the System Waits view.

The System Waits view displays critical details regarding all the wait events that Microsoft SQL Server has recorded. You can easily see what events that occurred the most often as well as the events that have accumulated the most wait time.

Tip: If you would like to see an expanded view of the System waits grid shown on the top of the System Waits view, click the Expand button located at the top right corner of the section. All section views in Performance Analyst can be maximized/minimized using this control.

Proceed to Viewing Detail for Session Bottlenecks Using Performance Analyst for Microsoft

Viewing Detail for Session Bottlenecks Using Performance Analyst for Microsoft

Not only can you obtain detail information for system bottlenecks, but you can also use Performance Analyst to get drill down information regarding session-level bottlenecks. There are several layers to session-level details you can view.

To see more information regarding session-level waits, do the following:

1 Open the Main home page.

2 Double-click the Top Session Bottlenecks area.

OR

3 Right-click anywhere in Performance Analyst, point to Users, and then click Session Waits.

Performance Analyst opens the Session Waits drill down view.

When you are in the Session Waits drill down view, to drill further into a session’s details, do the following:

1 In the Historical Waits or Current Waits grid, select a session that appears to be experiencing critical waits.

2 Double-click the session or right-click, and then select Details.

Performance Analyst opens a drill down view that shows key statistics for the selected session, including its current work activities, wait events, and SQL statements.

Proceed to Session 4: Using Workload Performance Analysis in Performance Analyst for Microsoft (DBArtisan)

Session 4: Using Workload Performance Analysis in Performance Analyst for Microsoft (DBArtisan)

Session 4: Using Workload Performance Analysis in Performance Analyst for Microsoft (DBArtisan)

Key ratios give you a global perspective on database activity. Bottleneck analysis gives you insight into things that are holding up user activity and throughput. But another technique is necessary if a database professional is to really get a handle on what’s occurring inside a badly performing database.

Workload analysis involves the investigation of two critical areas of a database’s performance:

- Session resource consumption and activity
- SQL execution

Without looking at these two key performance categories, you can miss a lot of what could be responsible for perceived performance problems. When performance on a database takes a sudden nosedive, it is not uncommon to find one or two sessions that are causing the bulk of the workload. Finding these processes can be easily accomplished by viewing session metadata coupled with resource consumption and statistical execution statistics.

Proceed to Pinpointing Top Resource Consumers Using Performance Analyst for Microsoft

Pinpointing Top Resource Consumers Using Performance Analyst for Microsoft

Performance Analyst identifies top resource consumers in a couple of different ways. On the main home page, Performance Analyst highlights the top resource sessions across I/O, memory usage, and CPU consumption. The percentage used across all statistical categories are displayed so a DBA can immediately pinpoint a session that is using all or most of a particular resource. The top resource sessions are also displayed on the memory, I/O, and other home pages as well with a listing for the top sessions for that category (for example, the top memory users appear on the memory home page, etc.)

Proceed to Getting Workload Details Using Performance Analyst for Microsoft

Getting Workload Details Using Performance Analyst for Microsoft

Performance Analyst offers several details on top resource sessions in addition to the summaries provided on each home page. A detailed and flexible top sessions view lets you view a summary of all the major resource sessions on the database as well as every critical detail regarding each session connected to the database.

Further drill downs are available that highlights specific details regarding each session's activities and submitted SQL. To see detailed information regarding a session identified in one of Performance Analyst's Workload Analysis sections, do the following:

1 In a Workload Analysis display, position the pointer over one of the processes.

2 Double-click the process or right-click, and then select Details.

If you want like to see all key workload analysis metadata for sessions, do the following:

1 On the toolbar, click the Top Sessions button.

OR

2 Right-click anywhere in Performance Analyst, point to Users, and then click Session Waits.

Proceed to Session 5: Using SQL Performance Analysis in Performance Analyst for Microsoft (DBArtisan)

Session 5: Using SQL Performance Analysis in Performance Analyst for Microsoft (DBArtisan)

SQL-based performance analysis is really a subset of Workload Analysis. Understanding current and historical SQL execution patterns lets you to have the second set of data points necessary to properly perform workload analysis. You may find that optimizing SQL code produces some of the best performance-enhancing boosts available for a database.

Proceed to Uncovering Resource-intensive SQL Using Performance Analyst for Microsoft

Uncovering Resource-intensive SQL Using Performance Analyst for Microsoft

Performance Analyst offers plenty of insight into SQL performance and execution patterns. Each one of the home pages depicts the most costly SQL statements as determined by a variety of different metrics for that performance category (most physical, logical I/O, etc.) For example, Performance Analyst might show that a single SQL statement is responsible for almost 60 percent of all the CPU consumption on the database.

Proceed to Getting Details on Top SQL with Performance Analyst for Microsoft

Getting Details on Top SQL with Performance Analyst for Microsoft

Getting details on identified SQL statements is easy in Performance Analyst. For example, if you would like to see further information on a SQL statement identified on the Performance Analyst Home page, do the following:

1 In the SQL Analysis section, position the pointer over Hash Values.

2 Double-click or right-click, and then select Details.

Performance Analyst opens the Top SQL summary page that shows a breakdown of all top SQL statements along with their execution statistics.

If you would like to see the full SQL text of a SQL statement, do the following

1 In the Top SQL Details section, double-click a statement.

Performance Analyst opens the **SQL Details** tab of the Top SQL view to display the entire SQL statement and the execution metrics. While on this pane, you can also perform an EXPLAIN of the SQL statement and examine its access path information.

To automatically tune a problem SQL statement if you have Embarcadero SQL Tuner installed on your workstation, do the following:

1 On the **SQL Details** tab, select a problem statement, and then click the Tune SQL button.

SQL Tuner opens and you can begin a tuning session.

Proceed to Session 6: Using Operating System Performance Analysis in Performance Analyst for Microsoft (DBArtisan)

Session 6: Using Operating System Performance Analysis in Performance Analyst for Microsoft (DBArtisan)

In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database sever for resources (CPU, I/O or Memory), a slow CPU, insufficient I/O devices, and insufficient memory. Performance Analyst allows the DBA to examine the relevant operating system metrics for Windows Platforms (NT /2000/XP/2003) as they relate to SQL Server.

To view the operating system metrics of the machine hosting the SQL Server you are monitoring, do the following:

1 Click the OS Performance Analyst toolbar button.

OR

2 Navigate to the OS Home Page by going to OS, and then Home.

Note: The first time you invoke the OS feature you will be prompted for an operating system login. Please specify the correct operating system, host name, username & password and protocol.

The Operating System home page has been designed in a similar fashion as the datasource home page. On this page you can see the pertinent operating system metrics displayed in five quadrants broken down into the following categories:

- Resource Usage
- Memory Analysis
- Disk Analysis
- Workload Analysis
- Bottleneck Analysis

Proceed to Obtaining Details on the Operating System Using Performance Analyst for Microsoft

Obtaining Details on the Operating System Using Performance Analyst for Microsoft

You can easily obtain more information regarding operating system problems that are degrading database performance by drilling down into a particular analysis quadrant. For example if you'd like to see more information regarding the processes running on the host machine, do the following:

1 Navigate to the Performance Analyst Operating System home page.

2 Double-click the Top Memory Processes in the Workload Analysis quadrant.

Performance Analyst opens the Processes Drilldown view.

OR

3 Right-click anywhere in Performance Analyst and select the Navigate to, OS, and then Processes.

The Processes detail tab shows all the processes on the host machine. From this view you can easily determine which processes are showing high marks in I/O or CPU, for example, by simply double clicking on the column headers and sorting the processes in ascending or descending order.

Proceed to Session 7: Identifying and Getting Help With Performance Problems Using Performance Analyst for Microsoft (DBArtisan)

Session 7: Identifying and Getting Help With Performance Problems Using Performance Analyst for Microsoft (DBArtisan)

You do not have to be an advanced expert in performance tuning to spot trouble in a database if you use Performance Analyst. There are a number of mechanisms in the tool that help you quickly locate areas of concern within a database.

Proceed to Using the Alarm Log in Performance Analyst for Microsoft

Using the Alarm Log in Performance Analyst for Microsoft

Performance Analyst has a central and flexible alarm log that lists every performance violation found in all monitored databases. Unlike some products that only show you alarms for a particular database, Performance Analyst lists all alarms for each database you are monitoring in one central log so you can see every problem in one place. To view the Performance Analyst Alarm log, do the following:

1 On the toolbar, click the Alarms button.

Once in the Alarm log, you can perform a number of actions including:

- Selecting an alarm and viewing the application page that shows details regarding the metric in violation.
- Filtering the types of alarms shown through the Options dialog.
- Clearing all current alarms.
- Copying or saving the Alarm log into a different format.

To select any of these options, do the following:

1 Position the pointer over any area of the Alarm Log.

2 Right-click and In the shortcut menu, select an option.

Proceed to Recognizing Performance Violations Using Performance Analyst for Microsoft

Recognizing Performance Violations Using Performance Analyst for Microsoft

Performance Analyst makes it easy to visually pinpoint performance violations in a database. Any statistic that does not conform to your predefined performance standards is colorized in any view that it appears so you can easily see potential troublespots.

The colors used to highlight a particular performance violation depend on the severity of the problem. Performance Analyst uses several layers of performance standards/threshold violations with each layer being represented by a different color. You can control everything regarding threshold violations, including how thresholds are determined, their color representation in

Performance Analyst's display, and more by using the Threshold Template feature of Performance Analyst.

How can Performance Analyst inform you of performance violations in your database if you minimize the product to your desktop and are working on other things? Even if Performance Analyst is not right in front of you, it can still let you know that a problem has been found in your database through the systray alarm icon. When Performance Analyst detects a problem in one of your databases, one of the things the product does is enable an alarm icon in your workstation's systray area that communicates the presence of alarms in your database. So even if the Performance Analyst interface is not in sight, the product can let you know that something in your database requires attention.

Proceed to Creating Threshold Templates for Performance Analyst for Microsoft

Creating Threshold Templates for Performance Analyst for Microsoft

How does Performance Analyst know what to signal as a problem in a database? It uses predefined performance standards that you define and enforce. All of this is accomplished by using Performance Analyst's Threshold templates. A template is a set of performance metrics and their thresholds that can be reused to enforce performance standards across any number of databases. Embarcadero provides a default performance template that contains nearly all of the most-watched performance statistics so you can begin monitoring with confidence right out of the box. However, you can create your own Threshold templates so you only are alerted to performance violations that are important to you and your particular environment.

To view or create Threshold templates, do the following:

1 On the toolbar, click the Thresholds button.

Performance Analyst opens the T**hreshold Template Properties** dialog.

The **Threshold Template Properties** dialog lets you select the options to add or edit templates, as well as other maintenance options. For example, you can copy the original Embarcadero Threshold template and modify it to fit your own needs. You can control what statistics are sampled for performance violations as well as the levels of severity you would like reported. Once you are satisfied with your template, you can save it and then apply it to one or more datasources in one action to immediately begin enforcing your performance standards.

Proceed to Getting Help With Performance Problems Using Performance Analyst for Microsoft

Getting Help With Performance Problems Using Performance Analyst for Microsoft

As databases become more complex, so do the problems that plague them. Knowing what to do for a particular performance problem can be hard in some cases, but Performance Analyst helps by providing a complete set of expert help that you can use as a guide when performance problems occur. To obtain expert help for any statistic shown in Performance Analyst, do the following:

1 Position the pointer over a statistic that you need help.

2 Right-click and select Help.

The Expert Help gives you a general description of the statistic, the range of values that are considered optimal, and some suggestions on how to fix a problem situation. Expert help can also be invoked for any of the Ratio, Bottleneck, SQL, or Workload Analysis sections shown on every home page by clicking on the help icon for that section, which is located at the top right corner of the section.

Proceed to Session 8: Fixing Identified Problems Using Performance Analyst for Microsoft (DBArtisan)

Session 8: Fixing Identified Problems Using Performance Analyst for Microsoft (DBArtisan)

Finding performance problems is one thing, but fixing them is another. Many performance monitors point out a particular problem in a database, but then offer no way to actually correct the situation. Such is not the case with Performance Analyst. Because of its tight integration with DBArtisan, certain problems found in Performance Analyst can be visually corrected by DBArtisan.

A Space Problem Correction Example

If an Microsoft SQL Server tablespace is about to run out of free space, Performance Analyst can issue an alarm that a space shortage exists. Once the space problem has been identified, do the following:

1 On the Space home page toolbar, click the Space button.

2 In the Space Analyst grid, right-click a tablespace, and then select Manage with DBArtisan.

DBArtisan opens an editor where you can perform corrective actions, such as enabling Autoextend for the tablespace, resizing the tablespace’s existing datafile(s), or adding a new datafile for the tablespace.

Microsoft SQL Server Performance Analyst Statistics (DBArtisan)

This section includes expert help for all Microsoft SQL Server statistics in the Embarcadero Performance Analyst views and pages:

- Home View (DBArtisan - SQL Server Performance Analyst)
- Memory View (DBArtisan - SQL Server Performance Analyst)
- I/O View (DBArtisan - SQL Server Performance Analyst)
- Space View (DBArtisan - SQL Server Performance Analyst)
- Users View (DBArtisan - SQL Server Performance Analyst)
- Database View (DBArtisan - SQL Server Performance Analyst)
- OS View (DBArtisan - SQL Server Performance Analyst)
- Top SQL View (DBArtisan - SQL Server Performance Analyst)

Home View (DBArtisan - SQL Server Performance Analyst)

The Embarcadero Performance Analyst Home page lets you review availability and overall performance of all monitored databases from a single window. Statistics on the Home view are organized into the following categories:

Key Ratio Analysis Pane Bottleneck Analysis Pane SQL Analysis Pane

Storage Analysis Pane Workload Analysis Pane

Key Ratio Analysis Pane

Database performance analysts typically use one of two methods for examining the performance levels of a database - ratio-based or wait/bottleneck-based. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as very good roll-up mechanisms for busy DBA's to use for at-a-glance performance analysis. Many DBAs have large database farms to contend with and cannot spend time checking detailed wait-based analysis outputs for each and every database they oversee. Succinctly presented performance ratios can assist in such situations by giving DBAs a few solid indicators that can be quickly scanned to see if any database needs immediate attention.

While there are certainly many opinions as to what rules to follow, there are some standards that should always be adhered to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that a DBA will examine come from the master.sysperfinfo performance table. This table maintains a count of various statistics for the server and particular databases since the server was brought up. For servers that are kept up for long periods of time, these values can grow quite large and will impact how a particular ratio that a DBA might be looking at is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each SQL Server's individual 'personality' with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead the DBA to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following ratios are used on the Performance Analyst Home page to succinctly communicate the general overall performance levels of the monitored database. The statistics available in this pane, are duplicates of statistics available on the Key Ratio Analysis Pane of the **Home View (DBArtisan - SQL Server Performance Analyst)** page:

Buffer Cache Hit Ratio

Data read from memory will produce end-user response times many times faster than when that same data is read from disk. Keeping physical I/Os to an absolute minimum is one of the purposes of the SQL Server buffer/data cache.

The buffer cache hit ratio is a terrific indicator of how often user requests for data are satisfied through memory vs. being physically read from disk.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Key Ratio Analysis PaneMicrosoft SQL Server Performance Analyst Statistics (DBArtisan) > Memory View (DBArtisan - SQL Server Performance Analyst) > Buffer Cache Tab
Metrics	To help ensure excellent performance, you want to keep your cache hit ratio in the neighborhood of 90% or higher. However, you should be aware that every server has its own 'personality' and might exhibit excellent performance with below average readings for the cache hit ratio. You should also be aware that excessive logical I/O activity can produce a very high cache hit ratio while actually degrading overall database performance. Consistently viewed low readings (60% or lower) might require tuning attention on the part of the DBA.
Correction	Ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server memory configuration option. Also, consider increasing your SQL Server Min. Memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed. Often a user process is taking a large amount of memory due to an inordinate amount of I/O.

Procedure Plan Hit Ratio

The SQL Server procedure cache is used to hold the execution plans for all Transact-SQL statements currently executing in the server. When a user executes a Transact-SQL statement, SQL Server looks in the procedure cache for a query plan to use.

This statistic is the percentage of query plan requests generated by stored procedures that are found in the procedure cache area. The percentage of times that a statement's plan and definition can be referenced in memory, the better the procedure execution time.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Key Ratio Analysis PaneMicrosoft SQL Server Performance Analyst Statistics (DBArtisan) > Memory View (DBArtisan - SQL Server Performance Analyst) > Home > Key Ratio Analysis Pane
Metrics	A high procedure cache hit rate is a desirable thing. You should strive for a hit ratio between 95-100%, with 95% being a good performance benchmark for code reference. Note that when a database is first started, the procedure cache hit rate will not be at an optimal level because all code being used will be relatively new, and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the procedure cache hit rate has not increased to desirable levels, you should look into the possibility of increasing the amount of memory allocated to the cache.
Correction	First, ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server Memory configuration option. Also, consider increasing your SQL Server Min Memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Ad Hoc SQL Hit Ratio

When an ad hoc SQL statement is issued, the query plan is then stored in the SQL Server procedure cache area. If the identical ad hoc statement is launched in the future, SQL Server uses the query plan already stored in the procedure cache if it is still there. The Ad Hoc SQL Hit Ratio statistic defines the percentage of times that a query plan for an ad hoc SQL statement is found in the procedure cache.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Key Ratio Analysis PaneMicrosoft SQL Server Performance Analyst Statistics (DBArtisan) > Memory View (DBArtisan - SQL Server Performance Analyst) > Home > Key Ratio Analysis Pane
Metrics	A high ad hoc hit rate is desirable, but is harder to maintain at a high level than something like a procedure cache hit rate. Therefore, an 80% or greater ad hoc cache hit rate is a good performance benchmark for code reference. Note that when a database is first started, the ad hoc cache hit rate will not be at an optimal level because all code being used will be relatively new, and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the ad hoc cache hit rate has not increased to desirable levels, you should look into the possibility of increasing the amount of memory allocated to the cache.
Correction	First, ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server Memory configuration option. Also, consider increasing your SQL Server Min Memory parameter to allocate more memory to SQL Server. (To obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Log Cache Hit Ratio

The Log Cache Hit Ratio represents the percentage of log cache reads satisfied from the log cache.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Key Ratio Analysis PaneMicrosoft SQL Server Performance Analyst Statistics (DBArtisan) > Memory View (DBArtisan - SQL Server Performance Analyst) > Log Cache Tab
Correction	A low percentage on this statistic is not necessarily a bad sign, as it is possible that the information needed from the log will not be readily available in memory.

CPU Busy

The CPU Busy statistic represents the percentage of time that the CPU has spent working since the last refresh.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Key Ratio Analysis Pane
Metrics	Seeing consistently high numbers might signal an overworked server.
Correction	If you consistently see high numbers for CPU busy, then drill down into session activity to see what processes are consuming the most CPU and look at their activity.

I/O Busy

The I/O Busy statistic represents the percentage of time that SQL Server has spent performing input and output operations since the last refresh.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Key Ratio Analysis PaneMicrosoft SQL Server Performance Analyst Statistics (DBArtisan) > I/O View (DBArtisan - SQL Server Performance Analyst) > Home > Bottleneck Analysis Pane

Server Idle

The Server Idle statistic represents the percentage of time that SQL Server has spent idle since the last refresh.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Key Ratio Analysis Pane

Bottleneck Analysis Pane

When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. And this is where wait-based or bottleneck analysis comes into play. DBAs use this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance because it helps a DBA track where a database has been spending its time. If lock contention or heavy table scan activity has been dragging down database performance, a DBA can use bottleneck analysis to confirm the actual root cause. Once one or more wait events or other bottlenecks have been pinpointed as possible performance vampires, the DBA can drill down and oftentimes discover a fair amount of detail about which sessions and objects are causing the problem. The Performance Analyst Home page identifies the top system and session waits that might be currently a cause of lessened performance.

When using bottleneck analysis, you cannot rely only on the information contained in the wait events that SQL Server provides. For example, a database might be suspect or offline. Such a failure will not be reflected in any wait event, but still represents a very real bottleneck to the database. In the same way that a DBA cannot depend on only a few ratios to properly carry out ratio-based performance analysis, an administrator must include other statistical metrics in their overall bottleneck analysis framework to obtain an accurate performance risk assessment. Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system. The Performance Analyst Home page displays space-related bottlenecks as well as other sources of contention/bottlenecks that might be dragging down the overall performance of your system.

The following bottleneck indicators are used on the Performance Analyst Home page to succinctly communicate the general overall performance level of the monitored database:

Current Object Blocks	DB's Low on Space	Deadlocks
Errors in Current Log	Logs Low on Space	Offline Databases
Suspect Databases	Table Lock Escalations	Top System Bottlenecks

Top System Bottlenecks

The Top System Bottlenecks display identifies the top waits that have occurred on SQL Server based on the wait time.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Bottleneck Analysis Pane
Correction	Wait events can be hard to interpret at times. If you see a particular event that has caused a lot of wait time, you can review the information in this link (Microsoft Knowledge Base Article - 244455) to help understand the cause and potential remedy: http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455

Current Object Blocks

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency. Unless a process times out via an application timeout mechanism, or the process has specified a timeout period via the SET LOCK_TIMEOUT command, a process waiting for a lock will wait indefinitely.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Bottleneck Analysis Pane
Metrics	You should immediately investigate any indicator above zero, before the situation has a chance to mushroom.
Correction	Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Then other user processes, almost always, complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is where it gets tricky. The DBA can drill down into Users View (DBArtisan - SQL Server Performance Analyst) detail and see all current blocking locks, learning exactly which sessions are holding the currently restrictive locks. The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do. By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process will wait for a lock before timing out.

Deadlocks

This metric shows the number of deadlocks per second detected by SQL Server. Deadlocks occur when processes cannot proceed because they are waiting on a set of resources held by each other or held by other processes.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Bottleneck Analysis Pane
Metrics	Consistently seeing deadlock counts greater than zero will indicate that some user processes are experiencing delays in completing their work. When SQL Server identifies a deadlock, it resolves the situation by choosing the process that can break the deadlock. This process is termed the deadlock victim. SQL Server rolls back the deadlock victim's transaction, and then notifies the process' application by returning an error message. It also cancels the process' request and allows the transactions of the remaining processes to continue. SQL Server always attempts to choose the least expensive thread running the transaction as the deadlock victim.
Correction	Because SQL Server automatically resolves deadlock situations, you should proactively work to prevent them in the first place. The culprit of most blocking lock and deadlock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do. You can change default deadlock behavior by using the SET DEADLOCK_PRIORITY command, which reprioritizes a process' position in a deadlock situation.

Table Lock Escalations

The Table Lock Escalations statistic represents the number of times locks on a table were escalated.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Bottleneck Analysis PaneMicrosoft SQL Server Performance Analyst Statistics (DBArtisan) > Users View (DBArtisan - SQL Server Performance Analyst) > Home > The following key statistics are used on the Performance Analyst Users home page to succinctly communicate the general session-based activity levels of the monitored database:
Metrics	Many table lock escalations could indicate contention problems. If increasing numbers of table lock escalations are viewed at the same time as blocking or deadlock problems, then the application design might be at fault.

Suspect Databases

The Suspect Databases statistic represents the number of databases SQL Server has marked as suspect. Databases are marked suspect by SQL Server if they fail during automatic recovery, which is performed at server startup. If serious damage is experienced by a database during regular uptime, SQL server will also mark a database suspect.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Bottleneck Analysis PaneMicrosoft SQL Server Performance Analyst Statistics (DBArtisan) > Database View (DBArtisan - SQL Server Performance Analyst) > Home > Bottleneck Analysis Pane
Metrics	You should not have any suspect databases found on any production server. You should immediately investigate any non-zero numbers for this statistic.
Correction	The steps to handling a suspect database will vary from one installation to another. However, there are some general guidelines you can use to troubleshoot a suspect database. *Begin by examining the SQL Server error log for clues as to what caused the database to be marked as suspect.It is not unusual for a server to run out of physical disk space on drives used by SQL Server. When this happens, recovery for databases can sometimes fail with the end result being SQL Server marking a database as suspect. To remedy this situation, you should free up space on the identified drives or add files to the newly marked suspect database. This can be accomplished by utilizing the two procedures - sp_add_data_file_recover_suspect_db and sp_add_log_file_recover_suspect_db stored. For version 7.0 of SQL Server, you must use the sp_resetstatus stored procedure to reset the suspect status flag for the database in question, use the alter database command to add new datafiles to the database, and then stop/start the SQL Server.Many times, suspect databases are caused by SQL Server not being able to access a database or log file. This happens if a particular physical hard drive has become unavailable, but also can occur if another operating system process has obtained exclusive access to a file. If this scenario proves to be true, once you have ensured that the file(s) are available once again to the operating system, you can use the sp_resetstatus stored procedure to reset the suspect status flag for the database and then stop/start the SQL Server.* If none of these solutions are possible, you will likely have to restore your database using the last full and transaction log backups.

Offline Databases

The Offline Databases statistic represents the number of databases SQL Server has offline, meaning that no database modifications can occur.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Bottleneck Analysis PaneMicrosoft SQL Server Performance Analyst Statistics (DBArtisan) > Database View (DBArtisan - SQL Server Performance Analyst) > Home > Bottleneck Analysis Pane
Metrics	You should not have any offline databases found on any production server. You should immediately investigate any non-zero numbers for this statistic.
Correction	Should an offline database be found by Performance Analyst, you can easily place it back online by utilizing either the sp_dboption stored procedure or the alter database command.

Errors in Current Log

SQL Server records various system events in its system or error log. The majority of messages placed into the log are informational in nature, however since some contain critical messages, you should immediately review them. Performance Analyst indicates the number of actual error messages in the most recent error log so you know if there are potential events that require your attention.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Bottleneck Analysis Pane
Correction	Drill down into the Error Log Tab within Performance Analyst to view the actual error messages.

DB's Low on Space

A SQL server contains many databases, some of which are devoted to system-level activities (the master and tempdb databases, for example) and others that hold user data. The Databases Low on Space statistic indicates databases that have fallen below a recommended percentage of free space.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Bottleneck Analysis PaneMicrosoft SQL Server Performance Analyst Statistics (DBArtisan) > Space View (DBArtisan - SQL Server Performance Analyst) > Home > Bottleneck Analysis Pane
Metrics	This might or might not be a problem. Some databases are not dynamic in nature (meaning they are not expected to grow in size) and are sized so very little free space is present. However, growing databases are another situation and might require DBA intervention if their free space amounts run low.
Correction	If the percent used amount of a database is approaching problematic levels, there are three ways a DBA can rectify the situation: *The DBA can resize the current file(s) used by the database via an ALTER DATABASE … MODIFY FILE commandThe DBA can add a new file to the database via the ALTER DATABASE … ADD FILE command.The DBA can modify the file(s) used by the database to automatically grow by using the ALTER DATABASE … MODIFY FILE … FILEGROWTH command. You should also ensure that the MAXSIZE setting for each file is set appropriately.* You should also ensure that enough physical space exists on the server to accommodate additional database space.

Logs Low on Space

Each database in SQL Server has a transaction log, which is a serial record of all modifications that have occurred in a database as well as the transactions that caused each change. The Logs Low on Space statistic indicates transaction logs that have fallen below a recommended percentage of free space.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Bottleneck Analysis PaneMicrosoft SQL Server Performance Analyst Statistics (DBArtisan) > Space View (DBArtisan - SQL Server Performance Analyst) > Home > Bottleneck Analysis Pane
Metrics	If any log's used space exceeds the Performance Analyst recommended thresholds, then the DBA should take action to ensure that the log does not run out of available free space.
Correction	There are several things a DBA can do to ensure that a database's log does not run out of available free space: *First, most transactional-oriented databases should have their logs assigned to a separate physical drive than the database. This prevents competition for space between the log and the database itself, allows the log to be monitored for space more effectively, and improves performance.If the database is not critical in nature, you can set the truncate log on checkpoint option trunc log on chkpt, which will eliminate any non-active space in the log when a database checkpoint occurs.Critical databases needing higher levels of recovery should have schedules established that regular perform transaction log dumps. Doing so ensures better recovery scenarios as well as a reduced risk of the transaction log running out of space.If a critical transaction log becomes full, it might be impossible to use standard procedures to dump transactions and reclaim space. The dump operation will likely have to incorporate the no log or truncate only options.If a transaction log continuously approaches dangerously low levels of free space, then the DBA should allow the underlying file(s) of the log to automatically grow to meet the demand. This can be accomplished by using the ALTER DATABASE … MODIFY FILE … FILEGROWTH command. You should also ensure that the MAXSIZE setting for each file is set appropriately.The DBA should also be on the lookout for large load or data modification operations that do not make use of prudently timed commit points. A single, large transaction has the ability to overwhelm any transaction log since only non-active space in the transaction log is removed from log dumps or truncation operations.*

SQL Analysis Pane

A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured server down in terms of end-user response times.

Before you can identify problem SQL in your database, you have to ask the question - "What is bad SQL?" What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every SQL Server professional's satisfaction. The table below lists some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

Criteria	Description
Reads	This is often used as the major statistic in terms of identifying good vs. bad SQL, this is a measure of how many disk reads the query caused to satisfy the user's request. While you certainly want to control disk I/O where possible, it is important that you not focus solely on physical I/O as the single benchmark of inefficient SQL. Make no mistake, disk access is slower than memory access and also consumes processing time making the physical to logical transition, but you need to look at the entire I/O picture of a SQL statement, which includes looking at a statements' logical I/O as well.
Writes	These can be caused by DML activity (INSERTs, etc.), but writes can also be indicative of heavy disk sort activity.
CPU	This is how much CPU time the query took to parse, execute, and fetch the data needed to satisfy the query.
Elapsed Time	This is how much time the query took to parse, execute, and fetch the data needed to satisfy the query. It should not include the network time needed to make the round trip from the requesting client workstation to the database server.

There are other criteria that you can examine like sort activity or access plan statistics (that show items like Cartesian joins and the like), but more often than not, these measures are reflected in the criteria listed above.

The SQL Analysis section provides you with a quick overview of the percentage of Reads, Writes, CPU, and Elapsed Time the most expensive SQL statements tracked by Performance Analyst have used. Depending on the page you are on, you might just see information regarding physical I/O activity or DML statements. For example, you might see that a SQL statement has caused 60% of all physical reads on the system, or that a procedure is responsible for 90% of all the CPU usage. To get information regarding the actual SQL text or stored procedure calls, drill down into the Top SQL details views.

Note: SQL Profiling is turned on by default in Performance Analyst so you can automatically collect SQL traffic. If you do not want to collect SQL execution information, use the options inside of Performance Analyst to disable SQL profiling.

Metrics

When you begin to look for inefficient SQL in a database, there are two primary questions you should answer:

- What has been the worst SQL that has historically been run in my database?
- What is the worst SQL that is running right now in my database?

When Correction a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It might be that the query is in an inefficient Transaction SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

There is the possibility that the SQL statement just is not tuned well. To determine that, you can drill down further into the Top SQL view and begin working with the query through EXPLAIN plan analysis and other techniques inside Embarcadero's DBArtisan.

Storage Analysis Pane

While DBAs focus on memory settings and tuning SQL, they frequently forget how dangerous and insidious storage problems can be. This is not a good mindset because storage headaches can play a major role in wrecking an otherwise well-running database.

Storage problems generally take one of two forms:

- The 'hit-the-wall' variety that can bring things to a complete standstill.
- The 'performance vampire' kind that slowly drains the performance of a database over time.

Storage problems can bring the curtain down on a database very quickly, as in the case of a transaction log running out of free space. But storage problems can also silently work behind the scenes to slowly, but surely rob a database of its performance. For example, a hub table in a busy database might be accessed very quickly when an application is first given life, but over time, if it develops a heavy forwarded row problem, it can cause things to run very differently.

The Storage Analysis section of the Performance Analyst Home page displays the Total Used/Total Free Space of all tablespaces in the database. It also provides a count of the number of databases, files, and file groups that exist on the server. For more detailed information on database space, you can use the Performance Analyst Space View (DBArtisan - SQL Server Performance Analyst) details.

Total Used/Total Free Space

These statistics represent the total used and free space available in all databases on the server. Although good to know, a more detailed listing by database is needed to determine where any actual space shortages exist on the server. This information can be viewed in the Performance Analyst Space View (DBArtisan - SQL Server Performance Analyst).

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Storage Analysis Pane
Metrics	If any one database or transaction log begins to approach the Performance Analyst recommended thresholds (and the objects contained within it are dynamic and growing as opposed to static), then the DBA should take action to prevent any future space allocation errors.
Correction	Here are a few things a DBA can do to prevent free space shortages: For Databases: *The DBA can resize the current file(s) used by the database via an ALTER DATABASE … MODIFY FILE command.The DBA can add a new file to the database via the ALTER DATABASE … ADD FILE command.The DBA can modify the file(s) used by the database to automatically grow by using the ALTER DATABASE … MODIFY FILE … FILEGROWTH command. You should also ensure that the MAXSIZE setting for each file is set appropriately.* For Transaction Logs: *First, most transactional-oriented databases should have their logs assigned to a separate physical drive than the database. This prevents competition for space between the log and the database itself, allows the log to be monitored for space more effectively, and improves performance.If the database is not critical in nature, you can set the truncate log on checkpoint option (trunc log on chkpt), which will eliminate any non-active space in the log when a database checkpoint occurs.Critical databases needing higher levels of recovery should have schedules established that regular perform transaction log dumps. Doing so ensures better recovery scenarios as well as a reduced risk of the transaction log running out of space.If a critical transaction log becomes full, it might be impossible to use standard procedures to dump transactions and reclaim space. The dump operation will likely have to incorporate the no log or truncate only options.If a transaction log continuously approaches dangerously low levels of free space, then the DBA should allow the underlying file(s) of the log to automatically grow to meet the demand. This can be accomplished by using the ALTER DATABASE … MODIFY FILE … FILEGROWTH command. You should also ensure that the MAXSIZE setting for each file is set appropriately.* You should also be on the lookout for large load or data modification operations that do not make use of prudently timed commit points. A single, large transaction has the ability to overwhelm any transaction log since only non-active space in the transaction log is removed from log dumps or truncation operations.

Workload Analysis Pane

When experiencing performance slowdowns, some of the first questions you should answer are:

- Who is currently connected to the database?
- What resources are they using?
- What are they currently executing?

The Workload Analysis section of the Performance Analyst Home page provides insight into the leading resource hogs of a server, as well as a count of total active and inactive processes. The Users View (DBArtisan - SQL Server Performance Analyst) page lets you easily get detailed information into what each leading session is currently doing.

Workload Analysis statistics include:

- Active User Processes
- Inactive User Processes

Metrics

If any one session appears to be using more than 50% of a total resource (CPU, memory, etc.), then you should drill down into the session to find out what they are currently executing.

Active User Processes

The Active User Processes statistic represents the total number of active and open threads reported on the server. This number displays the number of processes actively performing work.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Workload Analysis Pane

Inactive User Processes

This metric represents the total number of threads logged on to the server that are idle at the current time.

Note: This statistic is also available on the Home page under Users View (DBArtisan - SQL Server Performance Analyst).

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - SQL Server Performance Analyst)** > Workload Analysis Pane
Metrics	A large number of inactive users could indicate user sessions that have mistakenly been left logged on. Since each user thread consumes a portion of memory on the server, to reduce resource usage, you should sever any sessions not needing a connection.
Correction	Drill down into the Session Waits Tab and check sessions that have many seconds idle and/or that have been logged on for very long periods of time, as indicated by the logon time column. After verifying that a session is no longer necessary, you can KILL it.

Memory View (DBArtisan - SQL Server Performance Analyst)

In addition to a Home page, the Memory category of SQL Server Performance Analyst includes the following tabbed pages:

Buffer Cache Tab Log Cache Tab Procedure Cache Tab

Home

The Home page of the Memory category includes the following sections and statistics:

Key Ratio Analysis Pane Bottleneck Analysis Pane SQL Analysis Pane

Memory Analysis Pane Workload Analysis Pane

Key Ratio Analysis Pane

Database performance analysts typically use one of two methods for examining the performance levels of a database - ratio-based or wait/bottleneck-based. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as very good roll-up mechanisms for busy DBA's to use for at-a-glance performance analysis. Many DBAs have large database farms to contend with and cannot spend time checking detailed wait-based analysis outputs for each and every database they oversee. Succinctly presented performance ratios can assist in such situations by giving DBAs a few solid indicators that can be quickly scanned to see if any database needs immediate attention.

While there are certainly many opinions as to what rules to follow, there are some standards that should always be adhered to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that a DBA will examine come from the master.sysperfinfo performance table. This table maintains a count of various statistics for the server and particular databases since the server was brought up. For servers that are kept up for long periods of time, these values can grow quite large and will impact how a particular ratio that a DBA might be looking at is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each SQL Server's individual 'personality' with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead the DBA to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following memory ratios are used on the Performance Analyst Home page to succinctly communicate the general overall memory performance levels of the monitored database:

Cache Used Procedure Cache Used Used Cache %

Used Procedure Cache %

The following statistics, available in this pane, are duplicates of statistics available on the Key Ratio Analysis Pane of the Home View (DBArtisan - SQL Server Performance Analyst) page:

Buffer Cache Hit Ratio Procedure Plan Hit Ratio Ad Hoc SQL Hit Ratio

Log Cache Hit Ratio

Cache Used

The total MB of space that SQL Server is using in the buffer cache. Each instance of SQL Server has its own buffer cache where it stores recently used data pages to reduce physical I/O. The goal is to make the buffer cache large enough to maximize the ratio of logical reads to physical reads, but not so large that excessive memory swapping starts generating physical I/O to the pagefile.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - SQL Server Performance Analyst)** > Home > Key Ratio Analysis Pane
Metrics	A percentage consistently remaining close to 100% indicates a deficient amount of memory available to SQL Server.
Correction	First, ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server Memory configuration option. Also, consider increasing your SQL Server Min Memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Procedure Cache Used

The total MB of space that SQL Server is using in its SQL/procedure cache. Microsoft has begun to transition from the term "procedure cache" to "SQL cache" to define this area of memory. The reason being that in SQL Server's past, this area was devoted exclusively to holding query plans for stored procedures only.

The SQL Cache (procedure cache) is the part of the SQL Server memory pool that is used to store execution plans for Transact-SQL batches, stored procedures, and triggers. Execution plans record the steps that SQL Server must take to produce the results specified by the Transact-SQL statements contained in the batches, stored procedures, or triggers.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - SQL Server Performance Analyst)** > Home > Key Ratio Analysis Pane

Used Cache %

The Used Buffer Cache Percent statistic represents the total percentage of memory that is in use in the SQL Server buffer cache. Each instance of SQL Server has its own buffer cache where it stores recently used data pages to reduce physical I/O. The goal is to make the buffer cache large enough to maximize the ratio of logical reads to physical reads, but not so large that excessive memory swapping starts generating physical I/O to the pagefile.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - SQL Server Performance Analyst)** > Home > Key Ratio Analysis Pane
Metrics	A percentage used consistently remaining close to 100% indicates a deficient amount of memory available to SQL Server.
Correction	First, ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server Memory configuration option. Also, consider increasing your SQL Server Min Memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Used Procedure Cache %

The total percentage of memory that SQL Server is using in its SQL/procedure cache. Microsoft has begun to transition from the term “procedure cache” to “SQL cache” to define this area of memory. The reason being that in SQL Server's past, this area was devoted exclusively to holding query plans for stored procedures only.

The SQL Cache is the part of the SQL Server memory pool that is used to store execution plans for Transact-SQL batches, stored procedures, and triggers. Execution plans record the steps that SQL Server must take to produce the results specified by the Transact-SQL statements contained in the batches, stored procedures, or triggers.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - SQL Server Performance Analyst)** > Home > Key Ratio Analysis Pane

Bottleneck Analysis Pane

When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. And this is where wait-based or bottleneck analysis comes into play. DBAs use this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance because it helps a DBA track where a database has been spending its time. If lock contention or heavy table scan activity has been dragging down database performance, a DBA can use bottleneck analysis to confirm the actual root cause. Once one or more wait events or other bottlenecks have been pinpointed as possible performance vampires, the DBA can drill down and oftentimes discover a fair amount of detail about which sessions and objects are causing the problem.

Memory bottlenecks can cause performance degradation in an otherwise well-running database. Typically, these bottlenecks center around the SQL Server latches, which are lightweight locks used to protect certain resources in memory. To help you identify such problems, the following statistics are presented on the Performance Analyst Memory home page:

Latch Waits/Sec Total Latch Wait Time Top Latch Waits

Latch Waits/Sec

This metric represents the number of latches per second that could not be satisfied immediately by SQL Server. Latches are lightweight, mini-locks that are used to protect actions and resources used inside transactions.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Unless accompanied by long wait times, latch waits should not pose too much of a performance problem in normal SQL Server installations. If you have a high number of latch waits per second, then it is likely that SQL Server is also sporting a low buffer cache hit ratio and is being forced to perform extra physical I/O.

Total Latch Wait Time

Latches are lightweight, mini-locks that are used to protect actions and resources used inside transactions. The total latch wait time statistic represents the sum total of all latch wait time that has occurred on the system.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Unless accompanied by long wait times, latch waits should not pose too much of a performance problem in normal SQL Server installations. If you have a high number of latch waits per second, then it is likely that SQL Server is also sporting a low buffer cache hit ratio and is being forced to perform extra physical I/O.

Top Latch Waits

The Top Latch Waits graph depicts the top latch waits by wait time.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Latch waits rarely impact performance, however seeing latch waits with high wait time might indicate an area that needs further investigation.
Correction	Wait events can be hard to interpret. If you see a particular event that has caused a lot of wait time, you can review the information in this Microsoft Knowledge Base Article to help understand the cause and potential remedy: http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455

SQL Analysis Pane

When SQL Server executes a set of SQL (ad hoc, procedure, trigger, etc.), it places the code into its procedure cache. The SQL Analysis section of the Performance Analyst Memory page provides a summary of the total memory pages currently in use by the various SQL types (procedure, ad hoc, etc.)

The SQL Analysis section provides you with a quick overview of the percentage of Reads, Writes, CPU, and Elapsed Time the most expensive SQL statements tracked by Performance Analyst have used. Depending on the page you are on, you might just see information regarding physical I/O activity or DML statements. For example, you might see that a SQL statement has caused 60% of all physical reads on the system, or that a procedure is responsible for 90% of all the CPU usage. To get information regarding the actual SQL text or stored procedure calls, drill down into the Top SQL details views.

Note: SQL Profiling is turned on by default in Performance Analyst so you can automatically collect SQL traffic. If you do not want to collect SQL execution information, use the options inside of Performance Analyst to disable SQL profiling.

Memory Analysis Pane

SQL Server uses various memory regions to accomplish various tasks (like retain frequently used data, procedures, etc.) and improve overall performance. At start up, SQL Server is allocated memory that will fall between the range of two configurable parameters: Min Memory and Max Memory. These statistics show you the amount of SQL Server memory allocated at present. In a normal SQL Server system, this number fluctuates as SQL Server uses more memory and releases unused memory back to the operating system. SQL Server also works in conjunction with the Windows Memory Manager, deallocating space when it detects that the operating system is unable to satisfy memory requests of other operating system processes. Depending on the version of SQL Server you are using, expect to deal with different memory regions. The table below describes some of the SQL Server memory regions:

Memory Scan	Description
Buffer Cache	The buffer cache is a memory pool of buffer pages into which SQL Server reads data pages for quick access.
Procedure Cache	Beginning with SQL Server version 7.0, you can place other SQL-related code objects in memory for reuse. Once SQL Server parses through and places a set of SQL of program code in memory, response time can be increased for subsequent calls to the same set of SQL or SQL code objects.
Connection Memory	The total amount of dynamic memory the server is using for maintaining connections.
Granted Workspace Memory	The total amount of memory granted to executing processes such as sort, hash, bulk copy operations, and so forth.
Lock Memory	The total amount of dynamic memory the server is using for locks.
Optimizer Memory	The total amount of dynamic memory the server is using for query optimization.
SQL Cache Memory	The total amount of memory used to retrieve stored SQL caches.

Workload Analysis Pane

It is not uncommon for one or two users to cause the majority of runtime problems that plague a server. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation.

If your database server does not have an overabundance of memory, then you should periodically check to see who your heavy memory users are along with the total percentage of memory each takes up. If you see one or two users who have more than 25-50% of the total memory usage, then you should further investigate the sessions to see what activities they are performing.

Buffer Cache Tab

The Buffer Cache Hit Ratio tab includes the following sections:

Top Twenty Buffer Cache Objects Total vs. Target Memory Cached Pages By Database

Page Life Expectancy

The following statistic, available on this page, duplicates a statistic on the Key Ratio Analysis Pane of the Home View (DBArtisan - SQL Server Performance Analyst) page:

- Buffer Cache Hit Ratio

Top Twenty Buffer Cache Objects

SQL Server uses various memory regions to accomplish various tasks (like retain frequently used data, procedures, etc.) and improve overall performance. At start up, SQL Server is allocated memory that falls between the range of two configurable parameters: Min Memory and Max Memory. The Top Twenty Buffer Cache Objects chart shows you the amount of SQL Server memory allocated at present. In a normal SQL Server system, this number fluctuates as SQL Server uses more memory and releases unused memory back to the operating system. SQL Server also works in conjunction with the Windows Memory Manager, deallocating space when it detects that the operating system is unable to satisfy memory requests of other operating system processes. Depending on the version of SQL Server you are using, there are several memory regions you should expect to deal with. The table below describes the SQL Server memory regions:

Memory Scan	Description
Default Buffer Cache	The buffer cache is a memory pool of buffer pages into which SQL Server reads data pages for quick access.
Procedure/SQL Cache	Beginning with SQL Server version 7.0, you can place other SQL-related code objects in memory for reuse. Once SQL Server parses through and places a set of SQL of program code in memory, response time can be increased for subsequent calls to the same set of SQL or SQL code objects.
Connection Memory	The total amount of dynamic memory the server is using for maintaining connections.
Lock Memory	The total amount of dynamic memory the server is using for locks.
Optimizer Memory	The total amount of dynamic memory the server is using for query optimization.

The Active and Free Page chart in Performance Analyst displays a count of active memory pages (those being used by SQL Server) and free pages (pages not being used).

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - SQL Server Performance Analyst)** > Buffer Cache Tab
Metrics	Seeing consistent high volumes of free pages might indicate that SQL Server is using more memory than necessary.

Total vs. Target Memory

Total memory is the amount of memory that's currently in use by SQL server. Target memory is the amount of memory that could be used. SQL server dynamically adjusts its memory allocations depending either on workload up to the maximum memory configuration set by the DBA or the amount it thinks it should give other (non-SQL) server operations.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - SQL Server Performance Analyst)** > Buffer Cache Tab
Metrics	When total memory approaches or reaches the maximum amount of target (defined) memory, then SQL server is experiencing memory pressure. You can try adding more RAM to the machine or adjusting memory ceiling limits imposed at the SQL server configuration level.

Cached Pages By Database

The Cached Pages By Database metric gives the amount of pages in memory devoted to database content compared to all other content.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - SQL Server Performance Analyst)** > Buffer Cache Tab
Metrics	If you see that a large percentage of memory is devoted to database pages, the server is database page centric and not procedure plan centric. This can indicate that too many table scans are taking place, which might be flooding the buffer cache with infrequently used pages. You can determine if this is the case by looking at the page life expectancy metric. Page life expectancy readings of 300 seconds or less can indicate excessive table scan activity.

Page Life Expectancy

The page life expectancy metric indicates how long SQL server believes a page will stay in the buffer cache. Pages served from memory result in much shorter response times than pages read from disk into the cache. It's best if frequently used data is pinned in the buffer cache.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - SQL Server Performance Analyst)** > Buffer Cache Tab
Metrics	Readings of 300 seconds or less frequently mean that too many table scans are occurring and flooding the buffer cache with pages that are used rarely. Check the I/O access patterns on the I/O home page or the User I/O drill-down to confirm if this is the case.

Procedure Cache Tab

The Procedure Cache tab includes the following sections:

Procedure Cache Hit Ratio Procedure/SQL Cache Detail Procedure/SQL Cache Summary

Procedure Cache Hit Ratio

The SQL Server procedure cache holds the execution plans for all Transact-SQL statements currently executing in the server. When a user executes a Transact-SQL statement, SQL Server looks in the procedure cache for a query plan to use.

The Procedure Cache Hit Ratio statistic is the percentage of query plan requests generated by stored procedures that are found in the procedure cache area. The greater the percentage of times that a statement's plan and definition can be referenced in memory, the better the procedure execution time.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - SQL Server Performance Analyst)** > Procedure Cache Tab
Metrics	A high procedure cache hit rate is a desirable thing. You should strive for a hit ratio between 95-100%, with 95% being a good performance benchmark for code reference. Note that when a database first starts, the procedure cache hit rate will not be at an optimal level because all code being used will be relatively new, and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the procedure cache hit rate does not increase to desirable levels, you should look into the possibility of increasing the amount of memory allocated to the cache.
Correction	First, ensure SQL Server is configured to use as much physical memory as possible by checking the max server memory configuration option. Also, consider increasing your SQL Server min memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Procedure/SQL Cache Summary

The SQL Server procedure/SQL cache holds often-referenced code object definitions and ad-hoc SQL. The Procedure/SQL Cache Summary lists each type of code object, the amount of memory each code type (trigger, ad-hoc SQL, etc.) consumes, and a total count of each type of code object.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - SQL Server Performance Analyst)** > Procedure Cache Tab
Metrics	Seeing a lot of memory devoted to ad-hoc SQL plans might indicate an environment where sessions are submitting a lot of SQL requests. Many SQL Server DBAs like to control code submissions through stored procedures.

Procedure/SQL Cache Detail

Beginning with SQL Server version 7.0, other SQL-related code objects might be placed in memory for reuse. Once SQL Server parses through and places a set of SQL of program code in memory, response time can be increased for subsequent calls to the same set of SQL or SQL code object. The Procedure/SQL Cache Details section displays the top 1000 objects in the procedure/SQL cache. The table below describes the information available in the Procedure/SQL Cache Details section on the Procedure Cache tab of the Memory Detail:

Information	Description
Object Type	The type of object (ad hoc SQL, etc.)
Cache Type	The category of SQL code inside the cache (executable plan, etc.)
Object ID	The numeric ID of the object.
Object Name	The name of the object (if applicable).
Database ID	The numeric ID of the database.
Database	The database where the code originated.
User ID	The numeric ID of the user.
User	The owner of the object or code.
Use Count	The number of uses for the object.
SQL Bytes	The amount of SQL bytes used by the code object.
Size (KB)	The size used by the object in kilobytes.
SQL	The actual SQL statement or code being executed.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - SQL Server Performance Analyst)** > Procedure Cache Tab
Metrics	Seeing many objects of the ad hoc SQL type might indicate an environment where sessions are submitting a lot of SQL requests. Many SQL Server DBAs like to control code submissions through stored procedures.

Log Cache Tab

The Log Cache tab includes the following sections:

Database Log Cache Ratios Log Cache Details

The following statistic, available on this page, duplicates a statistic on the Key Ratio Analysis Pane of the Home View (DBArtisan - SQL Server Performance Analyst) page:

- Log Cache Hit Ratio

Database Log Cache Ratios

The Log Cache Hit Ratio statistic represents the percentage of log cache reads satisfied from the log cache. The database log cache ratios display breaks down log cache hit ratios by database. Note that some databases with inactive logs will not display a log cache hit ratio.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - SQL Server Performance Analyst)** > Log Cache Tab
Correction	A low percentage on this statistic is not necessarily a bad sign, as it is possible that the information needed from the log will not be readily available in memory.

Log Cache Details

Before ever writing transactions to disk, the log manager of SQL Server formats everything in memory. This area of memory is known as the log cache. The log writer of SQL Server moves through the log caches when transactions are committed (as well as other events) and flushes each cache out to disk. SQL Server also reads from the cache when log records are needed. The table below describes the information available in the Log Cache Details section on the Log Cache tab of the Memory Detail:

Information	Description
Database	The name of the SQL Server database.
Log Cache Reads	The number of reads from the log cache.
Log Flushes	The number of times data was flushed from the log to disk.
Log Flush Waits	The number of times the log had to wait before flushing data to disk.
Log Flush Wait Time	The amount of time the log waited before flushing data to disk in milliseconds.
Log Growths	The number of times the log had to physically grow in size to meet the need for more space.
Log Shrinks	The number of times the log contracted in physical size.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - SQL Server Performance Analyst)** > Log Cache Tab
Metrics	Seeing high amounts of wait time for log flushes could indicate a bottleneck at the disk level. A log that shows high number of growths likely indicates an undersized log. While automatic growth can alleviate out-of-space conditions, many growth operations can slow down overall operations. It is better to have a properly sized transaction log that allows SQL Server to continually enlarge it in size when needed.
Correction	Consider relocating logs showing high amounts of wait time to faster disks. For logs showing high numbers of growths, permanently enlarge the log(s) via an ALTER DATABASE command that will resize the log files.

I/O View (DBArtisan - SQL Server Performance Analyst)

In addition to a Home page, the I/O category of SQL Server Performance Analyst includes the following tabbed pages:

Database I/O Tab File Tab System I/O Tab

User I/O Tab

Home

The Home page of the I/O category includes statistics in the following categories:

Bottleneck Analysis Pane I/O Access Patterns Key System Analysis Pane

SQL Analysis Pane Workload Analysis Pane

Key System Analysis Pane

When complaints begin to surface about your database's performance, oftentimes the root cause can be traced to one or more issues with I/O. The thing to keep in mind when you begin to monitor the I/O of your database is that you are actually reviewing the success of your physical design model.

All the physical storage characteristics and placements, the table and index designs, and the speed that it all works are on display when I/O is monitored. Because a database's main index of performance is measured by how fast I/O needs are satisfied, it is your responsibility to quickly interrogate SQL Server to determine if a reported database slowdown is I/O related.

The following key I/O values are reported in this section:

Checkpoint Pages Lazy Writes Log Cache Reads

Log Flushes Workload Analysis Pane Page Writes

Page Reads Read Ahead Pages

Page Reads

This metric represents that number of physical database page reads that are issued per second by SQL Server.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > Home > Key System Analysis Pane
Metrics	Page reads are to be expected, especially after initial server start up. This is because SQL Server must first satisfy requests for data and meta-data by reading information in from physical disk. Numerous page reads can also be expected if the physical server does not contain an adequate amount of memory to hold repetitively requested blocks of information. No hard-and-fast rules exist for how many page reads per second is too much. You can cross-reference this statistic with the physical server disk statistics to see if physical page reads and accompanying physical disk I/O is approaching the server's premium capacity levels. And because logical I/O is always many times faster than physical I/O, you should also evaluate the buffer cache hit ratio to determine overall memory vs. physical read efficiency.
Correction	If you find that the server is becoming overworked from a physical I/O standpoint, there are several courses of action you can take: *Examine index usage to ensure that unnecessary table scans are not occurring.Check the physical database design to see if table objects have been over-normalized.Ensure that SQL Server is configured to use sufficient amounts of memory. Examine the min server memory (MB) and max server memory (MB) parameters to see if SQL Server is constrained on either end of the memory spectrum.Check for large pinned table objects that could be using excessive amounts of space in the buffer cache.Last, but not least, investigate the possibility of adding more RAM to the physical server.*

Page Writes

The Page Writes statistic represents the number of physical database page writes issued by SQL Server. Page Writes take place during operations such as checkpoints, lazywriter writes, index creations, and BCP routines.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > Home > Key System Analysis Pane
Metrics	Page Writes can give you an idea of overall physical write activity, however there are a number of statistics that pertain specifically to certain write activities like checkpoints, etc., that you can examine to determine the amount of physical writes caused by distinct SQL Server processes. With respect to performance, response times experienced by SQL Server users are normally not impacted by write operations unless the writes are synchronous in nature. These are typically BCPs, database recovery operations, and index creations.

Read Ahead Pages

The Read Ahead Pages statistic represents the number of physical database page read in anticipation of use by SQL Server. If SQL Server senses that database pages are being read in a sequential manner, it will institute a pre-fetch mechanism that moves pages into the buffer cache before they are actually needed by a process.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > Home > Key System Analysis Pane
Metrics	If data is accessed sequentially (for example, through the use of a clustered index), the read ahead mechanism of SQL Server can increase performance by needed database pages already in the buffer cache before they are actually requested. However, because the read ahead mechanism is typically triggered by full table or index range scans, if the read ahead pages are actually required to satisfy a user's query, then performance might actually suffer. In these cases, the judicious use of indexes is a better route to take.

Log Flushes

The Log Flushes statistic represents the total number of log pages for all databases written to disk by the log writer process. A log flush occurs when SQL Server writes all changes from the database's log cache out to the database's log files on disk.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > Home > Key System Analysis Pane
Metrics	Increasing numbers observed for log flushes should not cause concern unless the I/O subsystem of the server appears overwhelmed. In addition, to minimize I/O contention between a database and its accompanying log, it is wise to place database files and log files on separate disks.
Correction	If you have placed a very active database on the same physical file as its log, you can look into moving the log to a separate physical device by adding new log files to a new drive and subsequently removing the old log files when they are not being used.

Log Cache Reads

The Log Cache Reads statistic represents the reads performed per second through the log manager cache. Before ever writing transactions to disk, the log manager of SQL Server formats them in memory. This area of memory is known as the log cache and only contains log records. The log writer of SQL Server moves through the log caches when transactions are committed (as well as other events) and flushes each cache out to disk.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > Home > Key System Analysis Pane

Checkpoint Pages

The Checkpoint Pages statistic represents the number of pages flushed to disk per second by a checkpoint or other operation that require all dirty (modified) pages to be flushed.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > Home > Key System Analysis Pane
Metrics	Checkpoint operations are used by SQL Server to minimize the amount of work the server must perform when databases are recovered during system startup. Checkpoints periodically write out modified pages that are found in the buffer cache to disk. After, SQL Server records the operation in the log to signify that the operation succeeded. Checkpoints can be explicitly performed by a database owner issuing the checkpoint command. SQL Server also performs checkpoints automatically for databases that have the trunc log on chkpt option set. Large SQL Servers have the potential to generate lots of checkpoint write operations. Although SQL Server will do what it can to minimize checkpoint activity, you can also set the recovery interval server parameter to influence how often checkpoints should run.
Correction	If you believe excessive checkpoint activity is occurring, you can take the following steps: *Set the recovery interval server parameter to a larger value with sp_configure.Restart SQL Server so the change will take affect or use the RECONFIGURE option to make the change immediately.*

Lazy Writes

The lazy writer is a SQL Server system process that flushes out batches of dirty and aged buffers (buffers that contain changes that must be written back to disk before the buffer can be reused for a different page) and make them available to user processes. The Lazy Writes statistic represents the number of buffers written by the buffer manager's lazy writer process.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > Home > Key System Analysis Pane
Metrics	High values might indicate a thrashing situation with data scanned into the buffer cache (but referenced seldom) and then immediately moved back out.

Bottleneck Analysis Pane

When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. And this is where wait-based or bottleneck analysis comes into play. DBAs use this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance: it helps you track where a database has been spending its time. The Performance Analyst I/O home page identifies the top I/O related waits that might be currently causing performance issues. Also included are the current hottest database/log files and a summary of I/O activity.

Bottleneck Analysis includes the following information:

I/O Errors Hottest Database/Log Files Total I/O

Top I/O Waits

The following statistic, available on this pane, duplicates a statistic on the Key Ratio Analysis Pane of the Home View (DBArtisan - SQL Server Performance Analyst) page:

  - I/O Busy

I/O Errors

I/O Error Rate reflects total number of I/O errors (errors during read and write operations) encountered by the server since the last refresh inside Performance Analyst Center. I/O Error Rate is a percentage based on Total I/O (the sum the physical reads and writes).

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	You should observe few, if any errors.
Correction	If you notice any errors, you should check the SQL Server error log for details.

Hottest Database/Log Files

The Hottest Database/Log Files display presents the most active database files as indicated by physical I/O activity (reads and writes).

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Understanding the storage-level hot spots of a database is beneficial for a couple of reasons. *First, you can get a feel for overworked physical disks by viewing I/O statistics at the tablespace and datafile levels. If a particular disk or set of disks is under too much strain, you attempt to relocate the databases to other less-used devices or create new file groups on different disks and move hot objects to them (assuming, of course, you have extra disks available).Second, if you have used standard DBA practice and placed indexes in their own file group, you can view the I/O statistics for that file group and see if the indexes are actually being used.*
Correction	Some areas to consider when viewing the hottest database files display are: *Seeing a lot of activity in the master database and datafiles might indicate a lot of recursive calls.The tempdb database (devoted to sort activity) shows higher volumes of physical I/O which could indicate a problem with excessive disk sorts.You might want to quickly review all the physical I/O for each drive/file system and get a idea of the overworked disks on your server. If you have under-utilized disk drives (with their own controllers), you should consider relocating some databases or filegroups that exhibit high I/O characteristics to those drives.To minimize contention at the physical disk level, and improve performance overall, it is generally recommended that a SQL Server database have its database and log files physically separated onto different drives.*

Total I/O

The Total I/O statistic represents the total number of physical reads and writes.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane

Top I/O Waits

The Top I/O Waits statistics rank the top I/O related waits by wait time.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Correction	Wait events can be hard to interpret at times. If you see a particular event that has caused a lot of wait time, you can review the information in this link (Microsoft Knowledge Base Article - 244455) to help understand the cause and potential remedy: http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455

SQL Analysis Pane

A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured server down in terms of end-user response times.

Before you can identify problem SQL in your database, you have to ask the question - "What is bad SQL?" What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every SQL Server professional's satisfaction. The table below lists some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

Criteria	Description
Reads (Physical)	This is often used as the major statistic in terms of identifying good vs. bad SQL, this is a measure of how many disk reads the query caused to satisfy the user's request. While you certainly want to control disk I/O where possible, it is important that you not focus solely on physical I/O as the single benchmark of inefficient SQL. Make no mistake, disk access is slower than memory access and also consumes processing time making the physical to logical transition, but you need to look at the entire I/O picture of a SQL statement, which includes looking at a statements' logical I/O as well.
Writes	These can be caused by DML activity (INSERTs, etc.), but writes can also be indicative of heavy disk sort activity.

There are other criteria that you can examine like sort activity or access plan statistics (that show items like Cartesian joins and the like), but more often than not, these measures are reflected in the criteria listed above.

The SQL Analysis section provides you with a quick overview of the percentage of Reads, Writes, CPU, and Elapsed Time the most expensive SQL statements tracked by Performance Analyst have used. Depending on the page you are on, you might just see information regarding physical I/O activity or DML statements. For example, you might see that a SQL statement has caused 60% of all physical reads on the system, or that a procedure is responsible for 90% of all the CPU usage. To get information regarding the actual SQL text or stored procedure calls, drill down into the Top SQL details views.

Note: SQL Profiling is turned on by default in Performance Analyst so you can automatically collect SQL traffic. If you do not want to collect SQL execution information, use the options inside of Performance Analyst to disable SQL profiling.

Metrics

When you begin to look for inefficient SQL in a database, there are two primary questions you should answer:

- What has been the worst SQL that has historically been run in my database?
- What is the worst SQL that is running right now in my database?

When Correction a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It might be that the query is in an inefficient Transaction SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

There is the possibility that the SQL statement just is not tuned well. To determine that, you can drill down further into the Performance Analyst Top SQL view and begin working with the query through EXPLAIN plan analysis and other techniques inside Embarcadero's DBArtisan.

I/O Access Patterns

There are many different routes that SQL Server can take to access various objects contained in a database. The I/O Access Patterns display summarizes the methods SQL Server is using to satisfy end-user requests for data.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > Home > SQL Analysis Pane

Metrics

Depending on the version of SQL Server you are using, expect to deal with different access styles. The table below describes some of the SQL Server access styles:

Access Style	Description
Full Scans	Full Scans are the total number of full table or index scans. Full scans occur if a table is inadequately indexed or if SQL Server needs to access all rows in a table or index to satisfy a query. Other operations that can cause full scans include UPDATE STATISTICS calls. Avoid unnecessary scans on large tables is something to avoid, and can be a signal to you as a DBA to investigate the use of more indexes and to review SQL access through EXPLAIN plans. Small table scans are efficient because SQL Server can often cache the entire table in a single I/O operation. Large numbers of index scans are normally desirable too, since it typically indicates the fastest possible resolution to data access requests.
Range Scans	Range Scans are the total number of qualified range scans through indexes. Large numbers of index scans are normally desirable, since it typically indicates the fastest possible resolution to data access requests is being taken.
Probe Scans	Probe scans are used in SQL Server to directly find rows in an index or base table. Large numbers of probe scans are normally desirable, since they typically indicate the fastest possible resolution to data access requests is being taken.
Index Searches	Index searches are normally used to start range scans, for single index record fetches and also might be used to reposition an index.
Forward Record Fetches	The Forwarded Record Fetches statistic represents the total number of records fetched by reading forwarded record pointers. SQL Server will move rows in a table under certain conditions. One situation might arise when you update a row in a table that has a variable-length column to a larger size that will no longer fit on its original page. Another situation would be if SQL Server moves a row when the clustered index column changes. When SQL Server creates a forwarding pointer, it remains in place unless one of two things happens. The first is when a row shrinks enough to move back to its original location. The second is when the entire database shrinks. When a database file shrinks, SQL Server will reassign the row identifiers, which are used as the row locators, so the shrink process never generates forwarded rows. Forwarded records can reduce performance at times because additional I/O is involved to first obtain the record pointer to the relocated row, and then the row itself is read.

Correction

Actions to take after observing the ways SQL Server is accessing data depend on the actual situation you witness. If consistent numbers are present for Forward Record Fetches, you might want to examine your databases to see which tables have forwarded records. You can easily do this by utilizing the Embarcadero Space Analyst to view forwarded record data for tables.

If you do not have Space Analyst, then to see the total count of forwarded records in a table, you can enable trace flag 2509, and then execute the DBCC CHECKTABLE command. The output should display the number of forwarded records in that table. Tables with many forwarded records could be candidates for table reorganization.

Here are some methods you can use to avoid unnecessary full scans:

- Try not to use SQL statements that include the NOT IN, NOT LIKE, <>, IS NULL operators since they typically suppress the use of indexes.

- When referencing concatenated indexes with queries, be sure the leading column in the index is used. If it is not, the index will not be used at all.
- Avoid using functions in WHERE predicates.

Workload Analysis Pane

It is not uncommon for one or two users to cause the majority of runtime problems plaguing a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. The Top I/O Hogs display showcases the top sessions that have caused the most physical I/O on the system (both reads and writes).

Metrics

If you see one or two users who have caused more than 25-50% of total I/O, then you should drill down further into the sessions to see what activities they are performing.

System I/O Tab

The System I/O tab of the I/O Detail includes the following sections:

AWE I/O I/O Activity Physical I/O

Space I/O

Physical I/O

SQL Server performs many system-related I/O functions to keep data moving into and out of the server. The System I/O tab of the I/O Detail displays statistics that track various system-related I/O operations. The Physical I/O section details I/O tasks that involve physical disk activity. The table below describes the information available in the Physical I/O section:

Information	Description
Page Reads	The number of physical database page reads that are issued per second by SQL Server. It is collective in nature, meaning that it represents the total page reads per second across all databases that exist on the target SQL Server.
Page Writes	The number of physical database page writes issued by SQL Server. Page Writes take place during operations such as checkpoints, lazy writer writes, index creations, and BCP routines.
Checkpoint Pages	The number of pages flushed to disk per second by a checkpoint or other operation that require all dirty (modified) pages to be flushed.
Lazy Writes	The number of buffers written by the buffer manager's lazy writer. The lazy writer is a system process that flushes out batches of dirty and aged buffers and makes them available to SQL Server processes. The lazy writer eliminates the need to perform frequent checkpoints in order to create available buffers.
Read Ahead Pages	The number of physical database pages read in anticipation of use by SQL Server. If SQL Server senses that database pages are being read in a sequential manner, it institutes a pre-fetch mechanism that moves pages into the buffer cache before they are actually needed by a process.
Page Lookups	The number of requests to find a page in the buffer pool.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > System I/O Tab
Metrics	Page reads are to be expected, especially after initial server start up. SQL Server must first satisfy requests for data and metadata by reading information in from physical disk. Numerous page reads can also be expected if the physical server does not contain an adequate amount of memory to hold repetitively requested blocks of information. No hard and fast rules exist for how many page reads per second is too much. You can cross reference this statistic with the physical server disk statistics to see if physical page reads and accompanying physical disk I/O is approaching the server's premium capacity levels. And because logical I/O is always many times faster than physical I/O, you should also evaluate the buffer cache hit ratio to determine overall memory vs. physical read efficiency. Page Writes can give you an idea of overall physical write activity. There are a number of statistics, however, that pertain specifically to certain write activities like checkpoints, that can be examined to determine the amount of physical writes caused by distinct SQL Server processes. With respect to performance, response times experienced by SQL Server users are normally not impacted by write operations unless the writes are synchronous in nature. These are typically BCPs, database recovery operations, and index creations. With respect to read ahead pages, If data is accessed sequentially (for example, through the use of a clustered index), the read ahead mechanism of SQL Server can increase performance by needed database pages already in the buffer cache before they are actually requested. However, because the read ahead mechanism is typically triggered by full table or index range scans, if the read ahead pages are actually required to satisfy a user's query, performance can actually suffer. In these cases, the judicious use of indexes is a better route to take. Checkpoint operations are used by SQL Server to minimize the amount of work the server must perform when databases are recovered during system startup. Checkpoints periodically write out modified pages that are found in the buffer cache to disk. Afterward, SQL Server records the operation in the log to signify that the operation succeeded. A database owner issuing the checkpoint command can explicitly perform checkpoints. SQL Server also performs checkpoints automatically for databases that have the trunc log on chkpt option set. Large SQL Servers have the potential to generate lots of checkpoint write operations. Although SQL Server minimizes checkpoint activity, you can also set the recovery interval server parameter to influence how often checkpoints can run.
Correction	If you find that the server is becoming overworked from a physical I/O standpoint, here are a few things you can do: *Examine index usage to ensure that unnecessary table scans are not occurring.Check the physical database design to see if table objects have been over-normalized.Ensure that SQL Server is configured to use sufficient amounts of memory. Examine the min server memory (MB) and max server memory (MB) parameters to see if SQL Server is constrained on either end of the memory spectrum.Check for large pinned table objects that could be using excessive amounts of space in the buffer cache.Last, but not least, investigate the possibility of adding more RAM to the physical server.* If you believe excessive checkpoint activity is occurring, you can take the following steps: *Set the recovery interval server parameter to a larger value with sp_configure.Restart SQL Server or use the RECONFIGURE option to make the change immediately.*

Space I/O

SQL Server performs many system-related I/O functions to keep data moving into and out of the server. The Space I/O section of the I/O Detail details space-related I/O operations. The table below describes the information available in the Space I/O section:

Information	Description
Extents Allocated	The number of space extents that SQL Server allocated. Rapidly increasing numbers for these statistics indicates that SQL Server is receiving large volumes of incoming data and is allocating space to make room.
Extent Deallocations	This indicates that SQL Server is reclaiming space from database objects due to shrinking database volumes.
Freespace Page Fetches	The number of pages by free space scans used to satisfy requests to insert record fragments.
Freespace Scans	The number of scans performed by SQL Server to locate free space for an incoming record.
Mixed Page Allocations	The number of pages allocated from mixed extents. These are used for storing the first eight pages that are allocated to an index or table.
Page Deallocations	This indicates that SQL Server is reclaiming space from database objects due to shrinking database volumes.
Page Splits	When data is inserted or updated in a table, SQL Server might reorganize the storage of the data in the table's index pages. When an index page becomes full, but a DML operation demands room on that page, SQL Server moves about half the rows to a new page to accommodate the request. This reorganization is known as a page split. Performance for DML actions can be impaired from page split operations. In addition, more index pages can make for longer index scan times.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > System I/O Tab
Metrics	Increasing numbers for extent and page allocation, and freespace operations likely indicates aggressive volumes of data being inserted or modified in SQL Server. Page splits cause additional overhead in the form of CPU usage and I/O. Observing large numbers of page splits can signal a resource bottleneck in your server.
Correction	To avoid page splits, you can look into tuning the FILLFACTOR property of an index, which controls the percentage of the index page that is filled during creation. The default, 100, tells SQL Server to completely fill each page, whereas lower numbers tell SQL Server to leave room for additional index rows or updates to existing rows.

I/O Activity

SQL Server performs many system-related I/O functions to keep data moving into and out of the server. The I/O Activity section of the I/O Detail details maintenance-related I/O operations. The table below describes the information available in the I/O Activity section

Information	Description
DBCC Logical Scans	The number of logical read scan bytes per second caused by DBCC operations.
Bulk Copy Rows	The number of rows copied either into or out of the database via the BCP utility.
Bulk Copy Throughput	The amount of data (in KB) copied via BCP operations.
Transactions	The number of transactions that have occurred within the database.
Log Flushes	The number of log flushes for the server.
Backup/Restore T-Put	Defines the read/write throughput for backup and restore operations.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > System I/O Tab

AWE I/O

SQL Server performs many system-related I/O functions to keep data moving into and out of the server. The AWE I/O section of the I/O Detail details I/O relating to Address Windowing Extensions (the SQL Server mechanism for supporting very large amounts of RAM). The table below describes the information available in the AWE I/O section:

Information	Description
Lookup Maps	The number of times that a database page was requested by the server, found in the buffer pool, and mapped. When it is mapped, it is made a part of the server's virtual address space.
Stolen Maps	The number of times that a buffer was taken from the free list and mapped.
Unmap Cells	The number of calls to unmap buffers. When a buffer is unmapped, it is excluded from the virtual server address space. One or more buffers might be unmapped on each call.
Unmap Pages	The number of SQL Server buffers that are unmapped.
Write Maps	The number of times that it is necessary to map in a dirty buffer so it can be written to disk.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > System I/O Tab

User I/O Tab

The User I/O tab of the I/O Detail includes the following sections:

Lock Activity SQL Activity Temporary Object Activity

User Object Activity

User Object Activity

The User Object Activity section of the I/O Detail displays statistics that track various user-related I/O operations. The I/O function along with its counter value is presented. The User Object Activity section details performance statistics that reflect how SQL Server is performing object access operations. The table below describes the information available in the User Object Activity section:

Information	Description
Forwarded Records	The number of records per second fetched through forwarded record pointers. At times forwarded records can reduce performance because additional I/O is involved to first obtain the record pointer to the relocated row, and then the row itself is read.
Full Scans	Full scans of moderately sized indexes or tables are generally okay. SQL Server can scan and cache a small table much faster than using its index to navigate to any requested data. Full, unrestricted, large table scans, however, are typically not good and degrade overall system performance and response time.
Index Searches	The total number of index searches per second. Index searches are normally used to start range scans, for single index record fetches and can be used to reposition an index.
Probe Scans	The total number of probe scans per second. Probe scans are used in SQL Server to directly find rows in an index or base table.
Range Scans	The total number of qualified range scans through indexes per second.
Skipped Ghosted Records	The number of ghosted records per second skipped during scans.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > User I/O Tab
Metrics	Full scans occur if a table is inadequately indexed or if SQL Server truly needs to access all rows in a table or index to satisfy a query. UPDATE STATISTICS calls can also cause full scans. Unnecessary scans on large tables is something to avoid, and can be a signal to you as a DBA to investigate the use of more indexes and to review SQL access through EXPLAIN plans. Small table scans are actually a good thing because SQL Server can often cache the entire table in a single I/O operation. Large numbers of index scans are normally desirable too, because this typically indicates the fastest possible resolution to data access requests. When SQL Server creates a forwarding pointer, it remains in place unless one of two things happens. The first is when a row shrinks enough to move back to its original location. The second is when the entire database shrinks. When a database file shrinks, SQL Server reassigns the row identifiers, which are used as the row locators, so the shrink process never generates forwarded rows. Forwarded records can reduce performance at times because additional I/O is involved to first obtain the record pointer to the relocated row, and then the row itself is read. Large numbers of index searches and probe scans are normally desirable because they typically indicate the fastest possible resolution to data access requests is being taken.
Correction	Here are some methods you can use to avoid unnecessary large table scans: *Try not to use SQL statements that include the NOT IN, NOT LIKE, <>, IS NULL operators because they typically suppress the use of indexes.When referencing concatenated indexes with queries, be sure the leading column in the index is used. If it is not, the index will not be used at all.Avoid using functions in WHERE predicates.* If consistent numbers are present for Forward Record Fetches, examine your databases to see which tables have forwarded records. This can easily be done with the Embarcadero Space Analyst component. If you do not have Space Analyst, then to see the total count of forwarded records in a table, enable trace flag 2509, and then execute the DBCC CHECKTABLE command. The output should display the number of forwarded records in that table. Tables with many forwarded records could be candidates for table reorganization.

SQL Activity

The SQL Activity section of the User I/O tab of the I/O Detail displays statistics that track various user-related I/O operations. The I/O function along with its counter value is presented. The SQL Activity section details performance statistics that reflect SQL I/O-related operations. The table below describes the information available in the SQL Activity section:

Information	Description
Auto-Param Attempts	Auto-parameterization occurs when an instance of SQL Server attempts to reuse a cached plan for a previously executed query that is similar to, but not the same as, the current query. The Auto-param Attempts statistic shows the number of auto-parameterization attempts per second and includes failed, safe, and unsafe auto-parameterizations.
Batch Requests	A batch is a collection of one or more SQL statements sent in one unit by the client. Each batch is compiled into a single execution plan. If the batch contains multiple SQL statements, all of the optimized steps needed to perform all the statements are built into a single execution plan. The Batch Requests statistic shows the number of batch requests per second that are processed by the SQL Server instance.
Failed Auto-Params	Auto-parameterization occurs when an instance of SQL Server attempts to reuse a cached plan for a previously executed query that is similar to, but not the same as, the current query. The Failed Auto-params statistic shows the number of failed auto-parameterization attempts per second.
Safe Auto-Params	Auto-parameterization occurs when an instance of SQL Server attempts to reuse a cached plan for a previously executed query that is similar to, but not the same as, the current query. The Safe auto-params statistic shows the number of auto-parameterization attempts per second and includes only safe auto-parameterizations (ones where the cached plan can be shared).
Scan Point Revalidations	The number of times per second that the scan point had to be revalidated to continue the scan.
SQL Compilations	The number of SQL compilations performed, indicating the number of times the compile code path is entered. This also includes compiles due to recompiles. When SQL Server user activity levels become stable, this value reaches a steady state.
SQL Re-Compilations	The total number of recompiles triggered per second in a SQL Server instance. Recompiles occur when SQL Server determines that the currently defined execution plan for an executing stored procedure might no longer be the best possible plan. SQL Server pauses the query execution and recompiles the stored procedure.
Unsafe Auto-Params	Auto-parameterization occurs when an instance of SQL Server attempts to reuse a cached plan for a previously executed query that is similar to, but not the same as, the current query. The Unsafe auto-params statistic shows the number of auto-parameterization attempts per second and includes only safe auto-parameterizations (ones where the cached plan cannot be shared).

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > User I/O Tab
Metrics	SQL Server's ability to match new SQL statements with existing, unused execution plans is increased when parameters or parameter markers are used in Transact-SQL statements. If an SQL statement is executed without parameters, SQL Server parameterizes the statement internally to increase the possibility of matching it against an existing execution plan. A high number for auto-param attempts shows that SQL Server is efficiently reusing existing cached plans. A small number for failed auto-param attempts shows that SQL Server is efficiently reusing existing cached plans. Because compilation is a significant part of a query's turnaround time, you should strive to have as many compilations stored in the cache as possible. If this number does not stabilize in direct proportion to user activity stabilizing, you should investigate your SQL Cache to see if it has adequate memory assigned to it. Recompiles slow down the process that is executing the procedure and increases the load on the CPU. By extension, the more recompiles that are occurring on your system, the more overall load increases resulting in poor performance. In general, you want to keep the number of recompiles low. The most common reasons SQL Server would issue a recompile are: *Running sp_recompile against any table referenced in the stored procedure.Significant data changes in a referenced table.Schema changes to referenced objects.The use of the WITH RECOMPILE clause in the CREATE PROCEDURE or EXECUTE statement.A plan no longer available in the system cache.*
Correction	For failed auto-param attempts, you can increase the ability of the relational engine to match complex SQL statements to existing, unused execution plans, by explicitly specify the parameters using either sp_executesql or parameter markers in your T-SQL code. For high numbers of SQL re-compilations, try to practice coding standards that eliminate the most frequent causes detailed above. Also, try to: *Use temporary tables only in the stored procedure that created them.Minimize creating temporary tables in control block structures.Use the KEEP PLAN option on references to static temporary tables.Issue the CREATE TABLE statement before any other references to the created table.Minimize the use of temporary tables.*

Lock Activity

The Lock Activity section of the User I/O tab of the I/O Detail displays statistics that track various user-related I/O operations. The I/O function along with its counter value is presented. The Lock Activity section details performance statistics that reflect how SQL Server is handling lock operations. The table below describes the information available in the Lock Activity section:

Information	Description
Table Lock Escalations	The number of times locks on a table were escalated.
Locks	The total number of locks acquired on the server.
Blocked Users	The users currently being blocked by other processes.
Deadlocks	The number of deadlocks detected by SQL Server. Page Deadlocks occur when processes cannot proceed because they are waiting on a set of resources held by each other or held by other processes.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > User I/O Tab
Metrics	Many table lock escalations could indicate contention problems. If increasing numbers of table lock escalations are viewed at the same time as blocking or deadlock problems, the application design could be at fault. Consistently seeing positive numbers for the blocked statistic should also clue you into the fact that a bottleneck exists for some processes. You can easily drill down and discover the exact process(es) holding locks that are blocking out other user activity. Another situation to look for with respect to locking, is when the total number of acquired locks reaches the maximum lock limit currently set on SQL Server. Consistently seeing page deadlock counts greater than zero indicates that some user processes are experiencing delays completing their work. When SQL Server identifies a page deadlock, it resolves the situation by choosing the process that can break the deadlock. This process is termed the deadlock victim. SQL Server rolls back the deadlock victim's transaction, and then notifies the process' application by returning an error message. It also cancels the process' request and allows the transactions of the remaining processes to continue. SQL Server always attempts to choose the least expensive thread running the transaction as the deadlock victim.
Correction	Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the stranglehold on the objects the user was accessing. Other user processes then almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Embarcadero Performance Analyst, but preventing the blocking lock situation in the first place is tricky. The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do. By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process waits for a lock before timing out. Because SQL Server automatically resolves deadlock situations, you should work proactively to prevent them in the first place. You can change default deadlock behavior by using the SET DEADLOCK_PRIORITY command, which reprioritizes a process' position in a deadlock situation.

Temporary Object Activity

The Temporary Object Activity section of the User I/O tab of the Users Detail displays statistics that track various user-related I/O operations. The I/O function along with its counter value is presented. The Temporary Object Activity section details performance statistics that reflect how SQL Server is handling the creation of temporary objects, such as temporary tables used in stored procedures. The table below describes the information available in the Temporary Object Activity section:

Information	Description
Workfiles Created	The number of workfiles created by SQL Server.
Worktables Created	The total number of work tables created. Worktables are used many times by SQL Server to perform a logical operation specified in an end-user SQL statement. GROUP BY, ORDER BY, or UNION queries can cause worktables to be created as can specific CREATE statements used in Transact SQL processing. Worktables are built in the tempdb database and are dropped automatically at the end of the statement or procedure run.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > User I/O Tab
Metrics	With respect to worktables, keep in mind that the tempdb database should be large enough to hold large worktables.

Database I/O Tab

The Database I/O tab of the I/O Detail includes the following sections:

Database Bytes Read/Write Summary Location Database Read/Write Summary

Database Read/Write Summary

The Reads statistic represents that number of physical database page reads that are issued per second by SQL Server. The Writes statistic represents the number of physical database page writes issued by SQL Server. Writes take place during operations such as checkpoints, lazywriter writes, index creations, and BCP routines. Both statistics are collective in nature, meaning that they represents the total page reads and writes across all databases that exist on the target SQL Server.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > Database I/O Tab
Metrics	Page reads are to be expected, especially after initial server start up. This is because SQL Server must first satisfy requests for data and metadata by reading information in from physical disk. Numerous page reads can also be expected if the physical server does not contain an adequate amount of memory to hold repetitively requested blocks of information. No hard-and-fast rules exist for how many page reads per second is too much. You can cross reference this statistic with the physical server disk statistics to see if physical page reads and accompanying physical disk I/O is approaching the server's premium capacity levels. And because logical I/O is always many times faster than physical I/O, you should also evaluate the buffer cache hit ratio to determine overall memory vs. physical read efficiency. Page Writes can give you an idea of overall physical write activity. There are a number of statistics, however, that pertain specifically to certain write activities like checkpoints, that can be examined to determine the amount of physical writes caused by distinct SQL Server processes. Regarding performance, response times experienced by users are normally not impacted by write operations unless the writes are synchronous. These are typically BCPs, database recovery operations, and index creations.

Database Bytes Read/Write Summary

The Database Bytes Read/Write Summary statistics represent the number of bytes read and written by SQL Server.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > Database I/O Tab

Database I/O Detail

The Database I/O section of the Database I/O tab of the I/O Detail summarizes I/O activity for each database, letting you quickly spot the “hot” databases on your server. The table below describes the information available in the Database I/O Detail section:

Information	Description
Database	The name of the database.
Reads	The number of reads issued against the database.
Writes	The number of writes issued against the database.
Bytes Read	The total number of bytes read for the database.
Bytes Written	The total number of bytes written for the database.
I/O Stall	The total amount of time that processes have waited for I/O operations to complete, in milliseconds.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > Database I/O Tab
Metrics	Consider moving databases with lots of I/O activity and wait time onto separate drives/devices.

File Tab

The File tab of I/O Detail includes the following sections:

File Bytes Read/Write Summary File I/O Detail File Read/Write Summary

File Read/Write Summary

The Reads statistic represents that number of physical database page reads that are issued per second by SQL Server. The Writes statistic represents the number of physical database page writes issued by SQL Server. Writes take place during operations such as checkpoints, lazywriter writes, index creations, and BCP routines. Both statistics are collective in nature, meaning that they represents the total reads and writes across all databases that exist on the target SQL Server.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > File Tab
Metrics	Page reads are to be expected, especially after initial server start up. This is because SQL Server must first satisfy requests for data and metadata by reading information in from physical disk. Numerous page reads can also be expected if the physical server does not contain an adequate amount of memory to hold repetitively requested blocks of information. No hard and fast rules exist for how many page reads per second is too much. You can cross reference this statistic with the physical server disk statistics to see if physical page reads and accompanying physical disk I/O is approaching the server's premium capacity levels. And because logical I/O is always many times faster than physical I/O, you should also evaluate the buffer cache hit ratio to determine overall memory vs. physical read efficiency. Page Writes can give you an idea of overall physical write activity. There are a number of statistics, however, that pertain specifically to certain write activities like checkpoints, that can be examined to determine the amount of physical writes caused by distinct SQL Server processes. Regarding performance, response times experienced by users are normally not impacted by write operations unless the writes are synchronous in nature. These are typically BCPs, database recovery operations, and index creations.

File Bytes Read/Write Summary

The File Bytes Read/Written statistics communicate the number of bytes read and written by SQL Server.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > File Tab

File I/O Detail

The File I/O Detail section of the File tab of the I/O Detail summarizes I/O activity for each database file, letting you quickly spot the “hot” databases and files on your server. The table below describes the information available in the File I/O Detail section:

Information	Description
Database	The database name.
File ID	The file identifier for the target file.
Logical Name	The name given the file by the DBA.
File Name	The physical file name of the file.
Timestamp	The internal time stamp of when the data was obtained.
Reads	The number of reads issued against the database file.
Writes	The number of writes issued against the database file.
Bytes Read	The total number of bytes read for the database file.
Bytes Written	The total number of bytes written for the database file.
I/O Stall	The total amount of time that processes have waited for I/O operations to complete, in milliseconds.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **I/O View (DBArtisan - SQL Server Performance Analyst)** > File Tab
Metrics	Consider moving databases and/or files with lots of I/O activity and wait time onto separate drives/devices.

Space View (DBArtisan - SQL Server Performance Analyst)

In addition to a Home page, the Space category of SQL Server Performance Analyst includes the following tabbed pages:

File Groups Tab Files Tab Objects Tab

Virtual Log Files Tab Server Space Tab

Home

The Home page of the Space category includes statistics in the following categories:

Bottleneck Analysis Pane Disk Analysis Pane Space Analysis Pane

SQL Analysis Pane

Space Analysis Pane

A SQL server contains many databases, some of which are devoted to system-level activities (the master and tempdb databases, for example) and others that hold user data. The Key Space Analysis section of the Space home page displays details about the space situation for each database in a SQL server, including their total, used, and free space. The percent used amount for each database is also shown. In addition, each database's transaction log space is analyzed and presented.

Metrics

If a database or transaction log's free space goes below the Performance Analyst recommended threshold, (and either the database or transaction log does not have its automatic growth property enabled or the files have reached their growth limit) then the DBA should take action to ensure that the database or transaction log does not run out of available free space.

Correction

If the percent used amount of a database is approaching problematic levels, there are three ways a DBA can rectify the situation:

1 The DBA can resize the current file(s) used by the database via an ALTER DATABASE … MODIFY FILE command.

2 The DBA can add a new file to the database via the ALTER DATABASE … ADD FILE command.

3 The DBA can modify the file(s) used by the database to automatically grow by using the ALTER DATABASE … MODIFY FILE … FILEGROWTH command. You should also ensure that the MAXSIZE setting of each file is set appropriately.

Of course, the DBA should also ensure that enough physical space exists on the server to accommodate additional database space. There are several things a DBA can do to ensure that a database's log does not run out of available free space:

1 First, most transactional-oriented databases should have their logs assigned to separate physical drives other than the database. Reasons for doing this include:

- It prevents competition for space between the log and the database itself.
- It allows the log to be monitored for space more effectively.
- It improves performance.

2 If the database is not critical in nature, you can set the truncate log on checkpoint option (trunc log on chkpt), which eliminates any non-active space in the log when a database checkpoint occurs.

3 Critical databases needing higher levels of recovery should have schedules established that regular perform transaction log dumps. Doing so ensures better recovery scenarios as well as a reduced risk of the transaction log running out of space.

4 If a critical transaction log becomes full, it might be impossible to use standard procedures to dump transactions and reclaim space. The dump operation will likely have to incorporate the no log or truncate only options.

5 If a transaction log continuously approaches dangerously low levels of free space, then the DBA should allow the underlying file(s) of the log to automatically grow to meet the demand. This can be accomplished by using the ALTER DATABASE … MODIFY FILE … FILEGROWTH command. You should also ensure that the MAXSIZE setting of each file is set appropriately.

The DBA should also be on the lookout for large load or data modification operations that do not make use of prudently timed commit points. A single, large transaction has the ability to overwhelm any transaction log since only non-active space in the transaction log is removed from log dumps or truncation operations.

Bottleneck Analysis Pane

When using bottleneck analysis to troubleshoot a database, a DBA cannot rely only on the information contained in the wait event views that SQL Server provides. For example, a database might attempt to allocate another segment of space and yet be denied if no such free space exists on the server or if the underlying file has reached its growth limit. Such a failure will not be reflected in any wait event, but still represents a very real bottleneck to the database. In the same way that a DBA cannot depend on only a few ratios to properly carry out ratio-based performance analysis, an administrator must include other statistical metrics in their overall bottleneck analysis framework to obtain an accurate performance risk assessment. Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system. The Performance Analyst Space home page displays space-related bottlenecks that might be threatening the overall performance of your system.

The following bottleneck indicators are used on the Performance Analyst Space home page to succinctly communicate the general overall space situation of the monitored database:

Databases Without Auto-Grow
Databases on Same Disk as Log
Databases With Auto-Shrink
Logs Without Auto-Grow
Database Files Near Max Size
Log Files Near Max Size

The following statistics, available on this pane, duplicate statistics available on the Bottleneck Analysis Pane of the Home View (DBArtisan - SQL Server Performance Analyst) page:

DB's Low on Space Logs Low on Space

Database Files Near Max Size

A DBA can specify that one or more files of a SQL Server database can automatically grow by enabling the file's auto-growth property. A DBA can also specify that the file not be permitted to grow beyond a certain limit, and once that limit is reached, the file and/or the database might be permanently out of space. The Database Files Near Max Size statistic provides a count of how many database files are near their maximum file size.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Space View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	A DBA should not see any positive indicator that any file has reached its growth limit. While it might be okay from the standpoint that a file cannot simply grow anymore (if the database has other files that can grow or that no more space will be added to the database), the DBA should disable the auto-growth property of the file.
Correction	If a file has reached its growth limit, a DBA can take a number of actions, including: *Eliminate the growth limit for the file either by increasing the limit size or allowing to grow in an unlimited fashion.If the file is on a disk or file system that has no free space left, the DBA can add another file to the database to allow for space expansion and alter the problem file to have auto-growth property disabled.*

Databases on Same Disk as Log

To minimize I/O contention, a database should have its database files located on different physical disks than its log files. The Databases on Same Disk as Log statistic provides a count of how many databases have files that are on the same physical disk as their transaction log.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Space View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	For critical production systems, all databases should have their database and log files on different physical disks.
Correction	If a DBA has database and log files on the same physical disk, they can create new database filegroups on different disks, relocate objects to the new filegroups, and then delete the old files/filegroups.

Databases Without Auto-Grow

In Microsoft SQL Server, a DBA has the ability to tell SQL Server to automatically grow a database in size when more space is required. This feature can save a critical transaction or other database request from failing due to a lack of free space in the database. It is recommended that critical databases have this feature enabled. The Databases Without Auto-Grow statistic provides a count of databases that do not have their automatic growth property enabled.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Space View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Static databases (those not expected to grow in size) will likely not need their auto-growth property enabled. Growing, dynamic databases should almost always be allowed to automatically grow when needed.
Correction	If any critical, dynamic database is found to not have its auto-growth feature enabled, then the DBA can modify the file(s) used by the database to automatically grow by using the ALTER DATABASE … MODIFY FILE … FILEGROWTH command. You should also ensure that the MAXSIZE setting of each file is set appropriately.

Databases With Auto-Shrink

A database can be set up to automatically shrink in size if SQL Server finds the database with an abundance of unused free space. Setting a database up so that this occurs can help reduce wasted storage on a database server. The Databases With Auto-Shrink statistic provides a count of databases that are allowed to automatically shrink in size.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Space View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Dynamic databases that have the potential to balloon in size and then have large amounts of that same data removed should be considered candidates for auto-shrink.
Correction	If a DBA wants to configure a database to automatically shrink, they can issue a command similar to the following: **EXEC sp_dboption 'mydatabase','autoshrink',true**

Logs Without Auto-Grow

In Microsoft SQL Server, a DBA has the ability to tell SQL Server to automatically grow a database or transaction log in size when more space is required. This feature can save a critical transaction or other database request from failing due to a lack of free space in the database or transaction log. It is recommended that critical databases and their transaction logs have this feature enabled.

The Logs Without Auto-Grow statistic provides a count of transaction logs that do not have their automatic growth property enabled.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Space View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Static databases (those not expected to grow in size) will likely not need their transaction log's auto-growth property enabled. Growing, dynamic databases should almost always have their transaction log be set to automatically grow when needed.
Correction	If any critical, dynamic database is found to not have their transaction log auto-growth feature enabled, then the DBA can modify the file(s) used by the database's transaction log to automatically grow by using the ALTER DATABASE … MODIFY FILE … FILEGROWTH command. You should also ensure that the MAXSIZE setting for each file is set appropriately.

Log Files Near Max Size

A DBA can specify that one or more transaction log files of a SQL Server database can automatically grow by enabling the file's auto-growth property. A DBA can also specify that the file not be permitted to grow beyond a certain limit, and once that limit is reached, the file and/or the transaction log might be permanently out of space. The Log Files Near Max Size statistic provides a count of how many transaction log files are near their maximum file size.

You should not see any positive indicator that any file has reached its growth limit. While it might be okay from the standpoint that a file cannot simply grow anymore (if the database transaction log has other files that can grow or that no more transactions are expected to occur within the database), the DBA should disable the auto-growth property of the file.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Space View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Correction	If a file has reached its growth limit, a DBA can take a number of actions: *Eliminate the growth limit for the file either by increasing the limit size or allowing to grow in an unlimited fashion.If the file is on a disk or file system that has no free space left, the DBA can add another file to the database's transaction log to allow for space expansion and alter the problem file to have auto-growth property disabled.*

SQL Analysis Pane

A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end-user response times.

The SQL Analysis section of the Space home page shows a count of all SQL statements that have executed INSERT, UPDATE, DELETE, and TRUNCATE commands. Seeing these statistics should help you determine how much space-related SQL activity is occurring on your system.

The SQL Analysis section provides you with a quick overview of the percentage of Reads, Writes, CPU, and Elapsed Time the most expensive SQL statements tracked by Performance Analyst have used. Depending on the page you are on, you might just see information regarding physical I/O activity or DML statements. For example, you might see that a SQL statement has caused 60% of all physical reads on the system, or that a procedure is responsible for 90% of all the CPU usage. To get information regarding the actual SQL text or stored procedure calls, drill down into the Top SQL details views.

Note: SQL Profiling is turned on by default in Performance Analyst so you can automatically collect SQL traffic. If you do not want to collect SQL execution information, use the options inside of Performance Analyst to disable SQL profiling.

Metrics

The statistics in this section can be used in many different ways. For example, large numbers of inserts could indicate a growing database and that a DBA should keep a watchful eye on free space. Heavy update numbers could mean trouble for objects containing many indexes (indexes typically undergo splits and fragmentation when large volumes of updates are thrown their way). Many deletes might signal potential wasted space problems.

Disk Analysis Pane

The Disk Analysis section of the Performance Analyst Space home page provides a summary of the total database and log space used per server disk drive, as well as a summary of free disk space for each server drive.

Metrics

If any database or transaction log file has been set up to automatically grow, the DBA should ensure there is enough server disk space to accommodate any new, additional requests for space.

Correction

If you see any drive that has reached zero free space (or is close), you might want to add new files, on other disks with abundant free space, to any databases or transaction logs so that no out-of-space errors result.

File Groups Tab

SQL Server manages physical storage space through files and file groups. The file groups tab displays detailed information regarding storage usage in each of the SQL Server file groups. The table below describes the information available in the File Groups Summary section of the File Groups tab:

Information	Description
Database Name	The name of the database.
File Group ID	The numeric ID of the file group.
File Group	The name of the file group.
Can Grow	This indicates if one of more of the files making up the file group is allowed to grow automatically.
File Count	The number of files that make up the file group.
Size (MB)	The total physical size of the file group.
Table Res	The amount of reserved space consumed by tables.
Index Res	The amount of reserved space consumed by indexes.
Free (MB)	The amount of free space for the file group.
Free Percent	The percentage of free space for the file group.

Metrics

Unless server space is tight, it is normally wise practice to allow your files to automatically grow to meet demand for more incoming data. You should also physically separate your database and log files onto separate physical drives.

Correction

To let your files automatically grow until out of space, set the file's growth option in Embarcadero's DBArtisan, or by using the ALTER DATABASE... MODIFY FILE command.

Files Tab

SQL Server manages physical storage space through files and file groups. The Files tab displays detailed information regarding storage usage in each of the SQL Server files. The table below describes the information available in the Files Summary section of the Files tab:

Information	Description
Database Name	The name of the database.
Logical Name	The nickname given to the file by the DBA.
File Group	The name of the file group.
File Name	The name and location of the file.
Size (MB)	The total physical size of the file.
Can Grow	This indicates if the file can automatically grow in size.
Growth Amount	This indicates how much the file will grow in size.
Max File Size	This indicates the maximum file size that the file can grow.

Metrics

Unless server space is tight, it is normally wise practice to allow your files to automatically grow to meet demand for more incoming data. It is also smart to physically separate your database and log files onto separate physical drives.

Correction

To let your files automatically grow until out of space, set the file's growth option in Embarcadero's DBArtisan, or by using the ALTER DATABASE... MODIFY FILE command.

Virtual Log Files Tab

The Virtual Log Files tab of the Space Detail displays an internal structure of sorts for a each database's log. The presented information is helpful when trying to shrink a database's log because you can see how much of the log is active and exactly where the active portion resides. The table below describes the information available on the Virtual Log Files tab of the Space Detail:

Information	Description
Database Name	The name of database.
File Name	The name of the log file.
Status	This indicates if this portion of the log is active or inactive (not being used).
Size	The size of this portion of the log in MBs.

Objects Tab

The Objects tab of the Space/Databases Detail includes the following sections:

Database Object Detail Database Object Summary

Database Object Summary

Tables and indexes consume the storage in all databases. The Database Object Summary grid displays summary information regarding table and index storage for all databases. The table below describes the information available in the Database Object Summary section of the Objects tab of the Space Detail:

Information	Description
Database Name	The name of the database.
Table Count	The count of all tables in the database.
Index Count	The count of all indexes in the database.
Table Reserved (MB)	The amount of space (in MB) reserved by tables.
Index Reserved (MB)	The amount of space (in MB) reserved by indexes.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Space View (DBArtisan - SQL Server Performance Analyst)** > Objects TabMicrosoft SQL Server Performance Analyst Statistics (DBArtisan) > Database View (DBArtisan - SQL Server Performance Analyst) > Objects Tab
Metrics	Negative numbers viewed for index space information can be caused by inaccuracies contained in the SQL Server data dictionary. Frequently, running a DBCC UPDATEUSAGE command against the database resolves the problem. However, there are bugs in SQL Server that sometimes caused erroneous numbers to the reported for the reserved space amounts used by tables and indexes.

Database Object Detail

Tables and indexes consume the storage in all databases. The Database Object Detail grid displays object space details for the database selected in the Database Object Summary grid. The table below describes the information available in the Database Object Detail section of the Objects tab of the Space Detail:

Information	Description
Object Owner	The owner of the object.
Object Name	The name of the object.
Clustered	Indicates if the index is clustered.
File Group	The file group where the object resides.
Reserved (KB)	The amount of space reserved by the object in kilobytes.
Used (KB)	The amount of space used by the object in kilobytes.
Free (KB)	The amount of free space used by the object in kilobytes.
Percent Used	The percentage of space used by the object.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Space View (DBArtisan - SQL Server Performance Analyst)** > Objects TabMicrosoft SQL Server Performance Analyst Statistics (DBArtisan) > Database View (DBArtisan - SQL Server Performance Analyst) > Objects Tab
Metrics	Negative numbers viewed for index space information can be caused by inaccuracies contained in the SQL Server data dictionary. Frequently, running a DBCC UPDATEUSAGE command against the database resolves the problem. However, there are bugs in SQL Server that sometimes caused erroneous numbers to the reported for the reserved space amounts used by tables and indexes.

Server Space Tab

The Server Space tab of the Space Detail displays the amounts of space used by SQL Server per physical drive and by database per physical drive. The first section displays summary information and the second section contains space information per database.

The table below describes the information available in the Disk Summary By Space section on the Server Space tab of the Space Detail:

Column	Description
SQL Data (MB)	The amount of reserved database space on the drive.
SQL Log (MB)	The amount of reserved log space on the drive.
Free	The total amount of free space that remains on the drive.

The table below describes the information available in the Disk Summary By Database section on the Server Space tab of the Space Detail:

Column	Description
Disk Drive	The physical drive letter.
Database	The database name.
SQL Data (MB)	The amount of reserved database space on the drive.
SQL Log (MB)	The amount of reserved log space on the drive.

Metrics

If you allow your database and/or log files to automatically grow, and you see little or no free space left on their physical drives, an option is to add new files to the database or log on different drives to avoid any out of space errors. It is also smart to physically separate your database and log files onto separate physical drives.

Correction

If you need to add new files to your databases or logs, you can do so easily by using the ALTER DATABASE… ADD FILE and ALTER DATABASE… ADD LOG FILE commands.

Users View (DBArtisan - SQL Server Performance Analyst)

In addition to a Home page, the Users category of SQL Server Performance Analyst includes the following tabbed pages:

Locks Tab	Session Waits Tab	System Waits Tab
Top Sessions Tab		

Home

The Home page of the Users Performance Analyst statistics includes the following categories:

Bottleneck Analysis Pane	Key User Analysis Pane	Login Analysis
SQL Analysis Pane	Workload Analysis Pane	

Key User Analysis Pane

User database activity can be examined using key performance statistics and wait/bottleneck-based analysis.

The following key statistics are used on the Performance Analyst Users home page to succinctly communicate the general session-based activity levels of the monitored database:

Active Processes	The Active Processes statistic represents the total number of active and open threads reported on the server. This number displays the number of processes actively performing work.
Active Transactions	The active transactions statistic represents a count of the number of in-process transactions for SQL Server.
Inactive Processes	This metric represents the total number of threads logged on to the server that are idle at the current time.
System Processes	The System Processes statistic represents the total number of threads logged on to the server that are SQL Server internal processes.
Transaction/Sec	The transaction/sec statistic refers to the number of database transactions processed per second by SQL Server.
T-SQL Batches	The T-SQL batches statistic refers to the number of transact SQL batches processed by SQL Server.

Bottleneck Analysis Pane

When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. And this is where wait-based or bottleneck analysis comes into play. DBAs use this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance because it helps a DBA track where a database has been spending its time. If lock contention or heavy table scan activity has been dragging down database performance, a DBA can use bottleneck analysis to confirm the actual root cause. Once one or more wait events or other bottlenecks have been pinpointed as possible performance vampires, the DBA can drill down and oftentimes discover a fair amount of detail about which sessions and objects are causing the problem. The Performance Analyst Home page identifies the top system and session waits that might be currently a cause of lessened performance.

When using bottleneck analysis, you cannot rely only on the information contained in the wait events that SQL Server provides. For example, a database might be suspect or offline. Such a failure will not be reflected in any wait event, but still represents a very real bottleneck to the database. In the same way that a DBA cannot depend on only a few ratios to properly carry out ratio-based performance analysis, an administrator must include other statistical metrics in their overall bottleneck analysis framework to obtain an accurate performance risk assessment. Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system. The Performance Analyst Home page displays space-related bottlenecks as well as other sources of contention/bottlenecks that might be dragging down the overall performance of your system.

The following bottleneck indicators are used on the Performance Analyst Users home page to succinctly communicate the general overall performance level of the monitored database:

Current Blocked Processes
Failed Auto-Param Attempts
Lock Timeouts
Lock Waits
Lock Wait Time
Scan Point Revalidations
SQL Re-Compilations
Top Bottlenecked User Processes

The following statistic, available on this pane, dupicates a statistic appearing on the Bottleneck Analysis Pane of the Home View (DBArtisan - SQL Server Performance Analyst) page:

- Table Lock Escalations

Current Blocked Processes

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency. Unless a process times out via an application timeout mechanism, or the process has specified a timeout period via the SET LOCK_TIMEOUT command, a process waiting for a lock will wait indefinitely.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	You should immediately investigate any indicator above zero, before the situation has a chance to mushroom.
Correction	Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is where it gets tricky. The DBA can drill down into user detail and view all current blocking locks to see exactly which sessions are holding the currently restrictive locks. The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do. By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process will wait for a lock before timing out.

Failed Auto-Param Attempts

Auto-parameterization occurs when an instance of SQL Server attempts to reuse a cached plan for a previously executed query that is similar to, but not the same as, the current query. The Failed Auto-Param Attempts statistic shows the number of failed auto-parameterization attempts per second.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	SQL Server's ability to match new SQL statements with existing, unused execution plans increases when parameters or parameter markers are used in Transact-SQL statements. If an SQL statement is executed without parameters, SQL Server parameterizes the statement internally to increase the possibility of matching it against an existing execution plan. A small number for this statistic shows that SQL Server is efficiently reusing existing cached plans.
Correction	You can increase the ability of the relational engine to match complex SQL statements to existing, unused execution plans, by explicitly specify the parameters using either sp_executesql or parameter markers in your T-SQL code. Doing so helps lower this number.

Lock Timeouts

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency. Unless a process times out via an application timeout mechanism, or the process has specified a timeout period via the SET LOCK_TIMEOUT command, a process waiting for a lock will wait indefinitely. The lock timeouts represent the number of locks that have timed out due to the issuing of the aforementioned command.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Any indicator above zero might indicate the presence of lock contention on the server.
Correction	Once discovered, a blocking lock situation can be quickly remedied - the DBA issues a KILL against the offending process, eliminating the user's stranglehold on the objects they were accessing. Other user processes then almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is more difficult. You can drill down into user detail and view all current blocking locks to see exactly which sessions are holding the currently restrictive locks. The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do. By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process will wait for a lock before timing out.

Lock Waits

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency. Unless a process times out via a timeout mechanism, or the process has specified a timeout period via the SET LOCK_TIMEOUT command, a process waiting for a lock will wait indefinitely. The Lock Waits statistic represents the number of lock requests that required a process to wait.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Any indicator above zero might indicate the presence of lock contention on the server.
Correction	Once discovered, a blocking lock situation can be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then almost always complete in an instant. Discovering the blocked lock situation is made easier by tools like Performance Analyst, but preventing the blocking lock situation in the first place is more difficult. The DBA can drill down into user detail and view all current blocking locks to see exactly which sessions are holding the currently restrictive locks. The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do. By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process will wait for a lock before timing out.

Lock Wait Time

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency. Unless a process times out via an application timeout mechanism, or the process has specified a timeout period via the SET LOCK_TIMEOUT command, a process waiting for a lock will wait indefinitely. The lock wait time statistic represents the wait time (in milliseconds) that processes have had to wait to obtain a lock request.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Any indicator above zero might indicate the presence of lock contention on the server.
Correction	Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is where it gets tricky. The DBA can drill down into user detail and view all current blocking locks to see exactly which sessions are holding the currently restrictive locks. The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do. By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process will wait for a lock before timing out.

Scan Point Revalidations

The Scan Point Revalidations statistic represents the number of times per second that the scan point had to be revalidated to continue the scan.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane

SQL Re-Compilations

The SQL re-compilations statistic represents the total number of recompiles triggered per second in a SQL Server instance. Recompiles occur when SQL Server determines that the currently defined execution plan for an executing stored procedure might no longer be the best possible plan. SQL Server pauses the query execution and recompiles the stored procedure.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Recompiles slow down the process that is executing the procedure and increase the load on the CPU. By extension, the more recompiles that are occurring on your system, the more overall load increases, resulting in poor performance. In general, you want to keep the number of recompiles low. The most common reasons SQL Server would issue a recompile are: Running sp_recompile against any table referenced in the stored procedure, significant data changes in a referenced table, schema changes to referenced objects, the use of the WITH RECOMPILE clause in the CREATE PROCEDURE or EXECUTE statement, and a plan no longer available in the system cache.
Correction	Try to practice coding standards that eliminate the most frequent causes detailed above. Also, try to: *Use temporary tables only in the stored procedure that created them.Minimize creating temporary tables in control block structures.Use the KEEP PLAN option on references to static temporary tables.Issue the CREATE TABLE statement before any other references to the created table.Minimize the use of temporary tables.*

Top Bottlenecked User Processes

When viewing wait statistics, there are several levels of detail that a DBA can drill down into. The first level is the system view, which provides a global, cumulative snapshot of all the waits that have occurred on a system. The second level is the session view, which shows details on what events connected sessions are experiencing. Viewing these numbers can help a DBA determine which session-related wait events have caused the most commotion in a database thus far. The Top Bottlenecked User Processes display identifies the processes with the most current wait time along with the event causing their current wait.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane

SQL Analysis Pane

A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured server down in terms of end-user response times.

Before you can identify problem SQL in your database, you have to ask the question - "What is bad SQL?" What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every SQL Server professional's satisfaction. The table below lists some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

Criteria	Description
Reads (Physical)	This is often used as the major statistic in terms of identifying good vs. bad SQL, this is a measure of how many disk reads the query caused to satisfy the user's request. While you certainly want to control disk I/O where possible, it is important that you not focus solely on physical I/O as the single benchmark of inefficient SQL. Make no mistake, disk access is slower than memory access and also consumes processing time making the physical to logical transition, but you need to look at the entire I/O picture of a SQL statement, which includes looking at a statements' logical I/O as well.
Writes	These can be caused by DML activity (INSERTs, etc.), but writes can also be indicative of heavy disk sort activity.
CPU Time	This is how much CPU time the query took to parse, execute, and fetch the data needed to satisfy the query.
Elapsed Time	This is how much time the query took to parse, execute, and fetch the data needed to satisfy the query. It should not include the network time needed to make the round trip from the requesting client workstation to the database server.

There are other criteria that you can examine like sort activity or access plan statistics (that show items like Cartesian joins and the like), but more often than not, these measures are reflected in the criteria listed above.

The SQL Analysis section provides you with a quick overview of the percentage of Reads, Writes, CPU, and Elapsed Time the most expensive SQL statements tracked by Performance Analyst have used. Depending on the page you are on, you might just see information regarding physical I/O activity or DML statements. For example, you might see that a SQL statement has caused 60% of all physical reads on the system, or that a procedure is responsible for 90% of all the CPU usage. To get information regarding the actual SQL text or stored procedure calls, drill down into the Top SQL details views.

Note: SQL Profiling is turned on by default in Performance Analyst so you can automatically collect SQL traffic. If you do not want to collect SQL execution information, use the options inside of Performance Analyst to disable SQL profiling.

Metrics

When you begin to look for inefficient SQL in a database, there are two primary questions you should answer:

- What has been the worst SQL that has historically been run in my database?
- What is the worst SQL that is running right now in my database?

When Correction a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It might be that the query is in an inefficient Transaction SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

There is the possibility that the SQL statement just is not tuned well. To determine that, you can drill down further into the Performance Analyst Top SQL view and begin working with the query through EXPLAIN plan analysis and other techniques inside Embarcadero's DBArtisan.

Login Analysis

The Login Analysis section displays the number of successful login and logout operations.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Home > SQL Analysis Pane

Workload Analysis Pane

When the database population as a whole experiences a system slowdown, it is not uncommon to find one or two users who are responsible for bringing the system to its knees. In the best of worlds, users will have a pretty evenly divided amount of memory usage, disk I/O, CPU utilization, and parse activity. Unfortunately, this usually is not the case, and many times you will find users submitting large batch jobs during peak OLTP activity, or sessions that are firing off untuned queries on a critical system.

If you are seeing a slowdown in your database, and cannot seem to find a root cause, one thing to examine is the resource consumption of the leading sessions on a system. Oftentimes you will find one or a handful of users making things miserable for everyone else. They accomplish this by overwhelming the I/O capabilities of SQL Server (through untuned queries or runaway batch jobs) or hammering the CPU or memory structures of the database and server.

Performance Analyst makes it easy to pinpoint the top sessions by showing the leading processes at the physical I/O, memory, CPU, and transaction submission usage levels.

Metrics

If any one session appears to be using more than 50% of a total resource (CPU, memory, etc.), then you should drill down into the session to find out what they are currently executing.

Top Sessions Tab

The Top Sessions tab of the Users Detail includes the following processes:

Top CPU Process Top I/O Process Top Memory Process

Top Sessions

Top I/O Process

The Top I/O Process statistic identifies the SQL Server process that currently has caused the most I/O usage on the database.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane

Top Memory Process

The Top Memory Process statistic identifies the SQL Server process that currently is using the highest percentage of memory in the database.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane

Top CPU Process

The Top CPU Process statistic identifies the SQL Server process that currently has the highest percentage of CPU usage in the database.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane

Top Sessions

When a system experiences heavy activity, sometimes you will find that all the user connections are contributing somewhat equally to the overall load. More often than not, however, one or two user connections are causing most of the activity. It might be that a large batch load or other typical process is running that is perfectly okay for your system. Or it might be a runaway process or other rogue connection that needs to be tracked down and possibly eliminated.

Top Sessions displays information regarding all key statistics for all current sessions on a SQL Server. The table below describes the information available in the Top Sessions section on the Top Sessions tab of the Users Detail:

Information	Description
SPID	The unique identifier for the process.
Login	The login name for the process.
Windows User	The operating system name of the process.
Database	The name of the database the process is working in.
Status	The current status of the process.
Program	This identifies the program being run by the process.
Memory	The number of pages in the procedure/SQL cache allocated to the process.
CPU	The cumulative CPU time for the process.
Physical I/O	The cumulative total of all physical reads and writes for the process.
Blocked	This indicates if the process is blocked by another process.
Host	This identifies the workstation of the process.
Open Transaction	This indicates if the process has an open transaction.
Command	The command being currently issued by the process.
Login Time	This identifies when the process logged into SQL Server.
Last Batch	This indicates the date/time the process executed a remote stored procedure or issued an EXEC command.
Host Process	The process identifier on the SQL Server machine.
Net Address	The network address of the process.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane

Current SQL

This view shows a snippet of the SQL associated with the user in question.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane

System Waits Tab

The System Waits tab of the Users Detail includes the following sections:

System Waits Wait Percent by Time Waited Wait Percent by Total Waits

System Waits

When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The System Waits section identifies the waits that have occurred on SQL Server. The System Waits section identifies the waits that have occurred on SQL Server. The table below describes the information available in the System Waits section on the System Waits tab of the Users Detail:

Information	Description
Wait Type	The actual wait event name.
Requests	The number of waits.
Percent of Total	The percentage of the occurrences of this wait to all wait requests.
Time Waited	The number of seconds SQL Server spent waiting for this wait.
Percent Time Waited	The percentage of time SQL Server spent waiting on this wait.
Signal Wait Time	The signal wait time for the wait.
Percent Signal Wait Time	The percentage of signal wait time SQL Server spent waiting on this wait.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > System Waits Tab
Metrics	Wait events can be hard to interpret at times. If you see a particular event that has caused a lot of wait time, you can review the information in this link (Microsoft Knowledge Base Article - 244455) to help understand the cause and potential remedy: http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455

Correction

The table below describes some possible solutions to some of the most common wait events:

Wait Event	Description
CURSOR	This indicates SQL is waiting to sync up with asynchronous cursors and can point to the excessive use of cursors.
CMEMTHREAD	This indicates waits for memory to be freed up for use.
CXPACKET	This relates to SQL Server parallelism. Oftentimes the cost of SQL Server using parallelism for a query is high, with the end result being these types of waits. If high, you can adjust SQL Server and turn parallelism off for the server or for just low cost queries.
EXCHANGE	Related to CXPACKET. See information above.
IO_COMPLETION	This means certain I/O related actions are being delayed. Bulk insert operations, growth of database or log files, high physical I/O SQL, page splitting, and server paging are potential causes. You can check Performance Analyst's SQL Analysis section to ferret out heavy I/O SQL and the Database and File I/O Detail pages to examine disk and database hot spots with respect to I/O. The I/O Stall column is especially useful in determining which files are experiencing the most delays.
LOGMGR	This refers to waits for the Log Writer to start writing a transaction. High waits here might warrant transfer transaction logs to faster devices or breaking up long running DML transactions.
OLEDB	This indicates waits for an OLE DB operation to act on its requests. Slow connection speeds or very high transaction rates can cause these.
PAGEIOLATCH	Related to IO_COMPLETION. See information above.
PAGELATCH	Related to IO_COMPLETION. See information above.
PAGESUPP	Related to CXPACKET. See information above.
PSS_CHILD	This is SQL waiting for a child thread within an asynchronous cursor and can point to the excessive use of cursors.
TEMPOBJ	This wait occurs when temp tables and the like are dropped.
WAITFOR	Waits caused by the WAITFOR T-SQL command. Not a cause for concern.
WRITELOG	Related to LOGMGR. Refers to waits for writes to disk of transactions. See information above.
XACTLOCKINFO	Involves the release of escalation of locks during bulk operations.

Wait Percent by Total Waits

When SQL Server is running, every connected process is either busy doing work or waiting for work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The Wait Percent by Total Waits section graphically depicts the top waits as a percentage of all waits.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > System Waits Tab
Correction	Wait events can be hard to interpret at times. If you see a particular event that has caused a lot of wait time, you can review the information in this link (Microsoft Knowledge Base Article - 244455) to help understand the cause and potential remedy: http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455

Wait Percent by Time Waited

When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The Wait Percent by Time Waited section graphically depicts the top waits by their percentage of time waited to total time waited. 5

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > System Waits Tab
Correction	Wait events can be hard to interpret at times. If you see a particular event that has caused a lot of wait time, you can review the information in this link (Microsoft Knowledge Base Article - 244455) to help understand the cause and potential remedy: http://support.microsoft.com/default.aspx?scid=kb;en-us;Q24445

Session Waits Tab

The Session Waits tab of the Users Detail includes the following sections:

Session Wait Percent by Time Waited Session Wait Percent by Wait Type Session Waits

Session Waits

When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The Session Waits section shows processes that are currently waiting on SQL Server. The table below describes the information available in the Session Waits section on the Session Waits tab of the Users Detail:

Information	Description
SPID	The unique identifier for the process.
Login Name	The login name for the process.
Windows User	The operating system name of the process.
Status	The current status of the process.
Host	This identifies the workstation of the process.
Program	This identifies the program being run by the process.
Last Wait Type	The actual wait event name.
Wait Time	The number of seconds the process spent waiting for this wait.
Percent Total Wait Time	The percentage of wait time for this process vs. total processes.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Session Waits Tab
Metrics	Wait events can be hard to interpret at times. If you see a particular event that has caused a lot of wait time, you can review the information in this link (Microsoft Knowledge Base Article - 244455) to help understand the cause and potential remedy: http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455
Correction	he table below describes some possible solutions to some of the most common wait events:

T

Wait Event	Description
CURSOR	This indicates SQL is waiting to sync up with asynchronous cursors and can point to the excessive use of cursors.
CMEMTHREAD	This indicates waits for memory to be freed up for use.
CXPACKET	This relates to SQL Server parallelism. Oftentimes the cost of SQL Server using parallelism for a query is high, with the end result being these types of waits. If high, you can adjust SQL Server and turn parallelism off for the server or for just low cost queries.
EXCHANGE	Related to CXPACKET. See information above.
IO_COMPLETION	This means certain I/O related actions are being delayed. Bulk insert operations, growth of database or log files, high physical I/O SQL, page splitting, and server paging are potential causes. You can check Performance Analyst's SQL Analysis section to ferret out heavy I/O SQL and the Database and File I/O Detail pages to examine disk and database hot spots with respect to I/O. The I/O Stall column is especially useful in determining which files are experiencing the most delays.
LOGMGR	This refers to waits for the Log Writer to start writing a transaction. High waits here might warrant transfer transaction logs to faster devices or breaking up long running DML transactions.
OLEDB	This indicates waits for an OLE DB operation to act on its requests. Slow connection speeds or very high transaction rates can cause these.
PAGEIOLATCH	Related to IO_COMPLETION. See information above.
PAGELATCH	Related to IO_COMPLETION. See information above.
PAGESUPP	Related to CXPACKET. See information above.
PSS_CHILD	This is SQL waiting for a child thread within an asynchronous cursor and can point to the excessive use of cursors.
TEMPOBJ	This wait occurs when temp tables and the like are dropped.
WAITFOR	Waits caused by the WAITFOR T-SQL command. Not a cause for concern.
WRITELOG	Related to LOGMGR. Refers to waits for writes to disk of transactions. See information above.
XACTLOCKINFO	Involves the release of escalation of locks during bulk operations.

Session Wait Percent by Wait Type

When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The Session Wait Percent by Wait Type section graphically depicts the top waits by wait type.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Session Waits Tab
Correction	Wait events can be hard to interpret at times. If you see a particular event that has caused a lot of wait time, you can review the information in this link (Microsoft Knowledge Base Article - 244455) to help understand the cause and potential remedy: http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455

Session Wait Percent by Time Waited

When SQL Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The Session Wait Percent by Time Waited section graphically depicts the top session waits by their percentage of time waited to total time waited.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Session Waits Tab
Correction	Wait events can be hard to interpret at times. If you see a particular event that has caused a lot of wait time, you can review the information in this Microsoft Knowledge Base Article to help understand the cause and potential remedy: http://support.microsoft.com/default.aspx?scid=kb;en-us;Q244455

Locks Tab

The Locks tab of the Users Detail includes the following sections:

Locks Locks by Lock Type Locks by Database

Locks

To modify database information or structures, a user session must obtain a lock on the object to perform its task. In addition to user locks, SQL Server itself issues lock requests to carry out its internal duties. The Locks section gives information the locks currently on the system and also indicates if any blocking situations are occurring. The table below describes the information available in the Locks section:

Information	Description
SPID	The process id of the process holding the lock.
Login	The login name of the process.
Windows User	The operating system name of the process.
Database	The database in which the process is running.
Table Name	The name of the table involved in a lock. This will be NULL for non-table locks or table locks that take place in the tempdb database.
Index ID	The index ID involved in the lock.
Lock Type	The type of the lock (database, table, row id, etc.)
Lock Mode	The lock's mode (shared, exclusive, etc.)
Lock Status	The lock's status (waiting or granted).
Lock Owner Type	Whether the lock came from a regular session or a transaction.
User Program	The executable the process is using against the server.
Blocking SPID	If zero, the process is not being blocked. If non-zero, this column represents the process ID of the process blocking the requested lock.
Wait Time	The current amount of wait time for the process, in milliseconds.
SPID Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
SPID Command	The command the process is currently issuing.
NT Domain	The name of Windows 2000/NT domain.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Locks Tab

Metrics Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the stranglehold on the objects the user was accessing. Other user processes then almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Embarcadero Performance Analyst, but preventing the blocking lock situation in the first place is tricky. The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do. By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process waits for a lock before timing out.

Locks by Lock Type

To modify database information or structures, a user session must obtain a lock on the object to perform its task. In addition to user locks, SQL Server itself issues lock requests to carry out its internal duties. The Locks by Lock Type section of the Users Detail provides a graphical percentage breakdown of the different locks types being issued across all databases.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Locks Tab

Locks by Database

The Locks by Database section of the Lock tab provides a graphical representation of the percentage breakdown of locks by database.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Users View (DBArtisan - SQL Server Performance Analyst)** > Locks Tab

Database View (DBArtisan - SQL Server Performance Analyst)

In addition to a Home page, the Database category of SQL Server Performance Analyst includes the following tabbed pages:

Backups Tab	Configuration Tab	Error Log Tab
Overview Tab	Replication Tab	SQL Agent Tab
Objects Tab		

Home

The Home page of the Database Performance Analyst page includes the following categories:

Bottleneck Analysis Pane	Database Analysis Pane	SQL Analysis Pane
Workload Analysis Pane		

Database Analysis Pane

SQL Server is broken up into many different databases, all of which have their own levels of activity. The Database Analysis section displays a summary of activities that have occurred within each database. The table below describes the information available in the Database Analysis section:

Information	Description
Database Name	The name of the database.
DBCC Logical Scans	The number of DBCC scans that have occurred within the database.
Transactions	The number of transactions that have occurred within the database.
Active Transactions	The number of transactions currently running in the database.
Bulk Copy Rows	The number of rows inserted per second into the database via BCP operations.
Bulk Copy Throughput	The amount of data bulk (KB) copied into the database per second.
Log Cache Reads	The number of reads per second that have gone through the log manager cache.
Log Flushes	The number of transaction log flushes per second. A log flush occurs when SQL Server writes all changes from the database's log cache out to the database's log files on disk.
Log Growths	The number of times the transaction log has grown in size.
Log Shrinks	The number of times the transaction log has reduced itself in size.

Bottleneck Analysis Pane

When using bottleneck analysis to troubleshoot a database, a DBA cannot rely only on the information contained in the wait event views that SQL Server provides. For example, a critical database might be offline or be marked suspect. Such a failure will not be reflected in any wait event, but still represents a very real bottleneck to the database. In the same way that a DBA cannot depend on only a few ratios to properly carry out ratio-based performance analysis, an administrator must include other statistical metrics in their overall bottleneck analysis framework to obtain an accurate performance risk assessment. Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system. The Performance Analyst Database home page displays database-related bottlenecks that might be dragging down the overall performance of your system.

The following bottleneck indicators are used on the Performance Analyst Database home page to communicate the general overall conditions of databases on the monitored server:

Databases Requiring Backup
Extents Allocated
Databases Without Auto-Create Stats
Freespace Scans
Databases Without Auto-Update Stats
Page Splits

The following statistics, available on this page, duplicate statistics available on the Bottleneck Analysis Pane of the Home View (DBArtisan - SQL Server Performance Analyst) page:

Suspect Databases Offline Databases

Databases Requiring Backup

This metric represents the number of databases in SQL Server that have not been backed up for more than seven days. This statistic excludes the pubs, tempdb, Northwind, msdb, and model databases.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	To ensure proper protection for most databases, it is recommended that even the most static databases be backed up at least once a week. You should frequently backup critical, dynamic databases and include transaction log backups to enable point-in-time recovery ability.
Correction	Any critical databases found with obsolete or no backups should immediately be backed up. Moreover, to ensure proper data protection for each database, you should institute a planned backup schedule. The timing and repetition of the backups depend on the critical recovery needs of each database.

Databases Without Auto-Create Stats

The Databases Without Auto-Create Stats statistic represents the total number of databases defined on SQL Server that do not have the AUTO_CREATE_STATISTICS option enabled.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	When the AUTO_CREATE_STATISTICS option is enabled, statistics are automatically created on columns used in a SQL query predicate. Keeping object statistics fresh in the SQL Server data dictionary improves query performance because the optimizer can better determine how to evaluate a query and return the requested data. If the statistics are not used, SQL Server should automatically delete them.
Correction	If possible, a DBA should keep their databases in AUTO_CREATE_STATISTICS mode. If a database is found without this option set, you can easily change it by using the command: EXEC sp_dboption '<database name>','auto create statistics',true

Databases Without Auto-Update Stats

The Databases Without Auto-Update Statistics statistic represents the total number of databases defined on SQL Server that do not have the AUTO_UPDATE_STATISTICS option enabled.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	When the AUTO_UPDATE_STATISTICS option is enabled, SQL Server automatically updates existing statistics when the statistics become out-of-date because data in the tables has changed enough to affect the optimizer's decision-making process.
Correction	If possible, a DBA should keep databases in AUTO_UPDATE_STATISTICS mode. If a database is found without this option set, you can easily change its auto-update statistics to true by using the command: EXEC sp_dboption '(database name)',

Extents Allocated

SQL Server groups eight 8KB pages together to form an extent of space (64KB). Space is allocated to database objects (tables, indexes) in the form of extents. The Extents Allocated statistic provides a count of the number of extents allocated to tables and indexes.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	None. However, be aware that excessive counts of allocated extents might signal a large load job occurring in the database.

Freespace Scans

The Freespace Scans statistic provides a count of the number of scans SQL Server initiated to search for free space that will accommodate requests to insert new data rows.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	None. However, be aware that excessive counts of freespace scans might signal INSERT problems being performed on large tables.

Page Splits

When data is added or modified for a table that has indexes, the indexes must be updated as well. As index pages fill up, free space needed to keep index keys in their proper order can oftentimes run very low. If an index update needs to be made to a page, and no free space exists on that page, SQL Server will perform a page split where it moves approximately half the rows on a page to a new page. The Page Splits statistic provides a count of how many page splits are occurring in SQL Server.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Page splits cause additional overhead in the form of CPU usage and I/O. Observing large numbers of page splits can signal a resource bottleneck in your server.
Correction	To avoid page splits, you can look into tuning the FILLFACTOR property of an index, which controls the percentage of the index page that is filled during creation. The default, 100, tells SQL Server to completely fill each page, whereas lower numbers tell SQL Server to leave room for additional index rows or updates to existing rows.

SQL Analysis Pane

Certain databases on a SQL Server will experience more SQL activity than others. The SQL Analysis section displays a count of all statement executions for each database.

The SQL Analysis section provides you with a quick overview of the percentage of Reads, Writes, CPU, and Elapsed Time the most expensive SQL statements tracked by Performance Analyst have used. Depending on the page you are on, you might just see information regarding physical I/O activity or DML statements. For example, you might see that a SQL statement has caused 60% of all physical reads on the system, or that a procedure is responsible for 90% of all the CPU usage. To get information regarding the actual SQL text or stored procedure calls, drill down into the Top SQL details views.

Note: SQL Profiling is turned on by default in Performance Analyst so you can automatically collect SQL traffic. If you do not want to collect SQL execution information, use the options inside of Performance Analyst to disable SQL profiling.

Error Log Analysis pane

SQL Server records various system events in its system or error log. The majority of messages placed into the log are informational in nature, however since some contain critical messages, you should immediately review them. This section indicates the number of actual error messages in the most recent error log so you know if there are potential events that require your attention.

Note: Similar information is also available on the Bottleneck Analysis Pane of the Home View (DBArtisan - SQL Server Performance Analyst) page.

Metrics

You should investigate any positive values for errors.

Correction

If non-zero values are found for this statistic, you should drill down into the current error log and examine the detail found that accompanies each error issued by SQL Server.

Workload Analysis Pane

SQL Server is broken up into many different databases, all of which have their own levels of activity. The Workload Analysis section displays a summary of resource consumption across all databases. The table below describes the information available in the Workload Analysis section:

Information	Description
Database Name	The name of the database.
Connections	The number of processes that are connected to the database.
Blocked Users	The number of processes in the database experiencing blocks from lock activity.
Total Memory	The total number of memory pages in use by processes for each database.
Total I/O	The total amount of physical I/O being experienced by each database.
Total CPU	The total amount of cumulative CPU for all processes connected to the database.
Total Wait Time	The total wait time (in milliseconds) that all processes connected to the database have experienced.
DBCC Running Count	This indicates if any DBCC operations are currently running against the database.
BCP Running	This indicates if any BCP operations are currently running against the database.
Backup Restore Running	This indicates if any backup or restore operations are currently running against the database.

Overview Tab

The Overview tab of the Database Detail includes the following sections:

Database Object Analysis Database Size Analysis Database Summary

Database User Analysis

Database Size Analysis

The Database Size Analysis chart displays each database in terms of its total percentage of size to the total size of all databases on the SQL Server.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane

Database Object Analysis

The Database Object Analysis chart displays each database in terms of its total percentage of object to the total objects for all databases on the SQL Server.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane

Database User Analysis

The Database User Analysis chart displays each database in terms of its total percentage of user accounts to the total user accounts of all databases on the SQL Server.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane

Database Summary

The Database Summary Analysis section displays summarized metadata for all defined databases on SQL Server. The table below describes the information available in the Database Summary section:

Information	Description
Database	The name of the database.
Created	The date/time when the database was created.
Status	The status of the database (online, offline, suspect, etc.)
Database Size (MB)	The total size of the database in MB.
Log Size (MB)	The total size of the log in MB.
Tables	The number of all tables in the database.
Indexes	The number of all indexes in the database.
Users	The number of all defined user accounts in the database.
Last Backup	The date/time of the last full backup for the database.
Autoshrink	This indicates if the database will release space automatically back to the operating system when it does not need it.
DB Autogrow	This indicates if one or more of the database's underlying files can automatically grow in size if needed.
Log Autogrow	This indicates if one or more of the log's underlying files can automatically grow in size if needed.
Autocreate Stats	This indicates if object statistics are automatically created for objects in the database.
Autoupdate Stats	This indicates if object statistics are automatically updated for objects in the database.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	The following are things to take note of: *Any critical database that shows a last full backup date that is older than the database's required backup needs.Any database that shows a status of offline or suspect.Any growing database that does not have its database or log files set to automatically grow in size.Any dynamic database that does not have its object statistics set to automatically update.*
Correction	Depending on the situation, you should take the following actions: For databases that require a full backup, perform a full backup of the database when appropriate. For suspect databases, the steps to handling will vary from one installation to another. However, here are some general guidelines you can use to troubleshoot a suspect database: *Begin by examining the SQL Server error log for clues as to what caused the database to be marked as suspect.It is not unusual for a server to run out of physical disk space on drives used by SQL Server. When this happens, recovery for databases can sometimes fail with the end result being SQL Server marking a database as suspect. To remedy this situation, you should free up space on the identified drives or add files to the newly marked suspect database. This can be accomplished by utilizing the following stored procedures: sp_add_data_file_recover_suspect_db and sp_add_log_file_recover_suspect_db. For SQL Server version 7.0, you will need to use the sp_resetstatus stored procedure to reset the suspect status flag for the database in question, use the alter database command to add new datafiles to the database, and then stop/start SQL Server.* Many times, suspect databases are caused by SQL Server not being able to access a database or log file. This happens if a particular physical hard drive has become unavailable, but also can occur if another operating system process has obtained exclusive access to a file. If this scenario proves to be true, once you have ensured that the file(s) are available once again to the operating system, you can use the sp_resetstatus stored procedure to reset the suspect status flag for the database and then stop/start SQL Server. Should an offline database be found by Performance Analyst, you can easily place it back online by utilizing either the sp_dboption stored procedure or the alter database command. If any critical, dynamic database or log is found to not have their auto-growth feature enabled, then the DBA can modify the file(s) used by the database to automatically grow by using the ALTER DATABASE … MODIFY FILE … FILEGROWTH command. You should also ensure that the MAXSIZE setting for each file is set appropriately. If possible, a DBA should keep their databases in AUTO_CREATE_STATISTICS and AUTO_UPDATE_STATISTICS mode. If a database is found without this option set, you can easily change it by using the following commands: *EXEC sp_dboption '<database name>','auto create statistics',trueEXEC sp_dboption '<database name>','auto update statistics',true*

Error Log Tab

The Error Log tab of the Database Detail includes the following sections:

SQL Server Error Log Error Log Content Analysis Error Log Volume Analysis

SQL Server Error Log

SQL Server logs events pertinent to its operations to the SQL Server error log and the operating system application log. Errors encountered by SQL Server are clearly visible in the log, however large logs can be difficult to navigate. The SQL Server Error Log section displays all events for a selected log or just errors.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > Error Log Tab
Metrics	You should investigate any positive values reported for errors.

Error Log Volume Analysis

SQL Server logs events pertinent to its operations to the SQL Server error log and the operating system application log. Errors encountered by SQL Server as well as informational message are recorded in the log. The Error Log Volume Analysis section displays a graph that shows the volume amount of messages/errors over the last five days.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > Error Log Tab
Metrics	Seeing a disproportionate percentage of messages for one day might indicate that something has, or is occurring that needs investigation.

Error Log Content Analysis

SQL Server logs events pertinent to its operations to the error log and the operating system application log. Errors encountered by SQL Server as well as informational message are recorded in the log. The Error Log Content Analysis section displays a graph that shows percentage of informational messages to errors in the current log.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > Error Log Tab
Metrics	You should investigate any indicated errors.

Objects Tab

The following statistics, available on this tab, are duplicates of statistics available on the Objects Tab of the Space View (DBArtisan - SQL Server Performance Analyst) page:

Database Object Summary Database Object Detail

SQL Agent Tab

The SQL Agent tab of the Database Detail includes the following sections:

Job/Alert Detail SQL Server Alert Summary SQL Server Job Summary

Job/Alert Detail

SQL Server provides the ability to submit and run jobs as well as be notified about certain SQL Server-related events. The Job/Alert Detail section displays information regarding the status of the server's SQL Agent as well as details for all jobs and alerts that are defined to the system.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > SQL Agent Tab

SQL Server Job Summary

The SQL Server Job Summary section graphically displays the outcome of all jobs for the last two days, as well as current activity.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > SQL Agent Tab
Metrics	You should investigate any counts noted for failed jobs.

SQL Server Alert Summary

The SQL Server alert summary section graphically displays the number of times that performance and event alerts have fired.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > SQL Agent Tab
Metrics	You should investigate any counts noted for any event.

Backups Tab

The Backups tab of the Database Detail includes the following:

Backup Detail Database Backup Summary Log Backup Summary

Backup Detail

The Backup section displays the most recent 25 backups for a selected database. The table below describes the information available on the Backups tab of the Database Detail:

Information	Description
Database	The name of the database.
Backup Start	The time stamp when the backup began.
Backup Finish	The time stamp when the backup finished.
Backup Type	The type of backup (FULL, INCREMENTAL, etc.)
Backup Size (KB)	The size of the backup, in kilobytes.
Expiration Date	The expiration date for the backup, if any.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > Backups Tab
Metrics	Although the needs of an application determine the frequency and type of backup, it is generally recommended that most dynamic databases have a solid plan in place for full and differential backups. For databases requiring point-in-time recovery, a backup plan should also include log backups.

Database Backup Summary

The Database Backup Summary section graphically displays the backup successes and failures for database backups over the last week.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > Backups Tab
Metrics	You should investigate any backup failures.

Log Backup Summary

The Log Backup Summary section graphically displays the backup successes and failures for database log backups over the last week.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > Backups Tab
Metrics	You should investigate any backup failures.

Configuration Tab

The Configuration tab of the Database Detail displays metadata regarding the hardware characteristics of the SQL Server as well as the SQL Server configuration parameters. The table below describes the information available on the Configuration tab:

Information	Description
Parameter	The parameter name.
Value	The value set for the parameter.
Run	The value currently running for the parameter.
Minimum	The minimum value allowed for the parameter.
Maximum	The maximum value allowed for the parameter.
Dynamic	Whether the parameter is dynamic (can be changed without requiring SQL Server to be stopped/started) or not.

Replication Tab

The Replication tab of the Database Detail provides basic information regarding replication activities that are occurring on the monitored SQL Server.

Replication Agent Details

The Replication Agent Details section shows information regarding replication. The table below describes the information available on the Replication tab:

Information	Description
Agent	The agent performing the replication work.
Publication	The object being published to a subscribing database.
Publisher	The SQL Server providing the source material.
Publisher Database	The database of the SQL Server providing the source material.
Status	The current status of the replication agent's task.
Subscriber	The SQL Server requesting the source material.
Subscriber Database	The database of the SQL Server requesting the source material.
Type	The type of replication activity (push, pull, etc.)
Start Time	The start time for the replication task.
Duration	The duration of the replication task.
Last Action	The output message for the replication task.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > Replication Tab

Replication Throughput Details

The Replication Throughput Details section of the Replication tab displays:

- Replication Type
- Measure
- Instance Name
- Value

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Database View (DBArtisan - SQL Server Performance Analyst)** > Replication Tab

OS View (DBArtisan - SQL Server Performance Analyst)

In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database sever for resources (CPU, I/O, or Memory), a slow CPU, insufficient or slow I/O devices, and insufficient memory. The OS Statistics page of Performance Analyst lets you examine operating system metrics for the following platforms:

- AIX
- HP-UX

Note: To view processor info and swap disk info on an HP-UX box, you need to login as ROOT in the OS login.

- Linux
- Solaris
- Unix
- Windows XP and 2000

Note: The statistics available on the OS page depend on the operating system platform.

Tip: If you magnify a graph to change back to the original size or minimize the size, close the OS Page and then reopen it again. Performance Analyst displays the graph in its original form and size.

In addition to a Home page, the OS category of SQL Server Performance Analyst includes the following tabbed pages:

CPU Tab I/O Tab Memory Tab

Network Tab Processes Tab Space Tab

Home

The OS home page includes the following sections:

Key Resource Usage Pane Disk Analysis Pane Bottleneck Analysis Pane

Memory Analysis Pane Workload Analysis Pane

Key Resource Usage Pane

The following ratios are used on the Performance Analyst OS home page to communicate the general overall performance levels of the operating system:

Disk Time Swap Memory Used Processor Time

Swap Memory Used (Windows) Swap Memory Used (AIX, HP-UX, Linux, Solaris, Unix)

Tip: To open the CPU tab, right-click Processor Time. To open the I/O tab, right-click Disk Time. To open the Memory tab, right-click Paged Memory Used.

Disk Time

The Disk Time statistic is the percentage of elapsed time that the selected disk drive/device was busy servicing read or write requests.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	You should avoid consistently seeing values for this statistic greater then 90%.
Correction	Add more disk drives and partition the files among all of the drives.

Load Average

The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	High load averages usually mean that the system is being used heavily and response time is correspondingly slow.

Processor Time

The Processor Time statistic indicates the percentage of time the processor is working. This counter is a primary indicator of processor activity.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	If your computer seems to be running sluggishly, this statistic could be displaying a high percentage.
Correction	Upgrade to a processor with a larger L2 cache, a faster processor, or install an additional processor.

Paged Memory Used

The Paged Memory Used statistic is the ratio of Commit Memory Bytes to the Commit Limit. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk. The commit limit is determined by the size of the paging file. As the paging file increases, so does the commit limit.

Note: This statistic is available for the Windows platform.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	This value displays the current percentage value only and not an average. If the percentage of paged memory used is above 90%, you may be running out of memory.
Correction	Increase the size of page file.

Swap Memory Used

The Swap Memory Used statistic is the percentage of swap space currently in use.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	If the percentage of swap memory used is above 90%, you may be running out of memory.
Correction	Increase the size of your swap files.

Bottleneck Analysis Pane

The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

Average Disk Queue Length

Page Faults/Sec

Processor

Interrupts/Sec

Processor Queue Length

Network Output Queue Length/Network Queue Length

Processor Speed

Tip: To open the I/O tab, right-click any Details menu item. To open the CPU tab, right-click the Item Processor Queues Length Details menu item. To open the Network tab, right-click Network Output Queues Length.

Note: The statistics available in this section depend on the platform of operating system.

Average Disk Queue Length

The Average Disk Queue Length statistic is the average number of both read and write requests that were queued for the selected disk during the sample interval.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	This metric is useful in identifying I/O related bottlenecks. If the disk queue lengths for certain disks are consistently much higher than others, you may need to redistribute the load among available disks. If the disk queues lengths for all disks are consistently large, and you see a high amount of I/O activity, your disks may be inefficient.
Correction	Some things you can do if you have problems with this statistic include: *Redistribute the data on the disk with the large average disk queue to other disks.Upgrade to faster disk(s).*

Interrupts/Sec

Interrupts/Sec is the average rate, in incidents per second, at which the processor received and serviced hardware interrupts. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended. The system clock typically interrupts the processor every 10 milliseconds, creating a background of interrupt activity. This statistic shows the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis PaneMicrosoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > CPU Tab > CPU Utilization
Metrics	A high value indicates possible excessive hardware interrupts; justification is dependent on device activity.

Network Output Queue Length/Network Queue Length

The Network Output Queue Length statistic is the number of threads in the processor queue.

Note: The name of this statistic depends on the platform of the operating system.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.
Correction	A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Page Faults/Sec

The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis PaneMicrosoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Memory Tab
Metrics	This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
Correction	If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Processor Queue Length

The Processor Queue Length statistic is the number of threads in the processor queue.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis PaneMicrosoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > CPU Tab
Metrics	Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.
Correction	A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Processor Speed

The Processor Speed statistic displays the speed of the active processor in MHz. The speed is approximate.

Processor

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane

The Processor Statistic displays the type of processor currently in use, for example, GenuineIntel.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Bottleneck Analysis Pane

Memory Analysis Pane

The following metrics are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

Available Paged Memory (Windows)

Available Physical Memory

Available Swap Memory (AIX, HP-UX, Linux, Solaris, Unix)

Total Paged Memory/Total Swap Memory

Total Physical Memory

Tip: To open the Memory tab, right-click any Details menu item.

Note: The statistics available in this section depend on the platform of operating system.

Available Paged Memory

The Available Paged Memory statistic shows the amount of virtual memory available for the processes.

Note: This statistic is available for the Windows platform.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Memory Analysis Pane
Metrics	If the available virtual memory is less than 10% of the total virtual memory, your system may run out of memory.
Correction	Increase the size of page file.

Available Physical Memory

The Available Physical Memory statistic represents the amount of RAM available to all processes.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Memory Analysis Pane
Metrics	This counter displays the last observed value only and not an average. Use this value with the Total physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.
Correction	Some things you can do if you have problems with this statistic include: *Check the running processes to see if there are any memory leaks.Stop any services that are not required.Install additional RAM.*

Available Swap Memory

The Available Swap Memory statistic represents the amount of virtual memory available for the processes.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Memory Analysis Pane
Metrics	If the available Available Swap Memory is less than 10% of total Swap Memory, your system may run out of memory.
Correction	Increase the size of swap files.

Total Paged Memory/Total Swap Memory

The Total Paged Memory statistic shows the maximum amount of virtual memory available to all processes.

Note: The name of this statistic depends on the platform of the operating system.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Memory Analysis Pane
Metrics	It is recommended that this be 1½ - 3 times the amount of RAM on the system.

Total Physical Memory

The Total Physical Memory statistic shows the amount of physical memory installed on your computer.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Memory Analysis Pane
Metrics	This is an informational metric and displays the total amount installed on the machine. Use this value with the available physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Disk Analysis Pane

The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

Free Disk Space Total Disk Space Used Disk Space

Tip: To open the Space tab, right-click any Details menu item.

Note: The statistics available in this section depend on the platform of operating system.

Free Disk Space

The Free Disk Space statistic shows the unallocated space, in megabytes on all logical disk drives.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Disk Analysis Pane
Correction	There are many things you can do to ensure that a database does not encounter a space problem due to physical space limitations: *If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.*

Total Disk Space

Total Disk Space displays the total allocated and unallocated space, in megabytes on all logical disk drives.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Disk Analysis Pane
Correction	There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations, here are two: *If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.*

Used Disk Space

The Used Disk Space statistic shows the amount of allocated space, in megabytes on all logical disk drives.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Disk Analysis Pane
Correction	There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations: *If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.*

Workload Analysis Pane

The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

Number of Logins Number of Processes Top CPU Process

Top I/O Process Location

Tip: To open the Processes tab, right-click any Details menu item.

Note: The statistics available in this section depend on the platform of operating system.

Number of Logins

This statistic displays the total number of logins on the server.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Disk Analysis Pane

Number of Processes

This statistic displays the total number of processes on the server.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Disk Analysis Pane

Top Memory Process

Top Memory Process shows the current process that is consuming the most amount of memory. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Disk Analysis Pane
Metrics	If you are running out of memory on the system, this is a quick way to identify the top memory user. If the displayed process is using a significant portion of the total memory, it could be causing the memory issues.

Top CPU Process

Top CPU Process shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Disk Analysis Pane
Metrics	If the amount of CPU time used by this process is close to 100% and the CPU usage is very high, this process may be the bottleneck on the server.
Correction	Investigate the process further to see if it is in an inconsistent state. Also, look at minimum requirements for CPU speed for the process. You may need to upgrade your CPU.

Top I/O Process

The Top I/O Process statistic shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Home > Disk Analysis Pane

CPU Tab

The CPU tab of the OS Detail includes the following sections:

CPU Events CPU Utilization Load Averages

The following statistics, available on this tab, duplicate statistics available on the Bottleneck Analysis Pane of the **OS View (DBArtisan - SQL Server Performance Analyst)**Home page:

Interrupts/Sec Processor Queue Length

Note: The sections available on this tab depend on the platform of operating system.

CPU Utilization

The CPU Utilization section includes the following information:

% Privileged Time % User Time

% Privileged Time

This metric is the percentage of elapsed time that the process threads spent executing code in privileged mode.

Note: For Windows systems, when a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. These kernel commands, are considered privileged to keep the low-level commands executing and prevent a system freeze. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > CPU Tab
Metrics	The ideal range should be 0-40% (less than 40% indicates excessive system activity).
Correction	If your CPU consistently runs at less than 40% you may need to upgrade your system to include a faster processor(s).

% User Time

The % User Time statistic is the percentage of elapsed time the processor spends in the user mode. User mode is a restricted processing mode designed for applications, environment subsystems, and integral subsystems. The alternative, privileged mode, is designed for operating system components and allows direct access to hardware and all memory. The operating system switches application threads to privileged mode to access operating system services. This counter displays the average busy time as a percentage of the sample time.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > CPU Tab
Metrics	If the Privileged Time is high in conjunction with Physical Disk Reads, consider upgrading the disk I/O subsystem.

CPU Events

The CPU Events section includes the following information:

Context Switches/Sec Load Averages System Calls/Sec

Note: The statistics available in this section depend on the platform of operating system.

Context Switches/Sec

The Context Switches/Sec section shows the combined rate at which all processors on the computer are switched from one thread to another. Context switches occur when a running thread voluntarily relinquishes the processor, is preempted by a higher priority ready thread, or switches between user-mode and privileged (kernel) mode to use an Executive or subsystem service.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > CPU Tab
Metrics	The ideal range should be between 0-10,000. GA number greater then 10,000 may indicate too many threads contending for resources.

System Calls/Sec

System calls are functions which require the services of the operating system. Usually they involve access to some data that users must not be permitted to corrupt or even change.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > CPU Tab
Correction	Try to find a good ratio between the amount of time the operating system runs (fulfilling system calls and doing process scheduling), and the amount of time the application runs. Your goal should be running 60% to 75% of the time in application mode, and 25% to 40% of the time in operating system mode. If you find that the system is spending 50% of its time in each mode, then you should investigate to determine what is wrong.

Load Averages

The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > CPU Tab
Metrics	High load averages usually mean that the system is being used heavily and the response time is correspondingly slow.

Processes Tab

The Processes tab of the OS Detail page succinctly communicates the general overall performance levels of processes. The columns available in this table depend on the platform of operating system. The table below describes the information available in the table on this tab:

Column	Description
Process	The name of the process.
User	The user of the process.
ID	The ID Process is the unique identifier of this process. ID Process numbers are reused, so they only identify a process for the lifetime of that process.
CPU	The CPU is the percentage of elapsed time that all of process threads used the processor to execution instructions.
User Mode	The User Mode is the percentage of elapsed time that the process threads spent executing code in user mode.
Memory WINDOWS ONLY	Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Memory (MB)	Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Memory	Memory is the percentage of the memory used of the total memory.
Active Memory	Active Memory is the amount of committed virtual memory, in bytes for this process. Active memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive. This counter displays the last observed value only; it is not an average.
I/O Data	The rate at which the process is reading and writing bytes in I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.
Elapsed Time	The total elapsed time, in seconds, that this process has been running.
Thread Count	The number of threads currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.
Handle Count	The total number of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in this process.
Priority	The current base priority of this process. Threads within a process can raise and lower their own base priority relative to the process' base priority.
Creating Proc ID	The Creating Process ID value is the Process ID of the process that created the process. The creating process may have terminated, so this value may no longer identify a running process.
Page Faults/Sec	Page Faults/Sec is the rate at which page faults by the threads executing in this process are occurring. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This may not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with whom the page is shared.

Page File	Page File is the current number of kilobytes that this process has used in the paging file(s). Paging files are used to store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory.
Private	Private is the current size, in kilobytes, of memory that this process has allocated that cannot be shared with other processes.

I/O Tab

The I/O tab includes the following sections:

Average Disk Queue/Total Disk Queue	LocBytes per I/O Operation	Disk I/O Time
Disk Transfers/Sec	I/O Details	Transfer Rate

Note: The sections available on this tab depend on the platform of operating system.

I/O Details

The table below describes the information available in this section:

Column	Description
Disk	The disk number assignment.
Reading (KB/s)	The amount of bytes read from the device.
Writing (KB/s)	The amount of bytes written to the device.
Disk Read Time	Disk Read Time is the percentage of elapsed time that the selected disk drive was busy servicing read requests.
Disk Write Time	Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.
Disk Time	Disk Time is the percentage of elapsed time that the selected disk was busy servicing requests.
Avg. Read Queue	Avg. Disk Read Queue Length is the average number of read requests that were queued for the selected disk during the sample interval.
Avg. Write Queue	Avg. Disk Write Queue Length is the average number of write requests that were queued for the selected disk during the sample interval.
Disk Reads/Sec	Disk Reads/Sec is the rate of read operations on the disk.
Disk Writes/Sec	Disk Writes/Sec is the rate of write operations on the disk.

Note: The columns available in this table depend on the platform of operating system.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > I/O Tab

Transfer Rate

The Transfer Rate section succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

Statistic	Description	Metrics
Reading	The average number of bytes transferred from the disk during read operations.	It is measured in number of KBs per second.
Writing	The average number of bytes transferred to the disk during write operations.	It is measured in number of KBs per second.
Transferred	The amount of time required to transfer the data between the device's cache and the actual device media.	N/A

Note: The statistics available in this section depend on the platform of operating system.

LocBytes per I/O Operation

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > I/O Tab

The Bytes per I/O Operation section of the OS Detail succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

Statistic	Description	Metrics
Reading	The average number of bytes transferred from the disk during read operations.	It is measured in number of bytes per second.
Writing	The average number of bytes transferred to the disk during write operations.	It is measured in number of bytes per second.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > I/O Tab

Average Disk Queue/Total Disk Queue

The Average Disk Queue section of the OS Detail page succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

Statistic	Description	Information
Average Read Queue	Average Read Queue Length is the average number of read requests that were queued for a particular disk.	Sustained loads on any of the disk queue lengths could represent problems within your system.
Average Write Queue	Average Write Queue is the average number of write requests that were queued for a particular disk.	Sustained loads on any of the disk queue lengths could represent problems within your system. Contact your Windows System Administrator for assistance in Correction these problems.
Average Queue	Average time in ms. to service I/O requests which for disks includes seek, rotational latency and data transfer time).	Sustained loads on any of the disk queue lengths could represent problems within your system. Contact your Unix System Administrator for assistance in Correction these problems.

Note: The statistics available in this section depend on the platform of operating system.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > I/O Tab
Correction	Contact your Windows System Administrator for assistance in Correction these problems.

Disk I/O Time

The Disk I/O Time section of the OS Detail page succinctly communicates the general overall performance levels of the disk's I/O. The table below describes the statistics in this section:

Note: The statistics available in this section depend on the platform of operating system.

Statistic	Description	Metrics
% Disk Read Time	% Disk Read Time is the percentage of time during the sample interval that the disk is busy servicing read requests.	Use this in conjunction with the % Disk Write Time to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.
% Disk Write Time	% Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.	Use this in conjunction with the % Disk Read Time to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.
% Disk Time	% Disk Time is the percentage of time during the sample interval that the disk is busy servicing requests.	Use this statistic to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > I/O Tab

Disk Transfers/Sec

The Disk Transfers/Sec section of the OS Detail page succinctly communicates the general overall performance levels of the disk transfers. The table below describes the statistics in this section

Statistic	Description
Reads/Sec	The the rate of read operations on the disk.
Writes/Sec	The rate of write operations on the disk.
Transfers/Sec	The rate of transfer operations per second.

Note: The statistics available in this section depend on the platform of operating system.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > I/O Tab

Memory Tab

The Memory tab of the OS Detail page includes the following sections:

Buffer Size
Cache Size
Cache Efficiency
Memory Available
Paging Activity
Page Replacements

The following statistic, available on this pane, duplicates a statistic appearing on the Bottleneck Analysis Pane pf the **OS View (DBArtisan - SQL Server Performance Analyst)**Home page:

- Page Faults/Sec

Note: The sections available on this tab depend on the platform of operating system.

Paging Activity

The Paging Activity section includes the following statistics:

Blocks Input/Sec
Blocks Output/Sec
Paged In
Paged Out
Pages Input/Sec
Pages Output/Sec

Note: The statistics available in this section depend on the platform of operating system.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab

Blocks Input/Sec

The Blocks Input/sec statistic is the number of blocks sent to a block device per second.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab
Correction	Although it never hurts to have as much physical memory as your system can handle, here are two things you can check within your system to alleviate the memory bottleneck: *Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.Check to see if you have additional space on your disk drive that you could use to expand the size of your swap file. Normally, the bigger the initial size of your swap file, the better, in performance terms.*

Blocks Output/Sec

The Blocks Output/sec statistic is the number of blocks received from a block device per second).

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab
Correction	Although it never hurts to have as much physical memory as your system can handle, here are two things you can check within your system to alleviate the memory bottleneck. *Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.Check to see if you have additional space on your disk drive that you could use to expand the size of your swap file. Normally, the bigger the initial size of your swap file, the better, in performance terms*.

Paged In

The number of disk block paged into core from disk which occurs when the server is experiencing a shortage of RAM. Any nonzero value indicates excessive swap activity. In any virtual memory environment, it is not uncommon to see RAM pages moved out to a swap disk. This is a special disk area in UNIX that's reserved for holding memory pages so that the processor is capable of addressing RAM in excess of its full capability. While page-out of operations are a normal part of any server's operations, page-in operations indicate that the real amount of available RAM has been exceeded and that additional RAM pages are required on the server.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab
Correction	High values indicate the possible need to increase the amount of RAM on the system.

Paged Out

The number of disk block paged out of core from disk. In any virtual memory environment, it is not uncommon to see RAM pages moved out to a swap disk. This is a special disk area in UNIX that's reserved for holding memory pages so that the processor is capable of addressing RAM in excess of its full capability. While page-out of operations are a normal part of any server's operations, page-in operations indicate that the real amount of available RAM has been exceeded and that additional RAM pages are required on the server.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab

Pages Input/Sec

The Pages Input/Sec statistic is the number of pages read from disk to resolve hard page faults. Hard page faults occur when a process requires code or data that is not in its working set or elsewhere in physical memory, and must be retrieved from disk.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab
Metrics	This value was designed as a primary indicator of the kinds of faults that cause system-wide delays. It includes pages retrieved to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. This counter counts numbers of pages, and can be compared to other counts of pages, such as Memory: Page Faults/sec, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
Correction	Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck. *Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.*

Pages Output/Sec

The Pages Output/Sec statistic is the number of pages written to disk to free up space in physical memory. Pages are written back to disk only if they are changed in physical memory. A high rate of pages output might indicate a memory shortage.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab
Metrics	Windows NT writes more pages back to disk to free up space when low in physical memory. This counter counts numbers of pages, and can be compared to other counts of pages, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
Correction	Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck. *Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.*

Memory Available

The Memory Available section of the OS Detail page succinctly communicates the general overall performance levels of the server's memory. The statistics available in this section depend on the platform of operating system. The table below describes these statistics:

Statistic	Description	Metrics	Correction
Physical Memory	The Physical Memory statistic is the amount of committed virtual memory, in megabytes. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk.	This counter displays the last observed value only and not an average.	N/A
Physical	The Physical statistic is the total physical memory available.	N/A	N/A
Paged Memory	The Paged Memory statistic is the amount of committed virtual memory, in bytes. Used Memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive.	This counter displays the last observed value only; it is not an average.	N/A
Swap Memory	The Swap Memory statistic is the free, unreserved swap space.	N/A	Increase the size of your swap file or add additional files if you consistently see a shortage of free, unreserved swap space.
Swap	The Swap statistic is the total swap memory being used.	N/A	N/A
Total Physical	The Total Physical statistic is the total physical memory available.	N/A	N/A
Total Swap	The Total Swap statistic is the total swap memory available.	N/A	N/A
Swap-ins	The number of memory pages paged in from swapspace.	N/A	If the system is consistently low on free memory (over a 30 second average), the memory scheduler will start to swap out processes. Increase the size of your swap file or add additional files.
Swap-outs	The number of memory pages paged in from swapspace.	N/A	If the system is consistently low on free memory (over a 30 second average), the memory scheduler will start to swap out processes. Increase the size of your swap file or add additional files.
Buffers	The Buffers statistic represents the amount of buffers used by the kernel in kilobytes.	N/A	N/A

Page Faults/Sec	The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.	This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.	If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab

Buffer Size

The Buffer Size statistic represents the amount of cache used by the kernel in kilobytes.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab

Cache Size

The Cache Size statistic represents the amount of cache used by the kernel in kilobytes.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab

Page Replacements

The following statistics are available in this section:

Memory Freed (Pages/sec) Location

Memory Freed (Pages/sec)

Freed or destroyed (Kb/s).

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab

Clock Algorithm Scans (Pages/sec)

The VMM uses a technique known as the clock algorithm to select pages to be replaced. This technique takes advantage of a referenced bit for each page as an indication of what pages have been recently used (referenced). When the page-stealer routine is called, it cycles through the PFT, examining each page's referenced bit.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab

Cache Efficiency

The Cache Efficiency section of the Memory tab succinctly communicates the general overall performance levels of the server's memory. The following statistics are available in this section:

Copy Read Hits % Data Map Hits % MDL Read Hits %

Pin Read Hits %

Copy Read Hits %

The Copy Read Hits % statistic is the percentage of cache copy read requests that hit the cache and does not require a disk read to provide access to the page in the cache.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab
Metrics	When the page is pinned in the memory, the page's physical address in the file system cache will not be altered. A copy read is a file read operation where a page in the cache is copied to the application's buffer. Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.
Correction	Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate

Data Map Hits %

The Data Map Hits % statistic is the percentage of data maps in the file system cache that could be resolved without having to retrieve a page from the disk.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab
Metrics	Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.
Correction	Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

MDL Read Hits %

The MDL Read Hits % statistic is the percentage of Memory Descriptor List Read requests to the file system cache that hit the cache and does not require disk access to provide memory access to the pages in the cache.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab
Metrics	Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.
Correction	Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Pin Read Hits %

The Pin Read Hits % statistic is the percentage of pin read requests that hit the file system cache and does not require a disk read in order to provide access to the page in the file system cache.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab
Metrics	Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.
Correction	Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Space Tab

The Space tab of the OS Detail page includes the following sections:

Disk Space Free Disk Space Detail Device Summary

Device Detail

Disk Space Free

The Disk Space Free metric displays the amount of free disk space in megabytes.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab

Disk Space Detail

The Disk Space Detail section of the Space tab succinctly communicates the general overall performance levels of the server’s disks and space allotment. The table below describes the statistics in this section:

Statistic	Description
Partition	The drive letter of the disk.
Local Filesystem	The name of the file system.
Type	The type of file system.
Total Space	Total size of the disk/device's capacity expressed in MBs.
Used Space	Amount of MBs currently allocated on the particular disk/device.
Free Space	Amount of MBs currently unallocated and free on the particular disk/device.
Capacity	The percentage of space used on the device.
Mounted On	The mount point of the device.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab

Device Summary

The Device Summary section of the Space tab displays a bar chart for all devices. The Device Summary section succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

Statistic	Description
Used	Amount of MBs currently allocated on the particular disk/device.
Free	Amount of MBs currently unallocated and free on the particular disk/device.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab

Device Detail

The Device Detail section of the Space tab succinctly communicates the general overall performance levels of the server's disks and space allotment. The statistics available in this section depend on the platform of operating system. The table below describes the statistics in this section:

Statistic	Description
Partition	The drive letter of the disk.
Local Filesystem	The name of the file system.
Type	The type of file system.
Total Space	Total size of the disk/device's capacity expressed in MBs.
Used Space	Amount of MBs currently allocated on the particular disk/device.
Free Space	Amount of MBs currently unallocated and free on the particular disk/device.
Capacity	The percentage of space used on the device.
Mounted On	The mount point of the device.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Memory Tab

Network Tab

The Network tab of the OS Detail page succinctly communicates the general overall performance levels of the server's networking. The Network tab of the OS Detail page includes the following sections:

Network Details	Packet Collisions	Packet Discards
LPacket Errors	Transmission Rate	Transmission Rate (Bytes)
Transmission Queue Length		

Note: The sections available on this tab depend on the platform of operating system.

Network Details

The statistics available in this section depend on the platform of operating system. The table below describes the information available in this section:

Column	Description
Network Interface	The name of network interface.
INET Address/Address	The IP address assigned to the network interface.
Pkts Sent/Sec	The number of packets sent per second.
Pkts Received/Sec	The number of packets received per second.
Sent (KB/Sec)	The number of bytes sent per second.
Received (KB/Sec)	The number of bytes received per second.
Out Pkts Discarded	The number of outbound packets discarded.
In Pkts Discarded	The number of inbound packets discarded.
Out Pkt Errors	The number of outbound packet errors.
In Pkt Errors	The number of inbound packet errors.
Queue Length	The queue length.
Collisions	The number of collisions.
Packets Discarded	The number of packets discarded.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Network Tab

Transmission Rate

The Transmission Rate section of the Network tab succinctly communicates the packet transmission rate. The following statistics are available in this section:

- Location
- Packets Sent/Sec

Packets Sent/Sec

The Packets Sent/Sec statistic is the number of packets sent over each network adapter per second.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Network Tab

Packets Received/Sec

The Packets Received/Sec statistic is the number of packets received over each network adapter per second.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Network Tab

Transmission Rate (Bytes)

The Transmission Rate (Bytes) section of the Network tab succinctly communicates the packet transmission rate. The following statistics are available in this section:

Received (KB/Sec) Sent (KB/Sec)

Sent (KB/Sec)

The Sent (KB/Sec) statistic is the rate at which bytes are sent over each network adapter including framing characters.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Network Tab

Received (KB/Sec)

The Received (KB/Sec) statistic is the rate at which bytes are received over each network adapter, including framing characters.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Network Tab

Transmission Queue Length

The number of pending outgoing packets on either the software transmit queue or the hardware transmit queue

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Network Tab

Packet Collisions

A collision occurs when two devices on the network sense that the network is idle and end up trying to send data at the same time. Some collisions are normal on a repeated network, but excess collisions can cause serious performance problems.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Network Tab

Correction

Contact your network administrator if there is high rate of persistent Packet Collisions.

Packet Discards

The Packet Discards section of the Network tab succinctly communicates the packet discard rate. The following statistics are available in this section:

Inbound Packets Discarded Outbound Packets Discarded Packet Discards

Outbound Packets Discarded

The Outbound Packets Discarded statistic is the number of outbound packets that were discarded even though no errors had been detected to prevent from being transmitted.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Network Tab
Correction	A possible reason for discarding such a packet could be to free up buffer space.

Inbound Packets Discarded

Received Packets Discarded is the number of received packets that were discarded even though no errors had been detected to prevent from being transmitted.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Network Tab
Correction	A possible reason for discarding such a packet could be to free up buffer space.

Packet Discards

The Packet Discards statistic represents the number of network packets discarded.

LPacket Errors

ocation	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Network Tab
Correction	Contact your network administrator if there is high rate of persistent packet discards.

The Packet Errors section of the Network tab succinctly communicates the packet error rate. The following statistics are available in this section:

Inbound Packet Errors Outbound Packet Errors

Outbound Packet Errors

The Outbound Packet Errors statistic is the outbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Network Tab

Inbound Packet Errors

The Packets Received Errors statistic is the received packets that contained errors preventing them from being deliverable to a higher-layer protocol.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **OS View (DBArtisan - SQL Server Performance Analyst)** > Network Tab

Top SQL View (DBArtisan - SQL Server Performance Analyst)

One or two bad queries can cause a lot of trouble for the remaining sessions in a database, and so can one or two bad queries. It is important to find them before they get into a production system, but sometimes a few sneak through.

By applying custom filters and performance-related thresholds, the Top SQL view locates inefficient SQL. By applying filters to certain I/O and statistical counters, you hope to isolate queries that far exceed their nearest competitors in the same area (like disk reads). When you find them, you should reduce the number of sorts a query performs. Or, for a query that returns only a few records, you should try to minimize the number of rows a query processes.

Top SQL displays requested SQL for SQL Server datasources. The following tabs are available on the Top SQL page:

Summary Tab Criteria Tab SQL Details Tab

Summary Tab

The Summary tab of the Top SQL page includes the following sections:

Top SQL Statements - SQL Server Top SQL Details

Top SQL Statements - SQL Server

A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times.

Before you can identify problem SQL in your database, you have to ask the question of what 'bad SQL' is. What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every Oracle professional's satisfaction. The table below describes some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

Criteria	Description
Overall Response (Elapsed) Time	This is how much time the query took to parse, execute, and fetch the data needed to satisfy the query. It should not include the network time needed to make the round trip from the requesting client workstation to the database server.
CPU Time	This is how much CPU time the query took to parse, execute, and fetch the data needed to satisfy the query.
Physical I/O	Often used as the major statistic in terms of identifying good vs. bad SQL, this is a measure of how many disk reads the query caused to satisfy the user's request. While you certainly want to control disk I/O where possible, it is important that you not focus solely on physical I/O as the single benchmark of inefficient SQL. Make no mistake, disk access is slower than memory access and also consumes processing time making the physical to logical transition, but you need to look at the entire I/O picture of a SQL statement, which includes looking at a statements' logical I/O as well.
Logical I/O	This is a measure of how many memory reads the query took to satisfy the user's request. The goal of tuning I/O for a query should be to examine both logical and physical I/O, and use appropriate mechanisms to keep both to a minimum.
Repetition	This is a measure of how often the query has been executed. A problem in this area is not as easy to spot as the others unless you know your application well. A query that takes a fraction of a second to execute may still be causing a headache on your system if it is executed erroneously over and over.

There are other criteria that you can examine like sort activity or access plan statistics (that show items like Cartesian joins and the like), but more often than not, these measures are reflected in the criteria listed above.

Fortunately, Microsoft records some of these statistics and Performance Analyst computes the rest, which makes tracking the SQL that has been submitted against a SQL Server much easier.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Top SQL View (DBArtisan - SQL Server Performance Analyst)** > Summary Tab
Metrics	When you begin to look for inefficient SQL in a database, there are two primary questions you need to answer: *What has been the worst SQL that's historically been run in my database?What is the worst SQL that's running right now in my database?* When Correction a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It may be that the query is in an inefficient Transact SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

Top SQL Details

A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times.

Before you can identify problem SQL in your database, you have to ask the question of what 'bad SQL' is. What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every SQL Server professional's satisfaction. The Top SQL Details Section displays the top SQL statements across various performance categories. The table below describes the information available in this section:

Information	Description
Top For	Indicates the performance category for the SQL statement.
Usage	This displays the percent usage for this SQL statement in this performance category.
SQL text	A snippet of the actual SQL statement or code object (procedure, etc.)
Total Reads	The total physical reads caused by the statement since monitoring began.
Total Writes	The total physical writes caused by the statement since monitoring began.
Executions	The total number of times the statement or code object has been executed since monitoring began.
Avg CPU Time	The average CPU time used by the statement each time it is executed.
Avg Elapsed Time	The average elapsed response time experienced each time the statement is executed.

Location	Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Top SQL View (DBArtisan - SQL Server Performance Analyst)** > Summary Tab
Metrics	When you begin to look for inefficient SQL in a database, there are two primary questions you need to answer: *What has been the worst SQL that's historically been run in my database?What is the worst SQL that's running right now in my database?* When Correction a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It may be that the query is in an inefficient Transact SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

Criteria Tab

The Top SQL view displays all SQL running or executed in the current datasource. You can use the lists and boxes to reorder the display according to your specific needs. The Criteria tab of the Top SQL page includes the following sections that you can mix and match to give you the information you want to find.

- Demographic Criteria
- Statistical Criteria
- Sort Criteria

SQL Details Tab

The SQL Details tab of the Top SQL page includes the following sections:

Top SQL Results Location

Top SQL Results

Overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements or stored procedures can drag an otherwise well-configured database down in terms of end user response times.

Before you can identify problem SQL in your database, you have to ask the question of what 'bad SQL' is. What criteria do you use when hunting for problem SQL in critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every SQL Server professional's satisfaction. The Top SQL results grid displays the top SQL statements across various performance categories. The table below describes the information available in this section:

Information	Description
SQL Text	A snippet of the full SQL statement. Clicking the statement will cause the full statement to be presented in the Selected SQL Text grid.
Database	The database where the statement or code object's execution originated.
Avg Reads	The average physical reads caused by the statement each time it is executed.
Total Reads	The total physical reads caused by the statement since monitoring began.
Avg Writes	The average physical writes caused by the statement each time it is executed.
Total Writes	The total physical writes caused by the statement since monitoring began.
Executions	The total number of times the statement or code object has been executed since monitoring began.
Avg Elapsed Time	The average elapsed response time experienced each time the statement is executed.
Elapsed Time	The total elapsed response time over all executions.
Avg CPU Time	The average CPU time used by the statement each time it is executed.
Total CPU Time	The total CPU time over all executions.
First Start Time	The date/time when the statement was first executed.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Top SQL View (DBArtisan - SQL Server Performance Analyst)** > SQL Details Tab

Selected SQLText

The Selected SQL Text window allows you to select any of the Top SQL result rows and get the complete SQL text. From here you can open an Explain Plan or Tune the SQL.

Location Microsoft SQL Server Performance Analyst Statistics (DBArtisan) > **Top SQL View (DBArtisan - SQL Server Performance Analyst)** > Summary Tab

Performance Analyst for Sybase ASE Tutorial (DBArtisan)

Embarcadero Performance Analyst provides key insight into the complex world of database performance. Running inside the Embarcadero DBArtisan console, or standalone in a Web browser, Performance Analyst helps you pinpoint performance problems that threaten a database's availability or response times. The powerful analysis facilities will satisfy advanced experts, yet novices will find Performance Analyst's interface easy to use and the expert help a welcome aid to troubleshooting difficult performance issues.

About This Performance Analyst for Sybase ASE Tutorial

This tutorial is intended to help you get started with Embarcadero Performance Analyst.

After completion of this tutorial, you will have the foundation you need to explore the many features and benefits of Performance Analyst. You will have learned how to accurately monitor your database using the most popular performance methodologies in use today. You will know how to drill down into the many layers of database performance and also be able to locate performance issues in your database in a fast and effective manner. Finally, you'll understand how to fix performance problems using the integration that exists between Performance Analyst and Embarcadero DBArtisan.

This tutorial is divided into 7 sections.

Session 1: Getting Started With Performance Analyst for Sybase ASE (DBArtisan)

Session 2: Using Ratio-based Performance Analysis in Performance Analyst for Sybase ASE (DBArtisan)

Session 3: Using Bottleneck/wait-based Performance Analysis in Performance Analyst for Sybase ASE (DBArtisan)

Session 4: Using Workload Performance Analysis in Performance Analyst for Sybase ASE (DBArtisan)

Session 5: Using Operating System Performance Analysis in Performance Analyst for Sybase ASE (DBArtisan)

Session 6: Identifying and Getting Help With Performance Problems Using Performance Analyst for Sybase ASE (DBArtisan)

Session 7: Fixing Identified Problems (DBArtisan)

You can use this basic tutorial as a roadmap of product highlights.

After starting this tutorial, you can select Help from the menu to find many additional resources that complement and build on many of the activities shown in this brief guide.

For more information, see [DBATutorial.htm#AdditionalEvaluationResources Additional Evaluation Resources].

Session 1: Getting Started With Performance Analyst for Sybase ASE (DBArtisan)

Download and Install

You can obtain the latest version of Performance Analyst by downloading the DBArtisan software from the Embarcadero Web site at http://www.embarcadero.com/downloads/downloaddbartisan.asp. Performance Analyst runs fully contained within DBArtisan so you will need the DBArtisan product to use Performance Analyst (although Performance Analyst can also run in standalone fashion within a Microsoft Internet Explorer Web browser).

Click Download, and follow the steps indicated. Save the file on your computer and then double-click to launch the self-extracting file that will guide you through the installation process.

When you first install an evaluation copy of DBArtisan, you can use the tool for 14 days. After that time, a permanent license is needed.

Proceed to Overview of Performance Analyst for Sybase ASE

Overview of Performance Analyst for Sybase ASE

When DBArtisan is connected to a Sybase ASE database, you can open Performance Analyst two ways:

1 On the Analyst toolbar click the Performance Analyst icon.

OR

2 On Utilities menu, select Performance Analyst.

For Performance Analyst to work, several procedures need to be installed by the SA account. When Performance Analyst is first invoked, it will check for the presence of this package and if it has not been installed, the product will prompt you to install it. Once created, any account with access to these procedures can monitor databases using Performance Analyst. No other objects need to be installed on a monitored database.

Proceed to Performance Analyst for Sybase ASE Product Navigation

Performance Analyst for Sybase ASE Product Navigation

Navigating through Performance Analyst is easily accomplished through two main mechanisms:

1 The main Performance Analyst toolbar contains buttons that invoke the major performance analysis views within the product as well as frequently used actions like Refresh and Performance Threshold Customizations.

2 Positioning the mouse pointer on any area within Performance Analyst and clicking the right mouse button will bring up a navigation menu that allows you to easily move between any of the main Performance Analyst home pages and drill down views. The right mouse menu also contains

context sensitive options like drilling down further into a particular statistic, copying the display to a text file, and more.

Proceed to Session 2: Using Ratio-based Performance Analysis in Performance Analyst for Sybase ASE (DBArtisan)

Session 2: Using Ratio-based Performance Analysis in Performance Analyst for Sybase ASE (DBArtisan)

Ratio-based analysis has been around for many years, and used to be the only technique database administrators used when they were called on to diagnose the cause of a database slowdown. Performance ratios are very good roll-up mechanisms for busy DBAs, making possible the analysis-at-a-glance approach, however ratios must be computed correctly for them to be meaningful. Key ratios in Performance Analyst are calculated using delta statistics, which is superior to computing ratios with cumulative values. Delta statistics, which are values derived by using the before and after measures in each statistical refresh done in Performance Analyst, lets you see changes in performance metrics that would otherwise be invisible if cumulative statistics were used. Performance Analyst makes it easy to see the ratios that are computed with delta statistics vs. cumulative. A triangle symbol denotes a delta-based statistic, while square symbols represent a cumulative-based measure.

Ratio-based analysis is available for all performance categories (memory, etc.) where ratios can assist in determining the performance level of a database. For example, if you would like to see memory-specific ratios, on the Performance Analyst toolbar click Memory. Performance Analyst displays the Memory home page that showcases all important memory-related ratios.

Proceed to Obtaining Detail Information for a Ratio Using Performance Analyst for Sybase ASE

Obtaining Detail Information for a Ratio Using Performance Analyst for Sybase ASE

A ratio gives you a global picture for a particular performance measure. However, you often need more information regarding a statistical measure so you can make an informed decision. Performance Analyst has easy-to-use drill down features that make obtaining detailed information for various ratios a breeze.

For example, to view more information for the Data Cache Hit Ratio statistic, do the following:

1 On either the Main Home page or the Memory Home page, locate the Data Cache Hit Ratio statistic.

2 Position the pointer over the statistic.

The pointer changes to a hand symbol.

3 Double-click the left mouse button.

Performance Analyst opens the **Data Cache** tab of the memory drill down view.

OR

4 Right-click the statistic, select Details.

OR

5 Select Navigate to, Memory, Data Cache.

Performance Analyst opens the **Data Cache** tab of the memory drill-down view.

Proceed to Session 3: Using Bottleneck/wait-based Performance Analysis in Performance Analyst for Sybase ASE (DBArtisan)

Session 3: Using Bottleneck/wait-based Performance Analysis in Performance Analyst for Sybase ASE (DBArtisan)

When a Sybase ASE database is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things, or it can be an indicator that a database bottleneck exists. Bottleneck analysis is a valid method of measuring performance because it helps you track where a database and user sessions have been spending their time. If latch contention or heavy table-scan activity has been dragging a database’s performance down, you can use bottleneck analysis to confirm the actual root cause.

Performance Analyst uses three broad areas of metrics to form its bottleneck analysis. The first are contentions and conflicts as reported in Sybase's monitoring sp_sysmon procedure, as well as overall lock statistics. The second is the Sybase wait event interface (available only in Sybase 12.5.03 and higher), which displays where the database and user sessions have been bottlenecked. A third area of bottleneck information has to do with space related bottlenecks like databases and transaction logs that are running low on available free space.

Performance Analyst prominently displays bottleneck analysis on every performance home page. The Main home page, displays a summary of all key bottlenecks threatening the database. You can also see specific bottlenecks for memory, I/O, space, objects, and users by navigating to each home page and viewing the Bottleneck Analysis section for that home page. For example, to see bottlenecks that specifically relate to I/O, do the following

1 On the toolbar, click the I/O button.

2 Locate the Bottleneck Analysis section at the lower-left portion of the screen.

The I/O bottleneck analysis section shows the I/O related headaches, along with the most active devices.

Proceed to Obtaining Detail on System Bottlenecks Using Performance Analyst for Sybase ASE

Obtaining Detail on System Bottlenecks Using Performance Analyst for Sybase ASE

You can easily obtain more information regarding global bottlenecks that are plaguing a database by drilling down into a particular bottleneck metric. For example, if you would like to see more data regarding the current top system bottlenecks, do the following:

1 Right-click anywhere in Performance Analyst, point to Users, and then click System Waits.

Performance Analyst opens the System Waits view.

The System Waits view (available in Sybase ASE 12.5.03 and higher with the MDA monitoring views installed) displays critical details regarding all the wait events that Sybase has recorded. You can easily see what events that occurred the most often as well as the events that have accumulated the most wait time.

Tip: If you would like to see an expanded view of the System waits grid shown on the top of the System Waits view, click the Expand button located at the top right corner of the section. All section views in Performance Analyst can be maximized/minimized using this control.

Proceed to Viewing Detail for Session Bottlenecks Using Performance Analyst for Sybase ASE

Viewing Detail for Session Bottlenecks Using Performance Analyst for Sybase ASE

Not only can you obtain detail information for system bottlenecks, but you can also use Performance Analyst to get drill down information regarding session-level bottlenecks. There are several layers to session-level details you can view.

To see more information regarding session-level waits, do the following:

1 Right-click anywhere in Performance Analyst, point to Users, and then click Session Waits.

Performance Analyst opens the Session Waits drill-down view.

When you are in the Session Waits drill down view, to drill further into a session's details, do the following:

1 In the Historical Waits or Current Waits grid, select a session that appears to be experiencing critical waits.

2 Double-click the session or right-click, and then select Details.

Performance Analyst opens a drill-down view that shows key statistics for the selected session, including its current work activities, wait events, and SQL statements.

Proceed to Session 4: Using Workload Performance Analysis in Performance Analyst for Sybase ASE (DBArtisan)

Session 4: Using Workload Performance Analysis in Performance Analyst for Sybase ASE (DBArtisan)

Key ratios give you a global perspective on database activity. Bottleneck analysis gives you insight into things that are holding up user activity and throughput. But another technique is necessary if a database professional is to really get a handle on what's occurring inside a badly performing database.

Workload analysis involves the investigation of two critical areas of a database's performance:

- Session resource consumption and activity
- SQL execution (if available)

Without looking at these two key performance categories, you can miss a lot of what could be responsible for perceived performance problems. When performance on a database takes a sudden nosedive, it is not uncommon to find one or two sessions that are causing the bulk of the workload. Finding these processes can be easily accomplished by viewing session metadata coupled with resource consumption and statistical execution statistics.

Proceed to Pinpointing Top Resource Consumers

Pinpointing Top Resource Consumers

Performance Analyst identifies top resource consumers in a couple of different ways. On the main home page, Performance Analyst highlights the top resource sessions across I/O, memory usage, and CPU consumption. The percentage used across all statistical categories are displayed so a DBA can immediately pinpoint a session that is using all or most of a particular resource. The top resource sessions are also displayed on the memory, I/O, and other home pages as well with a listing for the top sessions for that category (for example, the top memory users appear on the memory home page, etc.)

Proceed to Getting Workload Details

Getting Workload Details

Performance Analyst offers several details on top resource sessions in addition to the summaries provided on each home page. A detailed and flexible top sessions view lets you view a summary of all the major resource sessions on the database as well as every critical detail regarding each session connected to the database.

Further drill downs are available that highlight specific details regarding each session's activities and submitted SQL. To see detailed information regarding a session identified in one of Performance Analyst's Workload Analysis sections, do the following:

1 In a Workload Analysis display, position the pointer over one of the processes.

2 Double-click the process or right-click, and then select Details.

If you want like to see all key workload analysis metadata for sessions, do the following:

1 On the toolbar, click the Top Sessions button.

OR

2 Right-click anywhere in Performance Analyst, point to Users, and then click Session Waits.

Proceed to Session 5: Using Operating System Performance Analysis in Performance Analyst for Sybase ASE (DBArtisan)

Session 5: Using Operating System Performance Analysis in Performance Analyst for Sybase ASE (DBArtisan)

In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database sever for resources (CPU, I/O or Memory), a slow CPU, insufficient I/O devices, and insufficient memory. Performance Analyst allows you to examine the relevant operating system metrics for Windows Platforms (NT /2000/XP/2003) as they relate to Sybase ASE.

To view the operating system metrics of the machine hosting the Sybase ASE server you are monitoring, do the following:

1 Click the OS Performance Analyst toolbar button.

OR

2 Navigate to the OS Home Page by going to OS, and then Home.

Note: The first time you invoke the OS feature you will be prompted for an operating system login. Please specify the correct operating system, host name, username and password, and protocol.

The Operating System home page has been designed in a similar fashion as the datasource home page. On this page you can see the pertinent operating system metrics displayed in five quadrants broken down into the following categories:

- Resource Usage
- Memory Analysis
- Disk Analysis
- Workload Analysis
- Bottleneck Analysis

Proceed to Obtaining Details on the Operating System Using Performance Analyst for Sybase ASE

Obtaining Details on the Operating System Using Performance Analyst for Sybase ASE

You can easily obtain more information regarding operating system problems that are degrading database performance by drilling down into a particular analysis quadrant. For example if you'd like to see more information regarding the processes running on the host machine, do the following:

1 Navigate to the Performance Analyst Operating System home page.

2 Double-click the Top Memory Process in the Workload Analysis quadrant.

Performance Analyst opens the Processes detail view.

OR

3 Right-click anywhere in Performance Analyst and select the Navigate to, OS, and then Processes.

The Processes detail tab shows all the processes on the host machine. From this view you can easily determine which processes are showing high marks in I/O or CPU, for example, by simply double clicking on the column headers and sorting the processes in ascending or descending order.

Proceed to Session 6: Identifying and Getting Help With Performance Problems Using Performance Analyst for Sybase ASE (DBArtisan)

Session 6: Identifying and Getting Help With Performance Problems Using Performance Analyst for Sybase ASE (DBArtisan)

You do not have to be an advanced expert in performance tuning to spot trouble in a database if you use Performance Analyst. There are a number of mechanisms in the tool that help you quickly locate areas of concern within a database.

Proceed to Using the Alarm Log in Performance Analyst for Sybase ASE

Using the Alarm Log in Performance Analyst for Sybase ASE

Performance Analyst has a central and flexible alarm log that lists every performance violation found in all monitored databases. Unlike some products that only show you alarms for a particular database, Performance Analyst lists all alarms for each database you are monitoring in one central log so you can see every problem in one place. To view the Performance Analyst Alarm log, do the following:

1 On the toolbar, click the Alarms button.

Once in the Alarm log, you can perform a number of actions including:

- Selecting an alarm and viewing the application page that shows details regarding the metric in violation.
- Filtering the types of alarms shown through the Options dialog.
- Clearing all current alarms.
- Copying or saving the Alarm log into a different format.

To select any of these options, do the following:

1 Position the pointer over any area of the Alarm Log.

2 Right-click and in the shortcut menu, select an option.

Proceed to Recognizing Performance Violations

Recognizing Performance Violations

Performance Analyst makes it easy to visually pinpoint performance violations in a database. Any statistic that does not conform to your predefined performance standards is colorized in any view that it appears so you can easily see potential troublespots.

The colors used to highlight a particular performance violation depend on the severity of the problem. Performance Analyst uses several layers of performance standards/threshold violations with each layer being represented by a different color. You can control everything regarding threshold violations, including how thresholds are determined, their color representation in

Performance Analyst's display, and more by using the Threshold Template feature of Performance Analyst.

How can Performance Analyst inform you of performance violations in your database if you minimize the product to your desktop and are working on other things? Even if Performance Analyst is not right in front of you, it can still let you know that a problem has been found in your database through the systray alarm icon. When Performance Analyst detects a problem in one of your databases, one of the things the product does is enable an alarm icon in your workstation's systray area that communicates the presence of alarms in your database. So even if the Performance Analyst interface is not in sight, the product can let you know that something in your database requires attention.

Proceed to Creating Threshold Templates

Creating Threshold Templates

How does Performance Analyst know what to signal as a problem in a database? It uses predefined performance standards that you define and enforce. All of this is accomplished by using Performance Analyst's Threshold templates. A template is a set of performance metrics and their thresholds that can be reused to enforce performance standards across any number of databases. Embarcadero provides a default performance template that contains nearly all of the most-watched performance statistics so you can begin monitoring with confidence right out of the box. However, you can create your own Threshold templates so you only are alerted to performance violations that are important to you and your particular environment.

To view or create Threshold templates, do the following:

1 On the toolbar, click the Thresholds button.

Performance Analyst opens the **Threshold Template Properties** dialog.

The **Threshold Template Properties** dialog lets you select the options to add or edit templates, as well as other maintenance options. For example, you can copy the original Embarcadero Threshold template and modify it to fit your own needs. You can control what statistics are sampled for performance violations as well as the levels of severity you would like reported. Once you are satisfied with your template, you can save it and then apply it to one or more datasources in one action to immediately begin enforcing your performance standards.

Proceed to Getting Help With Performance Problems

Getting Help With Performance Problems

As databases become more complex, so do the problems that plague them. Knowing what to do for a particular performance problem can be hard in some cases, but Performance Analyst helps by providing a complete set of expert help that you can use as a guide when performance problems occur. To obtain expert help for any statistic shown in Performance Analyst, do the following:

1 Position the pointer over a statistic that you need help.

2 Right-click and select Help.

The Expert Help gives you a general description of the statistic, the range of values that are considered optimal, and some suggestions on how to fix a problem situation. Expert help can also be invoked for any of the Ratio, Bottleneck, SQL, or Workload Analysis sections shown on every home page by clicking on the help icon for that section, which is located at the top right corner of the section.

Proceed to Session 7: Fixing Identified Problems (DBArtisan)

Session 7: Fixing Identified Problems (DBArtisan)

Finding performance problems is one thing, but fixing them is another. Many performance monitors point out a particular problem in a database, but then offer no way to actually correct the situation. Such is not the case with Performance Analyst. Because of its tight integration with DBArtisan, certain problems found in Performance Analyst can be visually corrected by DBArtisan.

A Space Problem Correction Example

If a Sybase ASE Server tablespace is about to run out of free space, Performance Analyst can issue an alarm that a space shortage exists. Once the space problem has been identified, do the following:

1 On the Space home page toolbar, click the Space button.

2 In the Space Analyst grid, right-click a tablespace, and then select Manage with DBArtisan.

DBArtisan opens an editor where you can perform corrective actions, such as enabling Autoextend for the tablespace, resizing the tablespace’s existing datafile(s), or adding a new datafile for the tablespace.

Sybase ASE Performance Analyst Statistics (DBArtisan)

This section includes expert help for all Sybase ASE categories and statistics in the Embarcadero Performance Analyst views and pages. This help is divided into the following sections:

Home View (DBArtisan - Sybase ASE Performance Analyst)	Memory View (DBArtisan - Sybase ASE Performance Analyst)	I/O (DBArtisan - Sybase ASE Performance Analyst)
Space (DBArtisan - Sybase ASE Performance Analyst)	Databases (DBArtisan - Sybase ASE Performance Analyst)	Users (DBArtisan - Sybase ASE Performance Analyst)
OS (DBArtisan - Sybase ASE Performance Analyst)	Top SQL Statistics (DBArtisan - Sybase ASE Performance Analyst)	

Note: Sybase ASE 12.5.0.3 users should make sure their MDA tables are properly configured so they can see all available statistics. Please refer to the instructions for Enabling MDA Tables (DBArtisan).

Enabling MDA Tables (DBArtisan)

In ASE 12.5.0.3, a new feature called "MDA tables" was introduced. These proxy tables are mapped to RPCs which provide access to low-level monitoring information about the server. Refer to Sybase documentation for details on MDA tables. For assistance, see Accessing Third Party Documentation.

DBArtisan's Performance Analyst for Sybase ASE requires these tables be installed to enable several statistics, for example the Top SQL view.

Please follow the instructions below to install and configure these tables.

1 Ensure the 'enable cis" configuration parameter is set to 1.

```
sp_configure 'enable cis'

go
```

2 Add the 'loopback' server name alias.

```
if not exists (select 1 from master..sysservers where srvname = "loopback")

BEGIN

exec sp_addserver "loopback", null, <enter your server name>

END
```

3 Install the MDA tables.

The script can be found in the following location:

$SYBASE/$SYBASE_ASE/scripts/installmontables.

4 Assign the 'mon_role' to logins allowed MDA access.

Note: The 'sa' login requires this grant as well.

```
use master

go

grant role mon_role to sa

go
```

5 Enable the following configuration parameters.

```
exec sp_configure "enable monitoring", 1

exec sp_configure "sql text pipe active", 1

exec sp_configure "sql text pipe max messages", 2000

exec sp_configure "plan text pipe active", 1

exec sp_configure "plan text pipe max messages", 2000
exec sp_configure "statement pipe active", 1
exec sp_configure "statement pipe max messages", 2000
exec sp_configure "errorlog pipe active", 1
exec sp_configure "errorlog pipe max messages", 2000
exec sp_configure "deadlock pipe active", 1
exec sp_configure "deadlock pipe max messages", 2000
exec sp_configure "wait event timing", 1
exec sp_configure "process wait events", 1
exec sp_configure "object lockwait timing", 1
exec sp_configure "SQL batch capture", 1
exec sp_configure "statement statistics active", 1
exec sp_configure "per object statistics active", 1
exec sp_configure "max SQL text monitored", 2000
```

Home View (DBArtisan - Sybase ASE Performance Analyst)

The Embarcadero Performance Analyst Home view lets you review availability and overall performance of all monitored databases from a single window. Statistics on the Home view are organized into the following categories:

Bottleneck Analysis Pane Key Ratio Analysis Pane I/O Analysis Pane

Storage Analysis Pane Workload Analysis Pane

Key Ratio Analysis Pane

Database performance analysts typically use one of two methods for examining the performance levels of a database - ratio-based or wait/bottleneck-based. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as very good roll-up mechanisms for busy DBAs to use for at-a-glance performance analysis. Many DBAs have large database farms to contend with and cannot spend time checking detailed wait-based analysis outputs for each and every database they oversee. Succinctly presented performance ratios can assist in such situations by giving DBAs a few solid indicators that can be quickly scanned to see if any database needs immediate attention.

While there are certainly many opinions as to what rules to follow, there are some standards that should always be adhered to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that a DBA will examine come from the master.sysperfinfo performance table. This table maintains a count of various statistics for the server and particular databases since the server was brought up. For servers that are kept up for long periods of time, these values can grow quite large and will impact how a particular ratio that a DBA might be looking at is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each SQL Server's individual 'personality' with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead the DBA to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following ratios are used on the Performance Analyst Home page to succinctly communicate the general overall performance levels of the monitored database:

Data Cache Hit Ratio
Procedure Plan Hit Ratio
Large I/O Hit Rate
Clean Buffer Grab Rate
CPU Busy
I/O Busy
Server Idle

Data Cache Hit Ratio

Data read from memory produces user response times many times faster than when that same data is read from disk. The Sybase ASE data cache assists with keeping physical I/Os to an absolute minimum.

The data cache hit ratio is an excellent indicator of how often user requests for data are satisfied through memory as opposed to being physically read from disk. The table below describes the three key counters in Sybase ASE used to arrive at this statistic:

Counter	Description
LOGICAL READS	Data read from memory for user requests.
PAGES PER I/O	The number of pages retrieved in a single I/O operation.
PHYSICAL READS	Data read physically from disk.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Key Ratio Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > Memory View (DBArtisan - Sybase ASE Performance Analyst) > Home> Key Ratio Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > Memory View (DBArtisan - Sybase ASE Performance Analyst) > Data Cache Tab
Metrics	To help ensure excellent performance, keep your cache hit rate in the neighborhood of 90% or greater. Lower amounts can be okay for user ad hoc databases where sporadic, large table scan operations occur. However, anything below this general threshold for normal databases can require tuning attention, and the adjustment of the Sybase ASE memory tuning parameters. If you are using named caches, you can drill down into the cache hit rates for each named cache. This helps you understand which objects/operations are depressing the overall cache hit rate for the server.
Correction	If a problem is found in Sybase ASE servers, versions 11-12, you can increase the amount of the total memory configuration parameter or reduce the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase ASE has met its kernel and procedure cache needs). Take care when reducing the procedure cache, as this could reduce performance in the server as it relates to reading procedures in from disk. For Sybase ASE 12.5, the total memory configuration parameter can again be increased to provide more memory for the data cache (and any named caches), but in 12.5, if you want to reduce the size of the procedure cache, note that it is now configured in terms of literal size instead of a percentage of the overall configured memory. Once the data cache has been adjusted, monitor Sybase ASE to see if the cache hit rate improves. If it does not, another increase may be necessary and examination of unnecessary large table scan operations. Also, keep a careful eye on the actual machine's memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.

Procedure Plan Hit Ratio

The Sybase ASE procedure cache is used to hold the definitions and query plans of stored procedures and triggers. It is used for short-term memory needs like statistics and query plans for parallel queries. When a user executes a stored procedure, Sybase ASE looks in the procedure cache for a query plan to use. If a query plan is available, Sybase ASE places it on the most recently used (MRU) end of the memory chain and the procedure begins to execute. If no execution plan is in memory, or if all copies of the plan are currently being used, the query tree for the procedure is read in again from the data dictionary, optimized, put on the MRU end of the chain, and executed. Note that other operations, like CREATE INDEX, can also use the procedure cache even when no procedure is referenced.

The more often that a procedure's plan and definition can be referenced in memory, the better the procedure execution time.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Key Ratio Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > Memory View (DBArtisan - Sybase ASE Performance Analyst) > Home> Key Ratio Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > Memory View (DBArtisan - Sybase ASE Performance Analyst) > Procedure Cache Tab
Metrics	A high procedure cache hit rate is a desirable thing. You should strive for a hit ratio between 95-100%, with 95% being a good performance benchmark for procedure code reference. Note that when a database is first started, the procedure cache hit rate is not at an optimal level because all code being used is relatively new, and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the procedure cache hit rate has not increased to desirable levels, you should look into the possibility of increasing the amount of memory allocated to the cache. You can drill down into the procedure cache to view the procedures currently in memory along with how much memory they are consuming. If there is not enough memory to load a requested procedure, or the maximum number of compiled objects is already in use, Sybase ASE returns an error (normally a 701).
Correction	If a problem is found in Sybase ASE servers, versions 11-12, you can increase the amount of the total memory configuration parameter or increase the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase ASE has met its kernel and procedure cache needs). You should be careful when increasing the procedure cache alone, as this could increase query response times due to more physical I/O being performed. For Sybase ASE 12.5, the total memory configuration parameter can again be increased to provide more memory for the Sybase ASE server, but in 12.5, if you want to increase the size of the procedure cache, note that it is now configured in terms of literal size instead of a percentage of the overall configured memory. Once the procedure cache has been adjusted, monitor Sybase ASE to see if the cache hit rate improves. If it does not, another increase may be necessary. Also, keep a careful eye on the actual machine's memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.

Large I/O Hit Rate

Large I/O can be enabled by splitting the default or any named cache into pools. By default, Sybase ASE performs I/O operations based on a 2-KB page size. For queries where pages are stored and accessed in a sequential manner, it is possible to read many more data pages in a single I/O operation. Large I/O can greatly reduce disk access time when the right situations exist. Operations that routinely perform large table scans, access image or text data, do bulk copies, scan the leaf level of nonclustered indexes, or initiate DBCC tasks can benefit from large I/O.

If large I/O has been configured and is being used, you should observe a high percentage of hits (the number of times large I/O could be performed vs. the number of times large I/O requests were denied by the server). If large I/O is not configured, no large I/O activity should be present.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Key Ratio Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > Memory View (DBArtisan - Sybase ASE Performance Analyst) > Home> Key Ratio Analysis Pane
Metrics	As you might expect, if large I/O is in use, a high hit rate is desirable. You should strive for a hit ratio between 90-100%, with 90% being a good performance benchmark.
Correction	If large I/O is configured, but you see a low hit rate, you should configure more caches for large I/O use.

Clean Buffer Grab Rate

As information is requested from users, buffers are moved into and out of the Sybase ASE data cache. Pages are also modified in the cache (termed dirty buffers) and need to be written out to disk. If Sybase ASE has to wait for a dirty buffer to be written out to disk before a requested buffer is placed into the cache, performance can suffer.

The clean buffer grab rate represents the percentage of time clean buffers were found and referenced in the cache as opposed to Sybase ASE finding dirty buffers.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Key Ratio Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > Memory View (DBArtisan - Sybase ASE Performance Analyst) > Home> Key Ratio Analysis Pane
Metrics	Ideally, the clean buffer grab rate should stay at or near 100%.
Correction	A poor clean buffer grab rate for either the default or named caches could indicate that the cache size is too small. You can look into adjusting the total memory configuration parameter higher. Monitor the actual machine's memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.

CPU Busy

The CPU Busy statistic represents the percent of the sample interval during which the Adaptive Server was busy.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Key Ratio Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > Databases (DBArtisan - Sybase ASE Performance Analyst) > Home> Server Analysis Pane
Correction	If you find that your CPU Busy value is consistently above 70%, consider adding additional hardware.

I/O Busy

This metric represents the number of clock ticks in the sample interval during which the user task performed I/O operations.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Key Ratio Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > I/O (DBArtisan - Sybase ASE Performance Analyst) > Home> Bottleneck Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > Databases (DBArtisan - Sybase ASE Performance Analyst) > Home> Server Analysis Pane
Metrics	High numbers indicate an I/O-intensive process. If idle time is also high, the application could be I/O bound.
Correction	To improve throughput: *Assign a higher priority to the application.Bind the application to a lightly loaded engine or engine group.Partition the application's data onto multiple devices.*

Server Idle

This metric represents the number of clock ticks in the sample interval during which the user task was idle.

Metrics

If you see high numbers for idle time and I/O Busy, the application could be I/O-bound.

Correction

To improve throughput:

- Assign a higher priority to the application.
- Bind the application to a lightly loaded engine or engine group.
- Partition the application's data onto multiple devices.

Bottleneck Analysis Pane

When Sybase ASE Adaptive Server is running, every connected process is either busy doing work or waiting to work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. And this is where wait-based or bottleneck analysis comes into play. DBAs use this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance because it helps a DBA track where a database has been spending its time. If lock contention or heavy table scan activity has been dragging down database performance, a DBA can use bottleneck analysis to confirm the actual root cause. Once one or more wait events or other bottlenecks have been pinpointed as possible performance vampires, the DBA can drill down and oftentimes discover a fair amount of detail about which sessions and objects are causing the problem. The Performance Analyst Home page identifies the top system and session waits that might be currently a cause of lessened performance.

When using bottleneck analysis, a DBA cannot rely only on the information contained in the wait events that Sybase Adaptive Server provides. For example, a database might be suspect or offline. Such a failure will not be reflected in any wait event, but still represents a very real bottleneck to the database. In the same way that a DBA cannot depend on only a few ratios to properly carry out ratio-based performance analysis, an administrator must include other statistical metrics in their overall bottleneck analysis framework to obtain an accurate performance risk assessment. Performance Analyst works to identify bottlenecks in your database that fall outside of pure wait events so you can get a total picture of all stoppages in your system. The Performance Analyst Home page displays space-related bottlenecks as well as other sources of contention/bottlenecks that might be dragging down the overall performance of your system.

The following bottleneck indicators are used on the Performance Analyst Home page to succinctly communicate the general overall performance level of the monitored database:

Device I/O Contention

Suspect Databases

Tempdb Percent Full

Blocked Connections

Modify Conflicts

Databases Low on Space

Task Context Switches

Deadlocks

Logs Low on Space

Group Commit Sleeps

Deadlocks

A deadlock occurs when two processes have a lock on a separate page or object and each wants to acquire a lock on the other process’ page or object. Each waits for the other to release the necessary lock. Sybase ASE constantly checks for deadlocks and, when found, chooses the transaction that has accumulated the least amount of CPU time and terminates the transaction. The server then rolls back that transaction and issues a notification of the event. The other process gets to move forward.

A deadlock occurs when two processes have a lock on a separate page or object and each wants to acquire a lock on the other process' page or object. Each waits for the other to release the necessary lock. Sybase ASE constantly checks for deadlocks and, when found, chooses the transaction that has accumulated the least amount of CPU time and terminates it (the transaction). The server then rolls back that transaction and issues a notification of the event. The other process gets to move forward.

The deadlock statistic displays the number of current deadlocks in a Sybase ASE server.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Bottleneck Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > Users (DBArtisan - Sybase ASE Performance Analyst) > Home> Bottleneck Analysis Pane
Metrics	A nonzero number should alert you to the fact that application conflicts are likely being experienced by your user community.
Correction	Well-designed applications can minimize deadlocks by always acquiring locks in the same order. You should always do updates to multiple tables in the same order. Once Sybase ASE discovers a deadlock, it takes action and remedies the situation. Embarcadero Performance Analyst makes it easier to discover how prevalent deadlock activity is on a system; preventing deadlocks from occurring in the first place is more difficult. Those responsible for writing systems can minimize deadlocks by ensuring that applications acquire locks in the same order. Likewise, you should always do updates and other DML that act on multiple tables in the same order. You can also shrink the amount of time that Sybase ASE waits to check for deadlocks by modifying the deadlock checking period configuration parameter.

Suspect Databases

It can be hard to recover quickly from a suspect database. There are times when the cache of the database is suspect and not the actual database itself. If you stop and start the Sybase ASE server, you can verify if the cache is the culprit. If the database itself is damaged, there could be a true recovery situation.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Bottleneck Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > Databases (DBArtisan - Sybase ASE Performance Analyst) > Home> Bottleneck Analysis Pane
Correction	The suspect database can be dropped using the DBCC DBREPAIR DROPDB command. You then need to re-create the database and perform a recovery operation using the most recent database dump.

Device I/O Contention

Device I/O Contention reflects the number of times a task or process was put to sleep while waiting for a semaphore for a particular database device.

When a task or process involves physical I/O, Sybase ASE first fills out the block I/O structure and links it to a per engine I/O queue. If two or more Sybase engines request an I/O structure from the same device at the exact same time, the server puts one of them to sleep where it waits for the semaphore it needs.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Bottleneck Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > I/O (DBArtisan - Sybase ASE Performance Analyst) > Home> Bottleneck Analysis Pane
Correction	If consistent high numbers are viewed for Device I/O Contention, you can try redistributing the tables across devices (that are located on separate hard disks) or by adding devices (again, that are located on separate hard disks) and moving tables and indexes to them.

Modify Conflicts

For some database operations, Sybase ASE uses a specialized protection mechanism to get exclusive access to a page without using actual restrictive page locks. Examples include accessing certain system tables and dirty reads on data pages. These actions need exclusive access to the page in question, even though they do not actually modify it. Modify conflicts record how many times these actions conflicted with other processes that need true database locks.

Tempdb Percent Full

The Tempdb Percent Full statistic shows the percentage of the Tempdb that is full.

Metrics

Tempdb is probably the most important resource in Adaptive Server shared by all applications and processes. It is a server-wide resource used for internal processing tasks like work tables, sorts, reformatting, and index builds.

Correction

Problems can be identified by a consistently high percentage registering or receiving a message indicating that your transactions are being suspended until space is made available in the log. In the short term, try using the “dump tran” DBCC command. If you suspect your application requires a very large tempdb, it should be placed on small fast disks. These disks should have no other data on them.

Note: On some systems, tempdb performs much faster using an operating system file rather than raw devices. This is because an operating system file uses buffering, which is faster than direct writes to a raw device.

Task Context Switches

Task Context Switches summarizes task-switch activity for all engines on SMP servers. You can use this statistic to observe the effect of reconfigurations.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Bottleneck Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > Databases (DBArtisan - Sybase ASE Performance Analyst) > Home> Bottleneck Analysis Pane
Metrics	You might reconfigure a cache or add memory if tasks appear to block on cache search misses and to be switched out often. Then, check the data to see if tasks tend to be switched out more or less often.

Group Commit Sleeps

For databases needing high throughput, a large log I/O size is very important to prevent disk queuing problems on the transaction log. Group commit sleeps reflect the number of times a task performed a transactional commit operation and was put to sleep by the server until data in the log was written to disk.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Bottleneck Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > I/O (DBArtisan - Sybase ASE Performance Analyst) > Home> Bottleneck Analysis Pane
Metrics	You should examine group commit sleeps in conjunction with the number of committed transactions (found on the Users performance category view).
Correction	A high count for group commit sleeps is not necessarily a problem if the server's transaction rate is low. If there are a significant number of transactions that result in group commit sleeps, and the log I/O size is greater than 2 KB, a smaller log I/O size can help to reduce commit time by causing more frequent page flushes. Other factors that can affect group commit sleeps are the size of the server run queue and the speed of the hard disk where the log is located.

Blocked Connections

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches on large systems. Although Sybase ASE supports flexible locking mechanisms, blocking lock situations do crop up. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency.

The blocked statistic displays the number of current processes blocked by other processes.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Bottleneck Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > Users (DBArtisan - Sybase ASE Performance Analyst) > Home> Bottleneck Analysis Pane
Metrics	While the lock contention rate is a better measure of the overall lock contention situation, if you consistently see positive numbers for the blocked statistic there may be a bottleneck for some processes. You can easily drill down and discover the exact process(es) holding locks that are blocking out other user activity.
Correction	Once you discover a blocking lock situation, you can normally remedy it by issuing a KILL against the offending process. This eliminates the user's stranglehold on the objects the user was accessing, and usually results in other user processes completing in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is tricky. The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do. By default, all processes wait indefinitely for locks in Sybase ASE. You can change this behavior by modifying the lock wait period configuration parameter, which limits the number of seconds that a process waits for a lock before timing out.

Databases Low on Space

The Databases Low on Space statistic indicates if any databases in the server are approaching low levels of available free space. Although good to know, you need a more detailed listing by database to determine where any actual space shortages exist in the server.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Bottleneck Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Home> Bottleneck Analysis Pane
Metrics	If any one database begins to approach 90% used space, and is continuing to dynamically grow, you should take action to prevent any future space allocation errors.
Correction	If a database is approaching its limit on space, you can either extend the database onto a new device, or if space exists on the device where the database currently resides, you can allocate more space for the database on the current device.

Logs Low on Space

This statistic indicates if any log for a database in the Sybase ASE server is approaching low levels of available free space. Although good to know, a more detailed listing by database is needed to determine where any actual space shortages exist in the server.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Bottleneck Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Home> Bottleneck Analysis Pane

Metrics If a database log's used space begins to approach 90%, you should take action to prevent any future space allocation errors. If the transaction log runs out of space, no transactions can take place in the database until you free up space in the log.

Correction If a database log consistently approaches its limit on used space, there are a few actions a DBA can take to prevent a database from freezing. A backup (dump) of the log can be taken: *If log backups are not required for disaster recovery, the truncate log on checkpoint option can be set for the database. Setting this option deletes any space devoted to transactions in the log that have already completed when a database checkpoint occurs.You can extend the log onto a new device. Or, if space exists on the device on which the database log currently resides, you can allocate more space for the log on the current device.*

I/O Analysis Pane

I/O statistics gives you a succinct overview of the datasource's overall performance. The ratios on the Home page are:

Server Reads Server Writes Transaction Log Writes

I/O Errors

Server Reads

Total Server Reads reflect the total number of physical reads performed by the database server since the last refresh.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > I/O Analysis Pane

Metrics Large numbers of physical reads could reflect a too small data or procedure cache. You should examine the data and procedure cache hit rates to determine the overall effectiveness of logical vs. physical I/O.

Server Writes

This value reflects the total number of physical writes performed by the database server since the last refresh.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > I/O Analysis Pane

Transaction Log Writes

Transaction Log Writes refers to the number of times Sybase ASE wrote a transaction log page to disk since the last refresh. When the current log page becomes full, Sybase ASE writes it out to disk. Sybase ASE also writes transaction log pages to disk after a transaction commits.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > I/O Analysis Pane

I/O Errors

The I/O Errors value reflects total number of I/O errors (errors during read and write operations) encountered by the database server since the last refresh.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > I/O Analysis Pane

Storage Analysis Pane

While DBAs focus on memory settings and tuning SQL, they frequently forget how dangerous and insidious storage problems can be. This is not a good mindset because storage headaches can play a major role in wrecking an otherwise well-running database.

Storage problems generally take one of two forms:

- The ‘hit-the-wall’ variety that can bring things to a complete standstill.
- The ‘performance vampire’ kind that slowly drains the performance of a database over time.

Storage problems have the capability to bring the curtain down on a database very quickly, as in the case of a transaction log running out of free space. But storage problems can also silently work behind the scenes to slowly, but surely rob a database of its performance. For example, a hub table in a busy database might be accessed very quickly when an application is first given life, but over time, if it develops a heavy forwarded row problem, it can cause things to run very differently.

The Storage Analysis section of the Performance Analyst Home page displays the Total Used and Free Space of all tablespaces in the database. It also provides a count of the number of databases, files, and file groups that exist on the server. For more detailed information on database space, you can use the Performance Analyst Space home page or the Space Detail.

The following ratios succinctly communicate the general overall performance levels of the datasource:

·Total Used Total Free Number of Databases

Number of Devices

Total Used

The Total Used statistic shows the total amount of space currently in use by the Sybase ASE Server.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Storage Analysis Pane

Total Free

The Total Free statistic shows the total amount of free space available to the Sybase ASE Server

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Storage Analysis Pane

Correction

A low percentage of free space indicates that additional space should be allocated to your database(s)/devices.

Number of Databases

This value indicates the total number of database currently defined on the Sybase ASE server.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Storage Analysis Pane

Number of Devices

This value indicates the total number of devices currently defined on the Sybase ASE server.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Storage Analysis Pane

Workload Analysis Pane

When your phone starts ringing with complaints of performance slowdowns, some of the first questions you should answer are:

- Who is currently connected to the database?
- What resources are they using?
- What are they currently executing?

The Workload Analysis section of the Performance Analyst Home page provides insight into the leading resource hogs of a server, as well as a count of total active and inactive processes.

Workload Analysis statistics include:

Top I/O Process Top Memory Process Top CPU Process

Active Processes Inactive Processes

Top I/O Process

The Top I/O Process statistic identifies the current SPID with the highest percentage of I/O activity.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Workload Analysis Pane
Metrics	If any one session uses more than 50% of a total resource (CPU, memory, etc.) drill-down into the session to investigate what work is currently executing.

Top Memory Process

The Top Memory Process statistic identifies the current SPID with the highest percentage of memory consumption.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Workload Analysis Pane
Metrics	If any one session uses more than 50% of a total resource (CPU, memory, etc.) drill-down into the session to investigate what work is currently executing.

Top CPU Process

The Top CPU Process statistic identifies the current SPID with the highest percentage of CPU consumption.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Workload Analysis Pane
Metrics	If any one session uses more than 50% of a total resource (CPU, memory, etc.) drill-down into the session to investigate what work is currently executing.

Active Processes

The Active Processes statistic is the total number of active and open threads reported on the Server. Active Processes displays the number of processes actively performing work.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Workload Analysis Pane

Inactive Processes

The Inactive Sessions statistic represents the total number of threads logged onto the server that are currently idle.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Home View (DBArtisan - Sybase ASE Performance Analyst)** > Workload Analysis Pane
Metrics	A large number of inactive processes could indicate user sessions that have mistakenly been left logged on. Since each user thread consumes a portion of memory on the server, to reduce resource usage, you should sever any sessions that do not need a connection.
Correction	Drill down into the Session Waits page and check sessions that have many seconds idle and/or that have been logged on for very long periods of time, as indicated by the logon time column. After verifying that a session is no longer necessary, you can KILL it.

Memory View (DBArtisan - Sybase ASE Performance Analyst)

In addition to a Home page, the Memory category of Sybase ASE Performance Analyst includes the following tabbed pages:

Data Cache Tab Procedure Cache Tab

Home

The Memory performance category view displays the following vital Sybase ASE memory statistics:

Key Ratio Analysis Pane Bottleneck Analysis Pane Metadata Cache Ratio Analysis Pane

Memory Analysis Pane Workload Analysis Pane

Key Ratio Analysis Pane

Database performance analysts typically use one of two methods for examining the performance levels of a database - ratio-based or wait/bottleneck-based. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as very good roll-up mechanisms for busy DBA's to use for at-a-glance performance analysis. Many DBAs have large database farms to contend with and cannot spend time checking detailed wait-based analysis outputs for each and every database they oversee. Succinctly presented performance ratios can assist in such situations by giving DBAs a few solid indicators that can be quickly scanned to see if any database needs immediate attention.

While there are certainly many opinions as to what rules to follow, there are some standards that should always be adhered to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that a DBA will examine come from the master.sysperfinfo performance table. This table maintains a count of various statistics for the server and particular databases since the server was brought up. For servers that are kept up for long periods of time, these values can grow quite large and will impact how a particular ratio that a DBA might be looking at is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each SQL Server's individual 'personality' with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead the DBA to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following memory ratios are used on the Performance Analyst Home page to succinctly communicate the general overall memory performance levels of the monitored database:

Dirty Read Restarts
Dirty Read Requests
Procedure Requests
Procedure Reads from Disk

The following statistics, available on this pane, are duplicates of statistics appearing on the Key Ratio Analysis Pane of the Home View (DBArtisan - Sybase ASE Performance Analyst) page:

Data Cache Hit Ratio
Procedure Plan Hit Ratio
Large I/O Hit Rate
Clean Buffer Grab Rate

Dirty Read Restarts

Sybase ASE allows dirty reads, which are reads of uncommitted data. To accomplish a dirty read, Sybase ASE uses a special lightweight protection mechanism to gain access to an object without using actual page locks. A dirty read restart occurs when a dirty read is active on an object page, and another process makes changes to the page that cause the page to be deallocated in memory. The scan for the dirty read must be restarted. The amount shown for dirty read restarts are the number of restarts that occurred since the last Embarcadero Performance Analyst refresh.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key Ratio Analysis Pane
Metrics	Values observed much above zero should serve as a signal that application modifications can be in order. Most applications should do everything possible to avoid restarts because of the large overhead they incur.
Correction	If the numbers observed for dirty read restarts are significant, you can want to look into modifying applications that use dirty reads to accomplish data acquisition.

Dirty Read Requests

Sybase ASE allows dirty reads, which are reads of uncommitted data. To accomplish a dirty read, Sybase ASE uses a special lightweight protection mechanism to gain access to an object without using actual page locks. This statistic displays the number of dirty reads that occurred since the last refresh in Embarcadero Performance Analyst.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key Ratio Analysis Pane
Metrics	Dirty read page requests can incur significant overhead if they are observed with many dirty read restarts.

Procedure Requests

The Procedure Requests statistic reports the number of times that stored procedures were executed since Embarcadero Performance Analyst was last refreshed. Such a request could use either an unused copy of the procedure's query plan in memory or if no such copy exists, the procedure must be read in from disk.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key Ratio Analysis Pane

Procedure Reads from Disk

The Procedure Reads from Disk statistic reports the number of times since Embarcadero Performance Analyst was last refreshed that stored procedures were read from disk rather than copied in the procedure cache.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key Ratio Analysis Pane
Metrics	You should examine this number in conjunction with the overall procedure cache hit rate. Observing large numbers in this statistic indicates a lower than ideal procedure cache hit rate. Note that when a database is first started, this statistic is likely larger than desired because all code being used is relatively new and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the procedure cache hit rate has not increased to desirable levels and this statistic continues to sport high numbers, you should look into the possibility of increasing the amount of memory allocated to the cache.
Correction	If a problem is found in Sybase ASE servers, versions 11-12, you can increase the amount of the total memory configuration parameter and/or increase the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase ASE has met its kernel and procedure cache needs). Take care when increasing the procedure cache alone, as this could increase query response times due to more physical I/O being performed. For Sybase ASE 12.5, the total memory configuration parameter can again be increased to provide more memory for the Sybase ASE server, but in 12.5, if you want to increase the size of the procedure cache, note that it is now configured in terms of literal size instead of a percentage of the overall configured memory. Once the procedure cache has been adjusted, monitor Sybase ASE to see if the cache hit rate improves. If it does not, another increase may be necessary. Also, keep a careful eye on the actual machine's memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.

Bottleneck Analysis Pane

When Sybase Adaptive Server is running, every connected process is either busy doing work or waiting to work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. And this is where wait-based or bottleneck analysis comes into play. DBAs use this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Bottleneck analysis is a valid method of measuring performance because it helps a DBA track where a database has been spending its time. If lock contention or heavy table scan activity has been dragging down database performance, a DBA can use bottleneck analysis to confirm the actual root cause. Once one or more wait events or other bottlenecks have been pinpointed as possible performance vampires, the DBA can drill down and oftentimes discover a fair amount of detail about which sessions and objects are causing the problem.

Memory bottlenecks can definitely cause performance degradation in an otherwise well-running database. Typically, these bottlenecks center around the server latches, which are lightweight locks used to protect certain resources in memory. To help you identify such problems, the following statistics are presented on the Performance Analyst Memory home page:

Log Semaphore Contention	Latch Contention	Waits on Execution Plans
Large I/Os Denied	Buffers Grabbed Dirty	Procedure Removals

Log Semaphore Contention

The Log Semaphore Contention statistic measures the total number of times tasks requested a log semaphore. This includes those granted immediately and those for which the task had to wait.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	In high throughput environments with a large number of concurrent users committing transactions, you should expect a certain amount of contention for the log semaphore.

Latch Contention

The Latch Contention statistic reports the number of times a task was switched out because it needed to wait for a latch. If your user tables use only allpages-locking, this latch contention is taking place either on a data-only-locked system table or on allocation pages. If your applications use data-only-locking, the contention reported here includes all waits for latches, including those on index pages and OAM pages as well as allocation pages.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	In SMP environments where inserts and expanding updates are extremely high, so that page allocations take place very frequently, contention for the allocation page latch can reduce performance. Normally, Adaptive Server allocates new pages for an object on an allocation unit that is already in use by the object and known to have free space. For each object, Adaptive Server tracks this allocation page number as a hint for any tasks that need to allocate a page for that object. When more than one task at a time needs to allocate a page on the same allocation unit, the second and subsequent tasks block on the latch on the allocation page.
Correction	You can specify a "greedy allocation" scheme, so that Adaptive Server keeps a list of eight allocation hints for page allocations for a table. The effect of dbcc tune(des_greedyalloc) are not persistent, so you need to reissue the commands after a reboot. You should use this command only if all of the following are true: *You have multiple engines. It is rarely useful with fewer than four engines.A large number of pages are being allocated for the object. You can use sp_spaceused or optdiag to track the number of pages.The latch contention counter shows contention.*

Waits on Execution Plans

The Waits on Execution Plans value represents the number of times that a process attempting to use sp_showplan had to wait to acquire read access to the query plan.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Query plans may be unavailable if you run sp_showplan before the compiled plan is completed or after the query plan finished executing. In these cases, Sybase ASE tries to access the plan three times and then returns a message to the user.

Large I/Os Denied

The Large I/Os Denied statistic represents the number of times that large I/O could not be performed.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Adaptive Server cannot perform large I/O for the following reasons: *If any page in a buffer already resides in another pool.When there are no buffers available in the requested pool.On the first extent of an allocation unit, since it contains the allocation page, which is always read into the 2K pool.*
Correction	If a high percentage of large I/Os were denied, it indicates that using larger pools might not be as effective as it could be. If a cache contains a large I/O pool, and queries perform both 2K and 16K I/O on the same objects, there will always be some percentage of large I/Os that cannot be performed because pages are in the 2K pool. If more than half of the large I/Os were denied, and you are using 16K I/O, try moving all of the space from the 16K pool to the 8K pool. Re-run the test to see if total I/O is reduced. Note that when a 16K I/O is denied, Adaptive Server does not check for 8K or 4K pools, but uses the 2K pool.

Buffers Grabbed Dirty

The Buffers Grabbed Dirty statistic represents the number of times Sybase ASE found dirty buffers since the last refresh in Embarcadero Performance Analyst.

As users request information, buffers are moved into and out of the Sybase ASE data cache. Pages are also modified in the cache (termed dirty buffers) and need to be written out to disk. If Sybase ASE has to wait for a dirty buffer to be written out to disk before a requested buffer is placed into the cache, performance can suffer.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Ideally, the dirty buffer grab statistic should stay close to zero.
Correction	Seeing high numbers for this statistic could indicate that the cache size is too small. You should look into carefully adjusting the total memory configuration parameter to be higher. However, keep a careful eye on the actual machine's memory limits and swap activity. Increased swap activity can indicate too little memory left for the server machine.

Procedure Removals

The Procedure Removals statistic reports the number of times that stored procedures were aged out of the procedure cache since Embarcadero Performance Analyst was last refreshed.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	High numbers, along with a lower than desired procedure cache hit rate, could indicate a procedure cache that is too small.
Correction	If a problem is found in Sybase ASE servers, versions 11-12, you can increase the amount of the total memory configuration parameter and/or increase the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase ASE has met its kernel and procedure cache needs). Take care when increasing the procedure cache only, as this could increase query response times due to more physical I/O being performed. For Sybase ASE 12.5, the total memory configuration parameter can again be increased to provide more memory for the Sybase ASE server, but in 12.5, if you wish to increase the size of the procedure cache, note that it is now configured in terms of literal size instead of a percentage of the overall configured memory. Once the procedure cache has been adjusted, monitor Sybase ASE to see if the cache hit rate improves. If it does not, another increase may be necessary. Also, keep a careful eye on the actual machine's memory limits and swap activity. Increased swap activity can indicate too little memory left for the server machine.

Metadata Cache Ratio Analysis Pane

The following statistics are used on the Performance Analyst for Sybase ASE Memory Home Page to succinctly communicate the general overall performance levels of the memory structures:

Open Objects
Open Indexes
Open Databases
Object Spinlock Contention
Index Spinlock Contention
Hash Spinlock Contention

Open Objects

The Open Objects statistic represents the use of the Adaptive Server metadata cache for open objects. The statistic includes number of active objects, free objects, and the maximum number of objects used.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Metadata Cache Ratio Analysis Pane
Metrics	When Adaptive Server accesses an object, it needs to read information about it in the corresponding system table: sysobjects. The metadata cache for objects lets Adaptive Server access the information that describes it in the sysobjects row directly in its in-memory structure. This improves performance because Adaptive Server bypasses expensive calls that require disk access. It also reduces synchronization and spinlock contention when Adaptive Server has to retrieve object information at runtime. Managing individual metadata caches for databases, indexes, or objects is beneficial for a database that contains a large number of indexes and objects, and where there is high concurrency among users.
Correction	You can use this information to set the configuration parameter's number of open objects.

Open Indexes

The Open Indexes statistic represents the use of the Adaptive Server metadata cache for open indexes. The statistic includes the number of active indexes, free indexes, and the maximum number of indexes.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Metadata Cache Ratio Analysis Pane
Metrics	When Adaptive Server accesses an index, it has to access information about it in the respective system table: sysindexes. The metadata cache for indexes lets Adaptive Server access the information that describes it in the sysindexes row directly in its in-memory structure. This improves performance because it allows the Adaptive Server to bypass expensive calls that require disk access. It also reduces synchronization and spinlock contention when Adaptive Server has to retrieve index information at runtime. Managing individual metadata caches for databases, indexes, or objects is beneficial for a database that contains a large number of indexes and objects, and where there is high concurrency among users.
Correction	You can use this information to set the configuration parameter's number of open indexes.

Open Databases

The Open Databases statistic shows the use of the Adaptive Server metadata cache for open databases. This statistic charts the number of active databases, free databases, and the maximum number of databases used.

When Adaptive Server opens a database, it has to access information about it in the respective system table: sysdatabases. The metadata cache for databases lets Adaptive Server access the information that describes it in the sysdatabases row directly in its in-memory structure. This improves performance because it allows the Adaptive Server to bypass expensive calls that require disk access. It also reduces synchronization and spinlock contention when Adaptive Server has to retrieve database information at runtime.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Metadata Cache Ratio Analysis Pane
Metrics	Managing individual metadata caches for databases, indexes, or objects is beneficial for a database that contains a large number of indexes and objects, and where there is high concurrency among users.
Correction	You can use this information to set the configuration parameter's number of open databases.

Object Spinlock Contention

The Object Spinlock Contention statistic represents spinlock contention on the object descriptor caches.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Metadata Cache Ratio Analysis Pane
Correction	You can use this information to tune the configuration parameter's open object spinlock and open index spinlock ratios. If the reported contention is more than 3%, decrease the value of the corresponding parameter to lower the number of objects or indexes that are protected by a single spinlock.

Index Spinlock Contention

The Index Spinlock Contention value represents spinlock contention on the index descriptor caches.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Metadata Cache Ratio Analysis Pane
Correction	You can use this information to tune the configuration parameter's open object spinlock and open index spinlock ratios. If the reported contention is more than 3%, decrease the value of the corresponding parameter to lower the number of objects or indexes that are protected by a single spinlock.

Hash Spinlock Contention

This metric represents the contention for the spinlock on the index metadata cache hash table.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Metadata Cache Ratio Analysis Pane
Correction	You can use this information to tune the open index hash spinlock ratio configuration parameter. If the reported contention is greater than 3%, decrease the value of the parameter.

Memory Analysis Pane

The following statistics are used on the Performance Analyst for Sybase ASE Memory Home Page to succinctly communicate the general overall performance levels of the memory structures:

Buffer Cache Procedure Cache Connection Memory
Open Database Memory Open Object Memory Open Index Memory

Buffer Cache

The Buffer Cache statistic displays the amount of memory currently allocated to the Buffer Cache.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Memory Analysis Pane

Procedure Cache

The Procedure Cache value displays the amount of memory currently allocated to the procedure cache.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Memory Analysis Pane
Metrics	The memory allocated for the procedure cache holds the optimized query plans (and occasionally trees) for all batches, including any triggers.

Connection Memory

The Connection Memory value indicates the amount of memory currently allocated to user connections.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Memory Analysis Pane
Metrics	Each connection requires approximately 60 KB.

Open Database Memory

The Open Database Memory value represents the current amount of memory allocated in the number of open databases parameter.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Memory Analysis Pane

Open Object Memory

This value represents the current amount of memory allocated in the number of open objects parameter.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Memory Analysis Pane

Open Index Memory

This value represents the current amount of memory allocated in the number of open indexes parameter.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Memory Analysis Pane

Workload Analysis Pane

The following statistics are used on the Performance Analyst for Sybase ASE Memory Home Page to succinctly communicate the general overall performance levels of the memory structures:

- Top Memory Hogs

Top Memory Hogs

This metric identifies the process that currently is using the highest percentage of memory in the database.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Home > Workload Analysis Pane

Metrics It is not uncommon for one or two users to cause the majority of runtime problems that plague a server. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. If your database server does not have an overabundance of memory, then you should periodically check to see who your heavy memory users are along with the total percentage of memory each takes up. If you see one or two users who have more than 25-50% of the total memory usage, then you should further investigate the sessions to see what activities they are performing.

Data Cache Tab

The Data Cache tab of the Memory Detail View includes the following sections:

Cache Configurations Cache Efficiency Cached Objects

Cache I/O Activity

The following statistic, available on this tab, duplicates a statistic available on the Key Ratio Analysis Pane of the Home View (DBArtisan - Sybase ASE Performance Analyst) page:

- Data Cache Hit Ratio

Cache Configurations

The Cache Configuration view represents the default and named data caches and the memory configuration in MB.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Data Cache Tab

Cache Efficiency

The Cache Efficiency section summarizes behavior for the default data cache and all named data caches combined. The table below describes the information available on the Cache Efficiency detail view:

Column	Description
Cache Name	The name of the specified cache
Hit Rate	The percentage of cache hits compared to the total number of cache searches.
Percent Used	The percentage of searches using this cache as a percentage of searches across all caches.
Spinlocks	The number of spinlocks.
LRU Buffers	The number of buffers that were retrieved and replaced in the Least Recently Used section of the pool.
MRU Buffers	The number of buffers that were retrieved and replaced in the Most Recently Used section of the pool.
Large I/Os	The size (bytes) of the I/O buffer for the pool.
Dirty Reads	The number of dirty reads.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Data Cache Tab
Metrics	Interpreting cache hit data requires an understanding of how the application uses each cache. In caches that are created to hold specific objects such as indexes or look up tables, cache hit ratios can reach 100%. In caches used for random point queries on huge tables, cache hit ratios can be quite low but still represent effective cache use. This data can also help you determine if adding more memory would improve performance. For example, if "Cache Hits" is high, adding memory probably would not help much.
Correction	You can compare the "percent used" value for each cache to determine if there are caches that are over- or under-utilized. If you decide that a cache is not well utilized, you can: *Change the cache bindings to balance utilization.Resize the cache to correspond more appropriately to its utilization.*

Cached Objects

The Cached Objects section shows the objects that are currently cached. The table below describes the information available on the Cached Objects detail view:

Column	Description
Cache Name	The name of the specified cache.
Database Name	The name of the database where the object resides.
Object Owner	The owner of the object.
Object Name	The name of the object.
Object Type	The type of the object.
Cached (KB)	The amount, in Kilobytes, of the cached object.
Processes Accessing	The number of processes accessing the specified object.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Data Cache Tab

Cache I/O Activity

The Cache I/O Activity view shows how many searches were directed to each cache. The table below describes the information available on the Cache I/O Activity view:

Column	Description
Cache Name	The name of the specified cache.
Physical Reads	The number of physical reads associated with the specified cache.
Pages Touched	The number of pages touched by the cache.
Pages Read	The number of pages read by the cache.
Buffers to MRU	The number of buffers that were retrieved and replaced in the Most Recently Used section of the pool.
Buffers to LRU	The number of buffers that were retrieved and replaced in the Least Recently Used section of the pool.
I/O Buffer Size	The size (bytes) of the I/O buffer for the pool.
Allocated (KB)	The number of KB that have been allocated for the pool.
Stalls	The number of times a process had to wait for a free procedure cache buffer when installing a stored procedure into cache.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Data Cache Tab
Metrics	You can compare the values in this view for each cache to determine if there are caches that are over- or under-utilized.
Correction	If you determine that a cache is not used as well as you would like, you can: *Change which objects are bound to each cache to balance use.Resize the cache to correspond more appropriately to its use.*

Procedure Cache Tab

The Procedure Cache tab of the Memory Detail View includes the following sections:

Procedure Cache Activity Procedure Cache Detail Current Procedure Usage

The following statistic, available on this tab, duplicates astatistic available on the Key Ratio Analysis Pane of the Home View (DBArtisan - Sybase ASE Performance Analyst) page:

- Procedure Plan Hit Ratio

Procedure Cache Activity

The Procedure Cache Activity view summarizes the key metrics concerning the Procedure Cache. The table below describes the information available on the Procedure Cache Activity view:

Column	Description
Requests	The number of stored procedures requested from the cache.
Loads	The number of stored procedures loaded into cache.
Writes	The number of times a stored procedure was normalized and the tree was written back to sysprocedures. This happens when a stored procedure (or trigger) is created during the interval between polls. A degradation in performance can occur if an application program generates stored procedures.
Stalls	The number of times a process had to wait for a free procedure cache buffer when installing a stored procedure into cache.
Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Procedure Cache Tab

Procedure Cache Detail

The Procedure Cache Detail view summarizes the key metrics concerning the Procedure Cache. The table below describes the information available on the Procedure Cache Detail view:

Column	Description
Object Name	The name of the object.
Object Owner	The object's owner.
Object Type	The type of object.
Database	The name of the database where the object resides.
Compile Date	The last month/day/year object data was compiled.
Memory Used (KB)	The amount of memory used by the object in kilobytes.
Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Procedure Cache Tab

Current Procedure Usage

The Current Procedure Usage view displays the number of procedures currently in use. The table below describes the information available in the Current Procedure Usage view:

Column	Description
SPID	The process name.
User Name	The name of the user attached to the process.
Database	The database attached to the process.
Object Owner	The object's owner.
Object Name	The object's name.
Object Type	The object's type.
Memory Used (KB)	The amount of memory the procedure used in kilobytes.
Program Name	The name of the program where the procedure is used.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Memory View (DBArtisan - Sybase ASE Performance Analyst)** > Procedure Cache Tab

I/O (DBArtisan - Sybase ASE Performance Analyst)

In addition to a Home page, the I/O category of Sybase ASE Performance Analyst includes the following tabbed pages:

Databases tab Devices tab Engines Tab

Network tab Objects tab

Home

The I/O performance category page displays the following vital Sybase ASE I/O statistics:

Bottleneck Analysis Pane Key System Analysis Pane Network Analysis Pane

Transaction Log Analysis Pane Workload Analysis Pane

Key System Analysis Pane

The following statistics are used on the Performance Analyst for Sybase ASE I/O Home Page to succinctly communicate the general overall performance levels of I/O:

Physical Reads APF Reads Physical Writes

Procedure Requests from Disk Data Cache Searches Checkpoints

Large I/Os Performed

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key System Analysis Pane

Physical Reads

Physical Reads represents the total number of physical reads performed by the database server since the last refresh.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key System Analysis Pane
Metrics	Large numbers of physical reads could reflect a too small data or procedure cache. You should examine the data and procedure cache hit rates to determine the overall effectiveness of logical vs. physical I/O.

APF Reads

This metric represents the number of pages that were read by asynchronous prefetch since the last refresh.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key System Analysis Pane
Metrics	Asynchronous prefetch tries to find a page it needs to read in the data cache using a quick scan without holding the cache spinlock. If that does not succeed, it then performs a thorough scan holding the spinlock.

Physical Writes

This value reflects total number of physical writes performed by the database server since the last refresh.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key System Analysis Pane

Procedure Requests from Disk

The Procedure Requests from Disk value represents the number of times that stored procedures were read from disk rather than found and copied in the procedure cache.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key System Analysis Pane
Metrics	If you see a relatively high number, it could indicate that the procedure cache is too small.
Correction	You should increase the size of the memory allocated to the procedure cache memory structure.

Data Cache Searches

The Data Cache Searches value represents the number of times that a needed page was found in any cache.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key System Analysis Pane
Correction	If you see a high number of cache misses, you should investigate statistics for each cache.

Checkpoints

The Checkpoints value represents the combined number of normal and free checkpoints that occurred during the sample interval.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key System Analysis Pane
Metrics	Checkpoints write dirty pages (pages that have been modified in memory, but not written to disk) to the database device. Adaptive Server's automatic (normal) checkpoint mechanism works to maintain a minimum recovery interval. By tracking the number of log records in the transaction log since the last checkpoint was performed, it estimates whether the time required to recover the transactions exceeds the recovery interval. If so, the checkpoint process scans all data caches and writes out all changed data pages. When Adaptive Server has no user tasks to process, a housekeeper task begins writing dirty buffers to disk. These writes are done during the server's idle cycles, so they are known as "free writes." They result in improved CPU use and a decreased need for buffer washing during transaction processing. If the housekeeper process finishes writing all dirty pages in all caches to disk, it checks the number of rows in the transaction log since the last checkpoint. If there are more than 100 log records, it issues a checkpoint. This is called a "free checkpoint" because it requires very little overhead. In addition, it reduces future overhead for normal checkpoints.

Large I/Os Performed

The Large I/Os Performed statistic measures the number of times that a requested large I/O was performed.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key System Analysis Pane
Metrics	Adaptive Server cannot perform large I/O for the following situations: *If any page in a buffer already resides in another pool.When there are no buffers available in the requested pool.On the first extent of an allocation unit, since it contains the allocation page, which is always read into the 2K pool.*
Correction	If a high percentage of large I/Os were denied, it indicates that the use of the larger pools might not be as effective as it could be. If a cache contains a large I/O pool, and queries perform both 2K and 16K I/O on the same objects, there will always be some percentage of large I/Os that cannot be performed because pages are in the 2K pool. If more than half of the large I/Os were denied, and you are using 16K I/O, try moving all of the space from the 16K pool to the 8K pool. Re-run the test to see if total I/O is reduced. Note that when a 16K I/O is denied, Adaptive Server does not check for 8K or 4K pools, but uses the 2K pool.

Bottleneck Analysis Pane

The following statistics are used on the Performance Analyst for Sybase ASE I/O Home Page to succinctly communicate the general overall performance levels of I/O:

Delayed Disk IOs
Outstanding Disk I/Os
Disk I/O Structure Contention
Disk I/O Structure Contention
Total I/O
I/O Errors
Hottest Devices

The following statistic, available on this pane, duplicates a statistic available on the Key Ratio Analysis Pane of the Home View (DBArtisan - Sybase ASE Performance Analyst) page:

- I/O Busy

The following statistics, available on this pane, are duplicates of statistics available on the Bottleneck Analysis Pane of the Home View (DBArtisan - Sybase ASE Performance Analyst) page:

Device I/O Contention
Group Commit Sleeps

Delayed Disk IOs

The Delayed Disk IOs value is the number of I/Os delayed when the limit on disk I/O structures is reached.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	When Adaptive Server exceeds the number of available disk I/O control blocks, I/O is delayed because Adaptive Server requires that tasks get a disk I/O control block before initiating an I/O request.
Correction	If you see a nonzero value for delayed disk I/Os, try to add to the number of available disk I/O control blocks by increasing the configuration parameter.

Outstanding Disk I/Os

The Outstanding Disk I/Os statistic reflects the maximum number of I/Os pending for Sybase ASE since the last refresh.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	If you see nonzero numbers for both outstanding and delayed disk I/Os, there could be a problem in many areas.

Disk I/O Structure Contention

When a Sybase ASE server begins to experience I/O delays, the result can be a very dissatisfied user community. When such problems begin to occur, you should investigate Sybase ASE or operating system limits. It could be that I/O operations are being blocked by one or both.

The Disk I/O Structures statistic represents the number of I/O delays caused by Sybase ASE reaching the limit on disk I/O structures. When Sybase ASE exceeds the number of available disk

I/O control blocks, I/O is deferred because Sybase ASE requires that any task get a disk I/O control block before beginning an I/O request.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Consistent numbers above zero should be a cause for concern.
Correction	If you continue to see nonzero numbers for this statistic, you can try increasing the number of available disk I/O control blocks by increasing the configuration parameter disk I/O structures.

Total I/O

The Total I/O statistic represents the total number of physical reads and writes.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane

I/O Errors

I/O error rate reflects total number of I/O errors (errors during read and write operations) encountered by the server since the last refresh inside Performance Analyst. The I/O errors rate is a percentage based on Total I/O (the sum of the physical reads and writes).

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	You should observe few, if any errors.
Correction	If you notice any errors, you should check the Sybase ASE error log for details.

I/O Busy

This metric represents the number of clock ticks in the sample interval during which the user task performed I/O operations.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	High numbers indicate an I/O-intensive process. If idle time is also high, the application could be I/O bound.
Correction	The application might achieve better throughput if you assign it a higher priority, bind it to a lightly loaded engine or engine group, or partition the application's data onto multiple devices.

Hottest Devices

The Hottest Devices view identifies the configured devices engaged in the most I/O activity. In Sybase ASE, when a task or process needs to perform physical I/O, the server fills out the actual block I/O structure and links it to a per-engine I/O queue. If two Sybase ASE engines request the same I/O structure from the exact same device at the identical time, one of them is put to sleep by the server and it waits for the semaphore it needs.

Device I/O Contention reflects the number of times a task or process was put to sleep while waiting for a semaphore for a particular database device.

The table below describes the information available on the Hottest Devices view:

Column	Description
Device Name	The name of the device.
APF Reads	The number of asynchronous pre-fetch buffers read.
Physical Reads	The total number of physical reads performed by the database server since the last refresh
Physical Writes	The total number of physical writes performed by the database server since the last refresh
Total I/O	The combined total of read and write operations.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Correction	If you consistently see high numbers for Device I/O Contention, you can try redistributing the tables across devices (that are located on separate hard disks) or by adding devices (again, that are located on separate hard disks) and moving tables and indexes to them.

Network Analysis Pane

The following statistics are used on the Performance Analyst for Sybase ASE I/O page to succinctly communicate the general overall performance levels of I/O:

Packets Sent
Packets Received
Network Delays
Network Errors

Packets Sent

Packets Sent displays the number of times Sybase ASE sent a packet to a client application.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Network Analysis Pane

Packets Received

Total Packets Received reflects the number of times Sybase ASE received a packet from a client application.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Network Analysis Pane

Network Delays

The Sybase ASE server normally sends and receives network packets at a regular rate. If the network begins to be saturated, Sybase ASE can experience delays in network I/O. The Network Delays statistic reflects the number of times network I/O activity was delayed.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Network Analysis Pane
Metrics	If you see a network contention rate greater than 1%, it could indicate a challenged network structure.
Correction	Beyond ensuring that the existing network is sufficient for handling the current load (database and non-database), other items to look at from a Sybase-specific standpoint include validating that only the necessary amount of data is being sent and returned to requesting users. You can also examine the default packet size and see if it is too small for the average packet size being sent/received by the Sybase ASE server.

Network Errors

The Network Errors statistic reflects the number of times that network errors were detected by Sybase ASE while reading and writing packets.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Network Analysis Pane
Metrics	If you consistently see a value much above zero, it could indicate the network structure is challenged.
Correction	Beyond ensuring that the existing network is sufficient for handling the current load (database and non-database), other items to look at from a Sybase-specific standpoint include validating that only the necessary amount of data is being sent and returned to requesting users. You can also examine the default packet size and see if it is too small for the average packet size being sent/received by the Sybase ASE server.

Transaction Log Analysis Pane

The following statistics are used on the Performance Analyst for Sybase ASE I/O page to succinctly communicate the general overall performance levels of I/O:

Log Writes Log Allocations ULC Flushes

Average Number of Writes per Log Page

Log Writes

Transaction Log Writes refers to the number of times Sybase ASE wrote a transaction log page to disk since the last refresh. When the current log page becomes full, Sybase ASE writes it out to disk. Sybase ASE also writes transaction log pages to disk after a transaction commits.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Transaction Log Analysis Pane

Log Allocations

Transaction Log Allocations refers to the number of times since the last refresh that additional pages were allocated to the transaction log. This statistic gives you a feel for the rate of transaction log growth.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Transaction Log Analysis Pane

ULC Flushes

ULC Flushes is the total number of times that ULCs (user log caches) were flushed to a transaction log during the sample interval.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Transaction Log Analysis Pane

Average Number of Writes per Log Page

The Average Number of Writes is the average number of times each log page was written to disk.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Transaction Log Analysis Pane

Metrics In high throughput applications, this number should be as low as possible. If the transaction log uses 2K I/O, the lowest possible value is 1; with 4K log I/O, the lowest possible value is .5, since one log I/O can write 2 log pages. In low throughput applications, the number will be significantly higher. In very low throughput environments, it may be as high as one write per completed transaction.

Workload Analysis Pane

The following statistics are used on the Performance Analyst for Sybase ASE I/O page to succinctly communicate the general overall performance levels of I/O:

- Top I/O Hogs

Top I/O Hogs

The Top I/O Process statistic identifies the processes that has currently caused the most I/O usage on the database. The table below describes the information available from the Top I/O Hogs view of the I/O page:

Column	Description
SPID	The process ID.
Login	The login name the session is using.
Physical I/O	The physical amount of I/O the process is using.
% Used	Percent of total I/O on the server this process is consuming.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Home > Workload Analysis Pane

Correction If any one session uses more than 50% of a total resource (CPU, memory, etc.) you should drill down into that particular session and investigate the cause.

Databases tab

The database I/O statistic displays the major I/O statistics as they relate to each associated database. The table below describes the information available from the I/O Databases detail tab:

Column	Description
Database Name	The name of the database.
Logical Reads	The number of logical reads associated with the database.
Physical Reads	The number of physical reads associated with the database.
APF Reads	The number of asynchronous pre-fetch buffer reads associated with the database.
Pages Read	The number of pages read associated with the database.
Physical Writes	The number of physical writes associated with the database.
Pages Written	The number of pages written to the associated database.
Rows Inserted	The number of rows inserted in the associated database.
Rows Updated	The number of rows updated in the associated database.
Rows Deleted	The number of rows deleted from the associated database.
Lock Requests	The number of requests for a lock on database objects.
Lock Waits	The number of times a task waited for a lock on a database object.

Correction

The major guidelines for improving I/O performance in Adaptive Server are as follows:

- Spread data across disks to avoid I/O contention.
- Isolate server-wide I/O from database I/O.
- Separate data storage and log storage for frequently updated databases.
- Keep random disk I/O away from sequential disk I/O.
- Mirror devices on separate physical disks.
- Partition tables to match the number of physical devices in a segment.

Objects tab

The following views are used on the Performance Analyst for Sybase ASE I/O Drilldown Objects to succinctly communicate the general overall performance levels of I/O:

Index Scans Open Object I/O

Index Scans

The following views are used on the Index Scans view of I/O Drilldown Objects to succinctly communicate the general overall performance levels of I/O:

Ascending Scans DOL Ascending Scans DOL Descending Scans

Descending Scans

Ascending Scans

The Ascending Scans value represents the number of forward scans on allpages-locked tables.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Objects tab

DOL Ascending Scans

The DOL value represents the number of forward scans on data-only-locked tables.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Objects tab

DOL Descending Scans

The DOL Descending Scans value represents the number of backward scans on allpages-locked tables.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Objects tab

Descending Scans

The Descending Scans value represents the number of backward scans on data-only-locked tables.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Objects tab

Open Object I/O

The Open Object I/O view displays a grid that shows a list of each object on the selected database that is either open, or has been open, and related I/O activity.

The table below describes the information available on the Open Object I/O grid:

Column	Description
Database Name	The name of the database.
Object Name	The name of the object.
Index ID	The ID number of the associated index.
Logical Reads	Total number of buffers read.
Physical Reads	Total number of buffers read from disk
APF Reads	The number of asynchronous pre-fetch buffer reads associated with the database.
Pages Read	The number of pages read associated with the database.
Physical Writes	The number of physical writes associated with the database.
Pages Written	The number of pages written to the associated database.
Rows Inserted	The number of rows inserted in the associated database.
Rows Updated	The number of rows updated in the associated database.
Rows Deleted	The number of rows deleted from the associated database.
Used Count	The number of rows used.
Lock Requests	The number of requests for a lock on database objects.
Lock Waits	The number of times a task waited for a lock on a database object.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Objects tab
Metrics	Objects are only removed from the list if the configured number of "open objects" is exceeded. Then they are aged out to make room for newly opened objects.

Network tab

The Network tab of the I/O Detail view presents the following statistics:

Network I/O Summary Network I/O by Engine Network I/O by Session

Network I/O Summary

Here you see a summary of data that includes the following statistics:

Packets Sent: Total Packets Sent reflects the number of times Sybase ASE sent a packet to a client application.

Packets Received: Total Packets Received reflects the number of times Sybase ASE received a packet from a client application.

Bytes Sent: Total Bytes Sent reflects the number of bytes sent to Sybase ASE since the last refresh.

Bytes Received: Total Bytes Received reflects the number of bytes received by Sybase ASE since the last refresh.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Network tab

Network I/O by Engine

The statistics, presented in tabular format, include:

Engine: The name of the engine under scrutiny.

Packets Sent: The number of packets sent by the engine.

Packets Received: The number of packets received by the engine.

Bytes Sent: The number of bytes sent by the engine.

Bytes Received: The number of bytes received by the engine.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Network tab

Network I/O by Session

This view shows the amount of network I/O committed by registered sessions on the Sybase ASE Server.

Presented in tabular format, you see:

Login: The log in name used by the session.

SPID: The process ID.

KPID: The kernel process ID.

Packets Sent: The number of packets sent by the session.

Packets Received: The number of packets received by the session.

Bytes Sent: The number of bytes sent by the session.

Bytes Received: The number of bytes received by the session.

Network Packet Size: Network packet size used by the current session.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Network tab
Metrics	This metric can be used to see if a user or batch job is generating more (or less) network traffic than expected. The process' network traffic, measured in conjunction with overall server network traffic, could help distinguish between process level or network-wide loads being the source of contention.

Devices tab

The following statistics are used on the Performance Analyst for Sybase ASE I/O Drilldown Devices to succinctly communicate the general overall performance levels of I/O:

Device I/O Summary Device I/O Type Real Time Device I/O

Device I/O Summary

Devices are accessed repeatedly in a dynamic Sybase ASE environment to satisfy user requests for data, to handle write activity that records transactions in a database's transaction log, and to manage other I/O operations. Viewing the I/O activity for each device is a good way to see what the "hot" devices are in a Sybase ASE server with respect to I/O usage. The same information can be used to spot heavy database usage in systems where the device-to-database mappings are one-to-one.

The table below describes the information available in the Device I/O Summary view:

Column	Description
Device	The device's name.
APF Reads	The number of asynchronous pre-fetch buffer reads associated with the database.
Physical Reads	The number of physical reads associated with the database.
Physical Writes	The number of physical writes associated with the database.
Total I/O	The combined total of read and write operations.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Devices tab
Metrics	If devices have one-to-one relationships with user databases, you can quickly tell which databases are experiencing the most demand on a server. If the device-to-physical drive/file system is a one-to-one relationship, you can also spot which physical drives on a server are getting the heaviest workouts. For servers that have many drives, it is normally desirable to spread devices across different physical drives and controllers so contention does not occur at the disk level. In addition, separating databases and their corresponding logs is normally recommended so that each is located on a distinct physical drive. If possible, write-intensive devices, like log devices, are best suited for non-RAID5 scenarios.
Correction	If device hit rates are low, you can add more devices or redistribute objects among different devices. Typically, segments can be used to redistribute objects among different devices or physical hard disks. Common techniques include placing tables and indexes on different segments and partitioning large tables.

Device I/O Type

The Device I/O Type view shows the type of I/O registering against each configured device.

Devices are accessed repeatedly in a dynamic Sybase ASE environment to satisfy user requests for data, to handle write activity that records transactions in a database's transaction log, and to manage other I/O operations. Viewing the I/O activity for each device is a good way to see what the "hot" devices are in a Sybase ASE server with respect to I/O usage. The same information can be used to spot heavy database usage in systems where the device-to-database mappings are one-to-one.

The table below describes the information available from the Device I/O Type detail view:

Column	Description
Device	The name of the device.
I/O Type	The type of I/O.
I/O Time (sec)	The amount of time, in seconds, spent performing I/O operations.
Total I/O	The combined total of read and write operations.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Devices tab
Metrics	If devices have one-to-one relationships with user databases, you can quickly tell which databases are experiencing the most demand on a server. If the device-to-physical drive/file system is a one-to-one relationship, you can also spot which physical drives on a server are getting the heaviest workouts. For servers that have many drives, it is normally desirable to spread devices across different physical drives and controllers so contention does not occur at the disk level. In addition, separating databases and their corresponding logs is normally recommended so that each is located on a distinct physical drive. If possible, write-intensive devices, like log devices, are best suited for non-RAID5 scenarios.
Correction	If device hit rates are low, you can add more devices or redistribute objects among different devices. Typically, segments can be used to redistribute objects among different devices or physical hard disks. Common techniques include placing tables and indexes on different segments and partitioning large tables. If device loads appear skewed (one device has much more activity than others), you should focus attention on that device. Again, redistributing objects can lessen the device's workload.

Real Time Device I/O

Devices are accessed repeatedly in a dynamic Sybase ASE environment to satisfy user requests for data, to handle write activity that records transactions in a database's transaction log, and to manage other I/O operations. Viewing the current, real-time I/O activity for each device is a good way to see what the "hot" devices are in a Sybase ASE server with respect to I/O usage. The same information can be used to spot heavy database usage in systems where the device-to-database mappings are one-to-one.

The table below describes the information available on the Real Time Device I/O detail view:

Column	Description
Device	The logical name of the device.
Physical Name	The physical name of the device.
APF Reads	The number of asynchronous pre-fetch buffer reads associated with the database.
Physical Reads	Number of buffers read from disk.
Physical Writes	Number of buffers written to disk.
I/O Time (Secs)	The amount of time, in seconds, spent performing I/O operations.
Semaphore Requests	The number of device semaphore requests immediately granted on the device.
Semaphore Waits	The number of tasks forced to wait for synchronization of an I/O request.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Devices tab
Metrics	The amount of Reads, APF Reads, and Writes can help you spot overworked devices. You should also keep your eye on the semaphore waits and I/O time for each device to help you determine devices that are experiencing I/O related delays.
Correction	If device hit rates are low, you can add more devices or redistribute objects among different devices. Typically, segments can be used to redistribute objects among different devices or physical hard disks. Common techniques include placing tables and indexes on different segments and partitioning large tables. If device loads appear skewed (one device has much more activity than others), you should focus attention on that device. Again, redistributing objects can lessen the device's workload.

Engines Tab

In symmetric multiprocessing (SMP) environments, a DBA can configure the Sybase ASE server to use more than one "engine," which represents a certain amount of CPU power. By default, Sybase ASE configures one engine for use. If you have a server machine with multiple CPUs, you can enable more engines to take advantage of the machine's additional processing ability.

The Engines tab of the I/O Detail view displays information with respect to how each engine is handling I/O. The table below describes the information available on the Engines tab of the I/O Detail view:

Column	Description
Engine	The name of the configured engine.
Completed I/Os	The number of I/Os completed during the sample interval.
Outstanding I/Os	The number of I/Os left outstanding during the sample interval.

Tip: To configure the grid to show/hide row numbers, use the Options Editor.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Devices tab
Metrics	If the number of outstanding I/Os remains high or increases during periods of heavy activity, there may not be enough engines configured for the system.
Correction	If the server machine has multiple CPUs, you can configure more engines for Sybase ASE to use by following this process: *Use the sp_configure procedure to change the current engine configuration. For example, to change the number of engines from one to two, you would run: 'sp_configure "max online engines",2'Stop and restart the Sybase ASE server.*

Index Scans Tab

Indexes are accessed frequently in dynamic Sybase ASE server environments. The type of index access often determines the response time an end user experiences. Single row index accesses are the quickest, and complete index scans are the most time consuming (for large indexes at least).

The Index Scans tab of the I/O Detail view presents information with respect to index scans. The table below describes the information available on the Index Scans tab of the I/O Detail view:

Column	Description
Scan Type	The type of index scan.
Count	The number of scans per type for the sample interval.
Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **I/O (DBArtisan - Sybase ASE Performance Analyst)** > Devices tab
Metrics	There are two basic scan types - ascending and descending. For ascending scans, Sybase ASE moves up the index page chain from beginning to end. Sybase ASE follows the page chain in reverse order for descending scans. Descending scans are normally the result of requests made for data by descending column order. Within ascending and descending scans, a data-only lock, or DOL, styled access can also occur.

Space (DBArtisan - Sybase ASE Performance Analyst)

In addition to a Home page, the Space category of Sybase ASE Performance Analyst includes the following tabbed pages:

Databases tab Devices tab

Home

The Space performance category view displays the following vital Sybase ASE space statistics:

Space Analysis Pane Transaction Log Analysis Bottleneck Analysis Pane

Device Analysis Pane

Space Analysis Pane

A Sybase ASE Server is broken up into many different databases, each of which has its own level of activity. The Space Analysis section displays a summary of activities that have occurred within each database. It also displays the total number of databases defined on the Sybase ASE server.

The table below describes the database information available on the Space Analysis view:

Column	Description
Database Name	The name of the Sybase ASE database.
Total Space (MB)	The total amount of space allocated to the database expressed in MB.
Free Space (MB)	The amount of space currently free expressed in MB.
Used Space (MB)	The amount of space currently in use, expressed in MB.
Percent Free	The available free space on the database, expressed as a percentage.
Log Total Space (MB)	
Log Free Space (MB)	
Log Used Space (MB)	
Log Percent Free	
Log Percent Used	

Transaction Log Analysis

The following statistics are used on the Performance Analyst for Sybase ASE Space page to succinctly communicate the general overall performance levels of space related metrics:

- Transaction Log Analysis

Transaction Log Analysis

The Transaction Log Analysis view indicates if any transaction log for a database in the Sybase ASE server is approaching low levels of available free space. The table below describes the information available on the Transaction Log Analysis grid:

Axis	Description
Percent Space Used	The amount of space currently used in the log, expressed as a percentage.
Databases	The name of the Sybase ASE database.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Space (DBArtisan - Sybase ASE Performance Analyst)** > Home > Space Analysis Pane
Metrics	If the transaction log runs out of space, no transactions can take place in the database until you free space in the log.
Correction	If a database log consistently approaches its limit on used space, there are a few actions a DBA can take to prevent a database from freezing. A backup (dump) of the log can be taken: *If log backups are not required for disaster recovery, the truncate log on checkpoint option can be set for the database. Setting this option deletes any space devoted to transactions in the log that have already completed when a database checkpoint occurs.You can extend the log onto a new device. Or, if space exists on the device on which the database log currently resides, you can allocate more space for the log on the current device.*

Bottleneck Analysis Pane

The following statistics are used on the Performance Analyst for Sybase ASE Space page to succinctly communicate the general overall performance levels of space related metrics:

- Devices low on space

The following statistics, available on this pane, are duplicates of statistics available on the Bottleneck Analysis Pane of he Home View (DBArtisan - Sybase ASE Performance Analyst) page:

- Databases Low on Space
- Logs Low on Space

Devices low on space

This metric indicates if any device in the Sybase ASE server is approaching low levels of available free space.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Space (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	The environment of the particular Sybase ASE server, as well as your work style dictate the metrics you should use to evaluating a device that is running into trouble with space. Many DBAs create devices that parallel a single, corresponding database in size, and therefore, such devices show 100% utilization. Other DBAs create large devices that are not completely utilized.
Correction	If a device has become too full, a DBA can begin the process of manually relocating databases from it onto other devices. The process of moving logs is easier and can be accomplished via singular commands (sp_logdevice).

Device Analysis Pane

The Device Analysis view presents space related details about a selected device located on the Sybase ASE server. The table below describes the information available on the Device Analysis view:

Column	Description
Devices	Total number of devices configured on the server.
Device Name	The name of the device.
Physical Name	The name used by the operating system to identify the device.
Total Space (MB)	Total space allocated on this device in MB.
Used Space (MB)	Total space, in MB, currently in use on the device.
Free Space (MB)	Amount of space that has not been allocated on this device.
Percent Free (MB)	The available free space on the database, expressed as a percentage.
Percent Used (MB)	The amount of space used on the database, expressed as a percentage.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Space (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	The environment of the particular Sybase ASE server as well as your work style dictate the metrics you should use to evaluate a device that is running into trouble with space. Many DBAs create devices that parallel a single, corresponding database in size, and therefore, such devices show 100% utilization. Other DBAs create large devices that are not completely utilized.
Correction	If a device has become too full, a DBA can begin the process of manually relocating databases from it onto other devices. The process of moving logs is easier and can be accomplished via singular commands (sp_logdevice).

Databases tab

The following views are used on the Performance Analyst for Sybase ASE Space drill-down page to succinctly communicate the general overall performance levels of space related metrics:

Database Object Summary Database Object Detail

Database Object Summary

A Sybase ASE server contains many databases, some of which are devoted to system-level activities (the master and tempdb databases, for example) and others that hold user data. The database overview displays details about the space situation for each database in a Sybase ASE server, including the total, used, and free space. The percentage used amount for each database is also shown.

The table below describes the information available on the Database Object Summary detail view:

Column	Description
Database Name	The name of the database.
Total DB Space (KB)	Total amount of space allocated to the database.
Free DB Space (KB)	Total amount of space allocated but free on the database.
Used DB Space (KB)	Total amount of space currently used by the database.
Percent DB Free	The amount of free space expressed as a percentage.
User Table Count	The number of user tables in the database.
Table Reserved (KB)	The amount of total space reserved (used and free) for tables.
Table Data (KB)	The amount of total space used by tables.
Table Unused (KB)	The amount of total unused space reserved for tables.
User Index Count	The number of user indexes.
Index Reserved (KB)	The amount of index space that is reserved.
Index Data (KB)	The amount of data stored in indexes.
Index Unused (KB)	The amount of index space that is unused.
Total Log Space (KB)	The total amount of log space.
Free Log Space (KB)	The amount of log space that is allocated but unused.
Used Log Space (KB)	The amount of log space that is currently in use.
Percent Log Free (KB)	The amount of free log space expressed as a percent.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Space (DBArtisan - Sybase ASE Performance Analyst)** > Databases tab
Metrics	If a database's used space percent amount goes above 90%, and the database is dynamic in nature (meaning that users are constantly adding and modifying data), then you should take action to ensure that the database does not run out of available free space.
Correction	If the percent used amount of a database is approaching problematic levels, there are two ways you can rectify the situation: *If the database device that the database currently resides on contains additional free space, you can ALTER the database to consume more available space on that device.If the database device that the database currently resides on does not contain additional free space, you can do one of the following:Create a new device and issue an ALTER for the database to use that device for space in addition to those currently used.Choose another existing device that has free space and ALTER the database to use that device in addition to those currently in use.*

Database Object Detail

Tables and indexes comprise every Sybase ASE database. The Database Object Detail view presents space-related information about tables and indexes. You can choose to view object details by table or index. Or, view objects matching a pattern you specify in the text box.

The table below describes the information available in the Database Object Detail view:

Column	Description
Owner	The owner of the table or index.
Table Name	The name of the table.
Index Name	The name of the index.
Index Type	The type of index.
Key Count	The number of keys.
Fill Factor	The percentage value of the how full each index page can be.
Reserved (KB)	The amount of space reserved for the object, in KB.
Index Size (KB)	The size of the index, in KB.
Unused (KB)	The amount of unused space (free space) that the object contains in KB.
Segment Name	The name of the segment.
Extent Estimate	An estimation as to the number of extents for the object.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Space (DBArtisan - Sybase ASE Performance Analyst)** > Databases tab

Devices tab

The following statistics are used on the Performance Analyst for Sybase ASE Space Drill-down Devices to succinctly communicate the general overall performance levels of space related metrics:

Device Summary Device Fragment Detail

Device Summary

The Device Summary view presents space related details about a selected device located on the Sybase ASE server. The table below describes the information available in the Device Summary detail view:

Column	**Description**
Device Name	The logical name of the device.
Physical Name	The physical name of the device.
Total Space (MB)	Amount of space configured when the device was created.
Used Space (MB)	Amount of space available to any database (via the create or alter database commands).
Free Space (MB)	The amount of free space left in the device.
Percent Free	The amount of free space for the device expressed as a percentage of total space.
Percent Used	The amount of used space for the device expressed as a percentage of total space.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Space (DBArtisan - Sybase ASE Performance Analyst)** > Devices tab
Metrics	If a device's used space percent amount goes above 90%, you should take action to ensure that the device does not run out of available free space.
Correction	If the percent used amount of a device is approaching problem levels, there are two ways you can rectify the situation: *Create a new device and issue an ALTER for the database(s) to use that device for space in addition to those currently used.Choose another existing device that has free space and ALTER the database(s) to use that device in addition to those currently in use.*

Device Fragment Detail

Space from a device is allocated among one or more databases on a Sybase ASE server. The Device Fragment Detail section presents all databases and their space fragments that reside on a particular device.

The table below describes the information available on the Device Fragment Detail view:

Column	Description
Database Name	The name of the database.
Type	The type of database fragment (data only, log only, data and log).
Size (MB)	The size of the fragment in MB.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Space (DBArtisan - Sybase ASE Performance Analyst)** > Devices tab

Databases (DBArtisan - Sybase ASE Performance Analyst)

In addition to a Home page, the Databases category of Sybase ASE Performance Analyst includes the following tabbed pages:

Overview Tab Objects Tab Configuration Tab

Engines Tab

Home

The Database performance category view displays the following vital Sybase ASE database statistics:

Server Analysis Pane Bottleneck Analysis Pane Backup Analysis Pane

Engine Analysis Pane Workload Analysis Pane

Server Analysis Pane

The following statistic is used on the Performance Analyst for Sybase ASE Databases Page to succinctly communicate the general overall performance levels of database specific metrics:

- Sample Query Response Time

The following statistics, available on this pane, are duplicates of statistics available on the Key Ratio Analysis Pane of the Home View (DBArtisan - Sybase ASE Performance Analyst) page:

CPU Busy I/O Busy

Sample Query Response Time

Performance Analyst allows you to define a query that is periodically run and timed by the product. The sample query helps you gauge the response time effectiveness of the Sybase ASE server.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Home > Server Analysis Pane
Metrics	Because you control the complexity of the sample query, you must determine what is a “valid” response time for the query. Once you have determined that, you can create an alarm in Performance Analyst that will alert you whenever the sample query falls outside of its acceptable baseline.

Bottleneck Analysis Pane

The following statistics are used on the Performance Analyst for Sybase ASE Databases page to succinctly communicate the general overall performance levels of database specific metrics:

Databases with Suspect Pages
Engine Configuration Limit
Index Maintenance from Deletes
Index Maintenance from Insert/Updates
Operating System Limit
Page Splits
Server Configuration Limit

The following statistics, available on this pane, are duplicates of statistics available on the Bottleneck Analysis Pane of the Home View (DBArtisan - Sybase ASE Performance Analyst) page:

Suspect Databases

Databases with Suspect Pages

Suspect objects normally indicate an internal problem in a server. They can also indicate that physical damage has occurred to part of a database.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Home > Bottleneck Analysis Pane
Metrics	Suspect objects have no place in a production database. If you identify any suspect objects, you should take immediate action.
Correction	If the suspect object is an index, you could try dropping and recreating it, or use the DBCC REINDEX command. Other damaged objects can complete rebuilding the database. A suspect database can be a difficult thing to recover from quickly. There are times when the cache of the database is suspect and not the database itself. Stopping and starting the Sybase ASE server can verify if this is the case. If the database itself is actually damaged, there could be a true recovery situation. The suspect database can be dropped using the DBCC DBREPAIR DROPDB command. You would then need to recreate the database and perform a recovery operation using the most recent database dump.

Engine Configuration Limit

An engine can exceed its limit for outstanding asynchronous disk I/O requests based on the maximum number of asynchronous I/Os per engine parameter.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Home > Bottleneck Analysis Pane
Metrics	You can change this limit with the maximum asynchronous I/Os per engine configuration parameter.

Index Maintenance from Deletes

Indexes can considerably speed data access operations. The trade-off, however, is that indexes require maintenance whenever indexed columns are added to, modified, or removed in the parent table. The index maintenance from insert/updates metric provides a count of how many index maintenance operations have occurred from insert or update operations.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Home > Bottleneck Analysis Pane
Metrics	Some index maintenance activity is to be expected (unless you are operating in a read-only environment), but excessive index maintenance can quickly become a bottleneck in heavy OLTP environments.
Correction	If you observe a lot of index maintenance activity, you should begin a periodic review of your indexing schemes to ensure that all indexes currently maintained on the system are used and necessary. If you are using Sybase ASE 12.5.03 or later (and have installed the monitoring tables), you can use Performance Analysts object I/O drill-down view to diagnose which tables and indexes are the targets of heavy maintenance activity.

Index Maintenance from Insert/Updates

Indexes can considerably speed data access operations. The trade-off, however, is that indexes require maintenance whenever indexed columns are added to, modified, or removed in the parent table. The index maintenance from insert/updates metric provides a count of how many index maintenance operations have occurred from insert or update operations.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Home > Bottleneck Analysis Pane
Metrics	Some index maintenance activity is to be expected (unless you are operating in a read-only environment), however excessive index maintenance can quickly become a bottleneck in heavy OLTP environments.
Correction	If you observe a lot of index maintenance activity, you should begin a periodic review of your indexing schemes to ensure that all indexes currently maintained on the system are used and necessary. If you are using Sybase ASE 12.5.03 or higher (and have installed the monitoring tables), you can use Performance Analysts object I/O drill down view to diagnose which tables and indexes are the targets of heavy maintenance activity.

Operating System Limit

When a Sybase ASE server begins to experience I/O delays, the result caSybase ASEn be a very dissatisfied user community. When such problems occur, you should investigate the Sybase ASE or operating system limits. It could be that I/O operations are being blocked by one or both.

The Operating System Limit statistic shows nonzero numbers if Sybase ASE detects that the limit for asynchronous I/Os has been exceeded.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Home > Bottleneck Analysis Pane
Metrics	You should be concerned if you consistently see numbers above zero.
Correction	In most UNIX operating systems, there is a kernel parameter that limits the number of asynchronous I/Os that can occur at one time. If you continue to see nonzero numbers for this statistic, you should look into raising this limit.

Page Splits

When data is inserted or updated in a table, Sybase ASE may reorganize the data storage in the table's index pages. When an index page becomes full, but a DML operation demands room on that page, Sybase ASE moves half the rows to a new page to accommodate the request. This reorganization is known as a page split.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Home > Bottleneck Analysis Pane
Metrics	Performance for DML actions can be impaired from page split operations. In addition, more index pages can make for longer index scan times.

Server Configuration Limit

When a Sybase ASE server begins to experience I/O delays, the result can be a very dissatisfied user community. When such problems begin to occur, you should investigate the Sybase ASE or operating system limits. It could be that I/O operations are being blocked by one or both limits.

The Server Configuration Limit statistic shows nonzero numbers if Sybase ASE has exceeded its limit for the number of asynchronous disk I/O requests that can be outstanding for the server at one time.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Home > Bottleneck Analysis Pane
Metrics	You should be concerned if you consistently see numbers above zero.
Correction	If you continue to see nonzero numbers for this statistic, you can raise this limit using sp_configure with the max async I/Os per server parameter.

Backup Analysis Pane

The following statistics are used on the Performance Analyst for Sybase ASE Databases page to succinctly communicate the general overall performance levels of database specific metrics:

Backup Server Status Last Transaction Log Dump

Backup Server Status

The Backup Server Status value indicates whether or not the Backup Server is up and running.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Home > Backup Analysis Pane

Last Transaction Log Dump

The Last Transaction Log Dump value represents the last time the transaction was dump for the associated database.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Home > Backup Analysis Pane

Engine Analysis Pane

The following statistics are used on the Performance Analyst for Sybase ASE Databases page to succinctly communicate the general overall performance levels of database specific metrics:

Engine Busy % Maximum Engines

Engine Busy %

The Engine Busy % value represents the percentage of time the Adaptive Server Kernel is busy executing tasks on each Adaptive Server engine (rather than time spent idle).

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Home > Engine Analysis Pane
Metrics	When Adaptive Server has no tasks to process, it enters a loop that regularly checks for network and completed disk I/Os, and tasks in the run queue. Operating system commands to check CPU activity may show high usage for an Adaptive Server engine because they are measuring the looping activity, while Engine Busy % does not include time spent looping it is considered idle time.
Correction	This category can help you decide whether there are too many or too few Adaptive Server engines. Adaptive Server's high scalability is due to tunable mechanisms that avoid resource contention. If values are consistently very high (more than 90%), it is likely that response time and throughput could benefit from an additional engine.

Maximum Engines

The Maximum Engines value represents the maximum number of engines currently connected.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Home > Engine Analysis Pane

Workload Analysis Pane

The following statistic is used on the Performance Analyst for Sybase ASE Databases page to succinctly communicate the general overall performance levels of database specific metrics:

- Workload Analysis

Workload Analysis

The Sybase ASE Server is broken up into many different databases, each of which has its own levels of activity. The Workload Analysis section displays a summary of resource consumption across all databases.

The table below describes the information available in the Workload Analysis section:

Column	Description
DB Name	The name of the database.
Connections	The number of connections to the database.
Blocked Users	The number of users currently blocked on the associated database.
Total Memory	The total amount of memory currently being used by the database.
Total I/O	The total amount of I/O currently being registered by the database.
Total CPU	The total amount of cumulative CPU for all processes connected to the database.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Home > Engine Analysis Pane

Overview Tab

The following statistics are used on the Performance Analyst for Sybase ASE Databases Detail Overview Page to succinctly communicate the general overall performance levels of database specific metrics:

Database Size Analysis Database Object Analysis Database User Analysis

Database Summary

Database Size Analysis

The Database Size Analysis chart displays each database in terms of its total percentage of size to the total size of all databases on the Sybase ASE Server.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Overview Tab

Database Object Analysis

The Database Object Analysis chart displays each database in terms of its terms of its total percentage of objects to the total objects for all databases on the Sybase ASE Server.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Overview Tab

Database User Analysis

The Database User Analysis chart displays each database in terms of its total percentage of user accounts to the total user accounts of all databases on the Sybase ASE Server.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Overview Tab

Database Summary

This section displays summarized metadata for all defined databases on the Sybase ASE Server.

Note: Right-click any column heading to sort the column(s) in ascending or descending order.

The information available in the Database Summary table is described below:

Column	Description
Database	The name of the database.
Created	Gives the month/day/year and hour/minutes/second on which the database was first generated.
Owner	Lets you see who owns the database.
Database Size (MB)	Displays the size of the database in megabytes.
Log Size (MB)	Displays the log size in megabytes.
Tables	Gives the number of tables in the database.
Indexes	Displays the number of indexes in the table.
Users	Lets you see the number of database users.
Last Trans Log Backup	Let you see the last month/day/year and hour/minutes/second on which the transaction logs were most recently backed-up.
Suspect	Lets you see in YES or NO format whether the database is suspect.
Suspect Pages	Lets you see in YES or NO format whether there are any suspect pages in the database.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Overview Tab
Metrics	The following items merit attention: *Any critical database that shows a last transaction log backup date that is older than the database's required backup needs.Any database that shows a status of offline or suspect.*

Objects Tab

he following statistics are used on the Performance Analyst for Sybase ASE Databases Detail Objects Page to succinctly communicate the general overall performance levels of database specific metrics:

Database Object Detail Suspect Objects Cache Bindings

Database Object Detail

This view gives you detailed information for each database. Select the database you want to scrutinize and choose between reviewing tables or indexes. You can also enter a preference to see objects matching at pattern you specify in the text box. The table below describes the information available in the Database Object Detail table:

Column	Description
Owner	Displays the name of the table or index owner, depending on the view you selected.
Table/Index Name	Displays the name of the table or index, depending on the view you selected.
Segment Name	Lets you see associated table/index segment.
Locking Scheme	TABLE ONLY: Lets you see the type of locking scheme the table uses.
Last Statistic Date	TABLE ONLY: Lets you see the most recent date and time table statistics were gathered.
Row Count	TABLE ONLY: Lets you see the number of rows in the target table.
Reserved (KB)	Lets you see the amount of space reserved for the target object in kilobytes.
Avg Row Size	TABLE ONLY: Lets you see the average row size for the target table.
Tree Depth	INDEX ONLY: Lets you see the number of layers, or depth, of the index.
Leaf Count	INDEX ONLY: Lets you see the number of leaf-level pages in the index.
Leaf Row Size	INDEX ONLY: Lets you see the row size for each index leaf.

Location [Sybase ASE Performance Analyst Statistics (DBArtisan)] > [Space (DBArtisan - Sybase ASE Performance Analyst)] > [Objects Tab]

Suspect Objects

Suspect objects normally indicate an internal problem in a server or that physical damage has occurred to part of a database. The table below describes the information available in the Suspect Objects grid:

Column	Description
Owner	Displays the name of the table owner.
Table Name	The affected table's name.
Index Name	The affected index's name.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Objects Tab
Metrics	Suspect objects have no place in a production database. If any suspect objects are identified, you should immediately take action.
Correction	If the suspect object is an index, you could try dropping and re-creating it, or use the DBCC REINDEX command. Other damaged objects can complete rebuilding the database. A suspect database can be a difficult thing to recover from quickly. There are times when the cache of the database is suspect and not the database itself. Stopping and starting the Sybase ASE server can verify if this is the case. If the database itself is actually damaged, there could be a true recovery situation. The suspect database can be dropped using the DBCC DBREPAIR DROPDB command. You would then need to re-create the database and perform a recovery operation using the most recent database dump.

Cache Bindings

The Sybase ASE cache areas are devoted to facilitating the transfer of data and information between clients and the Sybase ASE database. Tables and/or indexes can be assigned to various caches in hopes of speeding up access to table or index information. The Cache Bindings Detail section provides information about the current cache bindings of tables and/or indexes. The table below describes the information available in the Cache Bindings chart:

Column	Description
Owner	The object owner's name.
Table Name	The table's name.
Index Name	The index's name.
Cache Binding	The name of the cache associated with the table or index.Databa

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Objects Tab
Metrics	The objects best suited for being bound to a cache include frequently accessed objects that are oftentimes small in nature. Large tables that are scanned are normally not suitable for being bound to a cache and are best left to being indexed properly.

Configuration Tab

The following statistics are used on the Performance Analyst for Sybase ASE Databases Detail Configuration Page to succinctly communicate the general overall performance levels of database specific metrics:

Database Configuration Server Configuration

Database Configuration

The Database Configuration view allows you to see all the settings currently configured parameters and their settings on the associated database. Select a database from the drop-down list to see the parameter/setting information.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Configuration Tab

Server Configuration

The Server Configuration view allows you to see all the settings of currently configured parameters on the Sybase ASE Server. Select a database from the drop-down list to see the following information in the Server Configuration grid:

Column	Description
Parameter	Lets you see the name of the configured parameter.
Configured Value	Displays the configured value of the parameter.
Run Value	Displays the run value.
Minimum	Displays the minimum configuration value.
Maximum	Displays the maximum configuration value.
Dynamic	Indicates whether or not the parameter is dynamic or not.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Configuration Tab

Engines Tab

The following statistic is used on the Performance Analyst for Sybase ASE Databases Detail Engines Page to succinctly communicate the general overall performance levels of database specific metrics:

- Database I/O

Database I/O

The Database I/O view on the Engines tab of the Databases Detail view allows you to see statistics about database objects. The table below describes the information available to you about Database I/O:

Column	Description
Name	The name of the engine.
Start Date/Time	The date and time when the Sybase ASE engine started.
Stop Date/Time	The date and time when the Sybase ASE engine stopped
OSPID	The operating system process ID.
KPID	The kernel process ID.
Status	The engine’s current status and whether it is connected or not.
Context Switches	The current number of context switches registering on the associated engine.
Current Connections	The current number of connections registering on the associated engine.
CPU Time	The current amount of CPU time consumed by the associated engine.
CPU Time (System)	The current amount of system CPU time.
CPU Time (User)	The current amount of user CPU time.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > Space (DBArtisan - Sybase ASE Performance Analyst) > Configuration Tab

Users (DBArtisan - Sybase ASE Performance Analyst)

In addition to a Home page, the Users category of Sybase ASE Performance Analyst includes the following tabbed pages:

Top Sessions tab	Locks tab	Transactions tab
System Waits tab	Session Waits tab	

Home

The Users performance category view displays the following vital Sybase ASE user statistics:

Key User Analysis pane	Bottleneck Analysis Pane	Index Scan Analysis Pane
Transaction Analysis - Users	Workload Analysis Pane	

Key User Analysis pane

The following statistics are used on the Performance Analyst for Sybase ASE User page to succinctly communicate the general overall performance levels of user specific metrics:

Active User Processes	Extended Procedure Requests	Inactive User Processes
Parallel Queries	Procedure Requests	System Processes

Active User Processes

The Active User Processes statistic is the total number of active and open threads reported on the Server. Active Sessions displays the number of processes actively performing work.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key User Analysis pane

Extended Procedure Requests

The Extended Procedure Requests value represents the number of times extended procedures were executed in the sample interval.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key User Analysis pane

Inactive User Processes

The Inactive Sessions statistic represents the total number of threads logged onto the server that are idle at the current time.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key User Analysis pane
Metrics	A large number of inactive users could indicate user sessions that have mistakenly been left logged on. Since each user thread consumes a portion of memory on the server, to reduce resource usage, you should sever any sessions not needing a connection.
Correction	Drill down into the Session Waits page and check sessions that have many seconds idle or that have been logged on for very long periods of time, as indicated by the logon time column. After verifying that a session is no longer necessary, you can KILL it.

Parallel Queries

The Parallel Queries value represents the total number of queries eligible to be run in parallel.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key User Analysis pane

Metrics

The optimizer determines the best plan, and decides whether a query should be run serially or in parallel and how many worker processes should be used for parallel queries.

Procedure Requests

The Procedure Requests value represents the number of times stored procedures were executed in the sample interval.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key User Analysis pane

System Processes

The System processes value represents the total number of threads logged onto the server that are Sybase ASE internal processes.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key User Analysis pane

Bottleneck Analysis Pane

The following statistics are used on the Performance Analyst for Sybase ASE User Home Page to succinctly communicate the general overall performance levels of user specific metrics:

Average Hash Chain Length	Average Lock Contention %	Current Locks
Deadlock %	Lock Promotions	Lock Timeouts
Total Priority Changes		

The following statistics, available on this pane, are duplicates of statistics available on the Bottleneck Analysis Pane of the Home View (DBArtisan - Sybase ASE Performance Analyst) page:

Deadlocks Blocked Connections

Average Hash Chain Length

This value represents the average number of locks per hash bucket during the sample interval.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Correction	If the average number of locks per hash chain is more than four, consider increasing the size of the hash table. Large inserts with bulk copy are an exception to this guideline. Lock hash chain lengths may be longer during large bulk copies.

Average Lock Contention %

The Average Lock Contention % represents the average number of times there was lock contention as a percentage of the total number of lock requests.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Correction	If the lock contention average is high, investigate the locks to see if the application is causing a lock contention or deadlock-related problem.

Current Locks

There are varieties of operations in Sybase ASE that require the use of locks. The Current Locks statistic represents the number of total locks currently active in Sybase ASE.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	You should watch to see if the if the number approaches the Sybase ASE limit for available locks.
Correction	If the number of current locks in a Sybase ASE server approaches the Sybase ASE limit for available locks, you can look into increasing the Number of Locks configuration parameter.

Deadlock %

A deadlock occurs when two processes have a lock on a separate page or object and each wants to acquire a lock on the other process' page or object. Each waits for the other to release the necessary lock. Sybase ASE constantly checks for deadlocks and, when found, chooses the transaction that has accumulated the least amount of CPU time and terminates the transaction. The server then rolls back that transaction and issues a notification of the event. The other process gets to move forward.

The deadlock % rate displays the percentage of times deadlocks occurred vs. the percentage of locks requested and immediately granted.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Immediately investigate a percentage much above zero to prevent the situation from mushrooming. You can easily drill down and discover the exact process(es) holding locks and deadlocks that are blocking out other user activity.
Correction	Well-designed applications can minimize deadlocks by always acquiring locks in the same order. You should always do updates to multiple tables in the same order. Once Sybase ASE discovers a deadlock, it takes action and remedies the situation. Embarcadero Performance Analyst makes it easier to discover how prevalent deadlock activity is on a system; preventing deadlocks from occurring in the first place is more difficult. Those responsible for writing systems can minimize deadlocks by ensuring that applications acquire locks in the same order. Likewise, you should always do updates and other DML that act on multiple tables in the same order. You can also shrink the amount of time that Sybase ASE waits to check for deadlocks by modifying the deadlock checking period configuration parameter.

Lock Promotions

The Lock Promotions value represents the average number of lock promotion types combined per second and per transaction.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Lock promotions occur when the following escalations take place: *"Sh-Page to Sh-Table" - Shared page to shared table."Ex-Row to Ex-Table" - Exclusive row to exclusive table."Sh-R to Sh-Table - Shared row to shared table."Sh-Next-Key to Sh-Table" - Shared next-key to shared table."Ex-Page to Ex-Table" - Exclusive page to exclusive table.*
Correction	If lock contention is high and lock promotion is frequent, you should consider changing the lock promotion thresholds for the tables involved. You can configure the lock promotion threshold either server-wide or for individual tables.

Lock Timeouts

The Lock Timeouts value represents the total number of locks timing out during the sample period.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	You can specify the time that a task waits for a lock: *At the server level, with the lock wait period configuration parameterFor a session or in a stored procedure, with the set lock wait commandFor a lock table command*
Correction	Except for lock table, a task that attempts to acquire a lock and fails to acquire it within the time period returns an error message and the transaction is rolled back. Using lock timeouts can be useful for removing tasks that acquire some locks, and then wait for long periods of time blocking other users. However, since transactions are rolled back, and users may simply resubmit their queries, timing out a transaction means that the work needs to be repeated.

Total Priority Changes

Total Priority Changes represent the total number of priority changes during the sample period.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	It is normal to see some priority switching due to system-related activity. Such priority switching occurs, for example, when: *A task sleeps while waiting on a lock - Adaptive Server temporarily raises the task's priority.The housekeeper task sleeps - Adaptive Server raises the priority to medium while the housekeeper sleeps, and changes it back to low when it wakes up.A task executes a stored procedure - the task assumes the priority of the stored procedure and resumes its previous priority level after executing the procedure.*
Correction	If you are using logical process management and there are a high number of priority changes compared to steady state values, it may indicate that an application, or a user task related to that application, is changing priorities frequently. Check priority change data for individual applications. You should verify that applications and logins are behaving expected. If you determine that a high-priority change rate is not due to an application or to related tasks, then it is likely due to system activity.

Index Scan Analysis Pane

The following statistics are used on the Performance Analyst for Sybase ASE User page to succinctly communicate the general overall performance levels of user specific metrics:

Ascending
Descending
DOL Ascending
DOL Descending
Total Index Scans

Ascending

The Ascending value represents the number of forward scans on allpages-locked tables.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Index Scan Analysis Pane

Descending

The Descending statistic represents the number of backward scans on allpages-locked tables.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Index Scan Analysis Pane

DOL Ascending

The DOL Ascending value represents the number of forward scans on data-only-locked tables.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Index Scan Analysis Pane

DOL Descending

The DOL Descending value represents the number of backward scans on data-only-locked tables.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Index Scan Analysis Pane

Total Index Scans

The Total Index Scans statistic represents the total number of index scans (forward and backward).

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Index Scan Analysis Pane

Transaction Analysis - Users

The following statistics are used on the Performance Analyst for Sybase ASE User page to succinctly communicate the general overall performance levels of user-specific metrics:

Active Transactions Committed Transactions

Active Transactions

The Active Transaction statistic represents a count of the number on in-process transactions for SQL Server.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Index Scan Analysis Pane

Committed Transactions

Committed Transactions gives the number of transactions committed since the last refresh inside Performance Analyst. This includes transactions that meet explicit, implicit, and ANSI definitions for committed transactions. Note that multi-database transactions are counted.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Index Scan Analysis Pane
Metrics	Multi-database transactions generally incur more overhead than single database transactions (for example, a transaction that modifies two databases is counted as two transactions). They usually involve more log activity and two-phase commits between the different databases, as well as cause more ULC flushes. You can improve performance by reducing the number of multi-database transactions.

Workload Analysis Pane

The following statistics are used on the Performance Analyst for Sybase ASE User page to succinctly communicate the general overall performance levels of user specific metrics:

- Workload Analysis

Workload Analysis

When your phone starts ringing with complaints of performance slowdowns, some of the first questions you should answer are:

- What users are currently connected to the database?
- What resources are they using?
- What are they currently executing?

The Workload Analysis section of the Performance Analyst page provides insight into the leading resource hogs of a server. The table below describes the information available in the Workload Analysis section of the Users page:

Column	Description
Activity	Statistics are presented for the Top CPU Process, Top I/O Process, or Top Memory Process.
SPID	The process ID.
Login	The logon name the session is using.
% Used	The percentage of CPU, I/O, or Memory consumed by the process.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Home > Workload Analysis Pane
Metrics	If any one session appears to be using more than 50% of a total resource (CPU, memory, etc.), then you should drill down into the session to find out what they are currently executing.

Top Sessions tab

The following statistics are used on the Performance Analyst for Sybase ASE Users Drill-down Top Sessions to succinctly communicate the general overall performance levels of user specific metrics:

Top I/O Processes Top Memory Processes Top CPU Processes

Top Sessions

Top I/O Processes

The Top I/O Processes statistic identifies the current Sybase ASE process with the highest % of I/O activity.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Top Sessions tab
Correction	If any one session uses more than 50% of a total resource (CPU, memory, etc.), you should drill down into that particular session and investigate the cause.

Top Memory Processes

The Top Memory Process statistic identifies the Sybase ASE process that currently is using the highest percentage of memory in the database.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Top Sessions tab

Top CPU Processes

The Top CPU Process statistic identifies the Sybase ASE process that currently has the highest percentage of CPU usage in the database.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Top Sessions tab

Top Sessions

When a system experiences heavy activity, sometimes you will find that all the user connections are contributing somewhat equally to the overall load. More often than not, however, one or two user connections are causing most of the activity. It might be that a large batch load or other typical process is running that is perfectly okay for your system. Or it might be a runaway process or other rogue connection that needs to be tracked down and possibly eliminated.

The table below describes the information available on the Top Sessions detail view:

Column	Description
SPID	The process ID.
User	The name of the session user.
FID	The process ID of the worker process' parent.
Status	The current status of the session.
Database	The database the session is attached to.
Host	The client machine name the session is using.
Program	This identifies the program being run by the session.
Mem Usage	The amount of memory being used by the session.
CPU Time	The cumulative CPU time for the process.
Physical Reads	The number of physical reads produced by the session.
Logical Reads	The number of logical reads produced by the session.
Physical Writes	The number of physical writes generated by the session.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Top Sessions tab

Locks tab

The following statistics are used on the Performance Analyst for Sybase ASE User Drill-down Locks tab to succinctly communicate the general overall performance levels of user specific metrics:

Locks Locks by Lock Type Locks by Database

Locks

To modify database information or structures, a user session must obtain a lock on the object to perform its task. In addition to user locks, Sybase ASE itself issues lock requests to carry out its internal duties. The Locks section gives information about the locks currently on the system and also indicates if any blocking situations are occurring. You can elect to show only blocking locks in the Lock detail view.

The table below describes the information available on the Locks detail view:

Column	Description
SPID	The process ID.
User	The name of the current user
Database	The name of the database where the process is running.
Lock Type	The type of lock (database, table, row id, etc.).
Object Name	The name of the object involved in the lock.
Status	The lock's status (waiting or granted).
Lock Page	The number of pages, if any, that are experiencing locks.
Lock Class	Lets you see the lock class.
Host	The client machine name the session is using.
Program	The executable the process is using against the server.
Command	The command currently being issued by the process.
CPU Time	The amount of time the process took to parse, execute, and fetch the data needed to satisfy the process’ execution.
I/O	The amount of I/O activity for the target process.
Mem Usage	The amount of memory being used by the target process.
FID	The process ID of the worker process' parent.
Transaction	The current transactions generated by processes on the Sybase ASE Server

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Locks tabSybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Locks/Waits
Metrics	A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency. Unless a process times out via an application timeout mechanism, or the process has specified a timeout period, a process waiting for a lock will wait indefinitely.
Correction	Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects the user was accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is where it gets tricky. The DBA can drill down into user detail and view all current blocking locks to see exactly which sessions are holding the currently restrictive locks. The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

Locks by Database

The Locks by Database view represents the total locks occurring on the Sybase ASE Server stratified by which database they are occurring on.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Locks tab

Locks by Lock Type

To modify database information or structures, a user session must obtain a lock on the object to perform its task. In addition to user locks, Sybase ASE itself issues lock requests to carry out its internal duties. The Locks by Lock Type section provides a breakdown of the different locks types being issued across all databases.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Locks tab

Transactions tab

The Transactions view displays current transactions generated by processes on the Sybase ASE Server.

The table below describes the information available on the Transactions detail view:

Column	Description
SPID	The process ID.
User	The name of the user generating the session.
Status	The current status of the session.
Start Time	The time the session started.
Type	The type of session.
State	The current state of the session, for example, whether or not it is active.
Command	The command currently being issued by the session.
Connection	The amount of memory allocated to the session's connections
CPU Time	The amount of time the process took to parse, execute, and fetch the data needed to satisfy the process' execution.
Mem Usage	The amount of memory being used by the session.
Physical I/O	The amount of physical I/O for the session.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Transactions tab
Metrics	Transaction activity is a good measurement of general performance by a user. If the total or committed numbers do not increase over a period of time, either the process is inactive or it may be hung or blocked.
Correction	Look at CPU and I/O statistics to see if any activity for the process is being noted there.

System Waits tab

The following statistics are used on the Performance Analyst for Sybase ASE User Drill-down System Waits tab to succinctly communicate the general overall performance levels of user specific metrics:

System Waits Wait Percent by Total Waits (By Wait Class) Wait Percent by Time Waited (By Wait Class)

System Waits

When the Sybase ASE Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The System Waits section identifies the waits that have occurred on the Sybase ASE Server.

The table below describes the information available on the System Waits detail view. You can elect to display the internal system and scheduler events.

Column	Description
Wait Event	The name of the wait event.
Wait Class	The type of wait, often called a wait category or class.
Wait Count	The number of waits that have occurred.
Percent Waits	The percent of waits as compared to the total number of waits.
Wait Time	The amount of time waited in seconds.
Percent Wait Time	The percent of wait time as compared to the total wait time.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > System Waits tab

Wait Percent by Total Waits (By Wait Class)

When the Sybase ASE Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The Wait Percent by Total Waits section graphically depicts the top waits by their percentage to all waits.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > System Waits tab

Wait Percent by Time Waited (By Wait Class)

When the Sybase ASE Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The Wait Percent by Time Waited section graphically depicts the top waits by their percentage of time waited to total time waited.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > System Waits tab

Session Waits tab

The following statistics are used on the Performance Analyst for Sybase ASE User Drill-down Locks tab to succinctly communicate the general overall performance levels of user specific metrics:

Session Waits Wait Percent by Total Waits (By Wait Class) Wait Percent by Time Waited (By Wait Class)

Session Waits

Session waits is merely a subset of contention that is viewable at the global database level. Often it takes analysis at the session level to pinpoint the exact source of the wait that is occurring globally.

Therefore, you need to become accustomed to viewing contention statistics and waits at the user process level.

When monitoring waits with respect to user sessions, there are two areas of interest:

- What HAS the user session been waiting on?
- What IS the user session waiting on?

The table below describes the information available on the Session Waits detail view. You can elect to include background process in the detail display.

Column	Description
SPID	The unique identifier for the process.
User	The name of the process' user.
FID	The process ID of the worker process' parent.
Status	The current status of the process.
Database	The name of the database where the process is running.
Hostname	The client machine name where the session is running.
Program Name	The name of the program being run by the process.
Wait Event	The name of the wait event.
Wait Class	The type of wait, often called a wait category or class.
Waits	The number of waits that have occurred.
Process Waits	The percent of waits as compared to the total number of waits.
Wait Time (s)	The number of seconds the process spent waiting for this wait.
Percent Waits	The percent of wait time as compared to the total wait time.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Session Waits tab
Metrics	To determine the actual wait causes currently experienced by user connections, you should drill down from the global count of users waiting, into the actual system and user wait details.
Correction	If you find a problem, drill down into wait details to determine whether the waits are resource-related.

Wait Percent by Total Waits (By Wait Class)

The Wait Percent by Total Waits section graphically depicts the waits that have experienced the highest percentage of occurrences on the system.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Session Waits tab

Wait Percent by Time Waited (By Wait Class)

When the Sybase ASE Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting might mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The Session Wait Percent by Time Waited section graphically depicts the top session waits by their percentage of time waited to total time waited.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Session Waits tab

Session Details

The Sessions Detail view includes the following tabbed pages:

SQL tab I/O tab Object Access tab

Locks/Waits

SQL tab

The SQL tab of the Sessions Detail view offers the following information:

Current SQL Current Procedures

Current SQL

This view shows a snippet of the SQL associated with the user in question.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Session Waits tab

Current Procedures

Here, in tabular format, you find the following information about the user's current procedures:

Owner

Object Name

Object Type

Database: The name of the database on which the procedure resides.

Compile Date: The last date on which the object was compiled.

Memory (KB): The number of pages in the procedure cache that is currently allocated to the process. A negative number indicates that pages are being released (freed) from the process.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Session Waits tab

I/O tab

The I/O tab of the Session Detail view displays the following statistical information about the current user's activities in a graphical display:

Physical Reads

Logical Reads

Pages Read

Pages Written

Physical Writes

Table Accesses (scans): The number of table accesses (scans) by the user.

Index Accesses: The number of index accesses by the user.

Tempdb Objects: The number of times the Tempdb was accessed by the user.

Object Access tab

The Object Access tab of the Session Detail view displays information about the current objects being accessed. Here you'll see, in tabular format:

Database: The name of the database

Owner ID: The ID number assigned to the process accessing the object.

Object Name

Object Type: The type of object being accessed.

Index ID

Physical Reads: The number of physical disk reads issued against the object.

Physical APF Reads: Number of physical Asynchronous Pre-Fetch buffers read.

Logical Reads: The number logical reads issued against the object.

Locks/Waits

The Lock/Waits tab of the Session Details page offers statistics on:

- Waits

The following statistic, available on this tab, duplicates a statistic on the Locks tab of the **Users (DBArtisan - Sybase ASE Performance Analyst)** page:

- Locks

Waits

When the Sybase ASE Server is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. The Waits section shows processes that are currently waiting on the Sybase ASE Server.

Here you find, in tabular format, details for:

Login: The logon name used by the session.

Database: The name of the database the process is attached to.

Wait Event: The name of the wait event.

Wait Class: The type of wait, often called a wait category or class.

Waits: The number of waits that have occurred.

Wait Time (sec): The amount of time waited in seconds.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **Users (DBArtisan - Sybase ASE Performance Analyst)** > Session Waits tab

OS (DBArtisan - Sybase ASE Performance Analyst)

In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database sever for resources (CPU, I/O, or Memory), a slow CPU, insufficient or slow I/O devices, and insufficient memory. The OS Statistics page of Performance Analyst lets you examine operating system metrics for the following platforms:

- AIX
- HP-UX
- Linux
- Solaris
- Unix
- Windows XP and 2000

Note: The statistics available on the OS page depend on the operating system platform.

Tip: If you magnify a graph to change back to the original size or minimize the size, close the OS Page and then reopen it again. Performance Analyst displays the graph in its original form and size.

In addition to a Home page, the OS category of Sybase ASE Performance Analyst includes the following tabbed pages:

CPU Tab
I/O Tab
Memory Tab
Network Tab
Processes Tab
Space Tab

Home

The OS home page includes the following sections:

Bottleneck Analysis Pane
Disk Analysis Pane
Key Resource Usage Pane
Memory Analysis Pane
Workload Analysis Pane

Key Resource Usage Pane

The following ratios are used on the Performance Analyst OS home page to communicate the general overall performance levels of the operating system:

Disk Time
Load Average
Processor Time
Paged Memory Used (Windows)
Swap Memory Used (AIX, HP-UX, Linux, Solaris, Unix)

Tip: To open the CPU tab, right-click Processor Time. To open the I/O tab, right-click Disk Time. To open the Memory tab, right-click Paged Memory Used.

Disk Time

The Disk Time statistic is the percentage of elapsed time that the selected disk drive/device was busy servicing read or write requests.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	You should avoid consistently seeing values for this statistic greater then 90%.
Correction	Add more disk drives and partition the files among all of the drives.

Load Average

The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	High load averages usually mean the system is being used heavily and the response time is correspondingly slow.

Processor Time

The Processor Time statistic indicates the percentage of time the processor is working. This counter is a primary indicator of processor activity.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	If your computer seems to be running sluggishly, this statistic could be displaying a high percentage.
Correction	Upgrade to a processor with a larger L2 cache, a faster processor, or install an additional processor.

Paged Memory Used

The Paged Memory Used statistic is the ratio of Commit Memory Bytes to the Commit Limit. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk. The commit limit is determined by the size of the paging file. As the paging file increases, so does the commit limit.

Note: This statistic is available for the Windows platform.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	This value displays the current percentage value only and not an average. If the percentage of paged memory used is above 90%, you may be running out of memory.
Correction	Increase the size of page file.

Swap Memory Used

The Swap Memory Used statistic is the percentage of swap space currently in use.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	If the percentage of swap memory used is above 90%, you may be running out of memory.
Correction	Increase the size of your swap files.

Bottleneck Analysis Pane

The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

Average Disk Queue Length	Interrupts/Sec	Network Output Queue Length/Network Queue Length
Page Faults/Sec	Processor Queue Length	Processor Speed
Processor		

Tip: To open the I/O tab, right-click any Details menu item. To open the CPU tab, right-click the Item Processor Queues Length Details menu item. To open the Network tab, right-click Network Output Queues Length.

Note: The statistics available in this section depend on the platform of operating system.

Average Disk Queue Length

The Average Disk Queue Length statistic is the average number of both read and write requests that were queued for the selected disk during the sample interval.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	This metric is useful in identifying I/O related bottlenecks. If the disk queue lengths for certain disks are consistently much higher than others, you may need to redistribute the load among available disks. If the disk queues lengths for all disks are consistently large, and you see a high amount of I/O activity, your disks may be inefficient.
Correction	Some things you can do if you have problems with this statistic include: *Redistribute the data on the disk with the large average disk queue to other disks.Upgrade to faster disk(s).*

Interrupts/Sec

Interrupts/Sec is the average rate, in incidents per second, at which the processor received and serviced hardware interrupts. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended. The system clock typically interrupts the processor every 10 milliseconds, creating a background of interrupt activity. This statistic shows the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > CPU Tab
Metrics	A high value indicates possible excessive hardware interrupts; justification is dependent on device activity.

Network Output Queue Length/Network Queue Length

The Network Output Queue Length statistic is the number of threads in the processor queue.

Note: The name of this statistic depends on the platform of the operating system.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.
Correction	A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Page Faults/Sec

The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Memory Tab
Metrics	This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
Correction	If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Processor Queue Length

The Processor Queue Length statistic is the number of threads in the processor queue.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis PaneSybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > CPU Tab
Metrics	Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.
Correction	A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Processor Speed

The Processor Speed statistic displays the speed of the active processor in MHz. The speed is approximate.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane

Processor

The Processor Statistic displays the type of processor currently in use, for example, GenuineIntel.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Bottleneck Analysis Pane

Memory Analysis Pane

The following metrics are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

Available Paged Memory (Windows)
Used Disk Space
Available Swap Memory
Total Physical Memory
Available Swap Memory(AIX, HP-UX, Linux, Solaris, Unix)

Tip: To open the Memory tab, right-click any Details menu item.

Note: The statistics available in this section depend on the platform of operating system.

Available Paged Memory

The Available Paged Memory statistic shows the amount of virtual memory available for the processes.

Note: This statistic is available for the Windows platform.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Memory Analysis Pane
Metrics	If the available virtual memory is less than 10% of the total virtual memory, your system may run out of memory.
Correction	Increase the size of page file.

Available Physical Memory

The Available Physical Memory statistic represents the amount of RAM available to all processes.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Memory Analysis Pane
Metrics	This counter displays the last observed value only and not an average. Use this value with the Total physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.
Correction	Some things you can do if you have problems with this statistic include: *Check the running processes to see if there are any memory leaks.Stop any services that are not required.Install additional RAM.*

Available Swap Memory

The Available Swap Memory statistic represents the amount of virtual memory available for the processes.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Memory Analysis Pane
Metrics	If the available Available Swap Memory is less than 10% of the total Swap Memory, your system may run out of memory.
Correction	Increase the size of swap files.

Total Paged Memory/Total Swap Memory

The Total Paged Memory statistic shows the maximum amount of virtual memory available to all processes.

Note: The name of this statistic depends on the platform of the operating system.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Memory Analysis Pane
Metrics	It is recommended that this be 1½ - 3 times the amount of RAM on the system.

Total Physical Memory

The Total Physical Memory statistic shows the amount of physical memory installed on your computer.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Memory Analysis Pane
Metrics	This is an informational metric and displays the total amount installed on the machine. Use this value with the available physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Disk Analysis Pane

The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

Free Disk Space Total Disk Space Used Disk Space

Tip: To open the Space tab, right-click any Details menu item.

Note: The statistics available in this section depend on the platform of operating system.

Free Disk Space

The Free Disk Space statistic shows the unallocated space, in megabytes on all logical disk drives.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Disk Analysis Pane
Correction	There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations, here are two: *If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.*

Used Disk Space

The Used Disk Space statistic shows the amount of allocated space, in megabytes on all logical disk drives.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Disk Analysis Pane
Correction	There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations, here are two: *If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.*

Total Disk Space

Total Disk Space displays the total allocated and unallocated space, in megabytes on all logical disk drives.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Disk Analysis Pane
Correction	There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations, here are two: *If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.*

Workload Analysis Pane

The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

Number of Logins Number of Processes Top CPU Process

Top I/O Process Top Memory Process

Tip: To open the Processes tab, right-click any Details menu item.

Note: The statistics available in this section depend on the platform of operating system.

Top Memory Process

Top Memory Process shows the current process that is consuming the most amount of memory. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Workload Analysis Pane
Metrics	If you are running out of memory on the system, this is a quick way to identify the top memory user. If the displayed process is using a significant portion of the total memory, it could be causing the memory issues.

Top CPU Process

Top CPU Process shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Workload Analysis Pane
Metrics	If the amount of CPU time used by this process is close to 100% and the CPU usage is very high, this process may be the bottleneck on the server.
Correction	Investigate the process further to see if it is in an inconsistent state. Also, look at minimum requirements for CPU speed for the process. You may need to upgrade your CPU.

Top I/O Process

The Top I/O Process statistic shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Workload Analysis Pane

Number of Logins

This statistic displays the total number of logins on the server.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Workload Analysis Pane

Number of Processes

This statistic displays the total number of processes on the server.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Home > Workload Analysis Pane

CPU Tab

The CPU tab of the OS Detail includes the following sections:

Context Switches/Sec CPU Events CPU Utilization

Load Averages

The following statistics, available on this tab, are duplicates of statistics available on the Bottleneck Analysis Pane of the **OS (DBArtisan - Sybase ASE Performance Analyst)**Home page:

Interrupts/Sec Processor Queue Length

Note: The sections available on this tab depend on the platform of operating system.

CPU Utilization

The CPU Utilization section includes the following information:

% Privileged Time % User Time

% Privileged Time

The % Privileged Time statistic is the percentage of elapsed time that the process threads spent executing code in privileged mode.

Note: For Windows systems, when a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. These kernel commands, are considered privileged to keep the low-level commands executing and prevent a system freeze. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > CPU Tab
Metrics	The ideal range should be 0-40% (less than 40% indicates excessive system activity).
Correction	If your CPU consistently runs at less than 40% you may need to upgrade your system to include a faster processor(s).

% User Time

The % User Time statistic is the percentage of elapsed time the processor spends in the user mode. User mode is a restricted processing mode designed for applications, environment subsystems, and integral subsystems. The alternative, privileged mode, is designed for operating system components and allows direct access to hardware and all memory. The operating system switches application threads to privileged mode to access operating system services. This counter displays the average busy time as a percentage of the sample time.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > CPU Tab
Metrics	If the Privileged Time is high in conjunction with Physical Disk Reads, consider upgrading the disk I/O subsystem.

CPU Events

The CPU Events section includes the following information:

Context Switches/Sec Load Averages System Calls/Sec

Note: The statistics available in this section depend on the platform of operating system.

Context Switches/Sec

The Context Switches/Sec section shows the combined rate at which all processors on the computer are switched from one thread to another. Context switches occur when a running thread voluntarily relinquishes the processor, is preempted by a higher priority ready thread, or switches between user-mode and privileged (kernel) mode to use an Executive or subsystem service.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > CPU Tab
Metrics	The ideal range should be between 0-10,000. GA number greater then 10,000 may indicate too many threads contending for resources.

System Calls/Sec

System calls are functions which require the services of the operating system. Usually they involve access to some data that users must not be permitted to corrupt or even change.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > CPU Tab
Correction	Try to find a good ratio between the amount of time the operating system runs (fulfilling system calls and doing process scheduling), and the amount of time the application runs. Your goal should be running 60% to 75% of the time in application mode, and 25% to 40% of the time in operating system mode. If you find that the system is spending 50% of its time in each mode, then you should investigate to determine what is wrong.

Load Averages

The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > CPU Tab
Metrics	High load averages usually mean that the system is being used heavily and the response time is correspondingly slow.

Processes Tab

The Processes tab of the OS Detail page succinctly communicates the general overall performance levels of processes. The columns available in this table depend on the platform of operating system. The table below describes the information available in the table on this tab:

Column	Description
Process	The name of the process.
User	The user of the process.
ID	The ID Process is the unique identifier of this process. ID Process numbers are reused, so they only identify a process for the lifetime of that process.
CPU	The CPU is the percentage of elapsed time that all of process threads used the processor to execution instructions.
User Mode	The User Mode is the percentage of elapsed time that the process threads spent executing code in user mode.
Memory WINDOWS ONLY	Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Memory (MB)	Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Memory	Memory is the percentage of the memory used of the total memory.
Active Memory	Active Memory is the amount of committed virtual memory, in bytes for this process. Active memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive. This counter displays the last observed value only; it is not an average.
I/O Data	The rate at which the process is reading and writing bytes in I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.
Elapsed Time	The total elapsed time, in seconds, that this process has been running.
Thread Count	The number of threads currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.
Handle Count	The total number of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in this process.
Priority	The current base priority of this process. Threads within a process can raise and lower their own base priority relative to the process' base priority.
Creating Proc ID	The Creating Process ID value is the Process ID of the process that created the process. The creating process may have terminated, so this value may no longer identify a running process.
Page Faults/Sec	Page Faults/Sec is the rate at which page faults by the threads executing in this process are occurring. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This may not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with whom the page is shared.

Page File	Page File is the current number of kilobytes that this process has used in the paging file(s). Paging files are used to store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory.
Private	Private is the current size, in kilobytes, of memory that this process has allocated that cannot be shared with other processes.

I/O Tab

The I/O tab includes the following sections:

Average Disk Queue/Total Disk Queue	Bytes per I/O Operation	Disk I/O Time
Disk Transfers/Sec	I/O Details	Transfer Rate

Note: The sections available on this tab depend on the platform of operating system.

I/O Details

The table below describes the information available in this section:

Column	Description
Disk	The disk number assignment.
Reading (KB/s)	The amount of bytes read from the device.
Writing (KB/s)	The amount of bytes written to the device.
Disk Read Time	Disk Read Time is the percentage of elapsed time that the selected disk drive was busy servicing read requests.
Disk Write Time	Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.
Disk Time	Disk Time is the percentage of elapsed time that the selected disk was busy servicing requests.
Avg. Read Queue	Avg. Disk Read Queue Length is the average number of read requests that were queued for the selected disk during the sample interval.
Avg. Write Queue	Avg. Disk Write Queue Length is the average number of write requests that were queued for the selected disk during the sample interval.
Disk Reads/Sec	Disk Reads/Sec is the rate of read operations on the disk.
Disk Writes/Sec	Disk Writes/Sec is the rate of write operations on the disk.

Note: The columns available in this table depend on the platform of operating system.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > I/O Tab

Transfer Rate

The Transfer Rate section succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

Statistic	Description	Metrics
Reading	The average number of bytes transferred from the disk during read operations.	It is measured in number of KBs per second.
Writing	The average number of bytes transferred to the disk during write operations.	It is measured in number of KBs per second.
Transferred	The amount of time required to transfer the data between the device's cache and the actual device media.	N/A

Note: The statistics available in this section depend on the platform of operating system.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > I/O Tab

Bytes per I/O Operation

The Bytes per I/O Operation section of the OS Detail succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

Statistic	Description	Metrics
Reading	The average number of bytes transferred from the disk during read operations.	It is measured in number of bytes per second.
Writing	The average number of bytes transferred to the disk during write operations.	It is measured in number of bytes per second.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > I/O Tab

Average Disk Queue/Total Disk Queue

The Average Disk Queue section of the OS Detail page succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

Statistic	Description	Information
Average Read Queue	Average Read Queue Length is the average number of read requests that were queued for a particular disk.	Sustained loads on any of the disk queue lengths could represent problems within your system.
Average Write Queue	Average Write Queue is the average number of write requests that were queued for a particular disk.	Sustained loads on any of the disk queue lengths could represent problems within your system. Contact your Windows System Administrator for assistance in Correction these problems.
Average Queue	Average time in ms. to service I/O requests which for disks includes seek, rotational latency and data transfer time).	Sustained loads on any of the disk queue lengths could represent problems within your system. Contact your Unix System Administrator for assistance in Correction these problems.

Note: The statistics available in this section depend on the platform of operating system.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > I/O Tab
Correction	Contact your Windows System Administrator for assistance in Correction these problems.

Disk I/O Time

The Disk I/O Time section of the OS Detail page succinctly communicates the general overall performance levels of the disk's I/O. The table below describes the statistics in this section:

Statistic	Description	Metrics
% Disk Read Time	% Disk Read Time is the percentage of time during the sample interval that the disk is busy servicing read requests.	Use this in conjunction with the % Disk Write Time to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.
% Disk Write Time	% Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.	Use this in conjunction with the % Disk Read Time to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.
% Disk Time	% Disk Time is the percentage of time during the sample interval that the disk is busy servicing requests.	Use this statistic to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.

Note: The statistics available in this section depend on the platform of operating system.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > I/O Tab

Disk Transfers/Sec

The Disk Transfers/Sec section of the OS Detail page succinctly communicates the general overall performance levels of the disk transfers. The table below describes the statistics in this section

Statistic	Description
Reads/Sec	The the rate of read operations on the disk.
Writes/Sec	The rate of write operations on the disk.
Transfers/Sec	The rate of transfer operations per second.

Note: The statistics available in this section depend on the platform of operating system.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > I/O Tab

Memory Tab

The Memory tab of the OS Detail page includes the following sections:

Buffer Size
Memory Available
Cache Efficiency
Paging Activity
Cache Size
Page Replacements

The following statistic, available on this tab, duplicates a statistic available on the Bottleneck Analysis Pane of the **OS (DBArtisan - Sybase ASE Performance Analyst)**Home page:

- Page Faults/Sec

Note: The sections available on this tab depend on the platform of operating system.

Paging Activity

The Paging Activity section includes the following statistics:

Blocks Input/Sec
Paged Out
Blocks Output/Sec
Pages Input/Sec
Paged In
Pages Output/Sec

Note: The statistics available in this section depend on the platform of operating system.

Blocks Input/Sec

The Blocks Input/sec statistic is the number of blocks sent to a block device per second.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Memory Tab

Correction Although it never hurts to have as much physical memory as your system can handle, here are two things you can check within your system to alleviate the memory bottleneck: *Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.Check to see if you have additional space on your disk drive that you could use to expand the size of your swap file. Normally, the bigger the initial size of your swap file, the better, in performance terms.*

Blocks Output/Sec

The Blocks Output/sec statistic is the number of blocks received from a block device per second).

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Memory Tab

Correction Although it never hurts to have as much physical memory as your system can handle, here are two things you can check within your system to alleviate the memory bottleneck. *Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.Check to see if you have additional space on your disk drive that you could use to expand the size of your swap file. Normally, the bigger the initial size of your swap file, the better, in performance terms.*

Paged In

The number of disk block paged into core from disk which occurs when the server is experiencing a shortage of RAM. Any nonzero value indicates excessive swap activity. In any virtual memory environment, it is not uncommon to see RAM pages moved out to a swap disk. This is a special disk area in UNIX that's reserved for holding memory pages so that the processor is capable of addressing RAM in excess of its full capability. While page-out of operations are a normal part of any server's operations, page-in operations indicate that the real amount of available RAM has been exceeded and that additional RAM pages are required on the server.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Memory Tab

Correction High values indicate the possible need to increase the amount of RAM on the system.

Paged Out

The number of disk block paged out of core from disk. In any virtual memory environment, it is not uncommon to see RAM pages moved out to a swap disk. This is a special disk area in UNIX that's reserved for holding memory pages so that the processor is capable of addressing RAM in excess of its full capability. While page-out of operations are a normal part of any server's operations, page-in operations indicate that the real amount of available RAM has been exceeded and that additional RAM pages are required on the server.

Pages Input/Sec

The Pages Input/Sec statistic is the number of pages read from disk to resolve hard page faults. Hard page faults occur when a process requires code or data that is not in its working set or elsewhere in physical memory, and must be retrieved from disk.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Memory Tab
Metrics	This value was designed as a primary indicator of the kinds of faults that cause system-wide delays. It includes pages retrieved to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. This counter counts numbers of pages, and can be compared to other counts of pages, such as Memory: Page Faults/sec, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
Correction	Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck. *Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.*

Pages Output/Sec

This metric is the number of pages written to disk to free up space in physical memory. Pages are written back to disk only if they are changed in physical memory. A high rate of pages output might indicate a memory shortage.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Memory Tab
Metrics	Windows NT writes more pages back to disk to free up space when low in physical memory. This counter counts numbers of pages, and can be compared to other counts of pages, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
Correction	Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck. *Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.*

Memory Available

The Memory Available section of the OS Detail page succinctly communicates the general overall performance levels of the server's memory. The statistics available in this section depend on the platform of operating system. The table below describes these statistics:

Statistic	Description	Metrics	Correction
Physical Memory	The Physical Memory statistic is the amount of committed virtual memory, in megabytes. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk.	This counter displays the last observed value only and not an average.	N/A
Physical	The Physical statistic is the total physical memory available.	N/A	N/A
Paged Memory	The Paged Memory statistic is the amount of committed virtual memory, in bytes. Used Memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive.	This counter displays the last observed value only; it is not an average.	N/A
Swap Memory	The Swap Memory statistic is the free, unreserved swap space.	N/A	Increase the size of your swap file or add additional files if you consistently see a shortage of free, unreserved swap space.
Swap	The Swap statistic is the total swap memory being used.	N/A	N/A
Total Physical	The Total Physical statistic is the total physical memory available.	N/A	N/A
Total Swap	The Total Swap statistic is the total swap memory available.	N/A	N/A
Swap-ins	The number of memory pages paged in from swapspace.	N/A	If the system is consistently low on free memory (over a 30 second average), the memory scheduler will start to swap out processes. Increase the size of your swap file or add additional files.
Swap-outs	The number of memory pages paged in from swapspace.	N/A	If the system is consistently low on free memory (over a 30 second average), the memory scheduler will start to swap out processes. Increase the size of your swap file or add additional files.
Buffers	The Buffers statistic represents the amount of buffers used by the kernel in kilobytes.	N/A	N/A

Page Faults/Sec	The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.	This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.	If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Buffer Size

The Buffer Size statistic represents the amount of cache used by the kernel in kilobytes.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Memory Tab

Cache Size

The Cache Size statistic represents the amount of cache used by the kernel in kilobytes.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Memory Tab

Page Replacements

The following statistics are available in this section:

Memory Freed (Pages/sec) Clock Algorithm Scans (Pages/sec)

Memory Freed (Pages/sec)

Freed or destroyed (Kb/s).

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Memory Tab

Clock Algorithm Scans (Pages/sec)

The VMM uses a technique known as the clock algorithm to select pages to be replaced. This technique takes advantage of a referenced bit for each page as an indication of what pages have been recently used (referenced). When the page-stealer routine is called, it cycles through the PFT, examining each page's referenced bit.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Memory Tab

Cache Efficiency

The Cache Efficiency section of the Memory tab succinctly communicates the general overall performance levels of the server's memory. The following statistics are available in this section:

Copy Read Hits % Data Map Hits % MDL Read Hits %

Pin Read Hits %

Copy Read Hits %

The Copy Read Hits % statistic is the percentage of cache copy read requests that hit the cache and does not require a disk read to provide access to the page in the cache.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Memory Tab
Metrics	When the page is pinned in the memory, the page's physical address in the file system cache will not be altered. A copy read is a file read operation where a page in the cache is copied to the application's buffer. Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.
Correction	Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate

Data Map Hits %

The Data Map Hits % statistic is the percentage of data maps in the file system cache that could be resolved without having to retrieve a page from the disk.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Memory Tab
Metrics	Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.
Correction	Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

MDL Read Hits %

The MDL Read Hits % statistic is the percentage of Memory Descriptor List Read requests to the file system cache that hit the cache and does not require disk access to provide memory access to the pages in the cache.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Memory Tab
Metrics	Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.
Correction	Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Pin Read Hits %

The Pin Read Hits % statistic is the percentage of pin read requests that hit the file system cache and does not require a disk read in order to provide access to the page in the file system cache.

Location	Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Memory Tab
Metrics	Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Correction

Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Space Tab

The Space tab of the OS Detail page includes the following sections:

Device Summary Device Detail

Disk Space Free

The Disk Space Free metric displays the amount of free disk space in megabytes.

Disk Space Detail

The Disk Space Detail section of the Space tab succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

Statistic	Description
Partition	The drive letter of the disk.
Local Filesystem	The name of the file system.
Type	The type of file system.
Total Space	Total size of the disk/device's capacity expressed in MBs.
Used Space	Amount of MBs currently allocated on the particular disk/device.
Free Space	Amount of MBs currently unallocated and free on the particular disk/device.
Capacity	The percentage of space used on the device.
Mounted On	The mount point of the device.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Space Tab

Device Summary

The Device Summary section of the Space tab displays a bar chart for all devices. The Device Summary section succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

Statistic	Description
Used	Amount of MBs currently allocated on the particular disk/device.
Free	Amount of MBs currently unallocated and free on the particular disk/device.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Space Tab

Device Detail

The Device Detail section of the Space tab succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

Statistic	Description
Partition	The drive letter of the disk.
Local Filesystem	The name of the file system.
Type	The type of file system.
Total Space	Total size of the disk/device's capacity expressed in MBs.
Used Space	Amount of MBs currently allocated on the particular disk/device.
Free Space	Amount of MBs currently unallocated and free on the particular disk/device.
Capacity	The percentage of space used on the device.
Mounted On	The mount point of the device.

Note: The statistics available in this section depend on the platform of operating system.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Space Tab

Network Tab

The Network tab of the OS Detail page succinctly communicates the general overall performance levels of the server's networking. The Network tab of the OS Detail page includes the following sections:

Network Details	Packet Collisions	Packet Discards
Packet Errors	Transmission Rate	Transmission Rate (Bytes)
Transmission Queue Length		

Note: The sections available on this tab depend on the platform of operating system.

Network Details

The statistics available in this section depend on the platform of operating system. The table below describes the information available in this section:

Column	Description
Network Interface	The name of network interface.
INET Address/Address	The IP address assigned to the network interface.
Pkts Sent/Sec	The number of packets sent per second.
Pkts Received/Sec	The number of packets received per second.
Sent (KB/Sec)	The number of bytes sent per second.
Received (KB/Sec)	The number of bytes received per second.
Out Pkts Discarded	The number of outbound packets discarded.
In Pkts Discarded	The number of inbound packets discarded.
Out Pkt Errors	The number of outbound packet errors.
In Pkt Errors	The number of inbound packet errors.
Queue Length	The queue length.
Collisions	The number of collisions.
Packets Discarded	The number of packets discarded.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Network Tab

Transmission Rate

The Transmission Rate section of the Network tab succinctly communicates the packet transmission rate. The following statistics are available in this section:

Packets Received/Sec Packets Sent/Sec

Packets Sent/Sec

The Packets Sent/Sec statistic is the number of packets sent over each network adapter per second.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Network Tab

Packets Received/Sec

The Packets Received/Sec statistic is the number of packets received over each network adapter per second.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Network Tab

Transmission Rate (Bytes)

The Transmission Rate (Bytes) section of the Network tab succinctly communicates the packet transmission rate. The following statistics are available in this section:

Received (KB/Sec) Sent (KB/Sec)

Sent (KB/Sec)

The Sent (KB/Sec) statistic is the rate at which bytes are sent over each network adapter including framing characters.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Network Tab

Received (KB/Sec)

The Received (KB/Sec) statistic is the rate at which bytes are received over each network adapter, including framing characters.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Network Tab

Transmission Queue Length

The number of pending outgoing packets on either the software transmit queue or the hardware transmit queue

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Network Tab

Packet Collisions

A collision occurs when two devices on the network sense that the network is idle and end up trying to send data at the same time. Some collisions are normal on a repeated network, but excess collisions can cause serious performance problems.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Network Tab

Correction Contact your network administrator if there is high rate of persistent Packet Collisions.

Packet Discards

The Packet Discards section of the Network tab succinctly communicates the packet discard rate. The following statistics are available in this section:

Inbound Packets Discarded Outbound Packets Discarded Packet Discards

Outbound Packets Discarded

The Outbound Packets Discarded statistic is the number of outbound packets that were discarded even though no errors had been detected to prevent from being transmitted.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Network Tab

Correction A possible reason for discarding such a packet could be to free up buffer space.

Inbound Packets Discarded

Received Packets Discarded is the number of received packets that were discarded even though no errors had been detected to prevent from being transmitted.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Network Tab

Correction A possible reason for discarding such a packet could be to free up buffer space.

Packet Discards

The Packet Discards statistic represents the number of network packets discarded.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Network Tab

Correction Contact your network administrator if there is high rate of persistent packet discards.

Packet Errors

The Packet Errors section of the Network tab succinctly communicates the packet error rate. The following statistics are available in this section:

Inbound Packet Errors Outbound Packet Errors

Outbound Packet Errors

The Outbound Packet Errors statistic is the outbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Network Tab

Inbound Packet Errors

The Packets Received Errors statistic is the received packets that contained errors preventing them from being deliverable to a higher-layer protocol.

Location Sybase ASE Performance Analyst Statistics (DBArtisan) > **OS (DBArtisan - Sybase ASE Performance Analyst)** > Network Tab

Top SQL Statistics (DBArtisan - Sybase ASE Performance Analyst)

A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements or stored procedures can drag an otherwise well-configured database down in terms of user response times.

Before you can identify problem SQL in your database, you have to ask the question of what 'bad SQL' is. What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every SQL Server professional's satisfaction. The Top SQL results grid displays the top SQL statements across various performance categories.

The SQL Details tab of the Top SQL view presents information as follows:

Active SQL Results Selected SQL Text

Metrics

When you begin to look for inefficient SQL in a database, there are two primary questions you need to answer:

- What has been the worst SQL that’s historically been run in my database?
- What is the worst SQL that’s running right now in my database?

When Correction a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It may be that the query is in an inefficient loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

Active SQL Results

The Active SQL results section displays the following information in grid format:

SQL Text: A snippet of the full SQL statement. Clicking on the statement will cause the full statement to be presented in the Selected SQL Text grid.

Database: The name of the database.

SPID: The process ID.

CpuTime: The accumulated CPU time for the statement.

Wait Time: The time the process has been waiting to be serviced, in milliseconds. A value of zero indicates there was no waiting.

Memory Usage (KB): The number of pages in the procedure cache that is currently allocated to the process. A negative number indicates that pages are being released (freed) from the process.

Physical Reads: The current cumulative number of physical disk reads issued by the process.

Logical Reads: The current cumulative number logical reads issued by the process.

Pages Modified: The number of pages that were modified by the process.

Packets Received

Packets Sent

Start Time: The date/time when the statement was first executed.

Batch ID

Selected SQL Text

In this section of SQL details, you see the fully expanded SQL statement that you highlighted in the Active SQL Results grid. By clicking Explain SQL you open the Explain Plan with its ISQL window.

Performance Analyst for IBM DB2 for Linux, Unix and Windows Tutorial (DBArtisan)

Embarcadero Performance Analyst provides key insight into the complex world of database performance. Running inside the Embarcadero DBArtisan console, or standalone in a Web browser, Performance Analyst helps you pinpoint performance problems that threaten a database's availability or response times. The powerful analysis facilities will satisfy advanced experts, yet novices will find Performance Analyst's interface easy to use and the expert help a welcome aid to troubleshooting difficult performance issues.

About This Performance Analyst for IBM DB2 Tutorial

This tutorial is intended to help you get started with Embarcadero Performance Analyst.

After completion of this tutorial, you will have the foundation you need to explore the many features and benefits of Performance Analyst. You will have learned how to accurately monitor your database using the most popular performance methodologies in use today. You will know how to drill down into the many layers of database performance and also be able to locate performance issues in your database in a fast and effective manner. Finally, you'll understand how to fix performance problems using the integration that exists between Performance Analyst and Embarcadero DBArtisan.

This tutorial is divided into 8 sections.

Session 1: Getting Started With Performance Analyst for DB2 (DBArtisan)

Session 2: Using Ratio-based Performance (DBArtisan)

Session 3: Using Bottleneck/wait-based Performance Analysis in Performance Analyst for DB2 (DBArtisan)

Session 4: Using Workload Performance Analysis (DBArtisan)

Session 5: Using SQL Performance Analysis (DBArtisan)

Session 6: Using Operating System Performance Analysis (DBArtisan)

Session 7: Identifying and Getting Help With Performance Problems (DBArtisan)

Session 8: Fixing Performance Problems Using Performance Analyst for DB2 (DBArtisan)

You can use this basic tutorial as a roadmap of product highlights.

After starting this tutorial, you can select Help from the menu to find many additional resources that complement and build on many of the activities shown in this brief tutorial.

For more information, see [DBATutorial.htm#AdditionalEvaluationResources Additional Evaluation Resources].

Topics

- Session 1: Getting Started With Performance Analyst for DB2 (DBArtisan)

- Session 2: Using Ratio-based Performance (DBArtisan)
- Session 3: Using Bottleneck/wait-based Performance Analysis in Performance Analyst for DB2 (DBArtisan)
- Session 4: Using Workload Performance Analysis (DBArtisan)
- Session 5: Using SQL Performance Analysis (DBArtisan)
- Understanding SQL Activity (DBArtisan)
- Session 6: Using Operating System Performance Analysis (DBArtisan)
- Session 7: Identifying and Getting Help With Performance Problems (DBArtisan)
- Session 8: Fixing Performance Problems Using Performance Analyst for DB2 (DBArtisan)

Session 1: Getting Started With Performance Analyst for DB2 (DBArtisan)

Download and Install

You can obtain the latest version of Performance Analyst by downloading the DBArtisan software from the Embarcadero Web site at http://www.embarcadero.com/downloads/downloaddbartisan.asp. Performance Analyst runs fully contained within DBArtisan so you will need the DBArtisan product to use Performance Analyst (although Performance Analyst can also run in standalone fashion within a Microsoft Internet Explorer Web browser).

Click Download, and follow the steps indicated. Save the file on your computer and then double-click to launch the self-extracting file that will guide you through the installation process.

When you first install an evaluation copy of DBArtisan, you can use the tool for 14 days. After that time, a permanent license is needed.

Proceed to Overview for Performance Analyst for DB2

Overview for Performance Analyst for DB2

When DBArtisan is connected to an DB2 database, you can open Performance Analyst two ways:

1 On the toolbar, click the Performance Analyst icon.

2 On Utilities menu, select Performance Analyst.

DBArtisan opens Performance Analyst.

DB2 requires that certain monitor switches be turned on for collecting various performance metrics. When Performance Analyst is first invoked, you will be prompted to turn on the monitor switches. Also, you can increase or decrease the size of the monitor heap depending on the activity of your database.

If you do not want to see this dialog each time you start Performance Analyst, you can check the "Save these settings as default" check box.

Proceed to Product Navigation

Product Navigation

Navigating through Performance Analyst is easily accomplished through two main mechanisms:

1 The main Performance Analyst toolbar contains buttons and drop-down menus that invoke the performance analysis main views and drill-downs within the product as well as frequently used actions like Refresh and Performance Threshold Customizations.

2 Positioning the mouse pointer on any area within Performance Analyst and right-clicking brings up a navigation menu that allows you to easily move between any of the main Performance Analyst

home pages and drill-down views. The right mouse menu also contains context-sensitive options like drilling down further into a particular statistic, copying the display to a text file, and more.

Proceed to Session 2: Using Ratio-based Performance (DBArtisan)

Session 2: Using Ratio-based Performance (DBArtisan)

Ratio-based analysis has been around for many years, and used to be the only technique database administrators used when they were called on to diagnose the cause of a database slowdown. Performance ratios are very good roll-up mechanisms for busy DBAs, making possible the analysis-at-a-glance approach, however ratios must be computed correctly for them to be meaningful. Key ratios in Performance Analyst are calculated using delta statistics, which is superior to computing ratios with cumulative values. Delta statistics, which are values derived by using the before and after measures in each statistical refresh done in Performance Analyst, lets you see changes in performance metrics that would otherwise be invisible if cumulative statistics were used. Performance Analyst makes it easy to see the ratios that are computed with delta statistics vs. cumulative. A triangle symbol denotes a delta-based statistic, while square symbols represent a cumulative-based measure.

Ratio-based analysis is available for all performance categories (memory, etc.) where ratios can assist in determining the performance level of a database. For example, if you would like to see memory-specific ratios, on the Performance Analyst toolbar click Memory. Performance Analyst displays the Memory home page that showcases all important memory-related ratios.

Proceed to Obtaining Detail Information for a Ratio Using Performance Analyst for DB2

Obtaining Detail Information for a Ratio Using Performance Analyst for DB2

A ratio gives you a global picture for a particular performance measure. However, you often need more information regarding a statistical measure so you can make an informed decision. Performance Analyst has easy-to-use drill down features that make obtaining detailed information for various ratios a breeze.

For example, to view more information for the Buffer Cache Hit Ratio statistic, do the following:

1 On either the main Home page or the I/O home page, locate the Buffer Pool Hit Ratio statistic.

2 Position the pointer over the statistic.

The pointer changes to a hand symbol.

3 Double-click the left mouse button.

Performance Analyst opens the **Buffer Cache** tab of the memory drill down view.

4 Or right-click the statistic, select Details or Navigate to… > I/O > Buffer Pool.

Once either of these actions has been taken, you will be taken to a drill-down view that provides more information on the DB2 buffer pools.

Proceed to Session 3: Using Bottleneck/wait-based Performance Analysis in Performance Analyst for DB2 (DBArtisan)

Session 3: Using Bottleneck/wait-based Performance Analysis in Performance Analyst for DB2 (DBArtisan)

When a DB2 database is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things, or it can be an indicator that a database bottleneck exists. Bottleneck analysis is a valid method of measuring performance because it helps you track where a database and user sessions have been spending their time. If lock contentions or heavy table-scan activity has been dragging a database's performance down, you can use bottleneck analysis to confirm the actual root cause.

Performance Analyst uses several key areas of DB2 metrics to form its bottleneck analysis. These areas are usually the main sources of performance drag on the databases.

Performance Analyst prominently displays bottleneck analysis on every performance home page. On the main home page, you will see a summary of all key bottlenecks threatening the database. You can also see specific bottlenecks for memory, I/O, space, objects, and users by navigating to each home page and viewing the Bottleneck Analysis section for that home page.

For example, to see bottlenecks that specifically relate to user applications, follow these steps:

1 Click the Users Performance Analyst toolbar button

2 Locate the Bottleneck Analysis section at the lower-left portion of the screen

The Users Bottleneck Analysis section shows the top bottlenecked applications, concurrency issues, and bottleneck activity count.

Proceed to Obtaining Detail on System Bottlenecks

Obtaining Detail on System Bottlenecks

You can easily obtain more information regarding global bottlenecks that are plaguing a database by drilling down into a particular bottleneck metric. For example, if you would like to see more data regarding the current top concurrency bottlenecks, do the following:

1 Open the Performance Analyst main Home page.

2 Double-click the Concurrency area in the Bottlenecks section to be taken to the Lock Waits drill-down section.

Performance Analyst opens the System Waits view.

3 Or right-click anywhere in Performance Analyst, point to Users, and then click Lock Waits.

Performance Analyst opens the System Waits view.

The Lock Waits view displays critical details regarding all the lock wait events that DB2 has recorded. You can easily see what applications are waiting on locks and the average lock wait times for all applications.

Note: If you would like to see an expanded view of the Lock Waits grid shown on the top of the Lock Waits view, click the Expand button located at the top right corner of the section. All section views in Performance Analyst can be maximized/minimized using this control.

Proceed to Viewing Detail for Application Bottlenecks

Viewing Detail for Application Bottlenecks

Not only can you obtain detail information for system bottlenecks, but you can also use Performance Analyst to get drill-down information regarding the locks held by and waited on by applications.

To see more information regarding application locks, do the following:

1 Open the Performance Analyst main Home page.

2 Double-click the Concurrency Bottlenecks area to be taken to the Lock Waits drill-down view.

Performance Analyst opens the Lock Waits tab.

3 Or right-click anywhere in Performance Analyst, point to Users, and then click Locks.

Once on the Locks drill-down view, you can view the details for locks held and waited on by an application details by following these steps:

1 Examine the Application Details grid

2 Select an application that appears to be experiencing lock waits problem

You will see details on the locks held by, and the locks waited on, by the selected application. You can get further details on the application that is holding the locks by navigating to the Attributes drill-down and selecting the appropriate Application ID.

Proceed to Session 4: Using Workload Performance Analysis (DBArtisan)

Session 4: Using Workload Performance Analysis (DBArtisan)

Key ratios give you a global perspective on database activity. Bottleneck analysis gives you insight into things that are holding up user activity and throughput. But another technique is necessary if a database professional is to really get a handle on what's occurring inside a badly performing database.

Workload analysis involves the investigation of two critical areas of a database's performance:

- Application resource consumption and activity
- SQL execution

Without looking at these two key performance categories, you can miss a lot of what could be responsible for perceived performance problems. When performance on a database takes a sudden nosedive, it is not uncommon to find one or two sessions that are causing the bulk of the workload. Finding these processes can be easily accomplished by viewing session metadata coupled with resource consumption and statistical execution statistics.

Proceed to Pinpointing Top Resource Consumers

Pinpointing Top Resource Consumers

Performance Analyst identifies top resource consumers in a couple of different ways. On the main home page, Performance Analyst highlights the top resource consumers across lock usage, CPU usage, and I/O usage. The percentage used across all statistical categories are displayed so a DBA can immediately pinpoint a session that is using all or most of a particular resource. The top resource sessions are also displayed on the memory, I/O, and other home pages as well with a listing for the top sessions for that category (for example, the top IO consumers appear on the I/O home page, etc.).

Proceed to Getting Workload Details Using Performance Analyst for DB2

Getting Workload Details Using Performance Analyst for DB2

Performance Analyst offers several details on top resource applications in addition to the summaries provided on each home page. A detailed and flexible top applications view lets you view a summary of all the major resource applications on the database as well as every critical detail regarding each application connected to the database.

Further drill-downs are available that highlights specific details regarding each application's activities and submitted SQL. To see detailed information regarding an application identified in one of Performance Analyst's Workload Analysis sections, do the following:

1 In a Workload Analysis display, position the pointer over one of the processes.

2 Double-click the process or right-click, and then select Details.

If you want like to see all key workload analysis metadata for sessions, do the following:

1 On the toolbar, click the Top Sessions button.

2 Or right-click anywhere in Performance Analyst, point to Users, and then click Top Sessions.

Proceed to Session 5: Using SQL Performance Analysis (DBArtisan)

Session 5: Using SQL Performance Analysis (DBArtisan)

SQL-based performance analysis is really a subset of Workload Analysis. Understanding current and historical SQL execution patterns lets you to have the second set of data points necessary to properly perform workload analysis. You may find that optimizing SQL code produces some of the best performance-enhancing boosts available for a database.

Proceed to Understanding SQL Activity (DBArtisan)

Understanding SQL Activity (DBArtisan)

Performance Analyst offers plenty of insight into SQL performance and execution patterns. Each one of the home pages depicts the current levels of SQL activity as determined by a variety of different metrics for that performance category. For example, Performance Analyst might show that rows reads account for 75% of all SQL statement I/O on the database.

Proceed to Getting Details on Top SQL

Getting Details on Top SQL

Getting details on identified SQL statements is easy in Performance Analyst. For example, if you would like to see further information on a SQL statement identified on the Performance Analyst Home page, do the following:

1 In the SQL Analysis section, position the pointer over Hash Values.

2 Double-click or right-click, and then select Details.

Performance Analyst opens the Top SQL summary page that shows a breakdown of all top SQL statements along with their execution statistics

If you would like to see the full SQL text of a SQL statement, do the following

1 In the Top SQL Details section, double-click a statement.

Performance Analyst opens the **SQL Details** tab of the Top SQL view to display the entire SQL statement and the execution metrics. While on this pane, you can also perform an EXPLAIN of the SQL statement and examine its access path information.

Note: To automatically tune a problem SQL statement if you have Embarcadero SQL Tuner installed on your workstation, do the following:

2 On the **SQL Details** tab, select a problem statement, and then click the Tune SQL button.

SQL Tuner opens and you can begin a tuning session.

Proceed to Session 6: Using Operating System Performance Analysis (DBArtisan)

Session 6: Using Operating System Performance Analysis (DBArtisan)

In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database sever for resources (CPU, I/O or Memory), a slow CPU, insufficient I/O devices, and insufficient memory. Performance Analyst allows you to examine the relevant operating system metrics for DB2.

To view the operating system metrics of the machine hosting the DB2 server you are monitoring, do the following:

1 Click the OS Performance Analyst toolbar button.

OR

2 Navigate to the OS Home Page by going to OS, and then Home.

Note: The first time you invoke the OS feature you will be prompted for an operating system login. Please specify the correct operating system, host name, username and password, and protocol.

The Operating System home page has been designed in a similar fashion as the datasource home page. On this page you can see the pertinent operating system metrics displayed in five quadrants broken down into the following categories:

- Resource Usage
- Memory Analysis
- Disk Analysis
- Workload Analysis
- Bottleneck Analysis

Proceed to Obtaining Details on the Operating System

Obtaining Details on the Operating System

You can easily obtain more information regarding operating system problems that are degrading database performance by drilling down into a particular analysis quadrant. For example if you'd like to see more information regarding the processes running on the host machine, do the following:

1 Navigate to the Performance Analyst Operating System home page.

2 Double-click the Top Memory Process in the Workload Analysis quadrant.

Performance Analyst opens the Processes detail view.

OR

3 Right-click anywhere in Performance Analyst and select the Navigate to, OS, and then Processes.

The Processes detail tab shows all the processes on the host machine. From this view you can easily determine which processes are showing high marks in I/O or CPU, for example, by simply double clicking on the column headers and sorting the processes in ascending or descending order.

Proceed to Session 7: Identifying and Getting Help With Performance Problems (DBArtisan)

Session 7: Identifying and Getting Help With Performance Problems (DBArtisan)

You do not need to be an advanced expert in performance tuning to spot trouble in a database if you use Performance Analyst. There are a number of mechanisms in the tool that help you quickly locate areas of concern within a database.

Proceed to Using the Alarm Log

Using the Alarm Log

Performance Analyst has a central and flexible alarm log that lists every performance violation found in all monitored databases. Unlike some products that only show you alarms for a particular database, Performance Analyst lists all alarms for each database you are monitoring in one central log so you can see every problem in one place. To view the Performance Analyst Alarm log, do the following:

1 On the toolbar, click the Alarms button.

Once in the Alarm log, you can perform a number of actions including:

- Selecting an alarm and viewing the application page that shows details regarding the metric in violation.
- Filtering the types of alarms shown through the Options dialog.
- Clearing all current alarms.
- Copying or saving the Alarm log into a different format.

To select any of these options, do the following:

1 Position the pointer over any area of the Alarm Log.

2 In the shortcut menu, select an option.

Proceed to Recognizing Performance Violations

Recognizing Performance Violations

Performance Analyst makes it easy to visually pinpoint performance violations in a database. Any statistic that does not conform to your predefined performance standards is colorized in any view that it appears so you can easily see potential troublespots.

The colors used to highlight a particular performance violation depend on the severity of the problem. Performance Analyst uses several layers of performance standards/threshold violations with each layer being represented by a different color. You can control everything regarding threshold violations, including how thresholds are determined, their color representation in Performance Analyst’s display, and more by using the Threshold Template feature of Performance Analyst.

How can Performance Analyst inform you of performance violations in your database if you minimize the product to your desktop and are working on other things? Even if Performance Analyst isn't right in front of you, it can still let you know that a problem has been found in your database through the systray alarm icon. When Performance Analyst detects a problem in one of your databases, one of the things the product does is enable an alarm icon in your workstation's systray area that communicates the presence of alarms in your database. So even if the Performance Analyst interface isn't in sight, the product can let you know that something in your database requires attention.

Proceed to Creating Threshold Templates

Creating Threshold Templates

How does Performance Analyst know what to signal as a problem in a database? It uses predefined performance standards that you define and enforce. All of this is accomplished by using Performance Analyst's Threshold templates. A template is a set of performance metrics and their thresholds that can be reused to enforce performance standards across any number of databases. Embarcadero provides a default performance template that contains nearly all of the most-watched performance statistics so you can begin monitoring with confidence right out of the box. However, you can create your own Threshold templates so you only are alerted to performance violations that are important to you and your particular environment.

To view or create Threshold templates, do the following:

1 On the toolbar, click the Thresholds button.

Performance Analyst opens the **Threshold maintenance** dialog.

The **Threshold Template Properties** dialog lets you select the options to add or edit templates, as well as other maintenance options. For example, you can copy the original Embarcadero Threshold template and modify it to fit your own needs. You can control what statistics are sampled for performance violations as well as the levels of severity you would like reported. Once you are satisfied with your template, you can save it and then apply it to one or more datasources in one action to immediately begin enforcing your performance standards.

Proceed to Getting Help With Performance Problems

Getting Help With Performance Problems

As databases become more complex, so do the problems that plague them. Knowing what to do for a particular performance problem can be hard in some cases, but Performance Analyst helps by providing a complete set of expert help that you can use as a guide when performance problems occur. To obtain expert help for any statistic shown in Performance Analyst, do the following:

1 Position the pointer over a statistic that you need help with.

2 Right-click and select Help.

The Expert Help gives you a general description of the statistic, the range of values that are considered optimal, and some suggestions on how to fix a problem situation. Expert help can also be invoked for any of the Ratio, Bottleneck, SQL, or Workload Analysis sections shown on every home page by clicking on the help icon for that section, which is located at the top right corner of the section.

Proceed to Session 8: Fixing Performance Problems Using Performance Analyst for DB2 (DBArtisan)

Session 8: Fixing Performance Problems Using Performance Analyst for DB2 (DBArtisan)

Finding performance problems is one thing, but fixing them is another. Many performance monitors point out a particular problem in a database, but then offer no way to actually correct the situation. Such is not the case with Performance Analyst. Because of its tight integration with DBArtisan, certain problems found in Performance Analyst can be visually corrected by DBArtisan.

If a DB2 tablespace is about to run out of free space, Performance Analyst can issue an alarm that a space shortage exists. Once the space problem has been identified, do the following:

1 On the Space home page toolbar, click the Space button.

2 In the Space Analyst grid, right-click a tablespace, and then select Manage with DBArtisan.

You will then be taken to DBArtisan’s Tablespace Editor where you can perform corrective actions, such as adding new containers to the tablespace, or resizing the tablespace’s existing container(s)

IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan)

This section includes expert help for all DB2 categories and statistics in the Embarcadero Performance Analyst views and pages.This help is divided into the following sections:

Home View (DBArtisan - DB2 LUW Performance Analyst)
Memory View (DBArtisan - DB2 LUW Performance Analyst)
I/O
Space View (DBArtisan - DB2 LUW Performance Analyst)
Users View (DBArtisan - DB2 LUW Performance Analyst)
Cfg (DBArtisan - Database Configuration)
Instance View (DBArtisan - DB2 LUW Performance Analyst)
OS View (DBArtisan - DB2 LUW Performance Analyst)
Top SQL View (DBArtisan - DB2 LUW Performance Analyst)

Topics

- Home View (DBArtisan - DB2 LUW Performance Analyst)
- Memory View (DBArtisan - DB2 LUW Performance Analyst)
- Space View (DBArtisan - DB2 LUW Performance Analyst)
- Users View (DBArtisan - DB2 LUW Performance Analyst)
- Cfg (DBArtisan - Database Configuration)
- History View (DBArtisan - DB2 LUW Performance Analyst)
- Instance View (DBArtisan - DB2 LUW Performance Analyst)
- OS View (DBArtisan - DB2 LUW Performance Analyst)
- Top SQL View (DBArtisan - DB2 LUW Performance Analyst)

Home View (DBArtisan - DB2 LUW Performance Analyst)

The Embarcadero Performance Analyst Home view lets you review availability and overall performance of all monitored databases from a single window. Statistics on the Home view are organized into the following categories:

Key Ratio Analysis Pane	Bottleneck Analysis Pane	SQL Analysis Pane
Storage Analysis Pane	Workload Analysis Pane	

Key Ratio Analysis Pane

Database performance analysts typically use ratio-based or wait/bottleneck-based methods for examining the performance levels. Ratio-based analysis involves examining a number of key database ratios that indicate how well a database is running. Performance ratios serve as very good roll-up mechanisms for at-a-glance performance analysis. Many DBAs have large database farms to contend with and cannot spend time checking detailed wait-based analysis outputs for each and every database they oversee. Succinctly presented performance ratios can assist in such situations by giving solid indicators that can be quickly scanned to see if any database needs immediate attention.

While there are certainly many opinions as to what rules to follow, there are some standards that should always be adhered to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that a DBA will examine come from the DB2 Snapshot Monitor. The snapshot monitor returns metrics for the Instance and particular databases since the start of monitoring. Some metrics are cumulative (e.g., counters) and others are instantaneous (e.g., gauges).

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each SQL Server's individual 'personality' with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead the DBA to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following ratios are used on the Performance Analyst Home page to succinctly communicate the general overall performance levels of the monitored database:

Efficiency	Buffer Pool Hit Ratio	Catalog Cache Hit Ratio	Package Cache Hit Ratio
	Shared Workspace Hit Ratio		
Resource Utilization	Database Heap Utilization	Log Space Utilization	Lock List Utilization
	Shared Sort Memory Utilization		

Buffer Pool Hit Ratio

The DB2 database server reads and updates all data from a bufferpool because memory access is much faster than disk access. Data is copied from disk to a bufferpool as needed by the applications using the database. When the server needs to read/write data and the data is already in the bufferpool, no disk access is required. However, if the data is not in the bufferpool, it needs to be read from the disk, which is significantly slower process.

The buffer pool hit ratio indicates the percentage of time that the database server did not need to load a page to service the request for a page. The calculation takes into account all the index and data pages that were requested.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Key Ratio Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Memory View (DBArtisan - DB2 LUW Performance Analyst) > Home > Key Ratio Analysis Pane
Metrics	Avoiding disk I/O is the main issue when you try to improve the performance tuning. A high buffer pool hit ratio is desirable because it indicates a lower frequency of synchronous disk I/O. A database where data is accessed uniformly from very large tables will have a poor hit ratio. There is little you can do to improve the performance in such cases.
Correction	The buffer pool hit ratio on the database Home page is the overall hit ratio of the database. First, drill down to the bufferpool level and check the individual buffer pool hit ratios to identify the elements that may require tuning. Increasing the bufferpool size generally improves the buffer pool hit ratio. You can use one of the following methods to increase the bufferpool size depending on the layout of the data: *If the tablespaces using the bufferpool have tables and indexes on them, increase the bufferpool size incrementally until the index page hit ratio stops increasing for that bufferpool. You will not usually get any benefit increasing the bufferpool size after you reach that point.If the tablespaces associated with the bufferpool have only indexes on them, increase the bufferpool size until the index page hit ratio stops increasing.If the tablespaces associated with the bufferpool have only data on them, increase the bufferpool size until the data page hit ratio stops increasing.* One general design guideline is to try and use different tablespaces for indexes and tables and associate different bufferpools with them. This generally improves performance and makes bufferpool tuning much easier.

Catalog Cache Hit Ratio

Catalog cache is used to cache the following types of catalog information:

- Metadata for tables, views, and aliases.
- Database authorization information that is used to check authorization when performing operations like CONNECT, LOAD, CREATE, BIND, and so on.
- Execute privileges for user-defined functions and stored procedures.

When a database operation accesses catalog information, it inserts this information into the catalog cache so operations accessing the same information can read it from the cache and avoid disk reads. The catalog cache hit ratio indicates how well the catalog cache avoids accesses to the catalog.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Key Ratio Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Memory View (DBArtisan - DB2 LUW Performance Analyst) > Home > Key Ratio Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Memory View (DBArtisan - DB2 LUW Performance Analyst) >Caches Tab
Metrics	A high catalog cache hit ratio (> 80%) indicates that the catalog cache is working well. A smaller ratio can indicate that this parameter needs tuning. You may see a smaller ratio immediately following the first connection to the database and execution of DDL/DCL statements since these require heavy catalog accesses.
Correction	If the catalog cache hit ratio is consistently small, increase the database configuration parameter catalogcache_sz. When you increase this value, pause to consider whether it would be more effective to allocate the extra memory you are reserving to another purpose such as package cache or bufferpools. In a partitioned database environment, make the catalogcache_sz larger on the catalog partition than on other partitions because the catalog information required for all partitions will go through this partition. When tuning this parameter, it is advisable to monitor changes to the database catalog via DDL statements. During the execution of DDL statements, there may be a large drop in the catalog cache hit ratio due to invalidation of the cached data.

Package Cache Hit Ratio

Embarcadero DBArtisan's Database Manager caches the sections for frequently used dynamic and static SQL statements in the package cache. The package cache hit ratio indicates how well the package cache is avoiding catalog accesses to packages and recompilations.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Key Ratio Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Memory View (DBArtisan - DB2 LUW Performance Analyst) > Home > Key Ratio Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Memory View (DBArtisan - DB2 LUW Performance Analyst) > Caches Tab
Metrics	A high package cache hit ratio (> 80%) indicates that the package cache is working well. In the case of static SQL statements, package caching allows the Database Manager to reduce the internal overhead by eliminating the need to access system catalogs when reloading a package. For dynamic SQL, the benefit of package caching is even greater since it a cache hit eliminates the need for recompilation. The package cache hit ratio metric is particularly important for transaction-processing applications since a typical workload involves repeatedly executing the same SQL statements.
Correction	Executing DDL statements can invalidate sections of SQL statements in the cache, causing the hit ratio to decrease dramatically. Before attempting to tune this parameter, you should check the DDL activity to see if that is causing a skew. No amount of tuning will improve the package cache performance if the DDL activity is causing package invalidations in the cache. If the DDL activity is minimal and package cache hit ratio is consistently small, consider increasing the package cache size (pckcachesz) in the database configuration. When increasing the value of this parameter, consider whether it might be more effective to allocate the extra memory being reserved to another purpose such as catalog cache or bufferpools.

Shared Workspace Hit Ratio

When sections are required by an application for executing dynamic or static SQL statements, they are placed in the shared workspace. The shared workspace exists at the application level and is shared among applications using the database. The hit ratio is a percentage indicating how well the shared SQL workspace is helping to avoid initialization of sections for SQL statements that are about to be executed. A high ratio indicates the shared workspace is successful in avoiding this action.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Key Ratio Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Memory View (DBArtisan - DB2 LUW Performance Analyst) > Home > Key Ratio Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Memory View (DBArtisan - DB2 LUW Performance Analyst) >Caches Tab
Metrics	A shared workspace is shared by many applications. If applications have similar database access patterns, they can benefit greatly if they find required sections in the shared workspace. If an application finds a section in the shared workspace (e.g., a hit), that application avoids setup and initialization cost for that section. A high hit ratio is desirable for this metric. Typically, you should expect to see a high ratio (>80%) in transaction processing environments.
Correction	Shared workspace memory is allocated from the application control heap (app_ctl_heap_sz database configuration parameter) and increasing this may improve the hit ratio.

Database Heap Utilization

There is one database heap per database, and the database manager uses it on behalf of all applications connected to the database. The database heap utilization is the percentage of database heap that is currently being used.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Key Ratio Analysis Pane
Metrics	The database heap contains control block information for tables, indexes, table spaces, and bufferpools. It also contains space for event monitor buffers, the log buffer, and temporary memory used by utilities. The heap utilization can increase or decrease whenever any of the constituent elements change. If the utilization goes above 85% several times or stays above 85% for a sustained period, you may need to increase the maximum database heap size.
Correction	The dbheap database configuration parameter determines the maximum amount of heap memory that can be allocated. Examine the individual memory pools and how they are using the memory before increasing this parameter. An excessively large use of one of the memory pools may be the cause of this problem.

Note: On the main toolbar, click Memory and then Memory Pools to see the Memory Pools Details.

Log Space Utilization

Log space utilization is the percentage of total available log space on the database that is currently being used.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Key Ratio Analysis Pane

Metrics Very high log space utilization (>75%) can indicate that the database might run out of log space soon. Usually, consistently high log space utilization has one of the following causes: *Large transaction size. Applications are performing large numbers of updates on the database but are not committing the transactions frequently.The log space configured for the database is insufficient for the level of database activity* Very low log space utilization (< 5) %may indicate an excessively large log file and that you may be able to free some of the space occupied by log files on disk.

Correction If the log utilization is mostly low but increases to near 100% for short periods, then you can increase the number of secondary log files (logsecond database configuration parameter). This allocates and deallocates additional log files as needed. If however, the log size is consistently high and changes frequently, you can increase the primary log size and/or the log file size (logfilsiz and logprimary database configuration parameters). Also, check the SQL activity on the SQL Activity tab of the Users Detail section to see if any applications are performing infrequent commits. If this is the case, evaluate whether such applications can be changed to have shorter transactions and hence less need for log space.

Lock List Utilization

Lock list utilization is the percentage of total database memory allocated for locks that is currently being used.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Key Ratio Analysis PanelIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Memory View (DBArtisan - DB2 LUW Performance Analyst) > Lock Memory Tab
Metrics	There is only one lock list for each database and it contains the locks held by all applications connected to the database. Once the lock list is full, the database manager starts escalating row locks to table locks to free up space. This escalation may result in serious performance degradation because of reduced concurrency. Additionally, the number of deadlocks and transaction rollbacks may go up. If this metric reaches the 75% mark, you should consider bringing this percentage down with tuning.
Correction	Depending on the database's activity level, you may be able to reduce the lock utilization by following these recommendations: Increase size of lock list: If there is not enough lock list space available, lock escalations will occur, increasing contention and reducing concurrency. Update the locklist database configuration parameter to increase this value. Tune applications to reduce locking: On the Locks tab of the Users detail section, identify the applications that are holding many locks and then consider the following steps for controlling the size of the lock list: *Make the transactions shorter by increasing the COMMIT frequency. This ensures that locks are released frequently, thus freeing up lock space.Before you update many rows in a single table, lock the entire table (using the SQL LOCK TABLE statement). This prevents many row-level locks from being obtained (although this decreases concurrency)To control how locking is done for a specific table, use the LOCKSIZE parameter of the ALTER TABLE.To decrease the number of share locks held, use the Cursor Stability isolation level when possible. If the applications' integrity requirements are not compromised, use Uncommitted Read instead of Cursor Stability to further decrease the amount of locking.* Decrease percentage of lock list: If a small number of applications are consuming most of the lock space, decrease the percentage of lock list for each application. You can throttle back those applications by decreasing the maxlocks database configuration parameter. This reduces the amount of lock list memory available to each application thereby allowing for better distribution of lock list memory. NOTE: Decreasing the percentage of lock list should be a last resort, used only if you cannot decrease utilization with the other recommendations. It can cause a large number of lock escalations.

Shared Sort Memory Utilization

The amount of memory allocated for each sort may be some or all of the available sort heap size. Sort heap size is the amount of memory available for each sort as defined in the database configuration parameter sortheap. Shared sort memory utilization gives the percentage of the sort heap being used. It is possible for a single application to have concurrent sorts active. For example, in some cases a SELECT statement with a subquery can cause concurrent sorts.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Key Ratio Analysis Pane
Metrics	Memory estimates do not usually include sort heap space. If excessive sorting is occurring, the extra memory used for the sort heap should be added to the base memory requirements for running the database manager. Generally, the larger the sort heap, the more efficient the sort. Typically the shared sort utilization should be less than or equal to 70%. You should consider tuning the database if you see a utilization value greater than this.
Correction	To bring the sort heap utilization to an acceptable level, use the following guidelines: *Examine the queries you are running on the database to see if you can add indexes to columns frequently accessed in the WHERE clause. This minimizes the use of the sort heap.If you are using dynamic bitmaps or hash join buffers in your queries, or your queries frequently require large sorts, increase the sortheap size.If you adjust the sortheap size, also look at the sheapthres database manager configuration parameter to see if it too needs to be adjustedIf you are adjusting the sortheap size, you may also benefit from rebinding your packages since the optimizer takes this parameter into account when determining the access paths.*

Bottleneck Analysis Pane

When a DB2 database is active, applications accessing the database can either successfully access the database resources, or wait for resources to become available. Waiting applications can indicate normal database activity, or can indicate that a database bottleneck exists. A database administrator can use wait-based or bottleneck performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Performance Analyst Bottleneck analysis section helps a DBA isolate potential problem areas where there are waits, resource overloads or potential failures. If concurrency issues or heavy table scan activity is impacting database performance, you can use bottleneck analysis to isolate the root cause. Once one or more bottlenecks have been isolated as possible areas of interest, you can drill down and examine details on why the applications or resources are causing the problem. The Performance Analyst home page identifies the top system and application bottlenecks that may be currently a cause of lessened performance.

The following bottleneck indicators are used on the Performance Analyst Home page to succinctly communicate the general overall performance level of the monitored database:

Concurrency	Apps Waiting on Locks	Avg. Lock Wait Time	Deadlocks Detected
	Lock Escalations	Lock Timeouts	
I/O Activity	Log Pages I/O	Synchronous I/O	
Object Bottlenecks	Accesses to Overflowed Records	Page Reorganizations	
Sorts and Joins	Hash Join Overflows	Sort Overflows	
Storage Bottlenecks	DMS Tablespaces Low on Space	Inaccessible Containers	Node with Smallest Log
	Tablespaces in Abnormal State		

Apps Waiting on Locks

This metric gives the percentage of all currently connected applications that are waiting for locks.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Bottleneck Analysis Pane
Metrics	If this number is high, you should investigate whether the applications are having concurrency problems.
Correction	Compare this metric with the lock escalations metric to identify if the lock list memory is too small. Go to the Locks Waiting Details section of the Locks tab of the Users detail view and examine the lock activity at application level to identify the applications that are holding a large number of row locks and table-level locks. You may be able to tune the applications with high number of locks.

Avg. Lock Wait Time

The average lock wait time metric gives the average amount of time applications waited for a lock in the database.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Bottleneck Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Users View (DBArtisan - DB2 LUW Performance Analyst) > Lock Waits tab
Metrics	This metric gives an overview of the concurrency performance of the database. If this value is too high, it may mean that either too many applications are vying for the same resources or that applications are executing long transactions that make other applications wait longer. You should examine this metric over a period of time to determine the normal operational state of the database.
Correction	As a first step, examine the lock activity at the application level to identify if any particular application is causing excessive lock contentions. If you see a high value, consider tuning the application to provide better concurrency. If lock timeouts are excessive, and average lock wait times are very short, you can increase the locktimeout database configuration parameter to make the applications wait longer before timing out. NOTE: You can find the lock activity in the Applications Waiting on Locks section of the Lock Waits tab of the User detail page.

Deadlocks Detected

Deadlocks detected is the total number of deadlocks that have occurred since this instance of Performance Analyst started monitoring the database.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Bottleneck Analysis Pane
Metrics	If a large number of deadlocks are detected, it can indicate that applications are experiencing lock contention problems. Deadlocks are usually caused by one of the following situations: *Lock escalations on the databaseCatalog tables locked for Repeatable ReadApplications are using inappropriate isolation levels at bind timeApplications are obtaining the same locks in a different orderApplications are locking tables explicitly where row level locks are sufficient.*
Correction	You may be able to modify the applications causing lock contentions for better concurrency. To identify the applications that may be causing contentions, go to the Lock Waits tab of the Users detail page and review the Application Details section.

Lock Escalations

The lock escalations metric indicates the number of times that locks have been escalated from row locks to table locks, since this instance of Performance Analyst started monitoring the database.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Bottleneck Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Users View (DBArtisan - DB2 LUW Performance Analyst) > Bottleneck Analysis Pane
Metrics	A lock is escalated when the total number of locks held by an application reaches the maximum amount it is allowed in the lock list memory. There are several possible causes of lock escalations: *The database lock list size is too small for the concurrent workloadThe maximum percentage of lock list memory allowed for each application is too smallOne or more applications are obtaining an excessive number of locks* Monitor the lock escalations over a period of time to determine what levels are acceptable in your environment. If the escalations are excessive, or are accompanied by deadlocks or long lock waits, consider tuning the database.
Correction	Examine the locklist database configuration parameter together with the Lock List Utilization metric to see if the locklist size needs to be increased. If the lock list utilization is within acceptable range, try raising the maxlocks database configuration parameter to increase the percentage of lock list space available to each application. You should also examine the detailed lock usage for each application on the Locks tab of the Users detail page to see if applications are holding a high number of locks. In this case, tune your application for better concurrency.

Lock Timeouts

This metric identifies the number of times that a request to lock an object timed out without being granted.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Bottleneck Analysis Pane
Metrics	If the number of lock timeouts becomes excessive when compared to the acceptable range for your database, it can indicate that an application is holding locks for long durations. It can also indicate that the amount of time an application waits for a lock before timing out is too short. If you have too few lock timeouts and the average lock wait time is too high, it can indicate that the lock timeout configuration parameter is set to an excessively high value.
Correction	First you should examine the lock activity at the application level to identify any particular application that is causing excessive lock contentions. If so, you can tune the application to provide better concurrency. If lock timeouts are excessive, and average lock wait times are very short, you can increase the locktimeout database configuration parameter to make the applications wait longer before timing out.

Log Pages I/O

The log pages I/O statistic is the total number of pages that are read and written to the disk by the database logger.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Bottleneck Analysis Pane
Metrics	You can use this element with an operating system monitor to get a bird's eye view of the amount of I/O on a device that is attributable to database activity.

Synchronous I/O

This metric presents the number of I/O operations that had to be performed synchronously by the database agents.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Bottleneck Analysis Pane
Metrics	DB2 read and write operations are always performed through bufferpools. If the page being read or written is not present in the bufferpool, it is first read into the bufferpool and then the I/O operation is performed on it. DB2 pre-fetchers fetch the data page into the bufferpool asynchronously in anticipation of a read/write operation. If the page is not available at the time of I/O, the database manager agent has to perform synchronous I/O. Synchronous I/O can decrease responsiveness of a database, and if the database tablespaces are spread across multiple physical devices you may be able to reduce the number of synchronous operations by tuning the database configuration.
Correction	You can set the num_ioserver database configuration parameter to 1-2 more than the number of physical storage devices used by the database to ensure that parallel I/O is occurring. To minimize synchronous I/O, you may also need to layout your tablespaces such that the containers are defined on different physical devices.

Accesses to Overflowed Records

This metric is the number of accesses (reads and writes) to overflowed rows in the database tables.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Bottleneck Analysis Pane

Metrics A row overflows if it is updated and no longer fits in the data page where it was originally written. This usually happens as a result of an update of a VARCHAR or an ALTER TABLE statement. Overflowed rows indicate that data fragmentation has occurred. If this number is high, you can try to improve performance by reorganizing the tables using the Space Analyst.

Page Reorganizations

The page reorganization statistic is the number of page reorganizations executed in the database.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Bottleneck Analysis Pane

Metrics Too many page reorganizations can result in less than optimal insert performance. You can use the Space Analyst to reorganize tables and eliminate fragmentation. You can also use the APPEND parameter for the ALTER TABLE statement to indicate that all inserts are appended at the end of a table and so avoid page reorganizations. In situations where updates to rows cause the row length to increase, the page may have enough space to accommodate the new row, but a page reorganization can be required to defragment that space. Or if the page does not have enough space for the new larger row, an overflow record is created that causes overflow_accesses during reads. You can avoid both situations by using fixed length columns instead of varying length columns.

Hash Join Overflows

Hash join is an option for the DB2 optimizer. A hash join compares the hash codes before comparing the predicates for tables involved in a join. This reduces the number of comparisons. The hash join overflows metric gives the percentage of all hash join operations that ran out of sort heap and may have required disk space for temporary storage since the current instance of Performance Analyst started monitoring the database.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Bottleneck Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Memory View (DBArtisan - DB2 LUW Performance Analyst) > Home > Bottleneck Analysis Pane

Metrics An overflowing hash join incurs a large overhead because of the amount of disk I/O required to complete the operation. If this value crosses the 30% mark, the DBA should take action.

Correction Increase the sortheap database configuration parameter to reduce the number of overflows.

Sort Overflows

The sort overflows statistic is the total number of sorts that ran out of sort heap and that may have required disk space for temporary storage.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Bottleneck Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Memory View (DBArtisan - DB2 LUW Performance Analyst) > Home > Bottleneck Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Users View (DBArtisan - DB2 LUW Performance Analyst) > Bottleneck Analysis Pane
Metrics	When a sort overflows, additional overhead is incurred. The sort requires a merge phase and can potentially require more I/O if data needs to be written to disk.
Correction	Sort overflows can be reduced by increasing the sortheap database configuration parameter.

DMS Tablespaces Low on Space

The DMS Tablespaces Low on Space metric gives the number of DMS tablespaces in the database where more than 90% of the total available space is in use.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Bottleneck Analysis Pane
Metrics	This metric is an indicator that some of the tablespaces in your database may be running out of space. You should drill down to the Space metrics to identify the tablespaces that may be at the critical utilization level. In tablespaces where there are little on no inserts after the initial data is loaded, little or no free space may have been left on purpose. In such cases, it is normal to see a high utilization.
Correction	If this metric goes beyond the normal operational value for your database, consider creating additional space on the DMS tablespaces that are running out of space. You can do this by performing one or more one of the following tasks: *Clean up unused space by deleting the tables/indexes that are no longer required.Resize or extend the existing containers in the tablespaces.Add new containers to the tablespaces.*

Inaccessible Containers

The inaccessible containers metric identifies the number of containers on all tablespaces that are currently inaccessible.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Bottleneck Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Space View (DBArtisan - DB2 LUW Performance Analyst) > Home > Bottleneck Analysis Pane
Metrics	Inaccessible containers represent a serious problem on the database. The database is unable to access/write any data on an inaccessible container. Usually this is caused by either media errors or modifications to the container files/directories from outside DB2.
Correction	Drill down to the Space metrics to examine the inaccessible containers and the tablespaces on which they are contained. You may need to perform a redirected restore on the tablespace from a backup to correct this.

Node with Smallest Log

The node with smallest log metric is only returned for global snapshots and indicates the node with the least amount of available log space.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Bottleneck Analysis Pane
Metrics	Use this metric in conjunction with the log space utilization metrics to check if the log space or applications on the specified node need to be adjusted.
Correction	If the log utilization is mostly low but increases to near 100% for short periods, then you can increase the number of secondary log files (logsecond database configuration parameter). This allocates and deallocates additional log files as needed. If, however, the log size is consistently high and changes frequently, you can increase the primary log size and/or the log file size (logfilsiz and logprimary database configuration parameters). Also, check the SQL activity, under Users Drilldown > SQL activity to see if any applications are performing infrequent commits. If possible, such applications can be changed to have shorter transactions and less need for log space.

Tablespaces in Abnormal State

This metric is a count of the number of tablespaces in your database that are not in a 'Normal State'.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Bottleneck Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Space View (DBArtisan - DB2 LUW Performance Analyst) > Home > Bottleneck Analysis Pane
Metrics	Tablespaces in not normal state are not necessarily a cause for alarm. They may have been transitioned to that state intentionally by the DBA for maintenance related tasks. If this metric shows a value that is higher than you anticipated, you may need to drilldown to the Space metrics to see which tablespaces are not normal. A tablespace can be in one or more of the following states when it is not 'Normal': Quiesced: SHARE, UPDATE, EXCLUSIVE Pending: Load, delete, backup, roll forward, restore, disable, drop In progress: Roll forward, reorg, backup, TBS deletion, TBS creation Storage must be defined Restore in progress Offline and not accessible Storage may be defined Storage definition is in 'final' state Storage definition was changed prior to rollforward DMS rebalancer is active
Correction	If the number of 'Not Normal' tablespaces is higher than you expect, drill down to the space metrics to examine the abnormal tablespaces and take appropriate action.

SQL Analysis Pane

A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured server down in terms of user response times.

Before you can identify problem SQL in your database, you need to understand the level of SQL activity that is occurring in the database. A high level overview of the type of SQL activity helps determine not only they types of statements that are being executed but also the type of internal activity they may be causing.

The following indicators are used on the Performance Analyst Home page to succinctly communicate the general overall performance level of the monitored database:

Static SQL
Internal Rebinds
Dynamic SQL
Internal Rollbacks Due to Deadlocks
Units of Work

Static SQL

The static SQL metric is the number of static SQL statements that were executed on the database since the start of this instance of Performance Analyst.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > SQL Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Memory View (DBArtisan - DB2 LUW Performance Analyst) > Home > SQL Analysis Pane
Metrics	Together with dynamic and failed SQL statements, this metric represents the number of SQL statements executed on the database.

Dynamic SQL

This metric is the number of static SQL statements that were executed on the database since the start of this instance of Performance Analyst.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > SQL Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Memory View (DBArtisan - DB2 LUW Performance Analyst) > Home > SQL Analysis Pane
Metrics	Together with Static and Failed SQL statements, this metric represents the number of SQL statements executed on the database.

Failed SQL

The failed SQL metric is the number of failed SQL statements that were executed on the database since the start of this instance of Performance Analyst.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > SQL Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Users View (DBArtisan - DB2 LUW Performance Analyst) > Home > Bottleneck Analysis Pane
Metrics	Together with static and dynamic SQL statements, this metric represents the number of SQL statements executed on the database. This metric may also help you determine reasons for poor performance, since failed statements mean time wasted by the database manager. This results in lower throughput for the database.

Units of Work

This is the number of transactions executed on the database since the start of this instance of Performance Analyst.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > SQL Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Memory View (DBArtisan - DB2 LUW Performance Analyst) > Home > SQL Analysis Pane

Metrics A low number of units of work compared to the overall SQL activity (static + dynamic SQL statements) indicate long transactions. This may in turn be an indicator of poor concurrency and heavy log usage.

Internal Rebinds

The internal rebinds metric is the number of automatic rebinds (or recompiles) that have been attempted on the database since the beginning of this instance of the Performance Analyst.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > SQL Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Memory View (DBArtisan - DB2 LUW Performance Analyst) > Home > SQL Analysis Pane

Metrics Automatic rebinds are the internal binds the system performs when a package has been invalidated. The rebind is performed the first time the database manager needs to execute an SQL statement from the package. For example, packages are invalidated when you: *Drop an object, such as a table, view, or index, on which the plan dependsAdd or drop a foreign keyRevoke object privileges on which the plan depends.* You can use this element to determine the level of database activity at the application or database levels. Since internal rebinds can significantly degrade performance, they should be minimized whenever possible.

Internal Rollbacks Due to Deadlocks

The internal rollbacks due to deadlocks metric is the total number of forced rollbacks initiated by the database manager because of a deadlock for the current instance of Performance Analyst. A rollback is performed on the current unit of work in an application selected by the database manager to resolve the deadlock.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > SQL Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Memory View (DBArtisan - DB2 LUW Performance Analyst) > Home > SQL Analysis Pane

Metrics This metric shows the number of deadlocks that have been broken and can be used as an indicator of concurrency problems. It is important because this metric lowers the throughput of the database.

Storage Analysis Pane

While DBAs focus on memory settings and tuning SQL, they frequently forget how dangerous and insidious storage problems can be. This is not a good mindset because storage headaches can play a major role in wrecking an otherwise well-running database.

Storage problems generally take one of two forms:

- The 'hit-the-wall' variety that can bring things to a complete standstill.
- The 'performance vampire' kind that slowly drains the performance of a database over time.

Storage problems have the capability to bring the curtain down on a database very quickly, as in the case of a transaction log running out of free space. But storage problems can also silently work behind the scenes to slowly, but surely rob a database of its performance. For example, a hub table in a busy database might be accessed very quickly when an application is first given life, but over time, if it develops a heavy forwarded row problem, it can cause things to run very differently.

The Storage Analysis section of the Performance Analyst Home page displays the total used and free space of all DMS tablespaces in the database. It also provides the total and available log space for the database.

The following ratios succinctly communicate the general overall performance levels of the datasource:

DMS Space Total SMS Space Used Log Space

Last Backup

DMS Space

The total used and total free metrics show the space details for DMS tablespaces. Specifically, they show the amount of used and free space on the DMS tablespaces in the database.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Storage Analysis PanelIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Space View (DBArtisan - DB2 LUW Performance Analyst) > Home > Storage Analysis Pane
Metrics	Unlike SMS tablespaces, the total available space to the DMS tablespaces is defined at the time the database is defined. The DBA needs to explicitly increase the amount of space on DMS tables by adding/extending tablespace containers (using the ALTER TABLESPACE statement). If left unattended, a DMS tablespace can either remain underutilized or fill up completely. Keeping an eye on the DMS space is important because the once it fills up, the applications trying to write to the database will come to a halt suddenly.
Correction	Go to the Space home page or drill-down to see the usage metrics for individual tablespaces and allocate/deallocate space to containers accordingly using ALTER TABLESPACE command.

Total SMS Space Used

This metric shows the total amount of SMS space used by the database.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Storage Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Space View (DBArtisan - DB2 LUW Performance Analyst) > Home > Storage Analysis Pane
Metrics	The maximum size of SMS tablespaces is not set at the time the tablespaces are created. The maximum size that can be reached is the space available on the drives/volumes that the SMS tablespace containers are defined on. You need to periodically examine the available space of the drives/volumes (using OS metrics Space page) to make sure there is enough space available for your database requirements. Since the space is determined by the space available on the drives/volumes, remember that if other applications are reading and writing to the same devices (especially logs, traces, etc), DB2 may be periodically competing for space with these applications.
Correction	Ideally, try and isolate the DB2 SMS drives/volumes from other applications. If the OS level metrics show that you are running out of space on a volume that is used by a tablespace's containers, you can add more containers defined on volumes with more space, to the tablespace using the ALTER TABLESPACE command.

Log Space

The log space metrics show the amount of free and used log space on the database.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Storage Analysis Pane
Metrics	Very high log space utilization (>75%) indicates that the database could run out of log space soon. Usually, consistently high log space utilization has one of the following causes: Large transaction size. Applications are performing a large number of updates on the database but are not committing the transactions frequently. The log space configured for the database is insufficient for the level of database activity Very low log space utilization (< 5) % can indicate an excessively large log file. It’s possible you can free some of the space occupied by log files on disk.
Correction	If the log utilization is mostly low but increases to near 100% for short periods, you can increase the number of secondary log files (logsecond database configuration parameter). This allocates and deallocates log files as needed. If however, the log size is consistently high, and changes frequently, you can increase the primary log size and/or the log file size (logfilsiz and logprimary database configuration parameters). Also, check the SQL activity, under Users Drilldown>SQL activity to see if any applications are performing infrequent commits, and if possible such applications can be changed to have shorter transactions and hence less need for log space.

Last Backup

The last backup metric indicates when the database was last backed-up.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Storage Analysis Pane

Workload Analysis Pane

When experiencing performance slowdowns, some of the first questions you should answer are:

- Who is currently connected to the database?
- What resources are they using?
- What are they currently executing?

The Workload Analysis section of the Performance Analyst Home page provides insight into the leading resource hogs of a server, as well as a count of total active and inactive processes. Drill-down's are available so you can easily get detailed information into what each leading application is currently involved with.

Workload Analysis statistics include:

Application Activity Analysis Applications Currently Executing Idle Applications

Application Activity Analysis

Ideally, database resource usage should be spread evenly across connected applications. In many cases, however, a couple of renegade application can hog most of the resources and starve the other applications that are competing for the same resources. If you see a slowdown of the database, but cannot identify the root cause from ratio / bottleneck analysis, you need to examine any connected applications that are using more than their fair share of system resources (CPU, I/O, locklist, etc.)

Performance Analyst provides an easy way to identify the top users of the database resources and what percentage of total usage can be attributed to these applications.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Workload Analysis Pane

Metrics If any one application is using more than 50% of the total resource you should drilldown into that application (Users drill-down) to isolate the problem.

Applications Currently Executing

The applications currently executing statistic gives the number of applications for which the database manager is currently executing requests.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Workload Analysis Pane
Metrics	You can use this number to understand how many of the database manager agent tokens are being used by applications connected to this database

Idle Applications

The idle applications statistic the number of applications that are currently connected to the database for which the database manager is not executing any requests

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Home View (DBArtisan - DB2 LUW Performance Analyst)** > Workload Analysis Pane
Metrics	You can use this statistic to help you understand the level of activity within a database and the amount of system resources currently in use.

Memory View (DBArtisan - DB2 LUW Performance Analyst)

The Memory category of DB2 Performance Analyst includes the following tabbed pages:

Caches Tab Memory Pools Tab Lock Memory Tab

Home

The Home page of the Memory performance category view has statistics in the following categories:

Key Ratio Analysis Pane Bottleneck Analysis Pane Memory Analysis Pane

SQL Analysis Pane Workload Analysis Pane

Key Ratio Analysis Pane

Database performance analysts typically use one of two methods for examining the performance levels of a database - ratio-based or wait/bottleneck-based. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as very good roll-up mechanisms for busy DBAs to use for at-a-glance performance analysis. Many DBAs have large database farms to contend with and cannot spend time checking detailed wait-based analysis outputs for each and every database they oversee. Succinctly presented performance ratios can assist in such situations by giving DBAs a few solid indicators that can be quickly scanned to see if any database needs immediate attention.

While there are certainly many opinions as to what rules to follow, there are some standards you should always adhere to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Most of the global ratios that a DBA will examine come from the DB2 Snapshot Monitor. The Snapshot Monitor returns various metrics for the instance and database since the start of monitoring. Some of the metrics are cumulative (i.e., counters) and others are instantaneous (i.e., gauges).

In the case of cumulative metrics, the values can grow quite large over the monitoring period and the ratios calculated on these metrics can hide the skews that may occur during the monitoring. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the database activity over the sampling period can be had.

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each DB2 databases' individual personality with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead the DBA to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following memory ratios are used on the Performance Analyst Key Ratio Analysis section to succinctly communicate the general overall memory performance levels of the monitored database:

Data Page Hit Ratio Private Workspace Hit Ratio Index Page Hit Ratio

The following topics, available on this pane, duplicate statistics available on the Home View (DBArtisan - DB2 LUW Performance Analyst) page:

Buffer Pool Hit Ratio Catalog Cache Hit Ratio Package Cache Hit Ratio

Shared Workspace Hit Ratio

Data Page Hit Ratio

This metric is the percentage of all data reads satisfied because the page was already available in a bufferpool.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key Ratio Analysis Pane
Metrics	The index page hits are used in conjunction with data page hits to calculate the overall buffer pool hit ratio. Use this element, in conjunction with the index page hit ratio when tuning the overall buffer pool hit ratio.

Private Workspace Hit Ratio

Each application executing on the database has access to the private workspace of the agents working for it.

The private workspace hit ratio is a percentage indicating how well the private SQL workspace is helping to avoid initialization of sections for SQL statements that are about to be executed. A high ratio indicates it is successful in avoiding this action.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key Ratio Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Caches Tab
Correction	Shared workspace memory is allocated from the application heap (app_ heap_sz database configuration parameter) and increasing this may improve the private workspace hit ratio.

Index Page Hit Ratio

The index page hit ratio is the percentage of all index reads that were satisfied because the page was already available in a bufferpool.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key Ratio Analysis Pane
Metrics	The index page hits are used in conjunction with data page hits to calculate the overall buffer pool hit ratio. Use this element, in conjunction with the data page hit ratio when tuning the overall buffer pool hit ratio.

Bottleneck Analysis Pane

The Bottleneck Analysis section identifies the number of times when overflows or over-utilization may have occurred in areas of memory. An overflow occurs when the usage of a memory area goes beyond the allocated amount.

Overflows are serious bottlenecks and may cause unnecessary lock escalations, out of memory errors or performance degradations.

The following statistics are used on the Performance Analyst for DB2 Memory home page to succinctly communicate the general overall performance levels of the memory structures:

Catalog Cache Overflows	Hash Join Small Overflows	Package Cache Overflows
Private Workspace Overflows	Shared Workspace Overflows	

The following statistics, available on this pane, dupicte statistics availabe on the Bottleneck Analysis Pane of the Home View (DBArtisan - DB2 LUW Performance Analyst) page.

Hash Join Overflows Lock Escalations Sort Overflows

Catalog Cache Overflows

The catalog cache overflow statistic is the number of times the catalog cache overflowed the bounds of its allocated memory.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Catalog cache overflows can cause unnecessary lock escalations. This can result in loss of concurrency, or 'out of memory' errors from other heaps allocated to the database's shared memory. Overflows of the catalog cache can also cause performance degradation. DB2 reclaims the catalog cache space by evicting table descriptor information for tables, views, or aliases and/or authorization information that is not currently being used by any transaction.
Correction	Use this element with the catalog cache high watermark (Memory> Caches Drilldown) to determine whether the size of the catalog cache needs to be increased to avoid overflowing. If the number of overflows is large, the catalog cache may be too small for the workload. Enlarging the catalog cache can improve its performance. If the workload includes transactions that compile a large number of SQL statements referencing many tables, views, aliases, user-defined functions, or stored procedures in a single unit of work, then compiling fewer SQL statements in a single transaction can improve the performance of the catalog cache. Or, if the workload includes binding packages that contain many SQL statements referencing many tables, views, aliases, user-defined functions, or stored procedures, you can try splitting packages so that they include fewer SQL statements to improve performance.

Hash Join Small Overflows

The hash join small overflows statistic is the number of times that hash join data exceeded the available sort heap space by less than 10%.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	If the value of hash join small overflows is greater than 10% of hash join overflows, then you should consider increasing the sort heap size. Values at the application level can be used to evaluate hash join performance for individual applications.
Correction	Increase the sortheap database configuration parameter to reduce the number of overflows.

Package Cache Overflows

The package cache overflows metric is the number of times that the package cache overflowed the bounds of its allocated memory.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Package cache overflows can cause unnecessary lock escalations. This can result in loss of concurrency, or 'out of memory' errors from other heaps allocated to the database's shared memory. Overflows of the package cache can also cause performance degradation.
Correction	Use this element with the package cache high watermark (Memory > Caches Drilldown) to determine whether the size of the package cache needs to be increased to avoid overflowing.

Private Workspace Overflows

The private workspace overflows statistic is the number of times that private workspaces overflowed the bounds of their allocated memory.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Private workspace overflows can cause performance degradation as well as 'out of memory' errors from other heaps allocated to the agent's private memory.
Correction	Use this element with the private workspace high watermark (Memory -> Caches Drilldown) to determine whether the size of the private workspace needs to be increased to avoid overflowing.

Shared Workspace Overflows

The shared workspace overflows metric is the number of times that shared workspaces overflowed the bounds of their allocated memory.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Overflows of shared workspaces can cause performance degradation. Overflows can also 'out of memory' errors from the other heaps allocated out of application's shared memory.
Correction	Use this element with the shared workspace high watermark (Memory > Caches Drilldown) to determine whether the size of the shared workspaces need to be increased to avoid overflowing.

Memory Analysis Pane

This section helps you to track memory usage for database processes through Memory Analysis Heaps.

Note: The memory management for DB2 (and hence the metrics) varies between platforms. You only see the memory metrics at the database level for UNIX system. If you are tracking the memory metrics for a Windows system, please refer to Instance -> Memory Pool Drilldown.

Memory Analysis Heaps

Under normal operating conditions you can track the memory usage for the following elements:

Application Heap
Lock Manager Heap
Package Cache Heap
Query Heap
FCMBP Heap
Buffer Pool Heap
Database Heap
Backup/Restore/Utility Heap
Catalog Cache Heap
Database Monitor Heap
Import Pool
Application Control
Statistics Heap
DFM Heap
Statement Heap
Other Memory

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Memory Analysis Pane
Metrics	The various memory heap usages are useful when tuning the corresponding database configuration parameters.
Correction	To see greater details on the current utilizations, maximum available, and high watermarks, go to the Memory Pools tab of the Memory detail view. Based on these you can make decisions about whether you need to increase or decrease the maximum allocations for the heaps (mostly through database configuration parameters)

SQL Analysis Pane

A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured server down in terms of user response times.

Before you can identify problem SQL in your database, you need to understand the level of SQL activity that is occurring in the database. A high level overview of the type of SQL activity helps determine not only they types of statements that are being executed but also the type of internal activity they may be causing.

The following indicators are used on the Performance Analyst Memory page to succinctly communicate the general overall performance level of the monitored database. They duplicate statistics available on the SQL Analysis Pane of the Home View (DBArtisan - DB2 LUW Performance Analyst) page:

Static SQL
Internal Rebinds
Dynamic SQL
Internal Rollbacks Due to Deadlocks
Units of Work

Workload Analysis Pane

When your phone starts ringing with complaints of performance slowdowns, one of the first things you should get a handle on is:

- Who is currently connected to the database?
- What resources are they using?
- What are they currently executing?

The Workload Analysis section of the Performance Analyst Memory page provides insight into the leading applications that caused memory overflows to disk or lock escalations. Drill down's are available so you can easily get detailed information into what each leading application is currently involved with.

Application Activity Analysis Pane

Top memory issues are displayed in the following categories:

- Top Locks
- Top Lock Escalations
- Top Cache Overflows
- Top Sort Overflows
- Top Workspace Overflows

Metrics

The list of top memory hogs can help you quickly identify the applications that are causing memory usage problems. You can drill down into the Memory tab views to see how these applications are using memory compared to their peers and the overall memory being used by the database. This is a good place to start, if you think that the memory problems you are noticing on the database may be as a result of poorly written applications rather than database configuration or overall system memory availability.

Caches Tab

The Caches tab of the Memory Detail page offers detailed views of cache usage metrics and includes the following statistics:

Application Details
Catalog Cache Usage Details
Package Cache Usage Details
Private Workspace Usage Details
Shared Workspace Usage Details

The following statistics, available on this page, dupication statistics available on the Key Ratio Analysis Pane of the Home View (DBArtisan - DB2 LUW Performance Analyst) page:

Catalog Cache Hit Ratio
Package Cache Hit Ratio
Shared Workspace Hit Ratio

The following statistic, available on this page, dupication statistics available on the Caches Tab of the **Memory View (DBArtisan - DB2 LUW Performance Analyst)** page:

Private Workspace Hit Ratio

Application Details

The Application Details section shows the list of applications connected to the database and the various cache usage metrics for each application. It also shows the rolled up metrics at the database level in the same list. You can select the type of cache monitoring you intend to perform for the applications from the following:

- Catalog Cache: Shows the details of all catalog cache activity for connected applications
- Package Cache:Displays the details of all package cache activity for connected applications
- Shared Workspace:Gives the details of all shared workspace activity for connected applications

Private Workspace:This shows the details all of private workspace activity for connected applications

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Caches Tab
Correction	Depending on the category of cache details you select, you can view cache activity associated with the applications. This section is very useful in isolating the applications and the cache areas you need to tune by comparing the cache performance for all the applications and also the overall cache performance at the database level.

Catalog Cache Usage Details

This section shows detailed catalog cache usage metrics for the selected item. The metrics presented here include:

- Catalog Cache Lookups: The number of times that the catalog cache was referenced to obtain table descriptor information or authorization information
- Catalog Cache Inserts: The number of times that the system tried to insert table descriptor or authorization information into the catalog cache.
- Catalog Cache Overflows: The number of times that the catalog cache overflowed the bounds of its allocated memory
- Catalog Cache High Watermark: The largest size reached by the catalog cache

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Caches Tab
Metrics	These metrics are used for calculating the catalog cache hit ratio. They also give you an estimate of catalog access activity occurring in the database. You can use the catalog cache overflow metric with the catalog cache high watermark to determine if the catalog cache size is too small. If the catalog cache overflows frequently, you consider increasing the catalog cache size.
Correction	If catalog cache overflows are large, the catalog cache may be too small for the workload. Enlarging the catalog cache may improve its performance. If the workload includes transactions that compile a large number of SQL statements referencing many tables, views, aliases, user-defined functions, or stored procedures in a single unit of work, then compiling fewer SQL statements in a single transaction may improve the performance of the catalog cache. Or if the workload includes binding packages that contain many SQL statements referencing many tables, views, aliases, user-defined functions, or stored procedures, you can try splitting packages. If they include fewer SQL statements, performance can improve. Monitor the catalog cache metrics for your normal workload. If you see frequent overflows, you can increase the value of catalogcache_sz database configuration parameter to the catalog cache high watermark/4096 (rounded up to whole integer).

Package Cache Usage Details

The Package Cache Usage Details section shows the detailed package cache usage metrics for the selected item. The metrics presented here include:

- Package Cache Lookups: The number of times that an application looked for a section or package in the package cache. At a database level, it indicates the overall number of references since the database was started, or monitor data was reset.
- Package Cache Inserts: The total number of times that a requested section was not available for use and had to be loaded into the package cache. This count includes any implicit prepares performed by the system.
- Package Cache Overflows: The number of times the package cache overflowed the bounds of its allocated memory
- Package Cache High Watermark: The largest size reached by the package cache.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Caches Tab
Metrics	These metrics are used to calculate the package cache hit ratio. They also give you an estimate of package cache access activity occurring on the database. You can use the package cache overflow metric with the package cache high watermark to determine if the package cache size is too small. If the package cache overflows frequently, you may need to increase the package cache size.
Correction	Monitor the package cache metrics for your normal workload. If you see frequent overflows, you can increase the value of the pckcachesz database configuration parameter to the package cache high watermark/4096 (rounded up to whole integer).

Private Workspace Usage Details

This section shows the detailed private workspace usage metrics for the selected item. The metrics presented here include:

- Private Workspace Lookups: Lookups of SQL sections by an application in its agents' private workspace.
- Private Workspace Inserts: Inserts of SQL sections by an application into the private workspace.
- Private Workspace Overflows: The number of times that the private workspaces overflowed the bounds of its allocated memory.
- Private Workspace High Watermark: The largest size reached by the Private Workspace.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Caches Tab
Metrics	These metrics are used to calculate the private workspace hit ratio. They also give you an estimate of private workspace access activity occurring on the database. When the private workspace overflows, memory is temporarily borrowed from other entities in agent private memory. This can result in memory shortage errors from these entities or possibly performance degradation.
Correction	You can reduce the chance of overflow by increasing APPLHEAPSZ.

Shared Workspace Usage Details

This section shows the detailed shared workspace usage metrics for the selected item. The metrics presented here include:

- Shared Workspace Lookups: Lookups of SQL sections by applications in shared workspaces.
- Shared Workspace Inserts: Number of inserts of SQL sections by applications into shared workspaces.

- Shared Workspace Overflows: The number of times that shared workspaces overflowed the bounds of their allocated memory.
- Shared Workspace High Watermark: The largest size reached by shared workspaces.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Caches Tab
Metrics	These metrics are used to calculate the shared workspace hit ratio. They also give you an estimate of shared workspace access activity occurring on the database. When the shared workspace overflows, memory is temporarily borrowed from other entities in application shared memory. This can result in memory shortage errors from these entities or possibly performance degradation.
Correction	You can reduce the chance of overflow by increasing APPL_CTL_HEAP_SZ.

Memory Pools Tab

The Memory Pools tab offers the following detailed views:

Application Details Memory Pool Details Memory Pool Usage

Memory Pool Utilization

Application Details

This section shows the list of applications connected to the database. You can select an application to show the breakdown of the memory pool usage by the application.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Pools Tab

Memory Pool Details

This section shows the memory pool usage details for the selected application (or database level) in the application details section. The metrics shown here include:

- Pool ID: This is the kind of memory pool. Each pool type is only shown once
- Pool Current Size: The current size of a memory pool.
- Pool Max Size: The internally configured size of a memory pool in DB2.

Pool High Watermark: The largest size of a memory pool since its creation.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Pools TabIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Instance View (DBArtisan - DB2 LUW Performance Analyst) > Memory Pools tab

Metrics The nature of memory_pool data elements varies between platforms. On Windows systems, no memory usage is reported at the database level, while on UNIX systems, memory is reported at the database level. Instead of reporting this memory at the database level, the system monitor for Windows systems reports it in instance -level snapshots. This divergence in reporting is due to differences in the underlying memory architecture between Windows systems and UNIX systems.

Memory Pool Usage

This section shows the memory pool usage details of the selected memory pool over time including the current size of the memory pool, its high watermark, and the maximum size of the memory pool.

Location

- IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Pools Tab

Memory Pool Utilization

This section shows the percentage utilization of the selected memory pool.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Pools TabIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Instance View (DBArtisan - DB2 LUW Performance Analyst) > Memory Pools tab

Metrics You can use this metric to see if a memory pool is nearly full. You can diagnose the problems with specific memory pools by monitoring their utilization over time.

Correction If the value of the pool utilization is consistently close to or exceeds 100%, you may need to increase the configuration parameter associated with that pool.

Lock Memory Tab

The Lock Memory tab of the Memory Detail view shows the following statistics:

- Lock List Utilization(dupiicates a statistics available on the Key Ratio Analysis Pane of the Home View (DBArtisan - DB2 LUW Performance Analyst) page)
- Lock Escalations

Lock Escalations

This section shows the metrics for lock escalations in the database. The metrics shown here are:

- Lock Escalations: The number of times that locks have been escalated from several row locks to a table lock.

- Exclusive Lock Escalations: The number of times that locks have been escalated from several row locks to one exclusive table lock, or the number of times an exclusive lock on a row caused the table lock to become an exclusive lock.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Memory View (DBArtisan - DB2 LUW Performance Analyst)** > Lock Memory Tab
Metrics	A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application, or the lock list space consumed by all applications is approaching the total lock list space. The amount of lock list space available is determined by the maxlocks and locklist configuration parameters. When an application reaches the maximum number of locks allowed and there are no more locks to escalate, it will then use space in the lock list allocated for other applications. When the entire lock list is full, an error occurs. This data item includes a count of all lock escalations, including exclusive lock escalations. There are several possible causes for excessive lock escalations: *The lock list size (locklist) may be too small for the number of concurrent applicationsThe percent of the lock list usable by each application (maxlocks) may be too smallOne or more applications may be using an excessive number of locks.*
Correction	To resolve these problems, you may be able to: *Increase the locklist configuration parameter value. See the Administration Guide for a description of this configuration parameter.Increase the maxlocks configuration parameter value. See the Administration Guide for a description of this configuration parameter.Identify the applications with large numbers of locks (see locks_held_top), or those that are holding too much of the lock list, using the following formula:(((locks held * 36) / (locklist * 4096)) * 100)and comparing the value to maxlocks. These applications can also cause lock escalations in other applications by using too large a portion of the lock list. These applications may need to resort to using table locks instead of row locks, although table locks may cause an increase in lock_waits and lock_wait_time.*

I/O

In addition to a Home page, the I/O category of DB2 Performance Analyst includes the following tabbed pages are available on the I/O Detail view:

Applications Tab Bufferpools Tab Logging Tab

Tablespaces Tab

Home

The Home Page of the I/O performance category page displays the following vital DB2 I/O statistic categories:

Key System Analysis Pane Bottleneck Analysis Pane SQL Analysis Pane

Non Buffered I/O Activity Workload Analysis Pane

Key System Analysis Pane

The root cause of many database problems can be attributed to I/O-related operations. I/O monitoring is essentially a review of your physical database design. All physical storage characteristics and placements, table and index designs, and the speed with which all processes works are on display when I/O is monitored. Because a database's main index of performance is measured by how fast I/O needs are satisfied, it is your responsibility to quickly determine if a reported database slowdown is I/O related.

The following statistics are used on the Performance Analyst for DB2 I/O Home Page to succinctly communicate the general overall performance levels of I/O:

Efficiency

Async Read Ratio Async Write Ratio Victim Page Cleaners %

Resource Utilization

Log Space Cleaners Triggered Victim Page Cleaners Triggered Threshold Cleaners Triggered

Async Read Ratio

Async read ratio is the percentage of all index and data reads that performed asynchronously by the database manager prefetchers.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > Key System Analysis Pane
Metrics	Prefetching refers to the technique whereby the Database Manager can read several pages on the physical reads device simultaneously into a bufferpool in anticipation of access to these pages. Since each physical read operation is costly, a 50% async read ratio is desirable. It shows that prefetchers are working well and read waits are minimal.
Correction	The num_ioserver database configuration parameter defines the number of prefetchers that are available for the database. To get an optimal async read ratio, you should set the num_ioserver database configuration parameter to 1-2 more than the number of physical storage devices used by the database to ensure that asynchronous I/O is occurring. This parameter should be changed judiciously. Having too many prefetchers invariably results in high system I/O because prefetchers can read many more pages than required into the bufferpools.

Async Write Ratio

Async write ratio is the percentage of all index and data writes that were performed asynchronously by the database manager bufferpool page cleaners.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > Key System Analysis Pane

Metrics Page cleaners write changed pages from bufferpool to disk before the space in the bufferpool is needed by a database agent. If the page cleaners are not working well, the problem may manifest itself in two ways: *The database agents need to synchronously free space in bufferpools resulting in poorer response time.If the system crashes, recovery time is greater because there will be a large number of pages that would not have been committed to disk at the time of the crash and they require processing for a large number of log records.* An overall async write ratio of 50% is desirable for most transactional database systems. If your database is 'query only' (i.e., updates are rarely or never performed), it is fine to have an async write ratio of 0 to less than 50%. Even in 'query only' databases if the queries create temp tables on the database for intermediate results, a 30-40% async write ratio is desirable.

Correction The async write ratio can be tuned using the num_iocleaners database configuration parameter, which specifies the number of asynchronous page cleaners for a database. Increasing the number of page cleaners generally results in a higher async write ratio. The following rules of thumb should be followed when tuning this parameter: *For most transactional systems set this parameter to between one and the number of physical storage devices used by the database.For workloads that consists mainly of simple reads from the database, set this parameter to zero.When you define very large bufferpools on your database you may need to increase the num_iocleaners* Monitor the victim page cleaners% metric when tuning the num_iocleaners.

Victim Page Cleaners %

The victim page cleaners % metric is the percentage of times the page cleaners were invoked because synchronous write was needed during the victim buffer replacement in the database.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > Key System Analysis Pane

Metrics Usually this metric should be maintained around the 50-60% level. If this ratio much lower than 50%, it can indicate that you have defined too many page cleaners. If the chngpgs_thresh database configuration parameter is set too low, the page cleaners may be writing out pages that will be dirtied later. Aggressive cleaning defeats one purpose of the bufferpool, that is to defer writing to the last possible moment. If this ratio is high, you may have too few page cleaners defined. This can increase recovery time after failures.

Correction You may increase or decrease the chngpgs_thresh database configuration parameter to bring the victim page cleaner % to optimal range. Usually, for databases with heavy transaction workloads, chngpgs_thresh should be set to around 60%. For a database with only a few very large tables you can try and increase this parameter to a higher value to see if you get performance improvements.

Log Space Cleaners Triggered

The log space cleaners triggered metric is the number of times a page cleaner was triggered because log space usage reached a predefined threshold for the database.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > Key System Analysis Pane
Metrics	The log space cleaners are triggered every time the space used by the DB2 log reaches the limit set in the softmax database configuration parameter. This parameter specifies the percentage of primary log size at which the cleaners are triggered. By default this parameter is set to 100. A low rate of log space cleaners getting triggered may indicate that logs are not being written to disk frequently enough and that you may need to process a large number of log records and redundant log records in the event of a crash recovery. High rates of log space cleaners getting triggered can indicate that your primary log is too small or that you have set the softmax too high. In either case, a very high rate of log cleaners being triggered may adversely impact database performance. It is important to keep in mind, however, that more page cleaner triggers and more frequent soft checkpoints increase the overhead associated with database logging.This can have an impact on the performance of the Database Manager. Also, more frequent soft checkpoints may not reduce the time required to restart a database, if you have: *Very long transactions with few commit points.A very large bufferpool and the pages containing the committed transactions are not written back to disk very frequently. The use of asynchronous page cleaners can help avoid this situation.* In both of these cases, the log control information kept in memory does not change frequently and there is no advantage in writing the log control information to disk, unless it has changed
Correction	Examine this metric together with other page cleaner metrics and the async write ratio to determine if excessive numbers of log page cleaners are being triggered. If this is true, you need to either increase your primary log size or adjust the softmax parameter to a higher value.

Note: Decreasing the log page cleaner triggering rate can impact the overall crash recovery time so it needs to be done judiciously.

Victim Page Cleaners Triggered

The victim page cleaners triggered statistic is the number of times the page cleaners were invoked because synchronous write was necessary during the victim buffer replacement in the database.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > Key System Analysis Pane
Metrics	If the chngpgs_thresh database configuration parameter is set too low, the page cleaners may be writing out pages that will be dirtied later. Aggressive cleaning defeats one purpose of the bufferpool, which is to defer writing to the last possible moment. If this ratio is high, it may indicate that you have too few page cleaners defined. Too few page cleaners increases recovery time after failures.
Correction	You may increase or decrease the chngpgs_thresh database configuration parameter to bring the victim page cleaner triggering rate to optimal range. For databases with heavy transaction workloads, chngpgs_thresh should be set to around 60%. For a database with only a few very large tables you can try and increase this parameter to a higher value to see performance improvements.

Threshold Cleaners Triggered

The threshold cleaners triggered metric is the number of times a page cleaner was invoked because a bufferpool had reached the dirty page threshold criterion for the database.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > Key System Analysis Pane
Metrics	The dirty page threshold is set by the chngpgs_thresh database configuration parameter. It is a percentage applied to the bufferpool size. When the number of dirty pages in the pool exceeds this value, the cleaners are triggered. If this value is set too low, pages might be written out too early, requiring them to be read back in. If set too high, then too many pages may accumulate, requiring users to write out pages synchronously.
Correction	For databases with frequent update transactions, you can generally ensure that there are enough clean pages in the bufferpool by setting the chngpgs_thresh parameter value to be equal-to or less-than the default value. A percentage larger than the default can help performance if your database has a small number of very large tables.

Bottleneck Analysis Pane

High prefetch waits and file closures are good indicators of I/O waits at the database. Often, heavy I/O activity may not be due to non-optimal settings of the database configuration parameters. Rather, the underlying cause can be poor physical layout of a few tables/tablespaces or a few poorly written/suboptimal database applications.

It is usually best to try and tune individual objects/applications before attempting to tune database configuration parameters. Database configuration changes can have a profound impact on every application and object that resides in the database and if there are only a couple of apps/objects dragging down the performance of the entire database, it may be best to first isolate and tune these.

I/O Bottleneck analysis provides information on applications and tablespaces that are performing the highest number of read and write operations on the database at a glance. It also shows the bufferpool I/O metrics that point to direct I/O waits.

The following statistics are used on the Performance Analyst for DB2 I/O home page to succinctly communicate the general overall performance levels of I/O:

Hottest Tablespaces

Total Time Spent Waiting for Prefetchers

Hottest Apps Total Database Files Closed

Hottest Tablespaces

This section displays the list of top three tablespaces with the highest number of physical reads and writes.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > Bottleneck Analysis Pane
Metrics	Understanding the storage-level hot spots of a database is beneficial for a two reasons. First, you can get a feel for overworked physical disks by viewing I/O statistics at the tablespace level. If a particular tablespace is under too much strain, you can attempt to relocate tablespace containers to other less-used devices. You can also try to create new tablespaces on different disks and move hot objects to them (assuming, of course, you have extra disks available). Second, if you have used standard DBA practices and placed indexes in their own tablespace, you can review the I/O statistics for the hottest tablespaces and see if the indexes are actually being used.
Correction	Some areas to consider when reviewing the hottest tablespaces display are: *A temp tablespace showing high volumes of physical I/O could indicate a problem with excessive sort overflows.Quickly review all physical I/O for each drive/file system at the OS level and get a feel for the overworked disks on your server. If you have underutilized disk drives (with their own controllers), then you should consider relocating some tablespaces/containers that exhibit high I/O characteristics to those drivesTo minimize contention at the physical disk level, and to improve performance overall, it is generally recommended that a database have its tablespaces and log files physically separated onto different drives/file systems.*

Hottest Apps

This metric shows the top three applications connected to the database with the highest number of physical reads and writes.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > Bottleneck Analysis Pane
Metrics	A few applications with suboptimal queries/database accesses can drag down the performance of the entire database. The hottest apps section allows you to get a good feel for the applications that are causing the highest number of reads and writes on the database. It is also a good indicator of where you will reap the greatest benefit when tuning the application's performance.
Correction	Drill down to the application I/O pages and see how the high I/O applications are behaving compared to other applications. Check the SQL Snapshot Metrics for the high read/write apps to see what queries these applications are running on the database and try to optimize these queries.

Total Database Files Closed

Both SMS tablespaces and DMS tablespace file containers are treated as files in the Database Manager's interaction with the operating system, and file handles are required. There is a limit to the number of files that can be open concurrently by each database agent and a file is closed when the limit of open files for an agent is reached and the agent needs to open another file. This metric gives the number of times a database file was closed because the limit for concurrently open files was reached.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > Bottleneck Analysis Pane
Metrics	Both SMS table spaces and DMS tablespace file containers are treated as files in the Database Manager's interaction with the operating system. More files are generally used by SMS tablespaces compared to the number of containers used for a DMS file table space. Thus, if you use SMS tablespaces, the database needs to open a larger number of files compared to what would be required for DMS file tablespaces. Closing and opening files is a very expensive operation and a large number of closed files can adversely affect the performance of the database. It is recommended that the number of files closed be as small as possible (preferably 0).
Correction	If you notice that files are being closed during the monitoring, first check the rate at which the files are being closed at the database level, from the I/O drill-down. If you see files being closed frequently, increase the maxfilop database configuration parameter. Note that the operating system on which the database is running also has a limit on the number of files that can be opened concurrently. Make sure that you do not increase the maxfilop parameter beyond the operating system limit.

Total Time Spent Waiting for Prefetchers

This is the total time database applications spent waiting for prefetchers to finish loading.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > Bottleneck Analysis Pane
Correction	If you see a high number for this metric, it indicates there is a bottleneck in the prefetcher operations. Experiment with changing the number of I/O servers (num_ioserver) and the I/O server sizes.

SQL Analysis Pane

Most of database's I/O performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag down an otherwise well-configured server in terms of user response times. This section shows a snapshot of row level SQL operations that have been performed on the database since you started monitoring. This gives you an insight into the row level of read/write activity that is currently occurring within the database and comparing this to the prefetcher, and cleaner activity allows you to relate the two.

The following statistics are used on the Performance Analyst for DB2 I/O home page to succinctly communicate the general overall performance levels of I/O:

Rows Deleted Rows Inserted Rows Selected

Rows Updated Rows Read

Rows Deleted

The rows deleted metric is the total number of row deletions attempted on the database. It includes the internal row deletions that were caused by constraints or triggers.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > SQL Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Users View (DBArtisan - DB2 LUW Performance Analyst) > Home > SQL Analysis Pane

Rows Inserted

The rows inserted metric is the total number of row insertions attempted on the database. This metric includes the internal row insertions that were caused by triggers.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > SQL Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Users View (DBArtisan - DB2 LUW Performance Analyst) > Home > SQL Analysis Pane

Rows Selected

The rows selected statistic is the total number of rows selected at the database. This metric does not include a count of rows accesses for operations such as COUNT(*) or joins.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > SQL Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Users View (DBArtisan - DB2 LUW Performance Analyst) > Home > SQL Analysis Pane

Rows Updated

The rows updated metric is the total number of row updates attempted on the database. This metric includes the internal row updates that were caused by triggers or constraints

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > SQL Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Users View (DBArtisan - DB2 LUW Performance Analyst) > Home > SQL Analysis Pane

Rows Read

The rows read statistic is the number of rows read from the table. This count is not the number of rows that were returned to the calling application. Rather, it is the number of rows that had to be read in order to return the result set.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > SQL Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > Users View (DBArtisan - DB2 LUW Performance Analyst) > Home > SQL Analysis Pane

Non Buffered I/O Activity

Some database operations such as backups, loads, restores, LOB (large object I/O) bypass the bufferpools and read and write directly to storage. Sometimes, even if your database I/O configurations and all of the other activities are within the normal thresholds, the I/O performance may be poor because LOB reads/writes or load/backup/restore operations are causing or waiting on I/O contentions on your storage devices.

The non buffered I/O activity section allows you to quickly identify the level of direct I/O activity on the database. The level of direct I/O activity is succinctly presented using the following metrics:

Direct Reads from the Database Direct Writes to the Database Direct I/O Averages

Direct Reads from the Database

The direct reads from the database metric identifies the number of read operations that did not use a bufferpool. Direct reads are performed for the following database operations:

- Reading LONG VARCHAR columns
- Reading LOB (BLOB/CLOB/DBCLOB) columns
- Performing a backup

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > SQL Analysis Pane

Direct Writes to the Database

The direct writes to the database metric identifies the number of write operations that did not use a bufferpool. Direct writes are performed for the following database operations:

- Writing LONG VARCHAR columns
- Writing LOB (BLOB/CLOB/DBCLOB) columns
- Performing a load
- Performing a restore

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > SQL Analysis Pane

Direct I/O Averages

The following values show how well the direct I/O operations are performing:

- Avg. Sectors / Read: Average number of sectors read for each direct read operation
- Avg Sectors / Write: Average number of sectors written for each direct write operation
- Avg Read Time: Average read time per sector for direct reads

Avg Write Time: Average write time per sector for direct writes

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > SQL Analysis Pane
Metrics	Understanding the level of direct I/O averages is important. A high level of activity with a slowdown in database performance can indicate that you need to have dedicated and/or high performance storage devices for your LOB objects or backup and load images. These metrics, when viewed with the device metrics at the OS level can also indicate I/O resource conflicts with other processes.
Correction	You can monitor these metrics for the LOB storage devices at the OS level to see if they are inordinately high for the database compared to the OS level. If so, there may be an I/O conflict with another process.

Workload Analysis Pane

When experiencing performance slowdowns, some of the first questions you should answer are:

- Who is currently connected to the database?
- What resources are they using?
- What are they currently executing?

The Workload Analysis section of the Performance Analyst I/O page provides insight into the leading resource hogs of a server, as well as a count of total active and inactive processes. Drill-down's are available so you can easily get detailed information into what each leading application is currently involved with.

Workload Analysis statistics includes:

- Top I/O Apps

Top I/O Apps

The Top I/O Apps section gives a list of the apps with performing that are performing the highest number of most aggressive I/O activities The list includes:

- Application with most bufferpool misses
- Application with highest direct reads and writes
- Application with highest sort overflows
- Application with highest direct read/write time

Application with highest bufferpool read write time

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Home > Workload Analysis Pane
Correction	Drill down to the application I/O level to see the details and compare the metrics with the application's peers, and the database. In a lot of cases simple SQL rewrites, logic changes, and defining indexes based on access patterns can make a world of difference.

Applications Tab

This section shows the list of applications connected to the database and the various I/O metrics for each application. It also shows the rolled up metrics at the database level in the same list. You can select the type of I/O monitoring you intend to perform for the applications from the following:

- Direct I/O: This shows the details of all direct I/O activity for the applications.
- Buffered I/O: This shows the details of all bufferpool I/O activity for the applications connected to the database.
- Extended Storage: This shows the level of extended storage usage by the applications.
- Sorts & Joins:This shows the details for the number of files closed for the various applications.
- Block / Vectored I/O: This shows the detailed of block and vectored I/O performed by the applications.

Correction

Depending on the type of details you select, you can view the different types of I/O activity associated with the applications. This section is very useful in isolating the applications, and the I/O areas you need to tune by comparing the I/O performance for all the applications and also the overall I/O performance at the database level.

Direct I/O

The Direct I/O Application or Tablespace Details view lets you the following statistics:

Direct Reads and Writes Direct Read and Write Requests Direct Read and Writes Time

Direct Reads and Writes

This section shows the rate at which direct read/write operations are occurring for the selected element. Direct I/O operations are those that do not use the bufferpools. Direct read and write operations are performed when:

- Reading/Writing LONG VARCHAR columns
- Reading/Writing LOB data
- Performing a backup operation
- Performing a restore operation

- Performing a load operation

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Applications Tab

Metrics When using the performance analyst to track I/O, these elements help you distinguish the database I/O from the non-database I/O on storage devices.

Direct Read and Write Requests

This section shows the rate at which direct read/write operations requests are occurring for the selected element. Direct I/O operations are those that do not use the bufferpools. Direct read and write operations are performed when:

- Reading/Writing LONG VARCHAR columns
- Reading/Writing LOB data
- Performing a backup operation
- Performing a restore operation
- Performing a load operation

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Applications Tab

Metrics Use these metrics together with the Direct R/W metrics and Direct R/W request metrics to determine if the rate at which the read/write occurring is similar to the rate at which they are being satisfied. If there is a large discrepancy between the Direct I/O request rates and Direct I/O rates, it may mean that there are I/O contentions on the device(s) where the direct I/O operations are being performed.

Correction If you suspect there are I/O contentions, you can investigate whether the contentions are being caused by the DB2 processes or other processes by examining the I/O metrics at the operating system level. Consider using dedicated devices for the direct I/O operations to reduce the contention levels.

Direct Read and Writes Time

This section shows the elapsed time to perform the direct I/O operations. Direct I/O operations are those that do not use the bufferpools. Direct read and write operations are performed when:

- Reading/Writing LONG VARCHAR columns
- Reading/Writing LOB data
- Performing a backup operation
- Performing a restore operation

Performing a load operation

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Applications Tab
Metrics	Use these metrics together with the Direct R/W metrics to determine average time it is taking to perform the read write operations. If the level of Direct I/O rates is low and the Direct RW times, it may mean that there are I/O contentions on the device(s) where the direct I/O operations are being performed.
Correction	If you suspect there are I/O contentions, you can investigate whether the contentions are being caused by the DB2 processes or other processes by examining the I/O metrics at the operating system level. Consider using dedicated devices for the direct I/O operations to reduce the contention levels. If the contention levels are low, you may also benefit from using faster storage devices for direct I/O.

Buffered I/O

The Buffered I/O Application Details view lets you see the following statistics:

Hit Ratios Write/Read Ratios

Hit Ratios

The DB2 database server reads and updates all data from a bufferpool because memory access is much faster than disk access. Data is copied from disk to a bufferpool as needed by the applications using the database. When the server needs to read/write data and the data is already in the bufferpool, no disk access is required. However, if the data is not in the bufferpool, it needs to be read from the disk, which is significantly slower process.

The buffer pool hit ratio indicates the percentage of time that the database server did not need to load a page to service the request for a page. The calculation takes into account all the index and data pages that were requested

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Applications Tab
Metrics	Avoiding disk I/O is the main issue when trying to improve performance tuning. A high buffer pool hit ratio is desirable because it indicates a lower frequency of synchronous disk I/O. A database where data is accessed uniformly from very large tables will have a poor hit ratio. There is little you can do to improve the performance in such cases.
Correction	The buffer pool hit ratio on the database Home page is the overall hit ratio of the database. First, drill down to the bufferpool level and check the individual buffer pool hit ratios to identify the elements that may require tuning. Increasing the bufferpool size generally improves the buffer pool hit ratio. You can use one of the following methods to increase the bufferpool size depending on the layout of the data: *If the tablespaces using the bufferpool have tables and indexes on them, increase the bufferpool size incrementally until the index page hit ratio stops increasing for that bufferpool. You will not usually get any benefit increasing the bufferpool size after you reach that point.If the tablespaces associated with the bufferpool have only indexes on them, increase the bufferpool size until the index page hit ratio stops increasing.If the tablespaces associated with the bufferpool have only data on them, increase the bufferpool size until the data page hit ratio stops increasing.One general design guideline is to try and use different tablespaces for indexes and tables and associate different bufferpools with them. This generally improves performance and makes bufferpool tuning much easier.*

Write/Read Ratios

This section shows the overall level of read / write activity being performed on the selected element.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Applications Tab

Extended Storage

The Extended Storage Application or Transaction Details view lets you see the following statistics:

Index I/O on Extended Storage Data I/O on Extended Storage

Index I/O on Extended Storage

This section shows the Index Pages copied to/from extended storage for the selected element. On 32-bit platforms, virtual addressable memory is usually limited to between 2 and 4 GB. If your 32-bit machine has more real addressable memory than the maximum amount, you can configure any additional real addressable memory beyond virtual addressable memory as an extended storage cache. Any of the defined bufferpools can use an extended storage cache to improve performance. You define the extended storage cache as a number of memory segments.

The bufferpools perform first-level caching, and any extended storage cache is used by the bufferpools as secondary-level caching. Ideally, the bufferpools hold the data that is most frequently accessed, while the extended storage cache hold data that is accessed less frequently.

Pages are copied from the bufferpool to extended storage, when they are selected as victim pages. This copying is required to make space for new pages in the bufferpool.

Required index pages are copied from extended storage to the bufferpool, if they are not in the bufferpool, but are in extended storage. This copying may incur the cost of connecting to the shared memory segment, but saves the cost of a disk read.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Applications Tab
Metrics	Because an extended storage cache is an extension to a bufferpool, it must always be associated with one or more specific bufferpools. Therefore, only the bufferpools that are declared to take advantage of advantage of the extended storage use it. The CREATE and ALTER BUFFERPOOL statements have the attributes NOT EXTENDED STORAGE and EXTENDED STORAGE that control cache usage. By default neither IBMDEFAULTBP nor any newly created bufferpool will use extended storage. Use these metrics together with the Data I/O on Extended Storage metrics to determine the level of I/O being performed on extended storage. If there is significant activity on this section, you may improve performance by increasing the size of extended storage.
Correction	The amount of extended storage for a database is determined by the following two database configuration parameters: **num_estore_segs***defines the number of extended storage memory segments. The default for this configuration parameter is zero, which specifies that no extended storage cache exists.***estore_seg_sz***defines the size of each extended memory segment. This size is determined by the platform on which the extended storage cache is used.*

Data I/O on Extended Storage

This section shows the DataPages copied to/from extended storage for the selected element.

On 32-bit platforms, virtual addressable memory is usually limited to between 2 and 4 GB. If your 32-bit machine has more real addressable memory than the maximum amount, you can configure any additional real addressable memory beyond virtual addressable memory as an extended storage cache. Any of the defined bufferpools can use an extended storage cache to improve performance. You define the extended storage cache as a number of memory segments.

The bufferpools perform the first-level caching, and any extended storage cache is used by the bufferpools as secondary-level caching. Ideally, the bufferpools hold the data that is most frequently accessed, while the extended storage cache hold data that is accessed less frequently.

Pages are copied from the bufferpool to extended storage, when they are selected as victim pages. This copying is required to make space for new pages in the bufferpool. Required pages are copied from extended storage to the buffer pool, if they are not in the bufferpool, but are in extended storage. This copying may incur the cost of connecting to the shared memory segment, but saves the cost of a disk read.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Applications Tab
Metrics	Because an extended storage cache is an extension to a bufferpool, it must always be associated with one or more specific bufferpools. Therefore, only the bufferpools that are declared to take advantage of advantage of the extended storage use it. The CREATE and ALTER BUFFERPOOL statements have the attributes NOT EXTENDED STORAGE and EXTENDED STORAGE that control cache usage. By default neither IBMDEFAULTBP nor any newly created bufferpool will use extended storage. Use these metrics together with the Index I/O on Extended Storage metrics to determine the level of I/O being performed on extended storage. If there is significant activity on this section, you can improve performance by increasing the size of extended storage.

Bufferpools Tab

This section shows the list of bufferpools on the database and the various I/O metrics for each bufferpool. It also shows the rolled up metrics at the database level in the same list. You can select the type of monitoring you intend to perform for the bufferpools from the following:

- Direct I/O: This shows the details of all direct I/O activity for the various bufferpools
- Buffered I/O: This shows the details of all bufferpool I/O activity for the various bufferpools in the database
- Extended Storage: This shows the level of extended storage usage by the various bufferpools that use extended storage
- Total Database Files Closed: This shows the details for the number of files closed for the various bufferpools in the database
- Block/Vectored I/O: This shows the detailed of block and vectored I/O performed by the various bufferpools.

Correction

Depending on the type of details you select, you can view the different types of I/O activity associated with the bufferpools. This section is very useful in isolating the bufferpools, and the I/O areas you need to tune by comparing the I/O performance for all the bufferpools and also the overall I/O performance at the database level.

Block/Vectored I/O

The Block/Vectored I/O section of the Bufferpool Details view displays the following statistics:

Vectored I/O Block I/O

Vectored I/O

The Vectored I/O section gives the metrics for vectored I/O requests and reads.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Bufferpools Tab

Metrics Use this element to determine how often vectored I/O is being done. The number of vectored I/O requests is monitored only during sequential prefetching.

Block I/O

The Block I/O section gives the metrics for block I/O requests and reads. If block-based bufferpool is enabled, this section reports how often block I/O is being done. The number of block I/O requests is monitored only during sequential prefetching when using block-based bufferpools.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Bufferpools Tab

Metrics If block-based bufferpool is enabled and the block I/O activity is very low, or close to the values in the vectored I/O section, consider changing the block size. This state can be an indication of the following: *The extent size of one or more tablespaces bound to the bufferpool is smaller than the block size specified for the bufferpool.Some pages requested in the prefetch request are already present in the page area of the bufferpool.* The prefetcher allows some wasted pages in each bufferpool block, but if too many pages are wasted, then the prefetcher will decide to perform vectored I/O into the page area of the bufferpool. To take full advantage of the sequential prefetch performance improvements that block-based bufferpools provide, it is essential to choose an appropriate value for the block size. This can, however, be difficult because multiple tablespaces with different extent sizes can be bound to the same block-based bufferpool. For optimal performance, it is recommended that you bind tablespaces with the same extent size to a block-based bufferpool with a block size equal to the extent size. Good performance can be achieved when the extent size of the tablespaces are greater than the block size, but not when the extent size is smaller than the block size. For example, if extent size is 2 and block size is 8, vectored I/O would be used instead of block I/O (block I/O would have wasted 6 pages). A reduction of the block size to 2 would solve this problem.

Correction You can alter the block size for a buffer pool using the ALTER BUFFERPOOL ddl statement.

Logging Tab

The Logging tab of the I/O Detail view displays Logging I/O statistics. This section shows the level of read/write activity being performed by the logger. The information presented here is:

Log Reads: The number of log pages read from the disk by the logger per second.

Log Writes: The number of log pages written to the disk by the logger.

Metrics

You can use this page with the operating system level I/O metrics, to quantify the amount of I/O on a device that is attributable to database activity.

Tablespaces Tab

The **Tablespace** tab of the I/O detail view shows the list of tablespaces in the database and the I/O metrics for each tablespace. It also shows the rolled-up metrics at the database level in the same list. You can select the type of monitoring you intend to perform for the tablespaces from the following:

- Direct I/O: Shows the details of all direct I/O activity for the various tablespaces
- Buffered I/O: Shows the details of all bufferpool I/O activity for the various tablespaces in the database
- Extended Storage: Shows the level of extended storage usage by the tablespaces that use extended storage
- Total Database Files Closed: Shows details for the number of files closed for various tablespaces in the database

Correction

Depending on the type of details you select, you can view the different types of I/O activity associated with the tablespaces. This section is very useful in isolating the tablespaces, and the I/O areas you need to tune by comparing the I/O performance for all the tablespaces and also the overall I/O performance at the database level.

Buffered I/O

The following statistics are available on the Buffered I/O section of the Tablespace Details of the I/O view:

Hit Ratios Async I/O Write/Read Ratios

Async I/O

Async read ratio is the percentage of all index and data reads that were performed asynchronously by the database manager prefetchers. Async write ratio is the percentage of all index and data writes that were performed asynchronously by the database manager bufferpool page cleaners.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > I/O > Tablespaces Tab
Metrics	Prefetching refers to the technique whereby the Database Manager can read several pages on the physical reads device simultaneously into a bufferpool in anticipation of an access to these pages. Since each physical read operation is costly, a 50% async read ratio is desirable since it shows that the prefetchers are working well and read waits are being minimized. Page cleaners write changed pages from bufferpool to disk before the space in the bufferpool is needed by a database agent. If the page cleaners are not working well, the problem may manifest itself in two ways: *The database agents will need to synchronously free up space in bufferpools resulting in poorer response time.If the system crashes, the recovery time of the system will be greater because there will be a large number of pages that would not have been committed to disk at time of crash and will require processing of a large number of log records.* An overall async write ratio of 50% is desirable for most transactional database systems. If your database is 'query only' (i.e., updates are rarely or never performed), it is okay to have an async write ratio of 0 to less than 50%. Even in 'query only' databases if the queries create temp tables on the database for intermediate results, a 30-40% async write ratio is desirable.
Correction	num_ioserver database configuration parameter defines the number of prefetchers that are available for the database. To get an optimal async read ratio, you should set the num_ioserver database configuration parameter to 1-2 more than the number of physical storage devices used by the database to ensure that asynchronous I/O is occurring. This parameter should be changed judiciously since having too many prefecthers invariably results in high system I/O because prefecthers can read many more pages than required into the bufferpools. Async write ratio can be tuned using the num_iocleaners database configuration parameter. This parameter specifies the number of asynchronous page cleaners for a database. Increasing the number of page cleaners generally results in a higher async write ratio. The following rules of thumb can be followed when tuning this parameter: *For most transactional systems set this parameter to between 1 and the number of physical storage devices used by the database.For workloads that consists of mainly simple reads from the database, set this parameter to 0.When you define very large bufferpools on your database you may need to increase the num_iocleaners.Monitor the victim page cleaners% metric when tuning the num_iocleaners.*

Space View (DBArtisan - DB2 LUW Performance Analyst)

While DBAs focus on memory settings and tuning SQL, they often forget just how dangerous and insidious storage problems can be. This is not a good mindset because storage headaches can play a major role in wrecking an otherwise well-running database.

Storage problems generally take one of two forms:

- The 'hit-the-wall' variety that can bring things to a complete standstill
- The 'performance vampire' kind that slowly drains the performance of a database over time

Storage problems can quickly lead to performance problems, such as when a transaction log runs out of free space. But storage problems can also silently work behind the scenes to slowly but surely rob a database of its performance.

In addition to a Home page, the Space category of DB2 Performance Analyst includes the following tabbed pages:

Usage tab	Logging tab	Bufferpools tab
Objects tab	Quiesce tab	Rebalance tab
Reorganization tab	Rollforward tab	

Home

The Home page of the Space performance category view displays statistics i the following categories:

Tablespace Analysis Pane	Bottleneck Analysis Pane	SQL Analysis Pane
Storage Analysis Pane	Log Analysis Pane	

Tablespace Analysis Pane

The Tablespace Analysis section includes a list of all of the tablespaces in the database. The following information is presented for each tablespace:

- Tablespace Name: The name of the tablespace
- Type: system managed (SMS) or database managed (DMS) tablespace
- State: This element describes the current state of the tablespace
- Used Space: The space currently in use on tablespace
- Free Space: The space currently free on the tablespace. (This is not applicable to SMS tablespaces)
- Content Type: The type of data that can be stored on the tablespace. It can be one of the following:

- Any (any type of data)
- Long (LOB/ Structured type / Index data)
- System Temporary (work area used by database manager to perform operations like sorts and joins.
- User Temporary (Stores declared global temporary tables)
- Page Size: Size of pages used by the tablespace
- Extent Size: Number of pages that will be written to a container before writing to the next container on the tablespace
- Prefetch Size: Number of pages that will be read from the tablespace by the prefetchers in a single prefetch
- Total Space: Total space (Used + Free) used by the tablespace on the storage device(s). This will be the same as used space for SMS tables
- High Watermark: Highest amount of used space for the tablespace (This is not applicable to SMS tablespaces)

Other statistics appearing in this section include:

Total Tablespaces	The total tablespaces statistic is the total number of tablespaces in the database.
SMS Tablespaces	The SMS tablespaces statistic is the number of system managed (SMS) tablespaces in the database.
DMS Tablespaces	The DMS tablespaces statistic is the number of database managed (DMS) tablespaces in the database.

Bottleneck Analysis Pane

Total Tablespaces SMS Tablespaces DMS Tablespaces

Space shortfalls or inaccessible tablespaces can quickly bring a database to a halt. The space bottleneck analysis section provides an at a glance look at the items that may require your attention. You can drilldown to the details pages to see further details of the various metrics.

The metrics presented here are:

DMS Tablespaces at Utilization >80% Inaccessible Tablespaces

The following statistics, available on this pane, are duplicates of statistics found on the Bottleneck Analysis Pane of the Home View (DBArtisan - DB2 LUW Performance Analyst) page:

Inaccessible Containers Tablespaces in Abnormal State

DMS Tablespaces at Utilization >80%

The DMS tablespaces at utilization greater than 80% metric gives the number of DMS tablespaces in the database where the used space on the tablespace is more than 80% of the total available space.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	This metric is an indicator that some of the tablespaces in your database may be running out of space. You should drill down to the Space metrics to identify the tablespaces that may be at the critical utilization level. In tablespaces where there are little on no inserts after the initial data is loaded, little or no free space may have been left on purpose. In such cases, it is normal to see a high utilization
Correction	If this metric goes beyond the normal operational value for your database, consider creating additional space on the DMS tablespaces that are running out of space. You can do this by performing one or more one of the following tasks: *Clean up unused space by deleting the tables/indexes that are no longer required.Resize or extend the existing containers in the tablespaces.Add new containers to the tablespaces.*

Inaccessible Tablespaces

The inaccessible tablespaces statistic is the number of tablespaces that are offline.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Sometimes a problem (such as inaccessible containers) will cause a tablespace to go offline. Even when the problem is corrected, such tablespaces are not accessible by any application and need to be brought online explicitly.
Correction	From Tablespace Analysis section, identify the tablespaces whose state is set to Offline Drill down to space details to make sure all of the containers for these tablespaces are accessible. If they are not, first fix the problem with the containers Disconnect all applications from the database (you can use DBArtisan DB Monitor). Reconnect to the database. Use the ALTER TABLESPACE .. SWITCH ONLINE command to bring the tablespace back online. *If the tablespace can be brought up successfully after issuing the command, or if the tablespace was not in the OFFLINE state to begin with, DB2 returns an SQLCODE of 0.If the tablespace cannot be brought up successfully because there are still problems with one or more of the containers, DB2 returns an SQLCODE of --293. You can force the database to restart by using the RESTART ... DROP PENDING TABLESPACE, but have to drop any faulty table spaces afterward.*

SQL Analysis Pane

The metrics presented here are:

DDL Statements Executed Insert/Updates/Deletes Statements Executed Select Statements Executed

Total SQL Executed

DDL Statements Executed

This metric indicates the number of SQL Data Definition Language (DDL) statements that have been executed.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Home > SQL Analysis Pane
Metrics	You can use this element to determine the level of database activity at the application or database level. DDL statements are expensive to run because of their impact on the system catalog tables. As a result, if the value of this element is high, you should determine the cause, and possibly restrict this activity from being performed. DDL statements can also impact: *The catalog cache, by invalidating table descriptor information and authorization information that are stored there and causing additional system overhead to retrieve the information from the system catalogsThe package cache, by invalidating sections that are stored there and causing additional system overhead due to section recompilation.*

Insert/Updates/Deletes Statements Executed

This is the number of SQL UPDATE, INSERT, and DELETE statements that were executed.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Home > SQL Analysis Pane
Metrics	You can use this element to determine the level of database activity at the application or database level. This information can be useful for analyzing application activity and throughput.

Select Statements Executed

This statistic is the number of SQL SELECT statements that were executed.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Home > SQL Analysis Pane
Metrics	You can use this element to determine the level of database activity at the application or database level. This information can be useful for analyzing application activity and throughput.

Total SQL Executed

You can use this element to calculate the total number of SQL statements at the database level.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Home > SQL Analysis Pane
Metrics	This element can also help you in determine reasons for poor performance. Failed statements mean time wasted by the database manager, and as a result, lower throughput for the database. You can use this element in conjunction with the other SQL metrics to understand the portion of SQL activity that can be attributed to various statement types.

Storage Analysis Pane

While DBAs focus on memory settings and tuning SQL, they frequently forget how dangerous and insidious storage problems can be. This is not a good mindset because storage headaches can play a major role in wrecking an otherwise well-running database. Storage problems generally take one of two forms:

- The 'hit-the-wall' variety that can bring things to a complete standstill.
- The 'performance vampire' kind that slowly drains the performance of a database over time.

Storage problems have the capability to bring the curtain down on a database very quickly, as in the case of a transaction log running out of free space. But storage problems can also silently work behind the scenes to slowly, but surely rob a database of its performance. For example, a hub table in a busy database might be accessed very quickly when an application is first given life, but over time, if it develops a heavy forwarded row problem, it can cause things to run very differently.

The Storage Analysis section of the Performance Analyst Home page displays the total used and free space of all DMS tablespaces in the database. It also provides the total and available log space for the database.

The following ratios succinctly communicate the general overall performance levels of the datasource:

Log Space Last Backup

The following statistics, available on this pane, duplicate statistics available on the Storage Analysis Pane of the Home page:

DMS Space Total SMS Space Used

Log Analysis Pane

The Log Analysis section gives you at-a-glance information about all your log configuration parameters and the high watermarks for the log space usage. You can use this information to quickly determine the logging mechanism in place for your database and if the high watermarks are getting too close to the maximum allocations for the log space.

The metrics presented here are:

Log Configuration High Watermarks

Log Configuration

The log configuration metric gives a list of space related log configuration parameters for your database. The values of these parameters determine how the logger uses storage space on the database:

- Block on Log Disk Full(blk_log_dsk_ful): this configuration parameter can be set to prevent disk full errors from being generated when DB2 cannot create a new log file in the active log path. Instead, DB2 will attempt to create the log file every five minutes until it succeeds. After each attempt, DB2 will write a message to the administration notification log. The only way to confirm that your application is hanging because of a log disk full condition is to monitor the administration notification log. Until the log file is successfully created, any user application that attempts to update table data is not able to commit transactions. Read-only queries may not be directly affected; however, if a query needs to access data that is locked by an update request or a data page that is fixed in the bufferpool by the updating application, read-only queries also appear to hang.

Setting blk_log_dsk_ful to YES causes applications to hang when DB2 encounters a log disk full error. You are then able to resolve the error and the transaction can continue. A disk full situation can be resolved by moving old log files to another file system, or by increasing the size of the file system so that hanging applications can complete.

If blk_log_dsk_ful is set to NO, a transaction that receives a log disk full error fails and is rolled back. In some cases, the database comes down if a transaction causes a log disk full error.

- Log File Size(logfilsiz): specifies the size of each configured log, in number of 4-KB pages.

There is a 256-GB logical limit on the total active log space that you can configure. This limit is the result of the upper limit on logfilsiz, which is 262144, and the upper limit on (logprimary + logsecond), which is 256.

The size of the log file has a direct bearing on performance. There is a performance cost for switching from one log to another. So, from a pure performance perspective, the larger the log file, size the better. This parameter also indicates the log file size for archiving. In this case, a larger log file is size it not necessarily better, since a larger log file size may increase the chance of failure or cause a delay in log shipping scenarios. When considering active log space, it may be better to have a larger number of smaller log files. For example, if there are two very large log files and a transaction starts close to the end of one log file, only half of the log space remains available.

Every time a database is deactivated (all connections to the database are terminated), the log file that is currently being written is truncated. So, if a database is frequently being deactivated, it is better not to choose a large log file size because DB2 will create a large file only to have it truncated. You can use the ACTIVATE DATABASE command to avoid this cost, and having the bufferpool primed will also help with performance.

Assuming that you have an application that keeps the database open to minimize processing time when opening the database, the log file size should be determined by the amount of time it takes to make offline archived log copies.

Minimizing log file loss is also an important consideration when setting the log size. Archiving takes an entire log. If you use a single large log, you increase the time between archiving. If the medium containing the log fails, some transaction information will probably be lost. Decreasing the log size increases the frequency of archiving but can reduce the amount of information loss in case of a media failure since the smaller logs before the one lost can be used.

- Log Retain(logretain): if logretain is set to RECOVERY, archived logs are kept in the database log path directory and the database is considered to be recoverable, meaning that rollforward recovery is enabled.

Note: The default value for the logretain database configuration parameter does not support rollforward recovery and must be changed if you are going to use it.

- User Exit(userexit): causes the database manager to call a user exit program for archiving and retrieving logs. The log files are archived in a location that is different from the active log path. If userexit is set to ON, rollforward recovery is enabled.

Note: The userexit database configuration parameter must be enabled to set logsecond parameter to-1

- Primary Logs(logprimary): specifies the number of primary logs of size logfilsz that will be created.

A primary log, whether empty or full, requires the same amount of disk space. Thus, if you configure more logs than you need, you use disk space unnecessarily. If you configure too few logs, you can encounter a log-full condition. As you select the number of logs to configure, you must consider the size to make each log and whether your application can handle a log-full condition. The total log file size limit on active log space is 256 GB.

If you are enabling an existing database for rollforward recovery, change the number of primary logs to the sum of the number of primary and secondary logs, plus 1. Additional information is logged for LONG VARCHAR and LOB fields in a database enabled for rollforward recovery.

- Secondary Logs(logsecond): specifies the number of secondary log files that are created and used for recovery, if needed.

If the primary log files become full, secondary log files (of size logfilsiz) are allocated, one at a time as needed, up to the maximum specified by this parameter. If this parameter is set to -1, the database is configured with infinite active log space. There is no limit on the size or number of in-flight transactions running on the database.

Note: If this parameter is set to -1, crash recovery time may be increased since DB2 may need to retrieve archived log files.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Log Analysis Pane

High Watermarks

The high watermarks statistic shows the maximum total and secondary log space used by the database.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Log Analysis Pane
Metrics	The total log space used high watermark in conjunction with the total log space allocated, helps you determine if applications have come close to using the total log space available to your database. The total secondary log space used high watermark in conjunction with total log space used high watermark helps you determine your current dependency on secondary logs.
Correction	High values of either high watermarks can indicate that you either need to increase the log space on the database or that the applications using the database need to perform more frequent COMMITS. You can increase the overall log space available by adjusting one or more of the following database configuration parameters: **logfilsiz**, **logprimary**, **logsecond**, and **logretain**.

Usage tab

The Space Detail Usage tab section shows you the list of tablespaces for the database. The following information is presented for each tablespace:

Tablespace Details Container Details

Tablespace Details

ID: Internal DB2 Tablespace identifier.

Name: Tablespace Name

Type: Tablespace Type – SMS (System Managed Space) or DMS (Database Managed Space)

Content Type: Shows the type of data that can be stored on the tablespace. The valid types are:

- Any meaning can store any type of data
- Sys Temp meaning, used internally by DB2 to store intermediate result data
- User Temp meaning, can be used to create temporary tables
- Long meaning, can be used to store LOB/LONG/Index data

State: Can be one or more of the following:

- Normal (see the definition SQLB_NORMAL in sqlutil.h)
- Quiesced: SHARE
- Quiesced: UPDATE
- Quiesced: EXCLUSIVE
- Load pending
- Delete pending
- Backup pending

- Roll forward in progress
- Roll forward pending
- Restore pending
- Recovery pending (not used)
- Disable pending
- Reorg in progress
- Backup in progress
- Storage must be defined
- Restore in progress
- Offline and not accessible
- Drop pending
- Storage may be defined
- Storage Definition is in 'final' state
- Storage Definition was changed prior to rollforward
- DMS rebalancer is active
- TBS deletion in progress
- TBS creation in progress

Page Size: The size of tablespace pages

Extent Size: Number of pages that will be written to a container before writing to the next container on the tablespace

Prefetch Size: The maximum number of pages the prefetcher gets from the disk at a time

Usable Space: The total space in a table space minus overhead space. (Note: For SMS tablespaces, this element will have the same value as Used Space).

Used Space: The total space in the tablespace that is currently used in the tablespace (including overhead)

Free Space: The total space that is free to be used for storage of data. This element is not applicable for SMS tablespace. free space for an SMS tablespace is the OS file system free space for the SMS containers.

Total Space: The total space used by the tablespace

Pending Free Pages: The number of pages in the tablespace that would become free if all pending transactions are committed or rolled back and new space is requested for an object. (Applies only to DMS tablespaces)

1. Containers: The number of containers in the tablespace

Usable Pages: The total pages in a table space minus overhead pages. (Note: For SMS tablespaces, this element will have the same value as Used Pages).

Used Pages: The total pages in the tablespace that is currently used in the tablespace (including overhead)

Free Pages: The total pages that are free to be used for storage of data. This element is not applicable for SMS tablespace. Free pages for an SMS tablespace are dependent on the OS file system free space for the SMS containers.

Total Pages: The total pages used by the tablespace

Page HWM: For DMS, this element represents the page number of the first free extent following the last allocated extent of a table space. Note that this is not really a "high water mark", but rather a "current water mark", since the value can decrease. For SMS, this is not applicable.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Usage tabIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Rebalance tab

Container Details

This section shows the list of containers for the selected tablespace. The information presented here includes:

ID: A value that uniquely identifies the container within the tablespace

Name: The name of the container. Typically this is the full path of the file/directory/device where the container exists.

Type: The type of container. An SMS Container will be a directory. A DMS Containers will be a raw device/file/stripped disk/ or stripped file. Together with Container name, and partition, this metric identifies the physical location of the container.

DB Partition: The database partition where the container exists

Usage Pages: Usable pages in the container (Applicable to DMS Tablespaces only).

Total Pages: Total pages in the container

Stripe Set: The stripe set that a container belongs to.

Accessible: This element describes if a container is accessible or not

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Usage tab
Metrics	This section can be used to get a better understanding of the physical layout of a tablespace. Also, it lets you quickly identify if any containers are inaccessible.

Logging tab

The Logging tab of the Space Detail view displays Logging Space statistics:

Log Read/Writes Indoubt Transactions Log Usage

Secondary Logs

Log Read/Writes

This section shows the level of read/write activity being performed by the logger. The information presented here is:

Log Reads: The number of log pages read from the disk by the logger per second.

Log Writes: The number of log pages written to the disk by the logger.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Logging tab
Metrics	You can use this page with the operating system level I/O metrics, to quantify the amount of I/O on a device that is attributable to database activity.

Indoubt Transactions

This section shows the number of outstanding indoubt transactions in the database.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Logging tab
Metrics	Indoubt transactions hold log space for uncommitted transactions, which can cause the logs to become full. When the logs are full, further transactions cannot be completed. The resolution of this problem involves a manual process of heuristically resolving the indoubt transactions. This monitor element provides a count of the number of currently outstanding indoubt transactions that must be heuristically resolved.

Log Usage

This section shows the following log usage details:

Total Used Space: Total space used by log files

Total Free Space: Total space still available to the logger

Total Used Space Highwatermark: The maximum amount of total log space used.

Secondary Space Used High Watermark: The maximum amount of secondary log space used.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Logging tab
Metrics	Use these metrics to check your current log space usage and the highest amount of space used by the logger. If the usage is high and the high watermarks are also near the total available space, you may need to increase the amount of log space or end/commit/rollback the oldest transaction.
Correction	You can use the following database configuration parameters to change the amount of log space available: **logfilsiz**, **logprimary**, and **logsecond.** You can end (force) a connection with a long running transaction using the DBArtisan database monitor.

Secondary Logs

This section shows the secondary logs allocations by the database over the monitoring period.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Logging tab
Metrics	When the primary log files become full, the secondary log files are allocated one at a time as needed, up to a maximum number as controlled by the logsecond database configuration parameter. Secondary log files are useful in preventing occasional log fill ups but they may not be as efficient as primary log and they also increase the recovery time. If you see a constant reliance on secondary log files, it may indicate infrequent commits by database applications or insufficient primary log space.
Correction	First examine the database applications to see if secondary log files are due to long running transactions and whether these transactions can be shortened. If the transactions cannot be shortened or you still see a very frequent use of secondary logs after transaction tuning, increase the primary log size (logprimary,logfilsiz Database configuration parameters) to reduce the dependence on secondary logs. If there is an occasional long running transaction, and you see your see transaction log full (SQL0964C) error messages, you can either increase the number of secondary log files (logsecond database configuration parameter) or set the number to -1 (no limit on number of secondary log files).

Bufferpools tab

The Bufferpools tab of the Space Detail View shows the list of tablespaces, with the current bufferpool associated with each tablespace and also the bufferpool used by the database at next startup

Objects tab

The Objects tab of the Space Detail View shows the following statistics:

Tablespace Object Summary Tablespace Object Details

Tablespace Object Summary

This section shows information about how tablespaces are being used on the database. The information presented here includes:

Tablespace: Name of the tablespace

Table Count: Number of tables on the tablespace

Page Size: Tablespace page size

Index Count: Number of indexes on the tablespace

Table Space: Space used by tables on the tablespace

Index Space: Space used by indexes on the tablespace

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Bufferpools tab

Tablespace Object Details

This section shows the details of the objects on the tablespaces shown in the Tablespace Object summary section. The user can toggle between the list of tables and indexes for the tablespace.

The information presented here for tables includes:

Table Schema: Schema name of the table

Table Name: Name of the tables

Avg Row Length: Approximate length of the table row.

1. Rows: Table Cardinality
2. Pages Allocated: Number of pages allocated to the table

Table Size: Total space used by the rows on the table

Available: Total available space on the table

Total: Total space used by the table pages

Note: The information presented here is based on the table/index statistics and if the statistics haven’t been collected /updated, this information may be out of date.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Bufferpools tab

Quiesce tab

The Quiesce tab of the Space Detail view shows displays the following information:

Quiesced Tablespaces Quiescer Details

Quiesced Tablespaces

This section shows a list of quiesced tablespaces. A quiesce is a persistent lock. Its benefit is that it persists across transaction failures, connection failures, and even across system failures (such as power failure, or reboot).

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Quiesce tab
Metrics	A quiesce is owned by a connection. If the connection is lost, the quiesce remains, but it has no owner, and is called a phantom quiesce. For example, if a power outage caused a load operation to be interrupted during the delete phase, the tablespaces for the loaded table would be left in delete pending, quiesce exclusive state. With a database restart, this quiesce would be an unowned (or phantom) quiesce. The removal of a phantom quiesce requires a connection with the same user ID used when the quiesce mode was set. This section is useful in identifying the tablespaces that are in a quiesced state.
Correction	To remove a phantom quiesce: *Connect to the database from DBArtisan with the same user ID used when the quiesce mode was set.Re-quiesce the table space using the current quiesce state from DBArtisan.Reset the quicesce from DBartisan.*

Quiescer Details

This section shows the list of quiescers for the selected quiesced table.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Quiesce tab
Metrics	Any quiescers whose agent ids are not listed under Users Detail >Attributes are phantom.
Correction	To remove a phantom quiesce: *Connect to the database from DBArtisan with the same user ID used when the quiesce mode was set.Re-quiesce the table space using the current quiesce state from DBArtisan.Reset the quicesce from DBartisan.*

Rebalance tab

The Rebalance tab of the Space Detail page displays the following information:

- Rebalance Status

The following statistic, available on this pane, are duplicates of statistics on the **Space View (DBArtisan - DB2 LUW Performance Analyst)**Usage tab.

Tablespace Details

Rebalance Status

This section shows a graph of the percentage of extents that have been processed and the percentage that remains incomplete for the selected tablespace.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Rebalance tab

Reorganization tab

The Reorganization tab of the Space Details View displays the following statistics:

Table Details Reorganization Status Reorganization Attributes

Table Details

This section shows the details of all the tables on which reorganization has been attempted since the start of monitoring. It also shows the status of tables that are presently undergoing reorganization. The information presented in this includes:

Table Schema: Table Schema of the table on which reorganization is being/was performed

Table Name: Table Name of the table on which reorganization is being/was performed

Tablespace: Tablespace of the table on which reorganization is being/was performed

Start: Start time of the reorganization

End: End Time of the reorganization

Phase Start: Start time of a phase of reorganization

Current Counter: Percentage of reorganization operation completed (0 if the reorganization status is complete)

Max Phase: The maximum number of reorganization phases that will occur during reorganization processing. This applies to classic (offline) reorganizations only.

Status: The status of an in-place (online) table reorganization. This is not applicable to classic (offline) table reorganizations. An in-place table reorganization can be in one of the following states:

- Started/Resumed
- Paused
- Stopped

- Completed
- Truncate

Completion: Table reorganize success indicator.

Phase: Table reorganize phase. This applies to classic (offline) table reorganizations only).

For classic table reorganization, the following phases are possible:

- Sort
- Build
- Replace
- Index Recreate

% Complete: Shows the completion status of the reorganization jobs

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Reorganization tab

Reorganization Status

Shows a graphical status view of the currently selected reorganization.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Reorganization tab

Reorganization Attributes

This section shows the reorganization attributes of the currently selected reorganization. The attributes include:

- Allow Write Access
- Allow Read Access
- Allow No Access
- Recluster Via Index Scan
- Reorg Long Field LOB Data
- No Table Truncation
- Reclustering
- Reclaiming
- Inplace Table Reorg
- Table Reorg

- Recluster Via Table Scan
- Reorg Data Only.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Space View (DBArtisan - DB2 LUW Performance Analyst)** > Reorganization tab

Rollforward tab

Recovering database changes is time-consuming. You can use the database system monitor to monitor the progression of a recovery. This section provides the following elements to help you monitor the status of rollforwards:

Timestamp: Timestamp of the log being processed. This is an indicator of the data changes that will be recovered

Tablespace: Tablespace being rolled forward. This element identifies the tablespaces involved in the rollforward

Type: An indicator of whether the recovery is happening at a database or tablespace level

Log#: The log being currently processed. This identifies the log involved in the rollforward.

Status: The status of rollforward recovery. This metric indicates if the recovery is in an undo (rollback) or redo (rollforward) phase.

1. Tablespaces: Number of tablespaces involved in the rollforward

Node: Indicates the database partition number for the rollforward operation

Users View (DBArtisan - DB2 LUW Performance Analyst)

The database activity can be examined using key performance metrics and wait/bottleneck-based analysis. In addition to a Home page, the Users category of DB2 Performance Analyst includes the following tabbed pages:

Top Sessions tab	Cursors tab	SQL Activity tab
Attributes tab	Lock Waits tab	Locks tab

Home

The Home page of the Users performance category view displays the vital DB2 user statistics in the following categories:

Key User Analysis Pane	Bottleneck Analysis Pane	SQL Analysis Pane
Connection Analysis Pane		

Key User Analysis Pane

Database performance analysts typically use one of two methods for examining the performance levels of a database - ratio-based or wait/bottleneck-based. Ratio-based analysis involves examining a number of key database ratios that can be used to indicate how well a database is running. Performance ratios serve as very good roll-up mechanisms for busy DBAs to use for at-a-glance performance analysis. Many DBAs have large database farms to contend with and cannot spend time checking detailed wait-based analysis outputs for each and every database they oversee. Succinctly presented performance ratios can assist in such situations by giving DBAs a few solid indicators that can be quickly scanned to see if any database needs immediate attention.

While there are certainly many opinions as to what rules to follow, there are some standards that should always be adhered to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that a DBA will examine come from the DB2 Snapshot Monitor. The snapshot monitor returns metrics for the Instance and particular databases since the start of monitoring. Some metrics are cumulative (e.g., counters) and others are instantaneous (e.g., gauges).

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each SQL Server's individual 'personality' with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead the DBA to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following ratios are used on the Performance Analyst User page to succinctly communicate the general overall performance levels of the monitored database:

Efficiency

Transactions/sec Static SQL/sec Dynamic SQL/sec

Activity

Location Applications Executing Connections High Watermark

Transactions/sec

Transactions/sec is the number of transactions (units of work) completed per second on the database.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key User Analysis Pane
Metrics	A small rate of transactional activity on the database can indicate that applications are not doing frequent commits, which may lead to logging and concurrency problems.
Correction	Drill down to the Users Detail>SQL Activity page to check which applications are running their transactions for long periods of time

Static SQL/sec

The static SQL/sec metric is the number of static SQL statement executions attempt on the database each second.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key User Analysis Pane

Dynamic SQL/sec

This metric is the number of dynamic SQL statement executions being attempted on the database each second.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key User Analysis Pane

Applications Idle

The idle applications metric indicates the number of applications that are currently connected to the database for which the database manager is not executing any requests

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key User Analysis Pane

Metrics This can help you understand the level of activity within a database and the amount of system resource being used.

Applications Executing

The executing applications statistic indicates the number of applications for which the database manager is currently executing requests.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key User Analysis Pane

Metrics You can use this element to understand how many of the database manager agent tokens are being used by applications connected to this database.

Connections High Watermark

The connections high watermark is the highest number of simultaneous connections to the database since the database was activated.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key User Analysis Pane

Metrics You can use this element to evaluate the setting of the maxappls configuration parameter. If the value of this element is the same as the maxappls parameter, it is likely that some database connection requests were rejected, since maxappls limits the number of database connections allowed.

Correction Increase the value of the maxappls parameter to allow more connections.

Bottleneck Analysis Pane

When a DB2 database is active, the applications accessing the database can either successfully access the database resources, or they can wait for resources to become available. An application that is waiting can indicate normal database activity, or it can be an indicator that a database bottleneck exists. This is where wait-based or bottleneck analysis comes into play. A database administrator can use this form of performance analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Performance Analyst Bottleneck analysis section helps a DBA isolate potential problem areas where there are waits, resource overloads or potential failures. If concurrency issues or heavy table scan activity has been dragging a database's performance down, a DBA can use bottleneck analysis to isolate the root cause. Once one or more bottlenecks have been isolated as possible areas of interest, the DBA can drill down and examine details on why the applications or resources are causing the problem. The Performance Analyst Users page identifies the top system and application bottlenecks that may be currently a cause of lessened performance.

The following bottleneck indicators are used on the Performance Analyst Home page to succinctly communicate the general overall performance level of the monitored database:

Top Bottlenecked User Processes	Automatic Rebinds	Internal Rollbacks
Waits	Deadlocks	Timeouts

The following statistics, appearing on this page, duplicate statistics available on the Bottleneck Analysis Pane of the Home View (DBArtisan - DB2 LUW Performance Analyst) page:

Lock Escalations Sort Overflows Failed SQL

The following statistic, available on this pane, duplicates a statistic on the SQL Analysis Pane of the Home View (DBArtisan - DB2 LUW Performance Analyst) page:

Failed SQL

Top Bottlenecked User Processes

The top bottlenecked user processes grid shows the top bottlenecked applications in the system. These applications are either performing the top I/O operations or have idled the longest to obtain locks.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane

Automatic Rebinds

The automatic rebinds statistic is the number of automatic rebinds (or recompiles) that have been attempted.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane

Metrics Automatic rebinds are the internal binds the system performs when an package has been invalidated. The rebind is performed the first time the database manager needs to execute an SQL statement from the package. For example, packages are invalidated when you: *Drop an object, such as a table, view, or index, on which the plan dependsAdd or drop a foreign keyRevoke object privileges on which the plan depends.* Since automatic rebinds can have a significant impact on performance, they should be minimized where possible.

Internal Rollbacks

The internal rollbacks statistic is the total number of rollbacks initiated internally by the database manager.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	An internal rollback occurs when any of the following cannot complete successfully: A reorga*nizationAn importA bind or pre-compileAn application ends as a result of a deadlock situation or lock timeout situationAn applica*tion ends without executing an explicit commit or rollback statement (on Windows).

Waits

The lock waits metric is the total number of times that applications or connections waited for locks within the database.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane

Deadlocks

The deadlocks statistic is the total number of deadlocks that have occurred since this instance of Performance Analyst started monitoring the database.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	If a large number of deadlocks are detected, it can indicate that applications are experiencing lock contention problems. Deadlocks are usually caused by one of the following situations: *Lock escalations on the databaseCatalog tables locked for Repeatable ReadApplications are using inappropriate isolation levels at bind timeApplications are obtaining the same locks in a different orderApplications are locking tables explicitly where row level locks are sufficient.*
Correction	You may be able to modify the applications causing lock contentions for better concurrency. To identify the applications that may be causing contentions, go to the Lock Waits tab of the Users detail page and review the Application Details section.

Timeouts

The lock timeouts metric identifies the number of times that a request to lock an object timed out without being granted.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	If the number of lock timeouts becomes excessive when compared to the acceptable range for your database, it can indicate that an application is holding locks for long durations. It can also indicate that the amount of time an application waits for a lock before timing out is too short. If you have too few lock timeouts and the average lock wait time is too high, it can indicate that the lock timeout configuration parameter is set to an excessively high value
Correction	First you should examine the lock activity at the application level (Users>Lock, Lock Waits drilldown) to identify any particular application that is causing excessive lock contentions. If so, you can tune the application to provide better concurrency. If lock timeouts are excessive, and average lock wait times are very short, you can increase the locktimeout database configuration parameter to make the applications wait longer before timing out.

SQL Analysis Pane

Most of database's I/O performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag down an otherwise well-configured server in terms of user response times.

This section shows a snapshot of row level SQL operations that have been performed on the database since you started monitoring. This gives you an insight into the row level of read/write activity that is currently occurring within the database and comparing this to the prefetcher, and cleaner activity allows you to relate the two.

The following statistics are used on the Performance Analyst for DB2 User home page to succinctly communicate the general overall performance levels of I/O. They duplicate statistics appearing on the SQL Analysis Pane of the I/OHome page.:

Rows Deleted Rows Inserted Rows Selected

Rows Updated Rows Read

Connection Analysis Pane

The Connection Analysis section displays the level of application connect/disconnect activity occurring on the database.

The following statistics are used on the Performance Analyst for DB2 User home page to succinctly communicate general overall connectivity:

Current Connections Disconnects Connections Since Database Activation

High Watermark for Concurrent Connections

Current Connections

The current connections metric is the number of applications currently connected to the database.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Connection Analysis Pane
Metrics	You can use this metric to help you get an overview of the level of database activity and the amount of system resources in use.
Correction	This metric can help you adjust the setting of the maxappls and max_coordagents configuration parameters. For example, if this value is always the same as maxappls, you consider increasing the value of maxappls.

Disconnects

This metric shows the number of disconnects from the database since the first connect, activate, or last reset.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Connection Analysis Pane

Connections Since Database Activation

The connections since database activation statistic indicates the number of connections to the database since the first connect, activate, or last reset.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Connection Analysis Pane
Metrics	Use this metric in conjunction with the database connection time and the database start time to see if the application connection frequency is low. When there are no connections to db2, by default the database is automatically deactivated until the next connection. The connection that reactivates the database encounters slow connection time because it needs to wait for the database to reinitialize. If very few applications are connecting to the database, some connections can be slow because of this
Correction	If there are short periods when your database has no applications connected to it, activate the database explicitly using the ACTIVATE DATABASE command. This significantly improves the connection performance for the database.

High Watermark for Concurrent Connections

The connections high watermark is the highest number of simultaneous connections to the database since the database was activated.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Connection Analysis Pane
Metrics	You can use this element to evaluate the setting of the maxappls configuration parameter. If the value of this element is the same as the maxappls parameter, it is likely that some database connection requests were rejected, since maxappls limits the number of database connections allowed.
Correction	Increase the value of the maxappls parameter to allow more connections.

Workload Analysis Pane

When your phone starts ringing with complaints of performance slowdowns, one of the first things you should get a handle on is:

- Who is currently connected to the database?
- What resources are they using?
- What are they currently executing?

The Workload Analysis section of the Performance Analyst Users page provides insight into the leading resource hogs of a server. Drill-downs are available so you can easily get detailed information into what each leading application is currently involved with. The Workload Analysis section presents the following metrics:

Top Lock Escalations	The top lock escalation displays the application with largest number of lock escalations.
Top CPU Process	The top CPU process displays the application with the highest CPU usage.
Top I/O Process	The top I/O process displays the application with the highest I/O read/write time.
Oldest Transaction	The oldest transaction displays the application with the longest running current transaction.

Top Sessions tab

Very often, one renegade client application brings a healthy database down by hogging most of the available resources. This page shows the top resource hogs in each of the resource usage areas to help you quickly identify the worst offenders in each area. You see:

Top I/O Process Top Memory Process Top CPU Process

Top Sessions

Top I/O Process

This section shows the application with the top IO time and compares it with the IO time of all other applications.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Top Sessions tab
Metrics	The metrics are calculated by measuring the total time spent by each application doing reads and writes (buffered and non-buffered).

Top Memory Process

This section shows the application with the largest memory pool usage aggregate. It also shows the total memory pool usage for all other applications.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Top Sessions tab
Metrics	This metric is calculated by aggregating all of the memory pools for each application and showing the top application. This section is useful in determining if the top application is using a lot more resources

Top CPU Process

This section shows the application with the top CPU time usage and compares it with the CPU usage of all other applications.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Top Sessions tab
Metrics	Metrics are calculated using the total user and system CPU time for each application connected to the database.

Top Sessions

Shows a list of all applications connected to the database and the resource usage for these applications. The information presented for each application includes:

Auth ID: The authorization ID of the user who invoked the application that is being monitored

Name: The name of the application running at the client as known to the database manager

Handle: A system-wide unique ID for the application. On a single-partitioned database, it consists of a 16 bit counter. On a multi-partitioned database, it consists of the coordinating partition number concatenated with a 16 bit counter. In addition, this identifier will be the same on every partition where the application may make a secondary connection.

ID: This identifier is generated when the application connects to the database at the database manager

Client PID: The process ID of the client application that made the connection to the database

I/O Time: The total time spent by application in performing buffered and non-buffered reads and writes

Memory Usage: Total memory pool usage for the application

CPU Time: The total user + system cpu time used by the application agents

1. Agents: Number of agents for the application

Status: Current Status of the application. Values for this field are:

- Database Connect Pending: The application has initiated a database connection but the request has not yet completed.
- Database Connect Completed: The application has initiated a database connection and the request has completed.
- Unit of Work Executing: The database manager is executing requests on behalf of the unit of work.
- Unit of Work Waiting: The database manager is waiting on behalf of the unit of work in the application. This status typically means that the system is executing in the application's code.
- Lock Wait: The unit of work is waiting for a lock. After the lock is granted, the status is restored to its previous value.
- Commit Active: The unit of work is committing its database changes.
- Rollback Active: The unit of work is rolling back its database changes.
- Recompiling: The database manager is recompiling (that is, rebinding) a plan on behalf of the application.
- Compiling: The database manager is compiling an SQL statement or precompiling a plan on behalf of the application.
- Request Interrupted: An interrupt of a request is in progress.
- Database Disconnect Pending: The application has initiated a database disconnect but the command has not yet completed executing. The application may not have explicitly executed the database disconnect command. The database manager will disconnect from a database if the application ends without disconnecting.
- Decoupled from Agent: The application has been decoupled from an agent.
- Transaction Prepared: The unit of work is part of a global transaction that has entered the prepared phase of the two-phase commit protocol.
- Transaction Heuristically Committed: The unit of work is part of a global transaction that has been heuristically committed.
- Transaction Heuristically Rolled Back: The unit of work is part of a global transaction that has been heuristically rolled-back.
- Transaction Ended: The unit of work is part of a global transaction that has ended but has not yet entered the prepared phase of the two-phase commit protocol.

- Creating Database: The agent has initiated a request to create a database and that request has not yet completed.
- Restarting Database: The application is restarting a database in order to perform crash recovery.
- Restoring Database: The application is restoring a backup image to the database.
- Backing Up Database: The application is performing a backup of the database.
- Data Fast Load: The application is performing a "fast load" of data into the database.
- Data Fast Unload: The application is performing a "fast unload" of data from the database.
- Wait to Disable Tablespace: The application has detected an I/O error and is attempting to disable a particular table space. The application has to wait for all other active transactions on the table space to complete before it can disable the table space.
- Quiescing a Tablespace: The application is performing a quiesce table space request.
- Wait for Remote Partition: The application is waiting for a response from a remote partition in a partitioned database instance.
- Remote Request Pending: The application is waiting for results from a federated datasource.

Last UOW Start Time: The date and time that the unit of work first required database resource

Last UOW End Time: The date and time that the most recent unit of work completed, which occurs when database changes are committed or rolled back

Current UOW Elapsed Time: The elapsed execution time of the most recently completed unit of work.

Login Timestamp: The date and time that a connection request was granted.

Inbound Comm Address: This is the communication address of the client.

Client Protocol: The communication protocol that the client application is using to communicate with the server.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Top Sessions tab

Cursors tab

The Cursors tab of the Users Detail view displays the Application SQL Cursor Activity. It details the SQL cursors open for each application. The information presented here includes:

Open Cursors: The number of cursors (local and remote) currently open for an application.

Blocking Cursors: The number of blocking cursors (local and remote) currently open for the application

Block Requests Accepted: The percentage of all blocking I/O requests that were accepted by the database

Block Requests Rejected: The percentage of all blocking I/O requests that were rejected by the database

Metrics

The number of blocking cursors for an application should be close to the number of open cursors. If this is not the case the application performance may be affected. If there are many cursors blocking data, the communication heap may become full. When this happens, instead of returning an error, the database stops allocating I/O blocks for blocking cursors.

Correction

If the number of blocking cursors in an application is not close to the number of open cursors, you may be able to improve application performance by tweaking the precompile options for record blocking for the treatment of ambiguous cursors, and by redefining the cursors with FOR FETCH ONLY clause where possible

If a large number of cursors were unable to perform blocking, you may be able to improve performance by increasing the query_heap database manager configuration parameter.

SQL Activity tab

The SQL Activity tab of the Users Detail view displays the following statistics:

Application List

UOW Details

SQL Executed

Binds and Precompiles Executed

SQL Activity

Application List

This section shows a list of applications connected to the database. You can select an application from this list and see details for the SQL activity for this application.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > SQL Activity tab

SQL Executed

This section shows the details for the various types of SQL statements executed by the particular application. It also gives you the total application throughput. The information presented here includes:

Selects: Percentage of statements that are select statements

Updates/Inserts/Deletes: Percentage of statements that modify/write data on the database

DDLs: Percentage of CREATE/DROP/ALTER statements

Others: All other types of statements (Fetch, Open cursor etc)

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > SQL Activity tab
Metrics	Various types of statements have varying impact on the performance and concurrency of the database. For example - DDL statements can cause catalog and package cache invalidations, Insert/update/delete statements usually obtain locks and perform log IO. This section can help you determine the contributing statement types to the activity for an application.

SQL Activity

This section gives details on the sql activity for the selected application. It allows you to quick compare the number of static dynamic and failed sql statements with the overall application throughput. The metrics presented here include:

Static SQL: The number of static SQL statements that the application tried to execute

Dynamic SQL: The number of dynamic SQL statements that the application tried to execute

Failed SQL: The number of SQL statements (static and dynamic) that the application tried to execute but failed

SQL Throughput: Number of SQL statements (dynamic and static) that the application executed successfully

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > SQL Activity tab
Metrics	These elements may help you determine the nature of the application. Failed SQL statements metric may also help identify the cause of poor performance on the database since failed statements mean time wasted by the database manager.

UOW Details

Shows the units of work being completed by the application over the refresh time interval.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > SQL Activity tab
Metrics	A low rate of unit of work completion may be an indicator of one of the following: *The application is not committing transactions frequently and may be causing concurrency issues for other applicationsThe application is idling and not performing any work on the server*
Correction	Examine the locks obtained by the application (Users Detail >Locks page) to see if a low number here should be a cause for concern.

Binds and Precompiles Executed

This section shows the total number of implicit and explicit binds and precompiles executed by the application.

Implicit rebinds are performed by the system when the application tries to execute a sql statement from a package and the package has been invalidated. A package is usually invalidated when you perform DDL/DCL operations on objects on which the package plan is dependent. Explicit rebinds occur when the user executes the REBIND PACKAGE command.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > SQL Activity tab

Metrics Bind and precompile operations have significant overhead for the database operations and they should be minimized during periods of heavy usage when possible.

Attributes tab

The Attributes tab of the Users Detail page displays the following statistics:

Application List Application Attributes Direct Authorities

Indirect Authorities

Application List

This section shows the list of applications connected to the database. The user can obtain further details for an application from other sections on this page by selecting the application from this list.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Attributes tab

Application Attributes

Shows the main information for the selected application. This information is useful in determining the applications connection attributes and its state.The information shown here includes:

Application Status: The status of the application. Values for this field are:

- Database Connect Pending: The application has initiated a database connection but the request has not yet completed.
- Database Connect Completed: The application has initiated a database connection and the request has completed.
- Unit of Work Executing: The database manager is executing requests on behalf of the unit of work.
- Unit of Work Waiting: The database manager is waiting on behalf of the unit of work in the application. This status typically means that the system is executing in the application's code.
- Lock Wait: The unit of work is waiting for a lock. After the lock is granted, the status is restored to its previous value.
- Commit Active: The unit of work is committing its database changes.

- Rollback Active: The unit of work is rolling back its database changes.
- Recompiling: The database manager is recompiling (that is, rebinding) a plan on behalf of the application.
- Compiling: The database manager is compiling an SQL statement or precompiling a plan on behalf of the application.
- Request Interrupted: An interrupt of a request is in progress.
- Database Disconnect Pending: The application has initiated a database disconnect but the command has not yet completed executing. The application may not have explicitly executed the database disconnect command. The database manager will disconnect from a database if the application ends without disconnecting.
- Decoupled from Agent: The application has been decoupled from an agent.
- Transaction Prepared: The unit of work is part of a global transaction that has entered the prepared phase of the two-phase commit protocol.
- Transaction Heuristically Committed: The unit of work is part of a global transaction that has been heuristically committed.
- Transaction Heuristically Rolled Back: The unit of work is part of a global transaction that has been heuristically rolled-back.
- Transaction Ended: The unit of work is part of a global transaction that has ended but has not yet entered the prepared phase of the two-phase commit protocol.
- Creating Database: The agent has initiated a request to create a database and that request has not yet completed.
- Restarting Database: The application is restarting a database in order to perform crash recovery.
- Restoring Database: The application is restoring a backup image to the database.
- Backing Up Database: The application is performing a backup of the database.
- Data Fast Load: The application is performing a "fast load" of data into the database.
- Data Fast Unload: The application is performing a "fast unload" of data from the database.
- Wait to Disable Tablespace: The application has detected an I/O error and is attempting to disable a particular table space. The application has to wait for all other active transactions on the table space to complete before it can disable the table space.
- Quiescing a Tablespace: The application is performing a quiesce table space request.
- Wait for Remote Partition: The application is waiting for a response from a remote partition in a partitioned database instance.
- Remote Request Pending: The application is waiting for results from a federated datasource.

ID of code page used by the application: The code page identifier.

Application status change time: The date and time the application entered its current status.

Sequence Number: This is incremented whenever a unit of work ends (that is, when a COMMIT or ROLLBACK terminates a unit of work). Together, the application id and sequence number uniquely identify a transaction.

Client Product/Version ID: The product and version of DB2 that is running on the client.

Database Alias used by the Application: The alias of the database provided by the application to connect to the database.

User login id: The ID that the user specified when logging in to the operating system. This ID is distinct from auth_id, which the user specifies when connecting to the database

DRDA correlation token: The DRDA correlation token is used for correlating the processing between the application server and the application requester. It is an identifier dumped into logs when errors arise, that you can use to identify the conversation that is in error. In some cases, it will be the LUWID of the conversation

Client operating platform: The operating system on which the client application is running.

Database territory code: The territory code of the database for which the monitor data is collected

Application agent priority: The priority of the agents working for this application

Application priority type: Operating system priority type for the agent working on behalf of the application

Coordinating node: In a multinode system, the node number of the node where the application connected or attached to the instance

Connection request start timestamp: The date and time that an application started a connection request.

Connection request completion timestamp: The date and time that a connection request was granted.

Previous unit of work completion timestamp: This is the time the unit of work completed.

Unit of work start timestamp: The date and time that the unit of work first required database resources.

Unit of work stop timestamp: The date and time that the most recent unit of work completed, which occurs when database changes are committed or rolled back.

Most recent unit of work elapsed time: The elapsed execution time of the most recently completed unit of work.

Unit of work completion status: The status of the unit of work and how it stopped.

Application idle time: Number of seconds since an application has issued any requests to the server. This includes applications that have not terminated a transaction, for example not issued a commit or rollback.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Attributes tab

Direct Authorities

Shows the highest level of direct authority granted to the application

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Attributes tab

Indirect Authorities

Shows the highest level of indirect authority granted to the application

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Attributes tab

Lock Waits tab

The Lock Waits tab of the Users Detail view shows the details for all applications in the database that are waiting on locks. This information is useful in tracking concurrency issues in the database. The statistics available here are:

Application Details Applications Waiting on Locks

The following statistic, available on this tab, duplicates a statistic on the Bottleneck Analysis Pane of the Home View (DBArtisan - DB2 LUW Performance Analyst) page:

Avg. Lock Wait Time

Application Details

This section shows the list of applications waiting for locks to be released on objects they need to access. The information presented here includes:

Auth ID: The authorization ID of the user who invoked the application that is being monitored.

Name: Name of the application executable

Handle: A systemwide unique ID for the application

ID: This identifier is generated when the application connects to the database at the database manager

Client PID: The process ID of the client application that made the connection to the database

Total Lock Waits: The total number of times that applications or connections waited for locks

Total Lock Waiting Time: This is the total amount of elapsed time that this connection or transaction has waited for a lock to be granted to it.

Locks Waiting: Indicates the number of agents waiting on a lock

UOW Lock Waiting Time: The total amount of elapsed time current unit of work has spent waiting for locks

Wait Start Time: The date and time that this application started waiting to obtain a lock on the object that is currently locked by another application

Lock App Handle: System wide Unique ID of the application holding the lock

Lock App ID: ID of the application holding the lock

Lock Sequence ID: The sequence number of the application that is holding a lock on the object that this application is waiting to obtain.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Lock Waits tab
Metrics	AuthID, Name, Handle, ID, and Client PID help you identify the application waiting on the lock. Lock App Handle, Lock App ID, and Lock Sequence number help you identify the application that is holding the waited on lock. You can use this information on Users Detail Locks tab to get further details on the application holding the locks and the locks it's holding.
Correction	If an application is not performing well, and it has a high number of total lock waits, and a high total lock waiting time you may need to improve concurrency of the applications that are holding the locks.

Applications Waiting on Locks

This metric indicates the percentage of all connected applications waiting on locks.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Lock Waits tab
Metrics	If this number is high, the application may have concurrency problems.
Correction	You can identify the applications that are holding locks or exclusive locks for a long time from the Users Detail> Locks page and tune such applications for better concurrency.

Locks tab

The Locks tab of the Users Detail view gives you information for the following statistics:

Application Details Locks Held Details Locks Waiting Details

Application Details

This section shows a list of applications connected to the database. You can look at the lock details for an application in the list by selecting the application. The attributes presented for each application include:

Handle: A system-wide unique ID for the application

ID: This identifier is generated when the application connects to the database at the database manager

Seq No

Name: Name of the application executable

Auth ID: The authorization ID of the user who invoked the application that is being monitored.

Status

Code Page

Locks Held

Total Wait Time:

UOW Lock Waiting Time

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Locks tab

Locks Held Details

This section gives the details of all the locks that the selected application is currently holding. The details presented here include:

Lock Name: Internal binary lock name. This element serves as a unique identifier for locks

Lock Attributes: Gives the lock attribute settings

Release Flags: The lock release flags can either be untracked or SQL Compiler

Lock Count: The number of locks on the lock being held. This value ranges from 1 to 255. It is incremented as new locks are acquired, and decremented as locks are released. When lock_count has a value of 255, this indicates that a transaction duration lock is being held. At this point, lock_count is no longer incremented or decremented when locks are acquired or released. The lock_count element is set to a value of 255 in one of two possible ways:

1 lock_count is incremented 255 times due to new locks being acquired.

2 A transaction duration lock is explicitly acquired. For example, with a LOCK TABLE statement, or an INSERT.

Hold Count: The number of holds placed on the lock. Holds are placed on locks by cursors registered with the WITH HOLD clause and some DB2 utilities. Locks with holds are not released when transactions are committed.

Object Name: It is the name of the object for table-level locks is the file ID (FID) for SMS and DMS table spaces. For row-level locks, the object name is the row ID (RID). For table space locks, the object name is blank. For buffer pool locks, the object name is the name of the buffer pool.

Object Type: The type of object against which the application holds a lock (for object-lock-level information), or the type of object for which the application is waiting to obtain a lock (for application-level and deadlock-level information).

Tablespace: This is the name of the table space against which the lock is held.

Table Schema: Schema of the table that the lock is on

Table Name: Name of the table that the lock is on. This element is only set if Object Type indicates Table

Lock Mode: The type of lock being held.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Locks tab

Locks Waiting Details

This section gives the details of all the locks that the selected application is currently waiting on. The details include:

Subsection: Identifies the subsection associated with the waited on lock

Agent ID: The application handle of the agent holding a lock for which this application is waiting

App ID: The application ID of the application holding a lock on the object that this application is waiting to obtain.

Lock Name: Internal binary lock name. This element serves as a unique identifier for locks.

Lock Attributes: Gives the lock attribute settings

Release Flags: Lock release flags

Object Type: The type of object against which the application holds a lock

Lock Mode: The type of lock being held

Lock Mode Requested: The lock mode being requested by the application

Tablespace: This is the name of the table space against which the lock is held.

Table Schema: Schema of the table that the lock is on

Table Name: Name of the table that the lock is on. This element is only set if Object Type indicates Table

Wait Start Timestamp: The date and time that this application started waiting to obtain a lock on the object that is currently locked by another application

Escalation: Indicates whether a lock request was made as part of a lock escalation.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Users View (DBArtisan - DB2 LUW Performance Analyst)** > Locks tab

Cfg (DBArtisan - Database Configuration)

The Database Configuration view displays the following tabbed pages:

Configuration tab History View (DBArtisan - DB2 LUW Performance Analyst) Indoubt Transactions tab

Objects tab

Configuration tab

This tab lets you view the current values for the DB2 database configuration parameters.

History View (DBArtisan - DB2 LUW Performance Analyst)

The History tab of the Database Configuration Details viewlets you browse the history file entries for the database. The history file contains a record of recovery and administrative events. Recovery events include full database and table space level backup, incremental backup, restore, and rollforward operations. Additional logged events include create, alter, drop, or rename table space, reorganize table, drop table, and load.

On this tab, you'll find the following information:

History Details SQLCA Structure Tablespaces List

DDL View

History Details

This section allows you to select the operation for which you want to see the history details. You can also narrow the history list to operations since a certain date or operations involving a certain object name. You can show the history for the following types of operations:

Backups
Dropped Tables
Load
Quiesce
Reorganizations
Rollforward
Tablespace Alters
Tablespace Renames
Tablespace Creates
Tablespace Drops

You can see further detail on some operations in the lower sections, by selecting that operation from the history detail operation list.

SQLCA Structure	Shows the SQLCA structure for the selected operation from the history detail. This information is not available for Quiesce, Rollforward, and Reorganization operations.
Tablespaces List	Shows the list of tablespaces involved in the selected operation in history detail.
DDL View	Shows the DDL for the selected operation in the history detai

Indoubt Transactions tab

The Indoubt Transactions tab provides a list of transactions that are in doubt. The Indoubt Transaction Detail metrics displayed for each indoubt transaction include:

Application ID: Specifies the application identifier assigned by the database manager for this transaction

Sequence Number: Specifies the sequence number assigned by the database manager as an extension to the application ID

Status: Indicates the status of this indoubt transaction. Valid values are:

SQLXA_TS_PREP	The transaction is prepared. The connected parameter can be used to determine whether the transaction is waiting for the second phase of normal commit processing or whether an error occurred and resynchronization with the transaction manager is required.
SQLXA_TS_HCOM	The transaction has been heuristically committed.
SQLXA_TS_HROL	The transaction has been heuristically rolled back.
SQLXA_TS_MACK	The transaction is missing commit acknowledgement from a node in a partitioned database.
SQLXA_TS_END	The transaction has ended at this database. This transaction may be re-activated, committed, or rolled back at a later time. It is also possible that the transaction manager encountered an error and the transaction will not be completed. If this is the case, this transaction requires heuristic actions, because it may be holding locks and preventing other applications from accessing data.

Timestamp: Specifies the time when the transaction entered the indoubt state.

Auth ID: Specifies the authorization ID of the user who ran the transaction.

Log Full: Indicates whether or not this transaction caused a log full condition

Originator

XID: Specifies the XA identifier assigned by the transaction manager to uniquely identify a global transaction

Objects tab

Tables and indexes consume the storage in all databases. The Database Object Detail grid displays object space details for the database selected in the Database Object Summary grid.

The following statistics are presented for tables:

Table Schema: The owner of the object

Table Name: The name of the object

File Group: File group where the object resides

Reserved (KB): The amount of space (in KB) reserved by the object

Used (KB): The amount of space (in KB) used by the object

Free (KB): The amount of free space (in KB) used by the object

Percent Used: The percentage of space used by the object

Instance View (DBArtisan - DB2 LUW Performance Analyst)

The Embarcadero Performance Analyst Instance view lets you review availability and overall performance of all monitored databases from a single window.

In addition to a Home page, the Instance category of DB2 Performance Analyst includes the following tabbed pages:

Configuration tab
Agents & Connections tab
Utilities tab
FCM Resources tab
FCM Throughput tab
Memory Pools tab
Sorts & Joins tab

Home

Statistics on the Instance home view are organized into the following categories:

Agent Analysis Pane
Bottleneck Analysis Pane
Location
Instance Information Pane
Key Ratio Analysis Pane

Agent Analysis Pane

The Agent Analysis section presents the following metrics:

Connections
Connections Executing
Agents Registered
Agents Waiting
Idle Agents

Connections

The connections statistic gives the total Number of connections to databases in the current instance.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Agent Analysis Pane
Metrics	Shows the number of connections from remote/local clients to databases in this instance. This changes frequently, so you may need to sample it at specific intervals over an extended period of time to get a realistic view of system usage.
Correction	This element can help you adjust the setting of the max_coordagents configuration parameter.

Connections Executing

The connections executed statistic is the number if applications that are currently connected to a database and are currently processing a unit of work within the Database Manager instance being monitored.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Agent Analysis Pane
Metrics	The connections executing metric can help you determine the level of concurrent processing occurring on the Database Manager. This value can change frequently, so you may need to sample it at specific intervals over an extended period of time to get a realistic view of system usage
Correction	This element can help you adjust the setting of the maxcagents configuration parameter.

Agents Registered

The agents registered metric is the number of agents registered in the Database Manager instance that is being monitored (coordinator agents and subagents).

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Agent Analysis Pane
Correction	You can use this element to help evaluate your setting for the maxagents configuration parameter.

Agents Waiting

The number of agents waiting for a token so they can execute a transaction in the database manager.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Agent Analysis Pane
Metrics	Each application has a dedicated coordinator agent to process database requests within the Database Manager. Furthermore, each agent has to get a token before it can execute a transaction. The maximum number of agents that can execute Database Manager transactions is limited by the configuration parameter maxcagents.
Correction	You can use this element to help evaluate your setting for the maxcagents configuration parameter.

Idle Agents

The idle agents metric is the number of agents in the agent pool that are currently unassigned to an application and are, therefore, "idle."

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Agent Analysis Pane
Metrics	Having idle agents available to service requests for agents can improve performance.
Correction	You can use this element to help set the num_poolagents configuration parameter.

Bottleneck Analysis Pane

The Bottleneck Analysis section presents the following metrics:

Sorts and Joins

Post Threshold Sorts Piped Sorts Rejects Hash Join Thresholds

Agent Bottlenecks

Agents Waiting on a Token Agents Created Due to Empty Pool

Post Threshold Sorts

This metric shows the number of sorts that have requested heaps after the sort heap threshold has been exceeded.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Under normal conditions, the Database Manager allocates sort heap using the value specified by the sortheap configuration parameter. If the amount of memory allocated to sort heaps exceeds the sort heap threshold (sheapthres configuration parameter), the database manager allocates sort heap using a value less than that specified by the sortheap configuration parameter. Each active sort on the system allocates memory, which may result in sorting taking up too much of the system memory available. Sorts that start after the sort heap threshold has been reached may not receive an optimum amount of memory to execute. As a result, however, the entire system may benefit.
Correction	*By modifying the sort heap threshold and sort heap size configuration parameters, the performance of sort operations and/or the overall system can be improved. If this element's value is high, you can:Increase the sort heap threshold (sheapthres) or,Adjust applications to use fewer or smaller sorts via SQL query changes.*

Piped Sorts Rejects

The piped sorts rejects statistic is the number of piped sorts that were rejected by the database manager.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane

Metrics Each active sort allocates memory, and can result in sorting taking up too much available system memory. A piped sort is not accepted if the sort heap threshold is exceeded when the sort heap is allocated for the sort. The sort list heap (sortheap) and sort heap threshold (sheapthres) configuration parameters help control the amount of memory used for sort operations. These parameters are also used to determine whether a sort will be piped. Since piped sorts may reduce disk I/O, allowing more piped sorts can improve the performance of sort operations and possibly the performance of the overall system.

Correction If piped sorts are being rejected, you might consider decreasing your sort heap or increasing your sort heap threshold. You should be aware of the possible implications of either of these options. If you increase the sort heap threshold there is the possibility that more memory will remain allocated for sorting. This could cause paging memory to disk. If you decrease the sort heap, you might require an extra merge phase that could slow down the sort

Hash Join Thresholds

The hash join thresholds metric is the total number of times that a hash join heap request was limited because of concurrent use of shared or private sort heap.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane

Metrics Each active sort allocates memory, and can result in sorting taking up too much of the available system memory. A piped sort is not accepted if the sort heap threshold is exceeded when the sort heap is allocated for the sort. The sort list heap (sortheap) and sort heap threshold (sheapthres) configuration parameters help control the amount of memory used for sort operations. These parameters are also used to determine whether a sort will be piped. Since piped sorts may reduce disk I/O, allowing more piped sorts can improve the performance of sort operations and possibly the performance of the overall system.

Correction If this value is large (greater than 5% of Hash Join Overflows), the sort heap threshold should be increased.

Agents Waiting on a Token

The agents waiting statistic is the percentage of agents registered that are waiting for a token so they can execute a transaction in the Database Manager.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Each application has a dedicated coordinator agent to process database requests within the Database Manager. Each agent has to get a token before it can execute a transaction. The maximum number of agents that can execute database manager transactions is limited by the configuration parameter maxcagents. This parameter is used to control the load on the system during periods of high simultaneous application activity. For example, you may have a system requiring a large number of connections but with a limited amount of memory to serve those connections. Adjusting this parameter can be useful in such an environment, where a period of high simultaneous activity could cause excessive operating system paging. This parameter does not limit the number of applications that can have connections to a database. It only limits the number of Database Manager agents that can be processed concurrently by the Database Manager at any one time, thereby limiting the usage of system resources during times of peak processing.
Correction	In cases where the high concurrency of applications is causing problems, you can use benchmark testing to tune this parameter to optimize the performance of the database.

Agents Created Due to Empty Pool

The agents created due to empty pool metric shows the number of agents created because the agent pool was empty. It includes the number of agents started when the current instance started (num_initagents).

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane

FCM Analysis Pane

The fast communication manager (FCM) provides communication support for Enterprise Server Editions. Each database partition server has one FCM thread to provide communications between database partition servers to handle agent requests and to deliver message buffers. The FCM thread starts when you start the instance.

The following ratios are used on the FCM Analysis section to succinctly communicate the general overall FCM performance levels of the monitored instance:

Buffer Throughput
Buffer Utilization
Connection Entry Utilization
Message Anchor Utilization
Request Block Utilization

Buffer Throughput

The buffer throughput metric is the total number of buffers that have been sent and received by this database partition since the start of monitoring.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > FCM Analysis Pane
Metrics	This gives the level of traffic to and from the selected partition.
Correction	If this traffic level is high, you should consider redistributing one or more databases on the instance or move tables to reduce internode traffic.

Buffer Utilization

This element indicates the percentage of all FCM buffers currently being used by the fast communication manager.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > FCM Analysis Pane
Correction	You can use this information to tune fcm_num_anchors. If the utilization percentage is high, you should increase the fcm_num_anchors to ensure that operations do not run out of FCM message anchors. If the utilization is low, you can decrease fcm_num_anchors to conserve system resources.

Connection Entry Utilization

The connection entry utilization element indicates the percentage of all connection entries that are currently being used by the fast communication manager.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > FCM Analysis Pane
Correction	You can use this information to tune fcm_num_connect. If the utilization percentage is high, you should increase the fcm_num_connect to ensure that operations do not run out of FCM connection entries. If the utilization is low, you can decrease fcm_num_connect to conserve system resources.

Note: This metric is only applicable to DB2 version 7. In DB2 version 8, the maximum connection entries are adjusted dynamically and automatically

Message Anchor Utilization

This metric shows the percentage of all message anchors currently being used by the fast communication manager.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > FCM Analysis Pane
Correction	You can use this information to tune fcm_num_buffers. If the utilization percentage is high, you should increase the fcm_num_buffers to ensure that operations do not run out of FCM buffers. If the utilization is low, you can decrease fcm_num_buffers to conserve system resources.

Note: This metric is only applicable to DB2 version 7. In DB2 version 8, the maximum message anchors are adjusted dynamically and automatically

Request Block Utilization

This element indicates the percentage of all request blocks that are currently being used by the fast communication manager.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > FCM Analysis Pane
Correction	You can use this information to tune fcm_num_rqb. If the utilization percentage is high, you should increase the fcm_num_rqb to ensure that operations do not run out of FCM request blocks. If the utilization is low, you can decrease fcm_num_rqb to conserve system resources.

Note: This metric is only applicable to DB2 version 7. In DB2 version 8, the maximum request blocks are adjusted dynamically and automatically

Instance Information Pane

Here you see the following information:

Product Name
FixPack Level
Instance Type
Operating System
Instance Name
Number of Partitions
Active Local Databases

Key Ratio Analysis Pane

The Key Ratio Analysis section presents the following metrics:

Efficiency

Agents Registered
Agents Waiting
Monitor Heap
FCMBP
Private Sort Memory

Agents Registered

The agents registered metric shows the percentage of total allowable agents (coordinator agents and subagents) that are currently working in the instance.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key Ratio Analysis Pane
Metrics	When this value is close to 100% it means that the number of agents currently working is close to the hard limit set in the database manager configuration. If this value stays high for long periods of time it means that there is high level of activity occurring in the databases on the instance.
Correction	Use this element to help you evaluate your setting for the maxagents database manager configuration parameter. When you increase the value of this parameter, make sure that your system has enough memory available to accommodate the additional agents that may be created.

Agents Waiting

The agents waiting statistic is the percentage of agents registered that are waiting for a token so they can execute a transaction in the Database Manager.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key Ratio Analysis Pane
Metrics	Each application has a dedicated coordinator agent to process database requests within the Database Manager. Each agent has to get a token before it can execute a transaction. The maximum number of agents that can execute database manager transactions is limited by the configuration parameter maxcagents. This parameter is used to control the load on the system during periods of high simultaneous application activity. For example, you may have a system requiring a large number of connections but with a limited amount of memory to serve those connections. Adjusting this parameter can be useful in such an environment, where a period of high simultaneous activity could cause excessive operating system paging. This parameter does not limit the number of applications that can have connections to a database. It only limits the number of Database Manager agents that can be processed concurrently by the Database Manager at any one time, thereby limiting the usage of system resources during times of peak processing.
Correction	In cases where the high concurrency of applications is causing problems, you can use benchmark testing to tune this parameter to optimize the performance of the database.

Monitor Heap

This metric shows the percentage of total monitor heap that is currently being used by the Database Manager.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key Ratio Analysis Pane
Metrics	The amount of memory required for monitoring activity depends on the number of monitoring applications (applications taking snapshots or event monitors), which switches are set, and the level of database activity. Memory is allocated from the monitor heap when you perform database monitoring activities such as taking a snapshot, turning on a monitor switch, resetting a monitor, or activating an event monitor. If this metric is running at over 70% utilization and you have many applications performing snapshot /event monitoring, you may need to increase the size of the monitor heap.
Correction	Increase the value of mon_heap_sz database manager configuration parameter to increase the available monitor heap size.

FCMBP

The fast communication manager bufferpool (FCMBP) is the percentage of total amount of the FCMBP that is currently being used. This metric is only applicable to multipartitioned database environments.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key Ratio Analysis Pane
Metrics	You can use this information to tune the fcm_num_buffers database manager configuration parameter.

Note: FCM buffers are used for internal communication both among and within database servers. The FCMBU metric gives the percentage utilization of the available fcm buffers in the Database Manager.

Private Sort Memory

The private sort memory metric is the percentage of the total private sort memory heap that is currently in use.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key Ratio Analysis Pane
Metrics	You can use this information to tune the sheapthres database manager configuration parameter. If the value of this element is close to 100%, it means that the sorts for some databases are not getting the full sort heap as defined by the sortheap database configuration parameter.

Configuration tab

The Configuration tab of the Instance Detail view lets you view and change the instance configuration parameter. It also allows you to look at some information about the instance such as Product Name, Type, Fixpak Level, Operating System, instance name, partitions, and number of active databases.

Agents & Connections tab

The Agents & Connections tab of the Instance Detail view gives you the following analyses:

Connection Analysis Agent Pool Analysis Agent Analysis

Connection Analysis

This section presents the metrics for the current local and remote connections to databases in the instance. The information presented here includes:

Local Connections: The number of local applications that are currently connected to a database within the database manager instance being monitored.

Remote Connections: The current number of connections initiated from remote clients to the instance of the database manager that is being monitored.

Local Connections Executing: The number of local applications that are currently connected to a database within the database manager instance being monitored and are currently processing a unit of work.

Remote Connections Executing: The number of remote applications that are currently connected to a database and are currently processing a unit of work within the database manager instance being monitored.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Agents & Connections tab
Metrics	These metrics can help you determine the level of concurrent processing occurring in the database manager. These metrics will change frequently so it is important to obtain a sampling over an extended period. These metrics are useful when tuning the max_coordagents, and maxcagents database manager configuration parameters.

Agent Pool Analysis

The metrics in this section help determine how often the database agents are created due to empty pool and how often they are assigned from the agent pool.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Agents & Connections tab
Metrics	If the number of agents created due to empty pool is high compared to the agents assigned from pool, it may mean that the number of pool agents should be increased. It may also mean that the overall workload is too high.
Correction	You can change the maxcagents and num_poolagents database manager configuration parameters to tune these metrics.

Agent Analysis

The metrics presented here give a detailed view of how the database manager agents are working. The metrics presented here include:

Registered Agents: The number of agents registered in the database manager instance that is being monitored (coordinator agents and subagents).

Agents Waiting for Token: The number of agents waiting for a token so they can execute a transaction in the database manager.

Idle Agents: The number of agents in the agent pool that are currently unassigned to an application and are, therefore, "idle".

Stolen Agents: The number of times that agents are stolen from an application. Agents are stolen when an idle agent associated with an application is reassigned to work on a different application.

Maximum Agents Registered: The maximum number of agents that the database manager has ever registered, at the same time, since it was started (coordinator agents and subagents).

Maximum Agents Waiting: The maximum number of agents that have ever been waiting for a token, at the same time, since the database manager was started.

Maximum Coordinating Agents: The maximum number of coordinating agents working at one time.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Agents & Connections tab

Utilities tab

The Utilities tab of the Instance Detail View gives information on the utilities currently executing in the database manager. The Utility Execution Details are as follows:

Utility ID: The unique identifier corresponding to the utility invocation

Type: Class of the utility (Rebalance, Backup, Restore, Reorg etc..)

Utility Description: A brief description of the work the utility is performing

Utility Start Time: The time the utility was invoked

DBName: Database on which the utility is operating.

FCM Resources tab

The FCM Resources tab of the Instances Detail View presents statistics on the following:

Node Details
FCM Buffer Utilization
FCM Message Anchor Utilitization
FCM Connection Entries Utilization
FCM Connection Entries Utilization

Node Details

This section gives a detailed view of inter- node communication in a multipartition environment. The information shown here includes the connection status and buffers sent and received between various combinations of partitions in a multi-partitioned environment. The columns presented here include:

Source Node: Partition that sends the information

Target Node: Partition that received the information

Buffers Sent: Number of Buffers sent from source node to the target node

Buffers Received: Number of Buffers sent from source node to the target node

Connection Status

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > FCM Resources tabIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > FCM Throughput tab

FCM Buffer Utilization

This section provides details on the current utilization and low watermark of FCM buffers.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > FCM Resources tab
Metrics	You can monitor the buffer utilization for the selected partition together with the low watermark to determine the usage trend of the fcm buffers.
Correction	If you notice a very small value for the low watermark and similar values for free FCM buffers over a long period, you may need to increase the fcm_num_buffers database manager configuration parameter to increase the number of available FCM buffers. If on the other hand you notice a very high value for the low watermark – you may have allocated too many buffers and you can reduce the value of fcm_num_buffers database manager configuration parameter.

FCM Message Anchor Utilitization

This section provides details on the current utilization and low watermark of FCM message anchors.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > FCM Resources tab
Metrics	You can monitor the message anchor utilization for the selected partition together with the low watermark to determine the usage trend of the message anchors.
Correction	If you notice a very small value for the low watermark and similar values for free message anchors over a long period, you may need to increase the fcm_num_anchors database manager configuration parameter to increase the number of available message anchors. If on the other hand you notice a very high value for the low watermark – you may have allocated too many message anchors and you can reduce the value of fcm_num_anchors database manager configuration parameter.

FCM Connection Entries Utilization

This section provides details on the current utilization and low watermark of FCM connection entries.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > FCM Resources tab
Metrics	You can monitor the connection entry utilization for the selected partition together with the low watermark to determine the usage trend of connection entries.
Correction	If you notice a very small value for the low watermark and similar values for free connection entries over a long period, you may need to increase the fcm_num_connect database manager configuration parameter to increase the number of available connection entries. If on the other hand you notice a very high value for the low watermark – you may have allocated too many connection entries and you can reduce the value of fcm_num_connect database manager configuration parameter.

FCM Request Block Utilization

This section provides details on the current utilization and low watermark of FCM request blocks.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > FCM Resources tab
Metrics	You can monitor the request block utilization for the selected partition together with the low watermark to determine the usage trend of the request blocks.
Correction	If you notice a very small value for the low watermark and similar values for free request blocks over a long period, you may need to increase the fcm_num_rqb database manager configuration parameter to increase the number of available request blocks. If on the other hand you notice a very high value for the low watermark – you may have allocated too many request blocks and you can reduce the value of fcm_num_rqb database manager configuration parameter.

FCM Throughput tab

The FCM Throughput tab of the Instance Detail VIew includes the following statistics:

- FCM Throughput

The following statistic, available on this tab, duplicates a statistic available on the **Instance View (DBArtisan - DB2 LUW Performance Analyst)**FCM Resources tab:

- Node Details

FCM Throughput

This section shows the delta statistics for the FCM buffers sent and received between the selected nodes in the Node Details section

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > FCM Throughput tab
Metrics	If the total number of FCM buffers sent or received between selected nodes is high, you may want to redistribute the database, or move tables to reduce the inter-node traffic.

Memory Pools tab

The Memory Pools tab is divided into three panes. Unique to this tab is the Memory Pool List pane. The Memory Pool Details and the Memory Pool Utilization are duplicates of panes available in the Memory View (DBArtisan - DB2 LUW Performance Analyst) category.

Memory Pool List pane

This section shows the memory pool usage details of the selected memory pool over time.

Sorts & Joins tab

The Sorts & Joins tab of the Instance Detail View presents the following information:

Sort Heap Details Sort Threshold Details Hash Join Threshold Details

Piped Sort Details

Sort Heap Details

This section shows the metrics for the total number of allocated pages of sort heap space for all sorts. It represents the sum of sort heap space allocated for all sorts in all active databases in the database manager.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Sorts & Joins tab
Metrics	Normal memory estimates do not include sort heap space. If excessive sorting is occurring, the extra memory used for the sort heap should be added to the base memory requirements for running the database manager. Generally, the larger the sort heap, the more efficient the sort. Appropriate use of indexes can reduce the amount of sorting required.
Correction	You may use the information in this section to help you tune the sheapthres configuration parameter. If the element value is greater than or equal to sheapthres, it means that the sorts are not getting the full sort heap as defined by the sortheap parameter.

Sort Threshold Details

This section details the number of sorts that have requested heaps after the sort heap threshold has been exceeded.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Sorts & Joins tab
Metrics	Under normal conditions, the database manager will allocate sort heap using the value specified by the sortheap configuration parameter. If the amount of memory allocated to sort heaps exceeds the sort heap threshold (sheapthres configuration parameter), the database manager will allocate sort heap using a value less than that specified by the sortheap configuration parameter. Each active sort allocates memory, which may result in sorting taking up too much of the system memory available. Sorts that start after the sort heap threshold has been reached may not receive an optimum amount of memory to execute, but, as a result, the entire system may benefit. By modifying the sort heap threshold and sort heap size configuration parameters, the performance of sort operations and/or the overall system can be improved.
Correction	If this element's value is high, you can: *Increase the sort heap threshold (sheapthres) or,Adjust applications to use fewer or smaller sorts via SQL query changes.*

Hash Join Threshold Details

This section shows the total number of times that a hash join heap request was limited due to concurrent use of shared or private sort heap space.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Sorts & Joins tab
Correction	If this value is large (greater than 5% of hash join overflows), the sort heap threshold should be increased

Piped Sort Details

This section gives the details on the piped sorts that have been requested and accepted.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **Instance View (DBArtisan - DB2 LUW Performance Analyst)** > Sorts & Joins tab
Metrics	Each active sort on the system allocates memory, which may result in sorting taking up too much of the available system memory. When the number of accepted piped sorts is low compared to the number requested, you can improve sort performance by adjusting one or both of the following configuration parameters: `sortheap` and **sheapthres**.
Correction	If piped sorts are being rejected, you might consider decreasing your sort heap or increasing your sort heap threshold. You should be aware of the possible implications of either of these options. If you increase the sort heap threshold, then there is the possibility that more memory will remain allocated for sorting. This could cause the paging of memory to disk. If you decrease the sort heap, you might require an extra merge phase that could slow down the sort.

OS View (DBArtisan - DB2 LUW Performance Analyst)

In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database sever for resources (CPU, I/O, or Memory), a slow CPU, insufficient or slow I/O devices, and insufficient memory. The OS Statistics page of Performance Analyst lets you examine operating system metrics for the following platforms:

- AIX
- HP-UX

Note: To view processor info and swap disk info on an HP-UX box, you need to login as ROOT in the OS login.

- Linux
- Solaris
- Unix
- Windows XP and 2000

Note: The statistics available on the OS page depend on the operating system platform.

Tip: If you magnify a graph to change back to the original size or minimize the size, close the OS Page and then reopen it again. Performance Analyst displays the graph in its original form and size.

In addition to a Home page, the OS category of DB2 Performance Analyst includes the following tabbed pages:

CPU Tab I/O Tab Memory Tab

Network Tab Processes Tab Space Tab

Home

The OS home page includes the following sections:

Key Resource Usage Pane Bottleneck Analysis Pane Memory Analysis Pane

Disk Analysis Panes Workload Analysis Pane

Key Resource Usage Pane

The following ratios are used on the Performance Analyst OS home page to communicate the general overall performance levels of the operating system:

Disk Time	Load Average	Processor Time
Paged Memory Used(Windows)	Swap Memory Used (AIX, HP-UX, Linux, Solaris, Unix)	

Tip: To open the CPU tab, right-click Processor Time. To open the I/O tab, right-click Disk Time. To open the Memory tab, right-click Paged Memory Used.

Disk Time

The Disk Time statistic is the percentage of elapsed time that the selected disk drive/device was busy servicing read or write requests.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	You should avoid consistently seeing values for this statistic greater then 90%.
Correction	Add more disk drives and partition the files among all of the drives.

Load Average

The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	High load averages usually mean the system is being used heavily and the response time is correspondingly slow.

Processor Time

The Processor Time statistic indicates the percentage of time the processor is working. This counter is a primary indicator of processor activity.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	If your computer seems to be running sluggishly, this statistic could be displaying a high percentage.
Correction	Upgrade to a processor with a larger L2 cache, a faster processor, or install an additional processor.

Paged Memory Used

The Paged Memory Used statistic is the ratio of Commit Memory Bytes to the Commit Limit. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk. The commit limit is determined by the size of the paging file. As the paging file increases, so does the commit limit.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	This value displays the current percentage value only and not an average. If the percentage of paged memory used is above 90%, you may be running out of memory.
Correction	Increase the size of page file.

Note: This statistic is available for the Windows platform.

Swap Memory Used

The Swap Memory Used statistic is the percentage of swap space currently in use.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Key Resource Usage Pane
Metrics	If the percentage of swap memory used is above 90%, you may be running out of memory.
Correction	Increase the size of your swap files.

Bottleneck Analysis Pane

The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

Average Disk Queue Length	Interrupts/Sec	Network Output Queue Length/Network Queue Length
Page Faults/Sec	Processor Queue Length	Processor Speed
Processor		

Tip: To open the I/O tab, right-click any Details menu item. To open the CPU tab, right-click the Item Processor Queues Length Details menu item. To open the Network tab, right-click Network Output Queues Length.

Note: The statistics available in this section depend on the platform of operating system.

Average Disk Queue Length

The Average Disk Queue Length statistic is the average number of both read and write requests that were queued for the selected disk during the sample interval.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	This metric is useful in identifying I/O related bottlenecks. If the disk queue lengths for certain disks are consistently much higher than others, you may need to redistribute the load among available disks. If the disk queues lengths for all disks are consistently large, and you see a high amount of I/O activity, your disks may be inefficient.
Correction	Some things you can do if you have problems with this statistic include: *Redistribute the data on the disk with the large average disk queue to other disks.Upgrade to faster disk(s).*

Interrupts/Sec

Interrupts/Sec is the average rate, in incidents per second, at which the processor received and serviced hardware interrupts. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended. The system clock typically interrupts the processor every 10 milliseconds, creating a background of interrupt activity. This statistic shows the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > CPU Tab
Metrics	A high value indicates possible excessive hardware interrupts; justification is dependent on device activity.

Network Output Queue Length/Network Queue Length

The Network Output Queue Length statistic is the number of threads in the processor queue.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane
Metrics	Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.
Correction	A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Note: The name of this statistic depends on the platform of the operating system.

Page Faults/Sec

The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Tab
Metrics	This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
Correction	If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Processor Queue Length

The Processor Queue Length statistic is the number of threads in the processor queue.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis PaneIBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > CPU Tab
Metrics	Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.
Correction	A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Processor Speed

The Processor Speed statistic displays the speed of the active processor in MHz. The speed is approximate.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane

Processor

The Processor Statistic displays the type of processor currently in use, for example, GenuineIntel.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Bottleneck Analysis Pane

Memory Analysis Pane

The following metrics are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

Available Paged Memory (Windows)
Available Physical Memory
Available Swap Memory(AIX, HP-UX, Linux, Solaris, Unix)
Total Paged Memory/Total Swap Memory
Total Physical Memory

Tip: To open the Memory tab, right-click any Details menu item.

Note: The statistics available in this section depend on the platform of operating system.

Available Paged Memory

The Available Paged Memory statistic shows the amount of virtual memory available for the processes.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Memory Analysis Pane
Metrics	If the available virtual memory is less than 10% of the total virtual memory, your system may run out of memory.
Correction	Increase the size of page file.

Note: This statistic is available for the Windows platform.

Available Physical Memory

The Available Physical Memory statistic represents the amount of RAM available to all processes.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Memory Analysis Pane

Metrics This counter displays the last observed value only and not an average. Use this value with the Total physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Correction Some things you can do if you have problems with this statistic include: *Check the running processes to see if there are any memory leaks.Stop any services that are not required.Install additional RAM.*

Available Swap Memory

The Available Swap Memory statistic represents the amount of virtual memory available for the processes.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Memory Analysis Pane

Metrics If the Available Swap Memory is less than 10% of the total Swap Memory, your system may run out of memory.

Correction Increase the size of swap files.

Total Paged Memory/Total Swap Memory

The Total Paged Memory statistic shows the maximum amount of virtual memory available to all processes.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Memory Analysis Pane

Metrics It is recommended that this be 1½ - 3 times the amount of RAM on the system.

Note: The name of this statistic depends on the platform of the operating system.

Total Physical Memory

The Total Physical Memory statistic shows the amount of physical memory installed on your computer.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Memory Analysis Pane
Metrics	This is an informational metric and displays the total amount installed on the machine. Use this value with the available physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Disk Analysis Pane

The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

Free Disk Space Total Disk Space Used Disk Space

Tip: To open the Space tab, right-click any Details menu item.

Note: The statistics available in this section depend on the platform of operating system.

Free Disk Space

The Free Disk Space statistic shows the unallocated space, in megabytes on all logical disk drives.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Disk Analysis Pane
Correction	There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations: *If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.*

Total Disk Space

Total Disk Space displays the total allocated and unallocated space, in megabytes on all logical disk drives.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Disk Analysis Pane

Correction There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations, here are two: *If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.*

Used Disk Space

The Used Disk Space statistic shows the amount of allocated space, in megabytes on all logical disk drives.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Disk Analysis Pane

Correction There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations:

If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Workload Analysis Pane

The following ratios are used on the Performance Analyst OS home page to succinctly communicate the general overall performance levels of the operating system:

Number of Logins Number of Processes Top CPU Process

Top I/O Process Top Memory Process

Tip: To open the Processes tab, right-click any Details menu item.

Note: The statistics available in this section depend on the platform of operating system.

Number of Logins

This statistic displays the total number of logins on the server.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Workload Analysis Pane

Number of Processes

This statistic displays the total number of processes on the server.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Disk Analysis Pane

Top CPU Process

Top CPU Process shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Disk Analysis Pane
Metrics	If the amount of CPU time used by this process is close to 100% and the CPU usage is very high, this process may be the bottleneck on the server.
Correction	Investigate the process further to see if it is in an inconsistent state. Also, look at minimum requirements for CPU speed for the process. You may need to upgrade your CPU.

Top I/O Process

The Top I/O Process statistic shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Disk Analysis Pane

Top Memory Process

Top Memory Process shows the current process that is consuming the most amount of memory. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Home > Disk Analysis Pane
Metrics	If you are running out of memory on the system, this is a quick way to identify the top memory user. If the displayed process is using a significant portion of the total memory, it could be causing the memory issues.

Processes Overview

The Processes Overview of the OS Summary includes the following sections:

Top CPU Process Top I/O Process Top Memory Process

CPU Tab

The CPU tab of the OS Detail includes the following sections:

Context Switches/Sec CPU Events CPU Utilization

Load Averages

The following statistics, available on this tab, are duplicates of statistics on the **OS View (DBArtisan - DB2 LUW Performance Analyst)**Home page:

Processor Queue Length Interrupts/Sec

Note: The sections available on this tab depend on the platform of operating system.

CPU Utilization

The CPU Utilization section includes the following information:

- % Privileged Time
- % User Time

% Privileged Time

The metric shows the percentage of elapsed time that the process threads spent executing code in privileged mode.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > CPU Tab
Metrics	The ideal range should be 0-40% (less than 40% indicates excessive system activity).
Correction	If your CPU consistently runs at less than 40% you may need to upgrade your system to include a faster processor(s).

Note: For Windows systems, when a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. These kernel commands, are considered privileged to keep the low-level commands executing and prevent a system freeze. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.

% User Time

The % User Time statistic is the percentage of elapsed time the processor spends in the user mode. User mode is a restricted processing mode designed for applications, environment subsystems, and integral subsystems. The alternative, privileged mode, is designed for operating system components and allows direct access to hardware and all memory. The operating system switches application threads to privileged mode to access operating system services. This counter displays the average busy time as a percentage of the sample time.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > CPU Tab
Metrics	If the Privileged Time is high in conjunction with Physical Disk Reads, consider upgrading the disk I/O subsystem.

CPU Events

The CPU Events section includes the following information:

Context Switches/Sec System Calls/Sec

Note: The statistics available in this section depend on the platform of operating system.

Context Switches/Sec

The Context Switches/Sec section shows the combined rate at which all processors on the computer are switched from one thread to another. Context switches occur when a running thread voluntarily relinquishes the processor, is preempted by a higher priority ready thread, or switches between user-mode and privileged (kernel) mode to use an Executive or subsystem service.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > CPU Tab
Metrics	The ideal range should be between 0-10,000. GA number greater then 10,000 may indicate too many threads contending for resources.

System Calls/Sec

System calls are functions which require the services of the operating system. Usually they involve access to some data that users must not be permitted to corrupt or even change.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > CPU Tab
Correction	Try to find a good ratio between the amount of time the operating system runs (fulfilling system calls and doing process scheduling), and the amount of time the application runs. Your goal should be running 60% to 75% of the time in application mode, and 25% to 40% of the time in operating system mode. If you find that the system is spending 50% of its time in each mode, then you should investigate to determine what is wrong.

Load Averages

The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > CPU Tab
Metrics	High load averages usually mean the system is being used heavily and the response time is correspondingly slow.

Processes Tab

The Processes tab of the OS Detail page succinctly communicates the general overall performance levels of processes. The columns available in this table depend on the platform of operating system. The table below describes the information available in the table on this tab:

Column	Description
Process	The name of the process.
User	The user of the process.
ID	The ID Process is the unique identifier of this process. ID Process numbers are reused, so they only identify a process for the lifetime of that process.
CPU	The CPU is the percentage of elapsed time that all of process threads used the processor to execution instructions.
User Mode	The User Mode is the percentage of elapsed time that the process threads spent executing code in user mode.
Memory WINDOWS ONLY	Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Memory (MB)	Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Memory	Memory is the percentage of the memory used of the total memory.
Active Memory	Active Memory is the amount of committed virtual memory, in bytes for this process. Active memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive. This counter displays the last observed value only; it is not an average.
I/O Data	The rate at which the process is reading and writing bytes in I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.
Elapsed Time	The total elapsed time, in seconds, that this process has been running.
Thread Count	The number of threads currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.
Handle Count	The total number of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in this process.
Priority	The current base priority of this process. Threads within a process can raise and lower their own base priority relative to the process' base priority.
Creating Proc ID	The Creating Process ID value is the Process ID of the process that created the process. The creating process may have terminated, so this value may no longer identify a running process.
Page Faults/Sec	Page Faults/Sec is the rate at which page faults by the threads executing in this process are occurring. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This may not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with whom the page is shared.

Page File	Page File is the current number of kilobytes that this process has used in the paging file(s). Paging files are used to store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory.
Private	Private is the current size, in kilobytes, of memory that this process has allocated that cannot be shared with other processes.

I/O Tab

The I/O tab includes the following sections:

Average Disk Queue/Total Disk Queue	Bytes per I/O Operation	Disk I/O Time
Disk Transfers/Sec	I/O Details	Transfer Rate

Note: The sections available on this tab depend on the platform of operating system.

I/O Details

The table below describes the information available in this section:

Column	Description
Disk	The disk number assignment.
Reading (KB/s)	The amount of bytes read from the device.
Writing (KB/s)	The amount of bytes written to the device.
Disk Read Time	Disk Read Time is the percentage of elapsed time that the selected disk drive was busy servicing read requests.
Disk Write Time	Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.
Disk Time	Disk Time is the percentage of elapsed time that the selected disk was busy servicing requests.
Avg. Read Queue	Avg. Disk Read Queue Length is the average number of read requests that were queued for the selected disk during the sample interval.
Avg. Write Queue	Avg. Disk Write Queue Length is the average number of write requests that were queued for the selected disk during the sample interval.
Disk Reads/Sec	Disk Reads/Sec is the rate of read operations on the disk.
Disk Writes/Sec	Disk Writes/Sec is the rate of write operations on the disk.

Note: The columns available in this table depend on the platform of operating system.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > I/O Tab

Transfer Rate

The Transfer Rate section succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

Statistic	Description	Metrics
Reading	The average number of bytes transferred from the disk during read operations.	It is measured in number of KBs per second.
Writing	The average number of bytes transferred to the disk during write operations.	It is measured in number of KBs per second.
Transferred	The amount of time required to transfer the data between the device's cache and the actual device media.	N/A

Note: The statistics available in this section depend on the platform of operating system.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > I/O Tab

Bytes per I/O Operation

The Bytes per I/O Operation section of the OS Detail succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

Statistic	Description	Metrics
Reading	The average number of bytes transferred from the disk during read operations.	It is measured in number of bytes per second.
Writing	The average number of bytes transferred to the disk during write operations.	It is measured in number of bytes per second.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > I/O Tab

Average Disk Queue/Total Disk Queue

The Average Disk Queue section of the OS Detail page succinctly communicates the general overall performance levels of the server's I/O. The table below describes the statistics in this section:

Statistic	Description	Information
Average Read Queue	Average Read Queue Length is the average number of read requests that were queued for a particular disk.	Sustained loads on any of the disk queue lengths could represent problems within your system.
Average Write Queue	Average Write Queue is the average number of write requests that were queued for a particular disk.	Sustained loads on any of the disk queue lengths could represent problems within your system. Contact your Windows System Administrator for assistance in troubleshooting these problems.
Average Queue	Average time in ms. to service I/O requests which for disks includes seek, rotational latency and data transfer time).	Sustained loads on any of the disk queue lengths could represent problems within your system. Contact your Unix System Administrator for assistance in troubleshooting these problems.

Note: The statistics available in this section depend on the platform of operating system.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > I/O Tab
Correction	Contact your Windows System Administrator for assistance in troubleshooting these problems.

Disk I/O Time

The Disk I/O Time section of the OS Detail page succinctly communicates the general overall performance levels of the disk's I/O. The statistics available in this section depend on the platform of operating system. The table below describes the statistics in this section:

Statistic	Description	Metrics
% Disk Read Time	% Disk Read Time is the percentage of time during the sample interval that the disk is busy servicing read requests.	Use this in conjunction with the % Disk Write Time to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.
% Disk Write Time	% Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.	Use this in conjunction with the % Disk Read Time to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.
% Disk Time	% Disk Time is the percentage of time during the sample interval that the disk is busy servicing requests.	Use this statistic to identify the disks that may be running at capacity and may require upgrades. This metric may also be used to balance the load between under/over utilized disks.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > I/O Tab

Disk Transfers/Sec

The Disk Transfers/Sec section of the OS Detail page succinctly communicates the general overall performance levels of the disk transfers. The table below describes the statistics in this section

Statistic	Description
Reads/Sec	The the rate of read operations on the disk.
Writes/Sec	The rate of write operations on the disk.
Transfers/Sec	The rate of transfer operations per second.

Note: The statistics available in this section depend on the platform of operating system.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > I/O Tab

Memory Tab

The Memory tab of the OS Detail page includes the following sections:

Buffer Size	Cache Efficiency	Cache Size
Memory Available	Paging Activity	Page Replacements

The following statistic, available on this tab, duplicates a statistic on the **OS View (DBArtisan - DB2 LUW Performance Analyst)**Memory Tab.

- Page Faults/Sec

Note: The sections available on this tab depend on the platform of operating system.

Paging Activity

The Paging Activity section includes the following statistics:

Blocks Input/Sec	Blocks Output/Sec	Paged In
Paged Outt	Pages Input/Sec	Pages Output/Sec

Note: The statistics available in this section depend on the platform of operating system.

Blocks Input/Sec

The Blocks Input/sec statistic is the number of blocks sent to a block device per second.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Tab
Correction	Although it never hurts to have as much physical memory as your system can handle, here are two things you can check within your system to alleviate the memory bottleneck: *Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.Check to see if you have additional space on your disk drive that you could use to expand the size of your swap file. Normally, the bigger the initial size of your swap file, the better, in performance terms.*

Blocks Output/Sec

The Blocks Output/sec statistic is the number of blocks received from a block device per second).

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Tab
Correction	Although it never hurts to have as much physical memory as your system can handle, here are two things you can check within your system to alleviate the memory bottleneck. *Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.Check to see if you have additional space on your disk drive that you could use to expand the size of your swap file. Normally, the bigger the initial size of your swap file, the better, in performance terms.*

Paged In

The number of disk block paged into core from disk which occurs when the server is experiencing a shortage of RAM. Any nonzero value indicates excessive swap activity. In any virtual memory environment, it is not uncommon to see RAM pages moved out to a swap disk. This is a special disk area in UNIX that's reserved for holding memory pages so that the processor is capable of addressing RAM in excess of its full capability. While page-out of operations are a normal part of any server's operations, page-in operations indicate that the real amount of available RAM has been exceeded and that additional RAM pages are required on the server.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Tab
Correction	High values indicate the possible need to increase the amount of RAM on the system.

Paged Out

The number of disk block paged out of core from disk. In any virtual memory environment, it is not uncommon to see RAM pages moved out to a swap disk. This is a special disk area in UNIX that's reserved for holding memory pages so that the processor is capable of addressing RAM in excess of its full capability. While page-out of operations are a normal part of any server's operations, page-in operations indicate that the real amount of available RAM has been exceeded and that additional RAM pages are required on the server.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Tab

Pages Input/Sec

The Pages Input/Sec statistic is the number of pages read from disk to resolve hard page faults. Hard page faults occur when a process requires code or data that is not in its working set or elsewhere in physical memory, and must be retrieved from disk.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Tab
Metrics	This value was designed as a primary indicator of the kinds of faults that cause system-wide delays. It includes pages retrieved to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. This counter counts numbers of pages, and can be compared to other counts of pages, such as Memory: Page Faults/sec, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
Correction	Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck. *Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.*

Pages Output/Sec

The Pages Output/Sec statistic is the number of pages written to disk to free up space in physical memory. Pages are written back to disk only if they are changed in physical memory. A high rate of pages output might indicate a memory shortage.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Tab
Metrics	Windows NT writes more pages back to disk to free up space when low in physical memory. This counter counts numbers of pages, and can be compared to other counts of pages, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
Correction	Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck. *Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.*

Memory Available

The Memory Available section of the OS Detail page succinctly communicates the general overall performance levels of the server's memory. The statistics available in this section depend on the platform of operating system. The table below describes these statistics.

Statistic	Description	Metrics/Correction
Physical Memory	The Physical Memory statistic is the amount of committed virtual memory, in megabytes. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk.	This counter displays the last observed value only and not an average.
Physical	The Physical statistic is the total physical memory available.	N/A
Paged Memory	The Paged Memory statistic is the amount of committed virtual memory, in bytes. Used Memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive.	This counter displays the last observed value only; it is not an average.
Swap Memory	The Swap Memory statistic is the free, unreserved swap space.	Increase the size of your swap file or add additional files if you consistently see a shortage of free, unreserved swap space.
Swap	The Swap statistic is the total swap memory being used.	N/A
Total Physical	The Total Physical statistic is the total physical memory available.	N/A
Total Swap	The Total Swap statistic is the total swap memory available.	N/A
Swap-ins	The number of memory pages paged in from swapspace.	If the system is consistently low on free memory (over a 30 second average), the memory scheduler will start to swap out processes. Increase the size of your swap file or add additional files.
Swap-outs	The number of memory pages paged in from swapspace.	If the system is consistently low on free memory (over a 30 second average), the memory scheduler will start to swap out processes. Increase the size of your swap file or add additional files.
Buffers	The Buffers statistic represents the amount of buffers used by the kernel in kilobytes.	N/A

Page Faults/Sec	The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.	This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval. If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Tab

Buffer Size

The Buffer Size statistic represents the amount of cache used by the kernel in kilobytes.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Tab

Cache Size

The Cache Size statistic represents the amount of cache used by the kernel in kilobytes.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Tab

Page Replacements

The following statistics are available in this section:

Memory Freed (Pages/sec) Clock Algorithm Scans (Pages/sec)

Memory Freed (Pages/sec)

Freed or destroyed (Kb/s).

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Tab

Clock Algorithm Scans (Pages/sec)

The VMM uses a technique known as the clock algorithm to select pages to be replaced. This technique takes advantage of a referenced bit for each page as an indication of what pages have been recently used (referenced). When the page-stealer routine is called, it cycles through the PFT, examining each page's referenced bit.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Tab

Cache Efficiency

The Cache Efficiency section of the Memory tab succinctly communicates the general overall performance levels of the server's memory. The following statistics are available in this section:

Copy Read Hits % Data Map Hits % MDL Read Hits %

Pin Read Hits %

Copy Read Hits %

The Copy Read Hits % statistic is the percentage of cache copy read requests that hit the cache and does not require a disk read to provide access to the page in the cache.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Tab
Metrics	When the page is pinned in the memory, the page's physical address in the file system cache will not be altered. A copy read is a file read operation where a page in the cache is copied to the application's buffer. Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.
Correction	Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate

Data Map Hits %

The Data Map Hits % statistic is the percentage of data maps in the file system cache that could be resolved without having to retrieve a page from the disk.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Tab
Metrics	Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.
Correction	Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate

MDL Read Hits %

The MDL Read Hits % statistic is the percentage of Memory Descriptor List Read requests to the file system cache that hit the cache and does not require disk access to provide memory access to the pages in the cache.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Tab
Metrics	Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.
Correction	Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Pin Read Hits %

The Pin Read Hits % statistic is the percentage of pin read requests that hit the file system cache and does not require a disk read in order to provide access to the page in the file system cache.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Memory Tab
Metrics	Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.
Correction	Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Space Tab

The Space tab of the OS Detail page includes the following sections:

- Device Summary
- Device Detail

Disk Space Free

The Disk Space Free metric displays the amount of free disk space in megabytes.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Space Tab

Disk Space Detail

The Disk Space Detail section of the Space tab succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

Statistic	Description
Partition	The drive letter of the disk.
Local Filesystem	The name of the file system.
Type	The type of file system.
Total Space	Total size of the disk/device's capacity expressed in MBs.
Used Space	Amount of MBs currently allocated on the particular disk/device.
Free Space	Amount of MBs currently unallocated and free on the particular disk/device.
Capacity	The percentage of space used on the device.
Mounted On	The mount point of the device.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Space Tab

Device Summary

The Device Summary section of the Space tab displays a bar chart for all devices. The Device Summary section succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

Statistic	Description
Used	Amount of MBs currently allocated on the particular disk/device.
Free	Amount of MBs currently unallocated and free on the particular disk/device.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Space Tab

Device Detail

The Device Detail section of the Space tab succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

Statistic	Description
Partition	The drive letter of the disk.
Local Filesystem	The name of the file system.
Type	The type of file system.
Total Space	Total size of the disk/device's capacity expressed in MBs.
Used Space	Amount of MBs currently allocated on the particular disk/device.
Free Space	Amount of MBs currently unallocated and free on the particular disk/device.
Capacity	The percentage of space used on the device.
Mounted On	The mount point of the device.

Note: The statistics available in this section depend on the platform of operating system.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Space Tab

Network Tab

The Network tab of the OS Detail page succinctly communicates the general overall performance levels of the server's networking. The Network tab of the OS Detail page includes the following sections:

- Network Details
- Packet Collisions
- Packet Discards
- Packet Errors
- Transmission Rate
- Transmission Rate (Bytes)
- Transmission Queue Length

Note: The sections available on this tab depend on the platform of operating system.

Network Details

The statistics available in this section depend on the platform of operating system. The table below describes the information available in this section:

Column	Description
Network Interface	The name of network interface.
INET Address/Address	The IP address assigned to the network interface.
Pkts Sent/Sec	The number of packets sent per second.
Pkts Received/Sec	The number of packets received per second.
Sent (KB/Sec)	The number of bytes sent per second.
Received (KB/Sec)	The number of bytes received per second.
Out Pkts Discarded	The number of outbound packets discarded.
In Pkts Discarded	The number of inbound packets discarded.
Out Pkt Errors	The number of outbound packet errors.
In Pkt Errors	The number of inbound packet errors.
Queue Length	The queue length.
Collisions	The number of collisions.
Packets Discarded	The number of packets discarded.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Network Tab

Transmission Rate

The Transmission Rate section of the Network tab succinctly communicates the packet transmission rate. The following statistics are available in this section:

Packets Sent/Sec	The Packets Sent/Sec statistic is the number of packets sent over each network adapter per second.
Packets Received/Sec	The Packets Received/Sec statistic is the number of packets received over each network adapter per second.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Space Tab

Transmission Rate (Bytes)

The Transmission Rate (Bytes) section of the Network tab succinctly communicates the packet transmission rate. The following statistics are available in this section:

Sent (KB/Sec)	The Sent (KB/Sec) statistic is the rate at which bytes are sent over each network adapter including framing characters
Received (KB/Sec)	The Received (KB/Sec) statistic is the rate at which bytes are received over each network adapter, including framing characters.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Space Tab

Transmission Queue Length

The number of pending outgoing packets on either the software transmit queue or the hardware transmit queue

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Space Tab

Packet Collisions

A collision occurs when two devices on the network sense that the network is idle and end up trying to send data at the same time. Some collisions are normal on a repeated network, but excess collisions can cause serious performance problems.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Space Tab

Correction

Contact your network administrator if there is high rate of persistent Packet Collisions.

Packet Discards

The Packet Discards section of the Network tab succinctly communicates the packet discard rate. The following statistics are available in this section:

Outbound Packets Discarded	The Outbound Packets Discarded statistic is the number of outbound packets that were discarded even though no errors had been detected to prevent from being transmitted.
Inbound Packets Discarded	Received Packets Discarded is the number of received packets that were discarded even though no errors had been detected to prevent from being transmitted.
Packet Discards	The Packet Discards statistic represents the number of network packets discarded.

Location	IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Space Tab
Correction	A possible reason for discarding such a packet could be to free up buffer space. Contact your network administrator if there is high rate of persistent packet discards.

Packet Errors

The Packet Errors section of the Network tab succinctly communicates the packet error rate. The following statistics are available in this section:

Outbound Packet Errors	The Outbound Packet Errors statistic is the outbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.
Inbound Packet Errors	The Packets Received Errors statistic is the received packets that contained errors preventing them from being deliverable to a higher-layer protocol.

Location IBM DB2 (W/U/L) Performance Analyst statistics (DBArtisan) > **OS View (DBArtisan - DB2 LUW Performance Analyst)** > Space Tab

Top SQL View (DBArtisan - DB2 LUW Performance Analyst)

One or two bad queries can cause a lot of trouble for the remaining sessions in a database, and so can one or two bad queries. It is important to find them before they get into a production system, but sometimes a few sneak through.

By applying custom filters and performance-related thresholds, the Top SQL view locates inefficient SQL. By applying filters to certain I/O and statistical counters, you hope to isolate queries that far exceed their nearest competitors in the same area (like disk reads). When you find them, you should reduce the number of sorts a query performs. Or, for a query that returns only a few records, you should try to minimize the number of rows a query processes.

Metrics

When you begin to look for inefficient SQL in a database, there are two primary questions you need to answer:

- What has been the worst SQL that's historically been run in my database?
- What is the worst SQL that's running right now in my database?

When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It may be that the query is in an inefficient loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

The following tabs are available on the Top SQL page:

Summary tab Criteria tab SQL Details tab

Dynamic SQL Cache tab

Summary tab

The Summary tab is divided into two sections:

- Top SQL Statements
- Top SQL Details

Top SQL Statements

A lot of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements or stored procedures can drag an otherwise well-configured database down in terms of user response times.

Before you can identify problem SQL in your database, you have to ask the question of what 'bad SQL' is. What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient

SQL; so there is no way to sufficiently answer this question to every SQL Server professional's satisfaction. The Top SQL results grid displays the top SQL statements across various performance categories including Reads, Writes, CPU statistics, and Elapsed Time.

Top SQL Details

This section shows the details of the top SQL statements. The columns presented here include:

Usage: % of all operations of this type performed by the selected sql statement.

SQL text: Text of the selected SQL statement.

Total Reads: Total number of read operations performed by the selected SQL statement.

Total Writes: Total number of write operations performed by the selected SQL statement.

Executions: Number of times the selected SQL statement was executed.

First Start Time: First time the selected SQL statement was executed.

Avg CPU Time: Avg CPU time used by the selected SQL statement.

Avg Elapsed Time: Avg Elapsed time for selected SQL statement.

Criteria tab

The Top SQL view displays all SQL running or executed in the current datasource. You can use the lists and boxes to reorder the display according to your specific needs. The Criteria tab of the Top SQL page includes the following sections that you can mix and match to give you the information you want to find.

- Demographic Criteria
- Statistical Criteria
- Sort Criteria

SQL Details tab

The SQL Details tab presents the following sections:

Top SQL Results Selected SQL Text Selected SQL Last Snapshot Details

Top SQL Results

This section shows the details of the SQL statements and (their metrics) captured at each refresh of the performance analyst. The information presented here includes

SQL text: Partial text of the captured SQL statement. The full text is shown in the selected SQL text section.

Agents Top: The maximum number of agents that were used when executing the selected statement at the time of last refresh

Max Agents Top: the maximum number of agents that were used when executing the selected statement over all captured executions

Fetch Count: Number of successful fetches performed for the specific statement at the time of last refresh

Avg Fetch Count: Avg number of successful fetches for the selected statement over all its captured executions

Int. Rows Deleted: Number of internal rows deleted for the selected statement as a result of internal activity

Avg Int Rows Deleted: over all its captured executions

Int Rows Inserted

Avg Int Rows Inserted over all its captured executions

Int. Rows Updated

Avg Internal Rows Updated over all its captured executions

1. Agents
2. Rows Estimate

Avg # Rows Estimate over all its captured executions

Cost Estimate

Avg Cost Estimate over all its captured executions

Rows Read

Avg Rows Read over all its captured executions

Rows Written

Avg Rows Written over all its captured executions

Sorts

Avg Sorts over all its captured executions

Sys CPU Time

Avg Sys CPU Time over all its captured executions

Usr CPU Time

Avg Usr CPU Time over all its captured executions

Total Sort Time

Avg Total Sort Time over all its captured executions

Buffer pool Hit Ratio

Overall Buffer pool Hit Ratio over all its captured executions

Temp Data Hit Ratio

Overall Temp Data Hit Ratio over all its captured executions

Selected SQL Text

The Selected SQL Text window allows you to select any of the Top SQL result rows and get the complete SQL text. From here you can open an Explain Plan or Tune the SQL.

Selected SQL Last Snapshot Details

This displays attributes for the last snapshot of the selected SQL statement

Dynamic SQL Cache tab

The Dynamic SQL Cache tab of the Top SQL view, displays the following sections:

Dynamic SQL Cache Details	Dynamic SQL Statement	Execution and Preparation Details
Bufferpool Analysis	Execution Time Details	

Dynamic SQL Cache Details

The DB2 statement cache stores packages and statistics for frequently used SQL statements. This section shows the statements in this cache and the metrics associated with each statement.

Metrics

By examining the contents of this cache, you can identify the dynamic SQL statements that are most frequently executed and the queries that consume the most resources. Using this information, you can examine the most commonly executed and most expensive SQL operations, to determine if SQL tuning could result in better database performance.

Dynamic SQL Statement

The Dynamic SQL Text window allows you to select any of the Dynamic SQL Cache Detail and get the complete SQL text. From here you can open an Explain Plan or Tune the SQL.

Execution and Preparation Details

Here you see the number of times the selected SQL statement was compiled and executed.

Metrics

You can use the execution information to identify the most commonly executed SQL statements in your system. You will probably see the greatest performance benefit from tuning them.

You can use the compilation information to determine the number of different compilations for the selected SQL statement. Use this information together with the number of executions to determine if a bad compilation environment is skewing the metrics.

Bufferpool Analysis

This section shows the various bufferpool usage metrics for the selected SQL statement.

Metrics

This information helps you determine the scope when you tune bufferpools.

Execution Time Details

The information presented here may be used to identify the longest running SQL.

Metrics

The longest running SQL statements together with the most frequently used SQL statements help you identify the most likely candidates for SQL tuning.

Space Analyst (DBArtisan)

Space management plays a key role in a database's uptime and overall performance. Because of this, it's important that a database administrator understand the various types of space bottlenecks that can occur in a database.

Embarcadero Space Analyst provides sophisticated diagnostic capabilities to help you effectively locate and troubleshoot bottlenecks and performance inefficiencies that result from the challenges of space management. Running inside the Embarcadero DBArtisan console, Space Analyst automates all diagnostic and correction procedures so any space-related headache can be quickly detected and corrected before it reaches critical mass.

The table below describes the major sections of the help available for the Space Analyst.

Section	Description
Welcome to Space Analyst (DBArtisan)	Provides information about opening, and licensing Space Analyst.
Space Analyst Product Design (DBArtisan)	Provides conceptual information about Space Analyst's product design and navigation.
Using Space Analyst (DBArtisan)	Shows how to use each component of the application. Using Space Analyst breaks each process and procedure into simple step-by-step instructions.
Space Analyst for Oracle tutorial (DBArtisan)	Helps you get started with Embarcadero Space Analyst. Provides the foundation you need to explore the many features and benefits of Space Analyst.
Space Analyst for Microsoft SQL Server Tutorial (DBArtisan)	Helps you get started with Embarcadero Space Analyst. Provides the foundation you need to explore the many features and benefits of Space Analyst.
Space Analyst for Sybase ASE Tutorial (DBArtisan)	Helps you get started with Embarcadero Space Analyst. Provides the foundation you need to explore the many features and benefits of Space Analyst.
Space Analyst for IBM DB2 for Linux, Unix, and Windows Tutorial (DBArtisan)	Helps you get started with Embarcadero Space Analyst. Provides the foundation you need to explore the many features and benefits of Space Analyst.
Space Analyst Expert Guide (DBArtisan)	Provides detailed information on each tab of the Space Analyst and discusses fragmentation concepts.

Topics

- Welcome to Space Analyst (DBArtisan)
- Space Analyst Application Basics (DBArtisan)

- Using Space Analyst (DBArtisan)
- Space Analyst for Oracle tutorial (DBArtisan)
- Space Analyst for Microsoft SQL Server Tutorial (DBArtisan)
- Space Analyst for Sybase ASE Tutorial (DBArtisan)
- Space Analyst for IBM DB2 for Linux, Unix, and Windows Tutorial (DBArtisan)
- Space Analyst Expert Guide (DBArtisan)

Welcome to Space Analyst (DBArtisan)

Space Analyst provides sophisticated diagnostic capabilities to troubleshoot bottlenecks and performance problems that result from poor space management. Space Analyst helps ensure that database storage is being used as efficiently as possible.

Space Analyst also includes a Reorganization Wizard as part of the Space Management Wizard (DBArtisan) that handles full storage structure and object reorganization needs.

For more information, see:

- What You Should Know Before Installing Space Analyst (DBArtisan)
- Opening an Analyst (DBArtisan)
- Licensing Space Analyst (DBArtisan)
- Using Space Analyst (DBArtisan)

Topics

- What You Should Know Before Installing Space Analyst (DBArtisan)
- Opening an Analyst (DBArtisan)
- Uninstalling a Repository (DBArtisan)
- Licensing Space Analyst (DBArtisan)

What You Should Know Before Installing Space Analyst (DBArtisan)

When you install the Analyst, several Embarcadero-generated stored objects are pushed onto the database server.

These objects are as follows:

SQL Server

Procedures

sp_embt_sa_database_map	sp_embt_sa_database_overview	sp_embt_sa_dbbottlenecks
sp_embt_sa_filegroups	sp_embt_sa_files	sp_embt_sa_index_space_general
sp_embt_sa_index_space_reorg	sp_embt_sa_index_space_reorg_summary	sp_embt_sa_server_overview
sp_embt_sa_table_space_general	sp_embt_sa_table_space_reorg	sp_embt_sa_table_space_reorg_summary
sp_embt_sa_index_reorg_summary	sp_embt_sa_table_reorg_summary	sp_embt_sa_version

Sybase ASE

Procedures

sp_embt_sa_dbspace	sp_embt_sa_device_summary	sp_embt_sa_device_detail
sp_embt_sa_database_overview	sp_embt_sa_db_obj_space	sp_embt_sa_database_detail
sp_embt_sa_db_frag_summary	sp_embt_sa_index_diag_summary	sp_embt_sa_index_diag
sp_embt_sa_server_overview	sp_embt_sa_table_diag_summary	sp_embt_sa_table_diag
sp_embt_sa_version		

Note: No procedures are installed in Space Analyst for Oracle or DB2.

Opening an Analyst (DBArtisan)

To open the Space Analyst, do the following:

1 On the Analyst menu, click Space Analyst.

Note: The first time that you open the Space Analyst, DBArtisan alerts you that you need to install the Analyst's procedures. If you elect to install the procedures, an **Analyst Repository Installation** dialog opens, prompting for credentials and other security details. When you install the Analyst, several procedures and tables are pushed onto the database server. For a list of these procedures and tables, see What You Should Know Before Installing Space Analyst (DBArtisan).

Uninstalling a Repository (DBArtisan)

To uninstall an Analyst repository, do the following:

1 On the Command menu, select Uninstall Repository.

The Analyst Repository Uninstallation Wizard opens.

Analyst Repository Uninstallation Wizard - Panel 1

The table below describes the options and functionality on this panel of the Analyst Repository Uninstallation Wizard:

Option	Description
Administrator Login Information	Displays the login name, password, and login as information.
Database to hold the procedures	Displays the database that stores the procedures.
Uninstall	Uninstalls the repository and opens Panel 2. For more information, see Analyst Repository Uninstallation Wizard - Panel 2.

Analyst Repository Uninstallation Wizard - Panel 2

The table below describes the options and functionality on this panel of the Analyst Repository Uninstallation Wizard:

Option	Description
Repository Login	Displays repository login.
Execution Log	Displays the repository execution log and links to ETRepoLog.txt.
Execution Error(s)	Displays any errors.
Finish	Closes wizard.

Licensing Space Analyst (DBArtisan)

Availability of this feature depends on your licensing. For more information, see Licensing.

Space Analyst Application Basics (DBArtisan)

The Space Analyst utility assists DBAs with space diagnostics and reorganizations by displaying pertinent reorganization diagnostics and providing a Reorganization Wizard as part of the Space Management Wizard (DBArtisan) that builds scripts that you can execute or schedule to perform reorganizations.

For more information, see Product Design.

Space Analyst Product Design (DBArtisan)

The Space Analyst product design includes tabs for Oracle, Microsoft SQL Server and IBM DB2.

Oracle Space Analyst Tabs

The Space Analyst includes the following tabs:

Database Overview Tab Datafiles Tab Tablespace Map Tab

Tables Tab Indexes Tab

Microsoft SQL Server Space Analyst Tabs

The Space Analyst includes the following tabs:

Storage Overview Tab Files/File Groups Tab Map Tab

Databases Tab Objects Tab

IBM DB2 Space Analyst Tabs

The Space Analyst includes the following tabs:

Database Overview Tab Tablespaces Tab Tablespace Map Tab

Tables Tab Indexes Tab

Sybase ASE Space Analyst Tabs

The Space Analyst includes the following tabs:

Storage Overview Tab Database Devices Databases Tab

Tables Tab Indexes Tab

Database Overview Tab

The Database Overview tab displays the storage overview for the DB2 datasource.

There are four distinct sections displayed on the first **Database Overview** tab of Space Analyst:

- Database Overview
- Log Overview (DB2 Only)
- Tablespace Space Overview
- Tablespace Fragmentation Overview
- Tablespace Objects Overview

For more information, see Space Analyst Tabs.

Data Files Tab

The Data Files tab displays datafile fragmentation information, and displays how much space in each datafile is assigned to tables, indexes, clusters, and rollback segments.

For more information, see Space Analyst Tabs.

Tablespace Map Tab

The Tablespace Map tab allows for tablespace mapping.

For more information, see Space Analyst Tabs.

Tables Tab

The Tables tab lets you identify tables with wasted space. The Space Analyst computes a block inefficiency ratio that quickly communicates the severity of the space problem. Space Analyst provides a summary count of all tables in a database that violate a customizable block inefficiency threshold, details on the tables, and the extent of wasted space.

For more information, see Space Analyst Tabs.

Indexes Tab

The Indexes tab lets you select index reorganization candidates. The tab also offers the option to obtain deleted leaf row information for each index, although such data gathering can take some time, especially on large systems.

Space Analyst runs diagnostics against your database and graphically display a count of indexes that are found to contain various types of space problems.

For more information, see Space Analyst Tabs.

Tablespaces Tab

The Tablespaces tab displays the Tablespace Container allocations and Database Partition Breakdown for each Tablespace.

There are three distinct sections displayed on the **Tablespaces** tab of Space Analyst:

- Space Allocation by Container
- Space Allocation by Node
- Tablespaces

Using Space Analyst (DBArtisan)

Using Space Analyst covers how to use each component of the application. The Using chapter breaks each process and procedure into simple step-by-step instructions.

The Space Analyst includes the following tabs, by platform:

Space Analyst for Oracle

Database Overview Tab
Datafiles Tab
Tablespace Map Tab
Tables Tab
Indexes Tab

Space Analyst for Microsoft SQL Server

Storage Overview Tab
Files/File Groups Tab
Map Tab
Databases Tab
Objects Tab

Space Analyst for Sybase ASE

The Space Analyst includes the following tabs:

Storage Overview Tab
Database Devices Tab
Databases Tab
Tables Tab
Indexes Tab

Space Analyst for IBM DB2 for Linux, Unix, and Windows

The Space Analyst includes the following tabs:

Storage Overview Tab
Datafiles Tab
Tablespace Tab
Tables Tab
Indexes Tab

Topics

- Space Analyst for Oracle and IBM DB2 for Linux, Unix, and Windows - Database Overview Tab (DBArtisan)
- Space Analyst for Oracle and IBM DB2 for Linux, Unix, and Windows - Datafiles Tab (DBArtisan)
- Space Analyst for Oracle - Tablespace Map Tab (DBArtisan)
- Space Analyst for Oracle, Sybase ASE, and IBM DB2 for Linux, Unix, and Windows - Tables Tab (DBArtisan)
- Space Analyst for Oracle - Tables Tab (DBArtisan)
- Space Analyst - Tables Tab (DBArtisan)
- Space Analyst for Sybase ASE and IBM DB2 for Linux, Unix, and Windows - Indexes Tab (DBArtisan)

  - Space Analyst for Oracle - Indexes Tab (DBArtisan)
  - Space Analyst - Storage Overview Tab (DBArtisan)
  - Space Analyst for Microsoft - File Groups/Files Tab (DBArtisan)
  - Space Analyst for Microsoft - Map Tab (DBArtisan)
  - Space Analyst for Microsoft - Databases Tab (DBArtisan)
  - Space Analyst for Microsoft - Objects Tab (DBArtisan)
  - Space Analyst for Sybase ASE - Database Devices Tab (DBArtisan)
  - Space Analyst for Sybase ASE - Databases Tab (DBArtisan)
  - Space Analyst for IBM DB2 for Linux, Unix, and Windows - Tablespaces Tab (DBArtisan)
  - Space Analyst Functionality (DBArtisan)
  - Space Management Wizard (DBArtisan)

Space Analyst for Oracle and IBM DB2 for Linux, Unix, and Windows - Database Overview Tab (DBArtisan)

The **Database Overview** tab lets you view the global storage picture of your database.

The table below describes the options and functionality on the **Database Overview** tab of Space Analyst for Oracle:

Option	Description
Database Overview	Shows how much space in KB is taken up in the database by various object types (tables, indexes, etc.) along with a pie chart that represents the percentage of each object type. You can switch between the default graphical view and a grid view. NOTE: Objects owned by SYS and SYSTEM are not included in this breakdown.
Tablespace Space Overview	When in graphical mode, this section contains a bar chart graph that displays all tablespaces and how much total space each has allocated. When in grid mode, Space Analyst displays detailed information about all tablespaces (status, total, used, free, percent free, etc.) ORACLE ONLY: When in grid mode, Space Analyst displays tablespace status. You can switch between the default graphical view and a grid view.
Tablespace Fragmentation Overview	Contains fragmentation information for all tablespaces and displays how badly fragmented a tablespace or datafile is. Higher scores for the tablespace's fragmentation index indicate little or no fragmentation, while lower scores indicate fragmentation. You can switch between graphical and grid views of tablespace fragmentation. If fragmentation is present, you can view detailed object and fragmentation information for the tablespace in the tablespace map. You can correct fragmentation problems via Space Analyst's Reorganization Wizard. You can switch between the default graphical view and a grid view.
Tablespace Objects Overview	Contains a grid that displays how much space in each tablespace is assigned to tables, indexes, clusters, and rollback segments. ORACLE ONLY: When in grid mode, Space Analyst displays tablespace LOB data values.

The table below describes the options and functionality on the **Database Overview** tab of Space Analyst for IBM DB2 for Linux, Unix, and Windows:

Option	Description
Database Overview	Shows how much space in KB is taken up in the database by various object types (tables, indexes, etc.) along with a pie chart that represents the percentage of each object type. You can switch between the default graphical view and a grid view.
Log Space Overview	Displays all of the log options for the database: Total Used Total Free Block Log When Disk Full Log File Size User Exit Primary Logs Secondary Logs Maximum Secondary Log Space Used Maximum Total Log Space Used
Tablespace Space Overview	When in graphical mode, this section contains a bar chart graph that displays all tablespaces and how much total space each has allocated. When in grid mode, displays detailed information about all tablespaces: Name Type State Used Space (MB) Free Space (MB) Content Type (Any, Long, Temp, System temp) Number of containers NOTE: You can switch between the default graphical view and a grid view.
Tablespace Objects Overview	Contains a grid that displays how much space in each tablespace is assigned to each object. Also displays tablespace LOB data values. Displays detailed information about all tablespace objects: Tablespace Name Table - total space used by tables Index - total space used by indexes Extent Size - tablespace extent size Page size - Tablespace page size

Available Functionality

The following functionality is available in the Command menu, shortcut menu, and toolbar:

Note: Functionality differ by platform.

Analysis Report	Convert to Dictionary-Managed	Convert to Locally-Managed
Extract	Estimate Size	Map

Open Editor. For information on modifying database objects, see Modifying objects using editors.

Open Wizard	Reorganize	Update Statistics
Tablespaces Wizard (DB2 LUW)	Threshold Options	

For more information, see:

Correcting Tablespace 'out of space' Problems

Space Analyst Tabs

Space Analyst for Oracle and IBM DB2 for Linux, Unix, and Windows - Datafiles Tab (DBArtisan)

The **Data Files** tab lets you view fragmentation and object placement information for data files.

The table below describes the options and functionality on the **Datafiles** tab of the Space Analyst:

Option	Description
Datafile Fragmentation	Displays fragmentation about datafiles. Higher scores for a datafile's fragmentation index indicate little or no fragmentation, while lower scores indicate fragmentation. If fragmentation is present, you can view detailed object and fragmentation information for the datafile's tablespace in the tablespace map. You can correct fragmentation problems via Space Analyst's Reorganization Wizard. NOTE: Datafiles for temporary tablespaces are not included in the Datafile Fragmentation grid.
Datafile Objects Overview	Contains a grid that displays how much space in each datafile is assigned to tables, indexes, clusters, and rollback segments.

Tip: To open a Tablespace Editor, right-click a datafile in any grid, and then select Open. To extract the tablespace DDL in an ISQL editor, right-click a datafile in any grid, and then select Extract.

Available Functionality

The following functionality is available in the Command menu, shortcut menu, and toolbar:

Note: Functionalists differ by platform.

Analysis Report	Convert to Dictionary-Managed	Convert to Locally-Managed
Estimate Size	Extract	Map

Open Editor. For information on modifying database objects, see Modifying objects using editors.

Reorganize

Threshold Options

For more information, see Space Analyst Tabs.

Space Analyst for Oracle - Tablespace Map Tab (DBArtisan)

The **Tablespace Map** tab lets you view a graphical layout regarding the physical placement of every object inside a selected tablespace. The tablespace map also helps identify fragmentation problems for a tablespace (honeycomb and bubble). An object grid is also present that lets you view detailed information about every object's storage properties.

The Space Info section contains information regarding the total, used, and free space (in MB) for the tablespace.

The View By section allows you to switch between two different tablespace map views:

Option	Description
Object Type	The normal tablespace map mode, which is color-coded by object type. Pressing the legend button shows the various colors assigned to the object types contained within the tablespace.
Extent Problems	When in this mode, the tablespace map highlights in red any object containing one or both of these object extent problems:· - Max extent problem - the object has reached or is near its maximum extent limit. - Extent deficit - the object may not be able to allocate its next extent because the tablespace does not currently have enough contiguous free space. This problem will be highlighted even if the tablespace has its autoextend property for one or more datafiles set to true.

When in Object Type mode, the accompanying grid of the tablespace map contains the following data:

Data	Description
Name	The name of the object in owner.object name.partition name format.
Extents	The total number of allocated extents for the object.
Blocks	The total number of allocated blocks for the object.
Bytes	The total number of allocated bytes for the object.
Type	The object type (table, index, etc.)
Initial Extent	The initial space extent (in bytes) allocated to the object.
Next Extent	The next extent (in bytes) that the object will attempt to allocate when more space for the object is required.
Percent Increase	The percent increase property for the object.

When in Extent Problems mode, the accompanying grid of the tablespace map contains the following data:

Data	Description
Name	The name of the object in owner.object name.partition name format.
Extents	The total number of allocated extents for the object.
Blocks	The total number of allocated blocks for the object.
Bytes	The total number of allocated bytes for the object.
Type	The object type (table, index, etc.)
Initial Extent	The initial space extent (in bytes) allocated to the object.
Next Extent	The next extent (in bytes) that the object will attempt to allocate when more space for the object is required.
Percent Increase	The percent increase property for the object.
Max Extent Percent	A percentage indicating how close the object is to reaching its maximum extent limit. For example, a 100 indicates an object that has reached its maximum extent limit.
Extendibility	Indicates if the object can allocate its next extent. If 'N', this is because the tablespace does not currently have enough contiguous free space. This problem will be highlighted even if the tablespace has its autoextend property for one or more datafiles set to true.

When viewing a tablespace in the graphical map, there are two broad categories of fragmentation that you will want to be on the lookout for: free space honeycombs and free space bubbles. Honeycomb fragmentation is really not difficult to detect or handle and occurs when two free space extents in a tablespace reside next to one another. For example, rather than having one large free space extent of 6MB, a tablespace might have two extents that are next to each other (a honeycomb) that make up 3MB and 3MB. If an incoming object required a single 4MB extent allocation, Oracle would have to coalesce the honeycomb to make room for the object. The necessary coalesce operation would cause a small degree of performance degradation, however the situation would be resolved and no space-related bottleneck would remain.

Fragmentation bubbles, however, are another story and are the traditional reason DBAs have had to perform full tablespace reorganizations. Bubbles occur when objects are continuously added and dropped in a tablespace, with the end result being free space pockets (or bubbles) being interspersed between objects that remain in the tablespace. This can become problematic when, for example, an incoming object requires a single 5MB extent allocation, but there is not enough contiguous space in the tablespace to meet the demand. Even if the total amount of free space in the tablespace equals the required 5MB, it will do no good if that 5MB is made up of 5 non-contiguous blocks of free space. In this situation, Oracle has no alternative but to either extend the datafile to make room for the object (if the tablespace's datafile has the AUTOEXTEND property enabled), or return an error and deny the object entry into the tablespace.

Space Analyst makes it easy to recognize both of these bottleneck headaches. When the map is viewed in Object Type mode, pockets of green seen throughout the tablespace indicate bubble fragmentation, while two or more blocks of green segments viewed next to one another in the map indicate honeycomb fragmentation.

The **Tablespace Map** tab includes the following buttons:

Button	Description
Legend	Opens the Tablespace Map Legend.
Display	Opens the Display window. The Display window lets you specifies the magnification level for the Tablespace Map.
Fit to Window	Fits the Tablespace Map to the window.
Full Screen	Fits the Tablespace Map to the full screen.
Print Map	Opens the **Print** dialog to print the Tablespace Map.

Available Functionality

The following functionality is available in the Command menu, shortcut menu, and toolbar:

Analysis Report
Estimate Size
Open Wizard
Tablespaces Wizard (DB2 LUW)
Convert to Dictionary-Managed
Extract
Reorganize
Threshold Options
Convert to Locally-Managed
[DBAObjectEditors.htm#ORATablespacesEditor Open Editor]
Update Statistics

For more information, see Space Analyst Tabs.

Space Analyst for Oracle, Sybase ASE, and IBM DB2 for Linux, Unix, and Windows - Tables Tab (DBArtisan)

The T**ables** tab displays diagnostic information regarding table space inefficiencies that may exist in your database.

The Table Reorganization Candidate Summary section uses a bar chart to count the various types of table space inefficiencies in your database. Note that the reorganization threshold options are used to compute the counts for each type of space inefficiency.

The table below describes the four distinct types of table space inefficiencies that Space Analyst computes for Oracle and Sybase ASE:

Problem	Description
Block Inefficiencies	These tables have wasted space that exceeds the allowable percentage of wasted space set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Tables with wasted space can take longer to scan and therefore contribute to slower response times.
Next Extent Problems	These tables cannot obtain their next extent of free space due to a lack of needed contiguous free space in the tablespace. Tables suffering from this problem may produce error conditions in the database if more data is added to them. This problem will be highlighted even if the tablespace that contains the table has its autoextend property for one or more datafiles set to true.
Chain Problems	These tables suffer from a chained or migrated row problem. The percentage of chained or migrated rows to overall rows contained in the table exceeds the allowable percentage of chained rows set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Because chained/migrated rows cause more I/O to occur when they are accessed, overall response times can be degraded when these tables are referenced in SQL queries.
Max Extent Problems	These tables are nearing their maximum extent limit and have exceeded the allowable maximum extent percentage set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Tables suffering from this problem may cause space-related errors to result if more data is added to them.

The Table Reorganization Candidate Details section provides filters and a grid that you can use to view detailed space diagnostics for all or selected tables. The table below describes the grid data:

Data	Description
Table Owner	The owner of the table.
Table Name	The name of the table.
Segment Type	Indicates if the object is a table, table partition, etc.
Table (KB)	The total allocated space of the table in KB.
Number of Rows	The number of rows that the table contains. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Number of Blocks	The total number of blocks allocated to the table. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Empty Blocks	The number of empty blocks that the table contains. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Highwater Mark	The last block that Oracle will scan up to when searching the table for data. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Average Used Blocks	The average number of blocks that the table is using. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Block Inefficiency	A percentage that indicates how much wasted space (space that the table is not using) that the table contains. High percentages indicate tables that may be candidates for reorganization. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Chained Row Percent	A percentage that indicates how many rows in the table suffer from chaining or migration. High percentages indicate tables that may be candidates for reorganization. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Max Extent Percent	A percentage indicating how close an object is to reaching its maximum extent limit. High percentages indicate tables that may be candidates for reorganization.
Number of Extents	The number of extents that the table consumes.
Max Extents	The maximum extent limit for the table.
Extent Deficit	Indicates if the object can allocate its next extent. If 'N', this is because the tablespace does not currently have enough contiguous free space. This problem will be highlighted even if the tablespace has its autoextend property for one or more datafiles set to true.
Next Extent	The next extent (in bytes) that the table will attempt to allocate when more space for the table is required.

Max Free Space	The largest contiguous block of free space that exists in the tablespace where the table resides.
Tablespace Name	The tablespace that contains the table.

IBM DB2 for Linux, Unix, and Windows

The table below describes the four distinct types of Reorganization Candidate Summary that Space Analyst computes for IBM DB2 for Linux, Unix, and Windows:

Problem	Description
Overflow Problems	Displays the number of tables that fall outside the preset limits for the thresholds of this area. Can be addressed with reorganization of the affected table(s).
Space Utilization Problem	Displays the number of tables that fall outside the preset limits for the thresholds of this area. Can be addressed with reorganization of the affected table(s).
Page Utilization Problem	Displays the number of tables that fall outside the preset limits for the thresholds of this area. Can be addressed with reorganization of the affected table(s).
Index Contention	Displays the number of tables where the index space is shared by the Tablespace. To resolve, use the Space Management Wizard (DBArtisan)to change the Index Tablespace in table definition.

The table below describes the options and functionality on the Table Reorganization Candidate Details section:

Option	Description
Tablespace	Displays all Tablespaces (regular) in the database and the value [All].
Table Schema	Displays all schema in the database and the value [All].
Display	All Tables Tables with Overflow Problems - Displays a list of all tables where overflow threshold is exceeded. Tables with Space Utilization problems - Displays a list of all tables where space utilization threshold is exceeded. Tables with Page Utilization problems - Displays a list of all tables where the page utilization threshold is exceeded. Tables with Index Contention - Displays a list of all tables where the table and indexes on the table share the same Tablespace.
Refresh	Refreshes the table grid with criteria selected in the above lists.

The Table Reorganization Candidate Details section provides filters and a grid that you can use to view detailed space diagnostics for all or selected tables. The table below describes the grid data:

Data	Description
Table Schema	The schema name for the table.
Table Name	The name of the table.
Cardinality	The number of rows in the table.
Size	The size of Table in bytes.
Overflow	The number of overflow rows. Overflow rows can be created when rows are updated and the new rows contain more bytes than the old ones (VARCHAR fields), or when columns are added to existing tables.
Pages with Data	The number of allocated pages that contain data.
Total Pages	The total number of allocated pages.
Active Blocks	The number of Active Blocks in Tables with Multidimensional clustering).
Overflow %	Percentage of overflow rows. This should typical be a small value (5%).
Space Utilization%	A low space utilization is an indicator of excessive fragmentation (internal and external). This can occur when there are many pages with free space and/or many free pages. This should typically be > 68%.
Page Utilization %	A low page utilization is an indicator of excessive external fragmentation. This value should typically be more than 80%.
Reorg	This is an indicator of how severe the fragmentation problem for the table is. --- means Overflow%, Space Utilization% and Page Utilization % are all with in the thresholds. *-* means that Overflow%, and Page Utilization % are outside the threshold but Space Utilization% is with in the threshold.
Tablespace	Tablespace where the table data resides.
Indexspace	Tablespace where the index data resides.

Available Functionality

Note: Functionality differ by platform.

The following functionality is available in the Command menu, shortcut menu, and toolbar:

Analysis Report

Analyze

DBCC

Estimate Size

Extract

Map

Tables Wizard (DB2 LUW)

[DBAObjectEditors.htm#ORAEditorTablespaces Open Editor]

Open Wizard

Place

Quick Reorganize /Reorg

Reorganize

Threshold Options

Update Statistics

For more information, see:

Finding Space-related Table Problems

Space Analyst Tabs.

Space Analyst for Oracle - Tables Tab (DBArtisan)

The **Tables** tab displays diagnostic information regarding table space inefficiencies that may exist in your database.

The Table Reorganization Candidate Summary section uses a bar chart to count the various types of table space inefficiencies in your database. Note that the reorganization threshold options are used to compute the counts for each type of space inefficiency.

The table below describes the four distinct types of table space inefficiencies that Space Analyst computes for Oracle and Sybase ASE:

Problem	Description
Block Inefficiencies	These tables have wasted space that exceeds the allowable percentage of wasted space set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Tables with wasted space can take longer to scan and therefore contribute to slower response times.
Next Extent Problems	These tables cannot obtain their next extent of free space due to a lack of needed contiguous free space in the tablespace. Tables suffering from this problem may produce error conditions in the database if more data is added to them. This problem will be highlighted even if the tablespace that contains the table has its autoextend property for one or more datafiles set to true.
Chain Problems	These tables suffer from a chained or migrated row problem. The percentage of chained or migrated rows to overall rows contained in the table exceeds the allowable percentage of chained rows set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Because chained/migrated rows cause more I/O to occur when they are accessed, overall response times can be degraded when these tables are referenced in SQL queries.
Max Extent Problems	These tables are nearing their maximum extent limit and have exceeded the allowable maximum extent percentage set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Tables suffering from this problem may cause space-related errors to result if more data is added to them.

The Table Reorganization Candidate Details section provides filters and a grid that you can use to view detailed space diagnostics for all or selected tables. The table below describes the grid data:

Data	Description
Table Owner	The owner of the table.
Table Name	The name of the table.
Segment Type	Indicates if the object is a table, table partition, etc.
Table (KB)	The total allocated space of the table in KB.
Number of Rows	The number of rows that the table contains. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Number of Blocks	The total number of blocks allocated to the table. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Empty Blocks	The number of empty blocks that the table contains. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Highwater Mark	The last block that Oracle will scan up to when searching the table for data. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Average Used Blocks	The average number of blocks that the table is using. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Block Inefficiency	A percentage that indicates how much wasted space (space that the table is not using) that the table contains. High percentages indicate tables that may be candidates for reorganization. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Chained Row Percent	A percentage that indicates how many rows in the table suffer from chaining or migration. High percentages indicate tables that may be candidates for reorganization. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Max Extent Percent	A percentage indicating how close an object is to reaching its maximum extent limit. High percentages indicate tables that may be candidates for reorganization.
Number of Extents	The number of extents that the table consumes.
Max Extents	The maximum extent limit for the table.
Extent Deficit	Indicates if the object can allocate its next extent. If 'N', this is because the tablespace does not currently have enough contiguous free space. This problem will be highlighted even if the tablespace has its autoextend property for one or more datafiles set to true.
Next Extent	The next extent (in bytes) that the table will attempt to allocate when more space for the table is required.

Max Free Space	The largest contiguous block of free space that exists in the tablespace where the table resides.
Tablespace Name	The tablespace that contains the table.

Available Functionality

Note: Functionality differ by platform.

The following functionality is available in the Command menu, shortcut menu, and toolbar:

Analysis Report	Analyze	DBCC
Estimate Size	Extract	Map
Tables Wizard (DB2 LUW)	[DBAObjectEditors.htm#ORAEditorTablespaces Open Editor]	Open Wizard
Place	Quick Reorganize /Reorg	Reorganize
Threshold Options	Update Statistics	

For more information, see Space Analyst Tabs.

Space Analyst - Tables Tab (DBArtisan)

The **Tables** tab displays diagnostic information regarding table space inefficiencies that may exist in your database.

The Table Reorganization Candidate Summary section uses a bar chart to count the various types of table space inefficiencies in your database. Note that the reorganization threshold options are used to compute the counts for each type of space inefficiency.

The table below describes the four distinct types of table space inefficiencies that Space Analyst computes for Sybase ASE:

Problem	Description
Block Inefficiencies	These tables have wasted space that exceeds the allowable percentage of wasted space set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Tables with wasted space can take longer to scan and therefore contribute to slower response times.
Next Extent Problems	These tables cannot obtain their next extent of free space due to a lack of needed contiguous free space in the tablespace. Tables suffering from this problem may produce error conditions in the database if more data is added to them. This problem will be highlighted even if the tablespace that contains the table has its autoextend property for one or more datafiles set to true.
Chain Problems	These tables suffer from a chained or migrated row problem. The percentage of chained or migrated rows to overall rows contained in the table exceeds the allowable percentage of chained rows set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Because chained/migrated rows cause more I/O to occur when they are accessed, overall response times can be degraded when these tables are referenced in SQL queries.
Max Extent Problems	These tables are nearing their maximum extent limit and have exceeded the allowable maximum extent percentage set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Tables suffering from this problem may cause space-related errors to result if more data is added to them.

The Table Reorganization Candidate Details section provides filters and a grid that you can use to view detailed space diagnostics for all or selected tables. The table below describes the grid data:

Data	Description
Table Owner	The owner of the table.
Table Name	The name of the table.
Segment Type	Indicates if the object is a table, table partition, etc.
Table (KB)	The total allocated space of the table in KB.
Number of Rows	The number of rows that the table contains. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Number of Blocks	The total number of blocks allocated to the table. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Empty Blocks	The number of empty blocks that the table contains. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Highwater Mark	The last block that Oracle will scan up to when searching the table for data. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Average Used Blocks	The average number of blocks that the table is using. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Block Inefficiency	A percentage that indicates how much wasted space (space that the table is not using) that the table contains. High percentages indicate tables that may be candidates for reorganization. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Chained Row Percent	A percentage that indicates how many rows in the table suffer from chaining or migration. High percentages indicate tables that may be candidates for reorganization. NOTE: This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Max Extent Percent	A percentage indicating how close an object is to reaching its maximum extent limit. High percentages indicate tables that may be candidates for reorganization.
Number of Extents	The number of extents that the table consumes.
Max Extents	The maximum extent limit for the table.
Extent Deficit	Indicates if the object can allocate its next extent. If 'N', this is because the tablespace does not currently have enough contiguous free space. This problem will be highlighted even if the tablespace has its autoextend property for one or more datafiles set to true.
Next Extent	The next extent (in bytes) that the table will attempt to allocate when more space for the table is required.

Max Free Space	The largest contiguous block of free space that exists in the tablespace where the table resides.
Tablespace Name	The tablespace that contains the table.

IBM DB2 for Linux, Unix, and Windows

The table below describes the four distinct types of Reorganization Candidate Summary that Space Analyst computes for IBM DB2 for Linux, Unix, and Windows:

Problem	Description
Overflow Problems	Displays the number of tables that fall outside the preset limits for the thresholds of this area. Can be addressed with reorganization of the affected table(s).
Space Utilization Problem	Displays the number of tables that fall outside the preset limits for the thresholds of this area. Can be addressed with reorganization of the affected table(s).
Page Utilization Problem	Displays the number of tables that fall outside the preset limits for the thresholds of this area. Can be addressed with reorganization of the affected table(s).
Index Contention	Displays the number of tables where the index space is shared by the Tablespace. To resolve, use the Space Management Wizard to change the Index Tablespace in table definition.

The table below describes the options and functionality on the Table Reorganization Candidate Details section:

Option	Description
Tablespace	Displays all Tablespaces (regular) in the database and the value [All].
Table Schema	Displays all schema in the database and the value [All].
Display	All Tables Tables with Overflow Problems - Displays a list of all tables where overflow threshold is exceeded. Tables with Space Utilization problems - Displays a list of all tables where space utilization threshold is exceeded. Tables with Page Utilization problems - Displays a list of all tables where the page utilization threshold is exceeded. Tables with Index Contention - Displays a list of all tables where the table and indexes on the table share the same Tablespace.
Refresh	Refreshes the table grid with criteria selected in the above lists.

The Table Reorganization Candidate Details section provides filters and a grid that you can use to view detailed space diagnostics for all or selected tables. The table below describes the grid data:

Data	Description
Table Schema	The schema name for the table.
Table Name	The name of the table.
Cardinality	The number of rows in the table.
Size	The size of Table in bytes.
Overflow	The number of overflow rows. Overflow rows can be created when rows are updated and the new rows contain more bytes than the old ones (VARCHAR fields), or when columns are added to existing tables.
Pages with Data	The number of allocated pages that contain data.
Total Pages	The total number of allocated pages.
Active Blocks	The number of Active Blocks in Tables with Multidimensional clustering).
Overflow %	Percentage of overflow rows. This should typical be a small value (5%).
Space Utilization%	A low space utilization is an indicator of excessive fragmentation (internal and external). This can occur when there are many pages with free space and/or many free pages. This should typically be > 68%.
Page Utilization %	A low page utilization is an indicator of excessive external fragmentation. This value should typically be more than 80%.
Reorg	This is an indicator of how severe the fragmentation problem for the table is. --- means Overflow%, Space Utilization% and Page Utilization % are all with in the thresholds. *-* means that Overflow%, and Page Utilization % are outside the threshold but Space Utilization% is with in the threshold.
Tablespace	Tablespace where the table data resides.
Indexspace	Tablespace where the index data resides.

Available Functionality

Note: Functionality differ by platform.

The following functionality is available in the Command menu, shortcut menu, and toolbar:

Analysis Report	Analyze	DBCC
Estimate Size	Extract	Map
Tables Wizard (DB2 LUW)	[DBAObjectEditors.htm#ORAEditorTablespaces Open Editor]	Open Wizard
Place	Quick Reorganize /Reorg	Reorganize
Threshold Options	Update Statistics	

For more information, see Space Analyst Tabs.

Space Analyst for Sybase ASE and IBM DB2 for Linux, Unix, and Windows - Indexes Tab (DBArtisan)

The **Indexes** tab displays diagnostic information regarding index space inefficiencies that may exist in your database.

The Index Reorganization Candidate Summary section uses a bar chart to count the various types of index space inefficiencies in your database. Note that the reorganization threshold options are used to compute the counts for each type of space inefficiency.

The table below describes the three distinct types of table space inefficiencies that Space Analyst computes:

Problem	Description
BLevel	These indexes have extended beyond the allowable B-level size set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Indexes with extended levels can take longer to scan and can therefore contribute to slower response times.
Next Extent Problems	These indexes cannot obtain their next extent of free space due to a lack of needed contiguous free space in the tablespace. Indexes suffering from this problem may cause space-related errors if more data is added to their parent table. This problem will be highlighted even if the tablespace that contains the table has its autoextend property for one or more datafiles set to true.
Max Extent Problems	These indexes are nearing their maximum extent limit and have exceeded the allowable maximum extent percentage set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Indexes suffering from this problem may cause space-related errors to result if more data is added to their parent table.

The Index Reorganization Candidate Details section provides filters and a grid that you can use to view detailed space diagnostics for all or selected indexes. The grid contains the following data:

Data	Description
Index Owner	The owner of the index.
Index Name	The name of the index.
Index Type	Indicates if the type of index (index, index partition, etc.)
Index (KB)	The total allocated space of the index in KB.
Number of Rows	The number of rows that the parent table contains. **Note:** This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Clustering Factor	Indicates the amount of order of the rows in the table based on the values of the index. If the value is near the number of blocks, then the table is very well ordered. If the value is near the number of rows, then the table is very randomly ordered. In this case, it is unlikely that index entries in the same leaf block point to rows in the same data blocks.
Index Blevel	Indicates the level of the index. Numbers beyond 3-4 can indicate a fragmented index.
Number of Blocks	The total number of blocks allocated to the index. Note: this column may report NULL if the index has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Max Extent Percent	A percentage indicating how close an object is to reaching its maximum extent limit. High percentages indicate indexes that may be candidates for reorganization.
Number of Extents	The number of extents that the index consumes.
Max Extents	The maximum extent limit for the index.
Can Extent Space	Indicates if the object can allocate its next extent. If 'N', this is because the tablespace does not currently have enough contiguous free space. This problem will be highlighted even if the tablespace has its autoextend property for one or more datafiles set to true.
Next Extent	The next extent (in bytes) that the index will attempt to allocate when more space for the index is required.
Max Free Space	The largest contiguous block of free space that exists in the tablespace where the index resides.
Tablespace Name	The tablespace that contains the index.

Note: The grid can also contain a column for deleted leaf rows if the Get Deleted Leaf Row Statistics option has been selected. Be aware that the calculation of this statistic can take a lot of time for databases with large numbers of indexes. Large numbers of deleted leaf rows can indicate an index in need of reorganization.

IBM DB2 for Linux, Unix, and Windows

The Index Reorganization Candidate Summary section uses a bar chart to represent the various types of index space inefficiencies in your database:

- Indexes with Cluster Ratio Problems
- Indexes with Used Space% problems
- Indexes with Balance Problems
- Indexes with Deleted RID problems
- Indexes with Empty Leaf Problems
- Indexes with Low Uniqueness

Note: Space Analyst uses the options you select in the Reorganization Wizard to compute each type of space inefficiency.

Tip: All of the problems in the Index Reorganization Candidate Summary section except low cardinality can be corrected with index reorganization. The Low cardinality is an indicator of an index where the columns on which it is defined have low level of uniqueness. Such indexes should either be dropped or changed to include columns that will make the index more unique.

The table below describes the options and functionality on the Index Reorganization Candidate Details section:

Option	Description
Tablespace	Displays all Tablespaces (regular/large) in the database and the value [All].
Index Schema	Displays all schema in the database and the value [All].
Display	All Indexes Indexes with Cluster Ratio Problems Indexes with Used Space% problems Indexes with Balance Problems Indexes with Deleted RID problems Indexes with Empty Leaf Problems Indexes with Low Uniqueness
Refresh	Refreshes the index grid with criteria selected in the above lists.

The Index Reorganization Candidate Details section provides filters and a grid that you can use to view detailed space diagnostics for all or selected indexes. The grid contains the following data:

Data	Description
Index Schema	The schema name for the index.
Index Name	The name of the index.
Table Schema	The schema name for the base table
Table Name	The base table name.
Table Cardinality	The number of rows in the base table.
Avg Key Size	The index key size, calculated from the average column length of all columns participating in the index.
Leafs	The total number of index leaf pages.
Pseudo Empty Leafs	The number of pseudo empty index leaf pages. A pseudo empty index leaf page is a page on which all the RIDs are marked as deleted, but have not been physically removed.
Pseudo Deleted RIDs	A RID that is marked deleted. This statistic reports pseudo deletes RIDs on leaf pages that are not pseudo empty. It does not include RIDs marked as deleted on leaf pages where all the RIDs are marked deleted.
Levels	The number of index levels.
Unique Entries	The number of unique index entries that are not marked deleted.
Page Size	The page size of the table space in which the table indexes reside, specified at the time of table creation.
% Free	Specifies the percentage of each index page to leave as free space, a value that is assigned when defining the index.
Cluster Ratio	The clustering ratio of an index should be greater than 80 percent. When multiple indexes are defined on one table, some of these indexes have a low cluster ratio. (The index sequence is not the same as the table sequence.) This cannot be avoided. Be sure to specify the most important index when reorganizing the table. The cluster ratio is usually not optimal for indexes that contain many duplicate keys and many entries.
Used Space%	Less than 50 percent of the space reserved for index entries should be empty.
Balance	To determine if recreating the index would result in a tree having fewer levels. This formula checks the ratio between the amount of space in an index tree that has one less level than the current tree, and the amount of space needed. If a tree with one less level could be created and still leave PCTFREE available, then a reorganization is recommended. The actual number of index entries should be more than 90% (or 100-PCTFREE) of the number of entries an NLEVELS-1 index tree can handle (only checked if NLEVELS>1).

Deleted RIDs	The number of pseudo-deleted RIDs on non-pseudo-empty pages should be less than 20 percent.
Empty Leaf	The number of pseudo-empty leaf pages should be less than 20 percent of the total number of leaf pages.
Uniqueness	100 * Index Cardinality / Table Cardinality - A low index uniqueness may indicate the need to Alter or Drop the index.
Reorg	An indicator of how severe the fragmentation problem for the index is. Each hyphen (-) displayed in this column indicates that the calculated results were within the set bounds of the corresponding formula, and each asterisk (*) indicates that the calculated result exceeded the set bounds of its corresponding formula: - or * on the left column corresponds to Cluster Ratio - or * in the second from left column corresponds to Used Space% - or * in the middle column corresponds to Balance - or * in the second column from the right corresponds to Deleted RIDs - or * on the right column corresponds to Empty Leaf
Tablespace	The tablespace where the table data resides.
Indexspace	The tablespace where the index data resides.

Available Functionality

The following functionality is available in the Command menu, shortcut menu, and toolbar:

Note: Functionality differ by platform.

Analysis Report	Analyze	DBCC
Drop	Extract	Estimate Size
Map	Indexes Wizard (DB2 LUW)	[DBAObjectEditors.htm#ORAIndexesEditor Open Editor]
Open Wizard	Place	Quick Reorganize /Reorg
Reorganize	Threshold Options	Update Statistics

For more information, see:

Finding Space-related Table Problems

Space Analyst Tabs

Space Analyst for Oracle - Indexes Tab (DBArtisan)

The **Indexes** tab displays diagnostic information regarding index space inefficiencies that may exist in your database.

The Index Reorganization Candidate Summary section uses a bar chart to count the various types of index space inefficiencies in your database. Note that the reorganization threshold options are used to compute the counts for each type of space inefficiency.

The table below describes the three distinct types of table space inefficiencies that Space Analyst computes:

Problem	Description
BLevel	These indexes have extended beyond the allowable B-level size set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Indexes with extended levels can take longer to scan and can therefore contribute to slower response times.
Next Extent Problems	These indexes cannot obtain their next extent of free space due to a lack of needed contiguous free space in the tablespace. Indexes suffering from this problem may cause space-related errors if more data is added to their parent table. This problem will be highlighted even if the tablespace that contains the table has its autoextend property for one or more datafiles set to true.
Max Extent Problems	These indexes are nearing their maximum extent limit and have exceeded the allowable maximum extent percentage set in the customizable thresholds of Space Analyst (to see all the thresholds, click the toolbar option labeled Reorganization Threshold Options). Indexes suffering from this problem may cause space-related errors to result if more data is added to their parent table.

The Index Reorganization Candidate Details section provides filters and a grid that you can use to view detailed space diagnostics for all or selected indexes. The grid contains the following data:

Data	Description
Index Owner	The owner of the index.
Index Name	The name of the index.
Index Type	Indicates if the type of index (index, index partition, etc.)
Index (KB)	The total allocated space of the index in KB.
Number of Rows	The number of rows that the parent table contains. **Note:** This column may report NULL if the table has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Clustering Factor	Indicates the amount of order of the rows in the table based on the values of the index. If the value is near the number of blocks, then the table is very well ordered. If the value is near the number of rows, then the table is very randomly ordered. In this case, it is unlikely that index entries in the same leaf block point to rows in the same data blocks.
Index Blevel	Indicates the level of the index. Numbers beyond 3-4 can indicate a fragmented index.
Number of Blocks	The total number of blocks allocated to the index. Note: this column may report NULL if the index has not been analyzed via the ANALYZE command or through the DBMS_STATS package.
Max Extent Percent	A percentage indicating how close an object is to reaching its maximum extent limit. High percentages indicate indexes that may be candidates for reorganization.
Number of Extents	The number of extents that the index consumes.
Max Extents	The maximum extent limit for the index.
Can Extent Space	Indicates if the object can allocate its next extent. If 'N', this is because the tablespace does not currently have enough contiguous free space. This problem will be highlighted even if the tablespace has its autoextend property for one or more datafiles set to true.
Next Extent	The next extent (in bytes) that the index will attempt to allocate when more space for the index is required.
Max Free Space	The largest contiguous block of free space that exists in the tablespace where the index resides.
Tablespace Name	The tablespace that contains the index.

Note: The grid can also contain a column for deleted leaf rows if the Get Deleted Leaf Row Statistics option has been selected. Be aware that the calculation of this statistic can take a lot of time for databases with large numbers of indexes. Large numbers of deleted leaf rows can indicate an index in need of reorganization.

Available Functionality

The following functionality is available in the Command menu, shortcut menu, and toolbar:

Note: Functionality differ by platform.

Analysis Report	Analyze	DBCC
Drop	Extract	Estimate Size
Map	Indexes Wizard (DB2 LUW)	[DBAObjectEditors.htm#ORAIndexesEditor Open Editor]
Open Wizard	Place	Quick Reorganize /Reorg
Reorganize	Threshold Options	Update Statistics

For more information, see Space Analyst Tabs.

Space Analyst - Storage Overview Tab (DBArtisan)

The **Storage Overview** tab displays an overview of the storage/space situation on the target server:

The table below describes the sections on the **Storage Overview** tab of Space Analyst for Microsoft:

Section	Description
Server Overview	Overview of total database and log space allocated on the SQL Server, as well as a count of databases, files, and file groups.
Log Overview	Contains a graphical and grid control that works in the following way: - When the graph is displayed, a bar chart is shown that lists the total percent used of all logs on the server. - When the grid is displayed, detailed storage information for each database log is presented.
Database Overview	Contains a graphical and grid control that works in the following way: - When the graph is displayed, a bar chart is shown that lists the total space usage of all databases on the server. - When the grid is displayed, detailed storage information for each database is presented.
Server Disk Overview	Contains an overview of total free space, by physical server drive, of the server that houses SQL Server.

The table below describes the sections on the **Storage Overview** tab of Space Analyst for Sybase ASE:

Section	Description
Server Overview	Overview of total database and log space allocated on the Sybase ASE Server, as well as a count of databases and devices.
Log Overview	Contains a graphical and grid control that works in the following way: - When the graph is displayed, a bar chart is shown that lists the total percent used of all logs on the server. - When the grid is displayed, detailed storage information for each database log is presented. Database name Total log space Free log space Used log space Log percent free
Database Overview	Contains a graphical and grid control that works in the following way: - When the graph is displayed, a bar chart is shown that lists the total space usage of all databases on the server. - When the grid is displayed, detailed storage information for each database is presented: Database name Total database space Database free space Database used space Database percent free
Device Overview	Contains a graphical and grid control that works in the following way: - When the graph is displayed, a bar chart is shown that lists the total space used (in MB) of all devices defined to the server. - When the grid is displayed, detailed storage information for each database device is presented: Device name Physical name Total space Used space Free space Percent free Percent used

Available Functionality

The following functionality is available in the Command menu, shortcut menu, and toolbar for Microsoft:

Analysis Report	Extract	Map
Databases Wizard (SQL Server)	Object Details	Open Editor
Open Wizard	DBCC	Shrink
Threshold Options		

Available Functionality

The following functionality is available in the Command menu, shortcut menu, and toolbar for Sybase ASE:

Analysis Report	Extract	Index Details
Open Editor	Open Wizard	Databases Wizard (Sybase ASE)
Move Log	DBCC	Table Details
Threshold Options	For more information, see:	Pinpointing 'out of space' Problems
Space Analyst Tabs		

Space Analyst for Microsoft - File Groups/Files Tab (DBArtisan)

The **Files/File Groups** tab displays a graph and grid for filegroups on the server and a grid for files.

Tip: You can toggle between the graphs and grid views.

The **File Groups/Files** tab reports space usage on SQL Server by file group and by file.

When the File Groups radio button is pressed, Space Analyst presents a graphical layout of file group space usage across all databases. For large servers, not all file groups will initially be shown, but the graph is scrollable so you can click the graph and scroll to see all file groups on the chart. When in grid mode, more detailed information is shown for each file group including its auto-growth properties, how many files make up the group, the amounts (in MB) taken up by tables and indexes, and more.

When the Files radio button is pressed, detailed information is displayed for each individual file for every database on the server.

Available Functionality

The following functionality is available in the Command menu, shortcut menu, and toolbar:

Analysis Report
Databases Wizard (SQL Server)
Open Wizard
Open Wizard
Extract
Object Details
DBCC
Threshold Options
Map
Open Editor
Shrink

For more information, see Space Analyst Tabs.

Space Analyst for Microsoft - Map Tab (DBArtisan)

The **Map** tab displays a graphical file map that displays extent placement for objects within database files.

The **Map** tab lets you view a graphical layout regarding the physical placement of every object inside a selected database and file. The map helps identify allocated and unused extents throughout a database file. An object grid is also present that lets you view detailed information about every object's storage usage in that file. Keep in mind that an object can span more than one file.

The Space Info section at the top contains information regarding the total number of extents allocated to a database file, a count of extents that contain table data, a count of extents that contain index data, and a count of extents that contain text data.

Databases are comprised of logical pages, each of which is a fixed 8KB in size. Note that the 8KB page is also a unit of I/O, which affects performance and locking. Extents are collections of database pages. An extent is made up of eight, 8K pages and is therefore 64KB in size. Extents are the graphical "blocks" displayed in the map.

Extents come in two flavors - uniform and mixed. Uniform extents ascribe all eight database pages to a single object, whereas mixed extents have the potential to hold multiple objects. With mixed extents, SQL Server tries to preserve space by not letting very small objects take up an entire extent and therefore waste a lot of space. By placing your mouse pointer over an extent "block" in the map, Space Analyst will present a pop-up dialog displaying what object(s) is/are contained in that extent.

The table below describes the options and functionality on the **Map** tab of Space Analyst:

Option	Description
Space Info	Displays total extent count, extents with tables count, extents with indexes count, and extents with text count. Note that an extent can contain more than one object and more than one object type.
Map	Displays object extent data from beginning to end of the database or file. All blocks in the map are the same size. A Microsoft SQL Server extent is 64KB in size or 8, 8K pages. Purple - Indicate extents that have been allocated to objects (used), and extents that are free. Green - Free extents.
Owner	Displays table owner.
Table name	Displays table name.
Object type	Displays object type.
Index name	Displays index name.
Pages used	Displays pages used.
Pages allocated	Displays pages allocated.
In extents	How many extents each object is in.
Table id	Displays table id.
Index id	Displays index id.

The **Tablespace Map** tab includes the following buttons:

Button	Description
Legend	Opens the Tablespace Map Legend.
Display	Opens the Display window. The Display window lets you specifies the magnification level for the Tablespace Map.
Fit to Window	Fits the Tablespace Map to the window.
Full Screen	Fits the Tablespace Map to the full screen.
Print Map	Opens the **Print** dialog to print the Tablespace Map.

Available Functionality

The following functionality is available in the Command menu, shortcut menu, and toolbar:

Analysis Report	Extract	Databases Wizard (SQL Server)
Object Details	Open Editor	Open Wizard
DBCC	Shrink	Threshold Options

For more information, see Space Analyst Tabs.

Space Analyst for Microsoft - Databases Tab (DBArtisan)

The **Databases** tab lets you analyze databases.

Tip: You can toggle between the graphs and grid views.

The **Databases** tab displays information regarding space usage across all databases, potential space-related bottlenecks that are threatening databases, and recent database backup information.

When in graph form, the Database Summary section displays a graphical view of table, index, and free space usage (in MB) across all databases. For large servers, not all databases will initially be shown, but the graph is scrollable so you can click the graph and scroll to see all databases on the chart. When in grid form, the Database Summary section displays much more detail regarding database and log usage, object counts, and more.

The Database Bottlenecks section provides a graphical count of various space-related bottlenecks that are threatening the availability of your databases (some of which are controlled by Space Analyst's customizable thresholds). When in grid form, each database is listed along with information that relays whether or not a potential bottleneck exists for that database.

The Database Backups section details the most recent backup for each database and the size (in MB) of the backup.

Available Functionality

The following functionality is available in the Command menu, shortcut menu, and toolbar:

Analysis Report

Databases Wizard (SQL Server)

Open Wizard

Threshold Options

Extract

Object Details

DBCC

Map

Open Editor

Shrink

For more information, see Space Analyst Tabs.

Space Analyst for Microsoft - Objects Tab (DBArtisan)

The **Objects** tab displays database objects, including tables and indexes.

You can choose whether to view table or index information by clicking either the Tables or Indexes ratio buttons. You can also choose whether to view general space information for tables and indexes or reorganization metrics by selecting the General Space or Space Diagnostics radio buttons.

Note: To obtain space diagnostic information for each object, Space Analyst uses the SQL Server DBCC SHOWCONTIG command. This command may take a while to run on large databases.

The Object summary presents one of two views:

- When the General Space radio button is pressed, a graph is presented that visually displays that amount of allocated space (in KB) for tables, indexes, and free space. For large servers, not all objects for a database will initially be visible on the chart, but the graph is scrollable so you can click the graph and scroll to see all database objects on the chart.
- When the Space Diagnostics radio button is pressed, Space Analyst presents a count of all objects (tables or indexes) that exceed your customizable thresholds for various forms of fragmentation of other space-related inefficiency. See below for a thorough discussion of these types of space problems.

The Object Details grid presents one of two views:

- When the General Space radio button is pressed, general space usage for tables or indexes is shown. Base object metadata (object name, type, partitions, etc.) is shown along with the amounts of total, used, and free space (in KB) for the object. The percent of space used for the object is also shown along with the percent of space in the database that the object consumes.
- When the Space Diagnostics radio button is pressed, Space Analyst present fragmentation and other space-related inefficiency information for all objects in the selected database. Note that definitions for all the various forms of object fragmentation and space inefficiencies follow later in the help.

The table below describes the options and functionality on the **Objects** tab of Space Analyst:

View	Description
Database	Lets you switch between databases.
Tables	Lets you switch between viewing space for tables and indexes. Displays object details when you select.
Indexes	Lets you switch between viewing space for tables and indexes. Displays object details when you select.
General Space View	Displays general space information for tables and indexes in the selected database. Displays object details when you select.
Space Diagnostic View	Displays fragmentation and space information to help the DBA decide if a reorg is necessary. Displays object details when you select.
File Group	Lets you view object data for all filegroups for the database or only a selected filegroup.
Owner	Lets you view object data for all table owners or only a selected owner.
Display	Lets you view space diagnostic information for tables or indexes.

Available Functionality

The following functionality is available in the Command menu, shortcut menu, and toolbar:

Analysis Report

Open Editor

Threshold Options

Extract

Open Wizard

Update Statistics

Databases Wizard (SQL Server)

DBCC

For more information, see Space Analyst Tabs.

Space Analyst for Sybase ASE - Database Devices Tab (DBArtisan)

The **Database Devices** tab displays database device information.

Tip: You can toggle between graph and grid views.

The table below describes the sections on the **Database Devices** tab of Space Analyst:

Section	Description
Device Summary	Contains summary information for all Sybase ASE database devices. Displays the amount of data, log, data and log, and free space amounts (in MB).
Device Details	Displays detailed information for selected devices. Lets you click in a bar chart and populate the lower section of the tab with that device's information.
Database Fragments	Displays database fragments.

Available Functionality

The following functionality is available in the Command menu, shortcut menu, and toolbar:

Analysis Report	Resize (DBArtisan)	Extract
Index Details	Move Log	Databases Wizard (Sybase ASE)
Databases Device Wizard (DBArtisan - Sybase ASE)	Segment Wizard (Sybase ASE)	Open Editor
Open Wizard	DBCC	Reinitialize (DBArtisan)
Refit (DBArtisan)	Table Details	Threshold Options

For more information, see Space Analyst Tabs.

Space Analyst for Sybase ASE - Databases Tab (DBArtisan)

The **Databases** tab lets you analyze databases.

Tip: You can toggle between graph and grid views.

The table below describes the sections on the **Databases** tab of Space Analyst:

Section	Description
Database Summary	Contains chart that displays database used and free space, along with log used and free space information for databases. The summary section totals up the total database, log, table, and index space. Also displays a summary count of all user tables and indexes.
Database Details	Displays detailed information for selected databases. Lets you select whether the graph shows space utilization or object fragmentation scores.

Available Functionality

The following functionality is available in the Command menu, shortcut menu, and toolbar:

- Analysis Report
- Index Details
- Databases Device Wizard (DBArtisan - Sybase ASE)
- Open Wizard
- Refit (DBArtisan)
- Resize (DBArtisan)
- Move Log
- Segment Wizard (Sybase ASE)
- DBCC
- Table Details
- Extract
- Databases Wizard (Sybase ASE)
- Open Editor
- Reinitialize (DBArtisan)
- Threshold Options

For more information, see Space Analyst Tabs.

Space Analyst for IBM DB2 for Linux, Unix, and Windows - Tablespaces Tab (DBArtisan)

The **Tablespace** tab lets displays tablespace space allocation and tablespace.

Available Functionality

The following functionality is available in the Command menu, shortcut menu, and toolbar:

Note: Functionality differ by platform.

Analysis Report
Extract
Open Editor - see Modifying objects using editors
Open Wizard
Reorganize
Update Statistics
Tablespaces Wizard (DB2 LUW)
Threshold Options

For more information, see Space Analyst Tabs.

Space Analyst Functionality (DBArtisan)

Space Analyst offers functionality that help you analyze space. The Space Analyst support the following functionality:

Note: Functionality differ by platform.

- Analysis Report
- Convert to Dictionary Managed
- Convert to Locally Managed
- Resize (DBArtisan)
- DBCC
- Drop
- Estimate Size
- Extract
- Index Details
- Map
- Move Log
- Databases Wizard (Sybase ASE)
- Databases Device Wizard (DBArtisan - Sybase ASE)
- Segment Wizard (Sybase ASE)
- Tables Wizard (DB2 LUW)
- Tablespaces Wizard (DB2 LUW)
- Open Editor
- Open Wizard
- Options
- Place
- Quick Reorganize /Reorg
- DBCC
- Refit (DBArtisan)
- Reinitialize (DBArtisan)
- Reorganize
- Shrink
- Table Details
- Update Statistics

Space Analyst - Analysis Report

The **Analysis Report** dialog lets you build an HTML analysis report for the selected tablespace. You can run reports on a recurring basis and dynamically alter any saved thresholds that flag space problems in the tablespace being analyzed.

For Oracle, the report includes sections for particular reorganization needs, including:

- Fragmentation index for the tablespace
- Tables with high block inefficiencies
- Tables or indexes with next extent space deficit problems
- Tables or indexes with maximum extent limit problems
- Tables or indexes with percent increase greater than zero
- Tables with chained/migrated row problems
- Indexes with B-tree depth problems

For Microsoft, the report contains different sections that highlight particular reorganization needs, including:

- Free space levels for the database and log
- Tables with forwarded record problems
- Indexes with average fragmentation problems
- Tables or indexes with extent fragmentation problems
- Tables or indexes with logical fragmentation problems
- Tables or indexes with scan density problems
- Tables or indexes with average page density problems

For Sybase ASE, the report contains different sections that highlight particular reorganization needs, including:

- Free space levels for the database and log
- Tables with forwarded row problems
- Tables with empty page problems
- Tables with deleted row problems
- Tables with low fragmentation scores
- Indexes with unused space problems
- Indexes with empty leaf problems
- Indexes with height/level problems
- Indexes with low fragmentation scores

For IBM DB2 for Linux, Unix, and Windows, the report contains different sections that highlight particular reorganization needs, including:

- Tables with overflow problems
- Tables with space utilization problems
- Tables with page utilization problems
- Tables with index space contention
- Indexes with cluster ratio problems
- Indexes with used space% problems
- Indexes with balance problems
- Indexes with deleted RID problems
- Indexes with empty leaf problems
- Indexes with low uniqueness

- Free levels of tablespace and log space

To open the Analysis Report dialog box:

1 Open the Space Analyst.

2 On the Command menu, click the Analysis Report.

For all platforms, browse to or type a report title and make your desired selections from the drop-down lists. After the report executes, you can print it or save it.

To print the report:

1 Click Print from the Analysis Report toolbar.

To save the report:

1 Select Save from the File menu on the main toolbar.

Space Analyst - Convert to Locally-managed

The **Convert to Locally-Managed** dialog lets you convert a dictionary-managed tablespace to locally-managed. Objects in locally-managed tablespaces have unlimited extents; they will never fail to extend due to an imposed maximum extent limit.

Important Notes

There are a few restrictions on these conversion procedures (for example, Oracle9i UNDO tablespaces currently can't be converted, etc.), so you should check the Oracle documentation for the specifics on performing a conversion.

The Convert to Locally-Managed option is only available if the selected tablespace(s) are:

- ONLINE
- In Read/Write mode
- Dictionary managed
- Not UNDO tablespaces
- Not Temporary tablespaces
- Not the SYSTEM tablespace

The table below describes the options and functionality on the **Convert to Locally-Managed** dialog

Option	Description
Tablespaces to be Converted	Displays the tablespaces to be converted to locally-managed.
Conversion Options	OPTIONAL: Lets you type allocation unit and relative file number information.

Completing the Convert to Locally-Managed Dialog Box for Space Analyst

To complete this dialog box, do the following:

1 Select Convert to Locally-Managed.

OR

Right-click the target object, and then select Convert to Locally-Managed.

The Space Analyst opens the dialog box.

2 Do one of the following:

Button	Description
Preview	Opens the **Preview** dialog.
Schedule	Opens a scheduling application.
Execute	Executes the task.

For more information, see Convert to Locally-Managed.

Space Analyst - Convert to Dictionary-Managed

Note: The Convert to Dictionary-Managed functionality is for Oracle 9i only.

The **Convert to Dictionary-Managed** dialog lets you convert a locally-managed tablespace to dictionary.

Important Notes

The **Convert to Dictionary-Managed** dialog is only available if the selected tablespace(s) are:

- ONLINE
- In Read/Write mode
- Locally managed
- Not UNDO tablespaces
- Not Temporary tablespaces
- Not the SYSTEM tablespace

The table below describes the options and functionality on the **Convert to Dictionary-Managed** dialog:

Option	Description
Tablespaces to be Converted	Displays the tablespaces to be converted to dictionary-managed.

Completing the Dictionary-Managed Dialog Box for Space Analyst

To complete this dialog box, do the following:

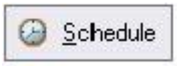
1 Select Convert to Dictionary-Managed.

OR

Right-click the target object, and then select Convert to Dictionary-Managed.

The Space Analyst opens the dialog box.

2 Do one of the following:

Button	Description
	Opens the **Preview** dialog.
	Opens a scheduling application.
	Executes the task.

For more information, see Convert to Dictionary-Managed.

Space Analyst - Extract

The Extract functionality lets you extract the database DDL in an ISQL editor.

To extract the database DDL in an ISQL editor, do the following:

1 Click the Extract DDL button.

OR

Select Extract.

OR

Right-click the target object, and then select Extract.

Space Analyst extract the database DDL in an ISQL editor.

Space Analyst - Map

The Map functionality opens the Tablespace Map tab or Map tabof the Space Analyst for the selected index.

Important Notes

None

For more information, see Opening the Tablespace Map tab.

Opening the Tablespace Map Tab for Space Analyst

To open the **Tablespace Map** or **Map** tab, do the following:

1 Select Map.

OR

Right-click the target object, and then select Map.

Space Analyst open the **Tablespace Map** tab or **Map** tab.

For more information, see Map.

Space Analyst - Open Editor

The Open functionality lets you modify objects in the object editors.

To open the object editor, do the following:

1 In any grid, select an object.

2 Click the Open Editor button.

OR

Select Open.

OR

Right-click the target object, and then select Open.

Space Analyst opens the object editor.

Space Management Wizard (DBArtisan)

The Space Management Wizard can create two specialized space management jobs that allow you to proactively eliminate space threats before they affect your database:

You can create a "reorganization analyst job" that allows you to specifically select various databases and/or objects and then set reorganization thresholds that determine when a reorganization will occur. Once constructed, you can schedule these jobs to run as often as you'd like. During execution, these jobs will interrogate your chosen database/objects and determine if a reorganization needs to occur by applying your customized reorganization thresholds. If any objects are found to need reorganization, they are dynamically reorganized using any customized options you specify.

You can build space maintenance jobs that will perform functions such as check the validity of your databases and filegroups, repair any found minor corruption, and eliminate unused space in your databases. You can also set the job up to notify you if any problems are found during job execution.

Creating or editing a space Management Job

The Space Management Wizard lets you create space reorganization or maintenance jobs without knowing the code. After creation, jobs can be reused as is, or edited.

To open the Space Management Wizard

1 Start the Space Analyst and click the Open Wizard button on the Space Analyst toolbar.

OR

Select Open Wizard from the Space Analyst menu.

The sections below display the fields you may find creating or editing a space management job. Be aware that the sections you see, and the order in which they appear in the wizard vary depending on the selections you make and the database platform.

New or Used Task
Type of Space Management Task
Database/Objects to Be Part of the Reorganization or Maintenance Task
Custom Reorganization
Save the OperationExecution Preferences for a Reorganization Operation
Global Customizations
Specific Object Customization
Threshold Criteria for Object Reorganization
Integrity Check Options
Integrity Check for Specific Database Objects
Unused Space Options
Problem Notification

New or Used Task

Here you choose whether to create a new space management task or whether to use a previously saved task. If you want to reuse a previously created job, you can type or browse to find the previously saved task definition file (*.sptsk).

Type of Space Management Task

Required Information	Description
Create standard reorganization job	Select to build a reorganization job for a database or the objects you specify.
Create reorganization analyst job	Select to create and deploy a reorganization job that runs periodically to detect and automatically correct space inefficiencies.
Create space maintenance job	Select to build a space maintenance job that validates database structures and performs storage maintenance tasks.

Database/Objects to Be Part of the Reorganization or Maintenance Task

Choose the databases and/or objects you want to include in the job you are creating. Simply move items from the tree in the right pane to the left pane by clicking the right arrow. You can amend your selections using the left arrow. The selection is limited to entire databases for a space maintenance job.

Custom Reorganization

Required Information	Description
Perform no customizations	When you choose this option, all default object and database attributes are used.
Perform global customizations	This option gives you the chance to define options that will be globally applied to all objects being reorganized.
Perform specific object customizations	Here you individualize options for specific objects being reorganized.

Save the Operation

If you want to save the operation, you can use the default name supplied or type another in the upper panel. If you don't want to save the operation, remove the default name. Browse to the location where you want to save the file. The job as it was created appears in the read-only preview panel. To make changes, you must go back and change your selections or open the job in an ISQL window prior to execution.

Execution Preferences for a Reorganization Operation

Option	Description
Open the reorganization script in an ISQL editor.	The reorganization script will be generated and then opened in a new ISQL editing session.
Perform reorganization now	The reorganization is run immediately with results presented afterward in an ISQL window.
Schedule the reorganization script for later execution.	The reorganization script will be generated and then scheduled to run at a specified date/time.

Global Customizations

Option	Description
Relocate to file group	Select to perform an extended alter and move the table(s) to the requested file group using the ON keyword in the table creation DDL.
Perform sorted reorg using table's clustered index	Select to rebuild of any table's clustered index.
Perform online reorganization where possible	MICROSOFT SQL SERVER ONLY: Select to use the DBCC INDEXDEFRAG command to perform an online rebuild of every table's index(es).
View reorganization output messages	Select for the WITH NO_INFOMSGS clause of any DBCC command to not be used.
Defragment tables in DROP/CREATE fashion	Select to defragment tables in DROP/CREATE fashion.
Relocate to file group	Select for DROP_EXISTING clause instead of doing a drop and then create.
Fill Factor	Select to specify a new fill factor for the DBCC DBREINDEX command or a create/drop of an existing index.
Index Reorganization	Rebuild:,Drop/Create, or Reorganize
Perform online reorganization where possible	Select to use the DBCC INDEXDEFRAG command to perform an online rebuild of every index.
View reorganization output messages	Select for the WITH NO_INFOMSGS clause of any DBCC command to not be used.

Specific Object Customization

Option	Description
Select	Drop-down list lets you switch between indexes and tables and see the particular objects for each.
Relocate to file group	Select a clause instead of doing a drop and then create.
Fill Factor	For indexes, select to specify a new fill factor.
Perform sorted reorganization using table's clustered index	For tables, select to perform sorted reorganization using table's clustered index.
Index reorganization	Rebuild: This is the default. Drop/Create Reorganize
Perform online reorganization where possible	Select to perform an online rebuild of every index.
View reorganization output messages	Select for the WITH NO_INFOMSGS clause of any DBCC command to not be used.
Defragment tables in DROP/CREATE fashion	For tables, select to defragment tables in DROP/CREATE fashion.

Threshold Criteria for Object Reorganization

Option	Description
Table: Average Page Density	Default is less than or equal to 50.
Scan Density Percent	Default is less than or equal to 0.
Logical Fragmentation	Greater than or equal to 25 is the default value.
Extent Fragmentation	Default is greater than or equal to 50.
Forwarded Record Percent	Default is greater than or equal to 50.
Index: Average Fragmentation	Default is greater than or equal to 25.
Scan Density Percent	Default is less than or equal to 0.
Logical Fragmentation	Greater than or equal to 25 is the default value.
Extent Fragmentation	Default is greater than or equal to 50.
Index Tree Depth	Default is greater than or equal to 3.
Average Page Density	Default is less than or equal to 50.

Note: All defaults are editable. Also, all current metrics for these criteria can be found on the Objects tab >Space Diagnostics view. The default thresholds are visible when you choose Threshold Options from the Space Analyst drop-down menu.

Integrity Check Options

Option	Description
Do not perform any integrity checks	Self-explanatory.
Check databases	Checks the database integrity for database(s) you identified earlier.
Check file groups	Self-explanatory.
Check specific database objects	In a later step you are able to select the particular objects.
Include index checks for all tables	This is the default and isn't actually optional.
Attempt to repair minor problems	Enabled as a default for database or database objects integrity checks. You can unselect.
Include check for database catalogs	Enabled as a default for database integrity checks. You can unselect.
Correct data dictionary inaccuracies	Enabled as a default for database integrity checks. You can unselect.
Check data purity	Verifies that columns only have data in the correct ranges, and so on.

Integrity Check for Specific Database Objects

Here you select objects for the database(s) you have included in this maintenance job.

Unused Space Options

Option	Description
Do not eliminate unused space	Self-explanatory
Eliminate unused space for databases	Select to only truncate space for databases larger than a specific size (in MB).
Only eliminate space when database is larger than	Available if you selected Eliminate unused space for databases option. Lets you specify maximum database size in MB.
Amount of free space to leave for database	Lets you specify how much free space is to remain after a space truncation.
Eliminate unused space for specific database tables	Self-explanatory.

Problem Notification

Option	Description
No, do not notify me	Select to not have Space Analyst notify you of any errors.
Yes, notify me in the following ways.	Select to have Space Analyst notify you of any errors and then type an e-mail or net send address.

Space Analyst - Threshold Options

The Space Analyst **Reorganization Thresholds Options** dialog lets you customize the various reorganization thresholds that can be used to flag storage and data objects that need to be reorganized. The thresholds you specify will apply to the Space Analysis Reports.

Important Notes

None

The table below describes the options and functionality on the Space Analyst **Thresholds Options** dialog for IBM DB2 for Linux, Unix, and Windows:

Option	Description
Table Reorganization Thresholds	The table thresholds apply to the displays used in tab 4 of the Space Analyst and the queries that are run to present the various customized displays. They are also used for the tablespace analysis report.
Index Reorganization Thresholds	The index thresholds apply to the displays used in tab 5 of the Space Analyst and the queries that are run to present the various customized displays. They are also used for the tablespace analysis report.

The table below describes the options and functionality on the Space Analyst **Thresholds Options** dialog for Oracle:

Option	Description
Tablespace and Datafile Fragmentation Thresholds	The tablespace and datafile thresholds apply to the tablespace and datafile fragmentation index. The default for each is 50 - meaning, flag any tablespace or datafile whose fragmentation index is less than 50. They are also used for the tablespace analysis report.
Table Reorganization Thresholds	The table thresholds apply to the displays used in tab 4 of the Space Analyst and the queries that are run to present the various customized displays. They are also used for the tablespace analysis report.
Index Reorganization Thresholds	The index thresholds apply to the displays used in tab 5 of the Space Analyst and the queries that are run to present the various customized displays. They are also used for the tablespace analysis report.

The table below describes the options and functionality on the Space Analyst **Thresholds Options** dialog for Sybase ASE:

Option	Description
Database and Log Space Thresholds	The database and log space thresholds apply to the database and log fragmentation index. The default for each is 80 - meaning, flag any database or log whose fragmentation index is less than 80. The database and log space thresholds apply to any grid that shows free space percent for those entities.
Table Reorganization Thresholds	The table thresholds apply to the displays used in tab 5 of the Space Analyst and the procedures that are run to present the various customized displays. They are also used for the database analysis report.
Index Reorganization Thresholds	The index thresholds apply to the displays used in tab 5 of the Space Analyst and the queries that are run to present the various customized displays. They are also used for the database analysis report.

The table below describes the options and functionality on the Space Analyst **Thresholds Options** dialog for SQL Server:

Option	Description
Database and Log Space Thresholds	The database and log space thresholds apply to the database and log fragmentation index. The default for each is 80 - meaning, flag any database or log whose fragmentation index is less than 80. The database and log space thresholds apply to any grid that shows free space percent for those entities.
Table Reorganization Thresholds	The table thresholds apply to the displays used in tab 5 of the Space Analyst and the procedures that are run to present the various customized displays. They are also used for the database analysis report.
Index Reorganization Thresholds	The index thresholds apply to the displays used in tab 5 of the Space Analyst and the queries that are run to present the various customized displays. They are also used for the database analysis report.
Fragmentation Analysis Mode	Detailed is the default. The other options are Sampled and Limited. This option serves as a parameter that’s passed to the space diagnostic procedures use on the Objects tab for the reorganization summary and the detail views for indexes and tables.

Completing the Space Analyst Thresholds Options Dialog Box for Space Analyst

To complete this dialog box, do the following:

1 On the Analysts Toolbar, click the Options button.

The Space Analyst opens the dialog box.

2 Review thresholds.

3 Click OK.

Space Analyst for Oracle tutorial (DBArtisan)

Embarcadero Space Analyst provides sophisticated diagnostic capabilities to troubleshoot bottlenecks and performance inefficiencies that result from poor space management. Running inside the Embarcadero DBArtisan console, Space Analyst helps you identify and correct storage problems by supplying powerful analysis facilities and corrective utilities and that will satisfy advanced experts, yet are easy enough to use for database novices.

About This Space Analyst for Oracle Tutorial

This tutorial helps you get started with Embarcadero Space Analyst.

After completion of this tutorial, you will have the foundation you need to explore the many features and benefits of Space Analyst. You will have learned how to get a summary view of storage for your database as well as how to identify potential space-related threats. Finally, you'll use the reorganization wizard to correct any space problems that you find.

This tutorial is divided into 5 sections.

Session 1: Getting started With Space Analyst for Oracle (DBArtisan)

Session 2: Obtaining a Global Vew of Storage Using Space Analyst for Oracle (DBArtisan)

Session 3: Identifying Storage Structure Problems Using Space Analyst for Oracle (DBArtisan)

Session 4: Pinpointing Object Performance Inefficiencies Using Space Analyst for Oracle (DBArtisan)

Session 5: Correcting Space-Related Problems Using Space Analyst for Oracle (DBArtisan)

You can use this basic tutorial as a roadmap of product highlights.

After starting this tutorial, you can select Help from the menu to find many additional resources that complement and build on many of the activities shown in this brief tutorial.

For more information, see [DBATutorial.htm#AdditionalEvaluationResources Additional Evaluation Resources].

Session 1: Getting started With Space Analyst for Oracle (DBArtisan)

Download and Install

You can obtain the latest version of Space Analyst by downloading DBArtisan from the Embarcadero Web site at http://www.embarcadero.com/downloads/downloaddbartisan.asp. Space Analyst runs fully contained within DBArtisan so you will need the DBArtisan product to use Space Analyst.

Click Download, and follow the steps indicated. Save the file on your computer and then double-click to launch the self-extracting file that will guide you through the installation process.

When you first install an evaluation copy of DBArtisan, you can use the tool for 14 days. After that time, a permanent license is needed.

Proceed to Space Analyst for Oracle Overview

Space Analyst for Oracle Overview

Once you are connected to an Oracle database, you can open Space Analyst from within DBArtisan two ways:

1 On the Analyst menu, select Space Analyst.

Note: You do not need to install anything for Space Analyst to work.

Proceed to Session 2: Obtaining a Global Vew of Storage Using Space Analyst for Oracle (DBArtisan)

Session 2: Obtaining a Global Vew of Storage Using Space Analyst for Oracle (DBArtisan)

When it opens, Space Analyst shows a complete overview of the storage situation of a particular database or database server. The first tab of Space Analyst displays all pertinent global statistics regarding storage structure usage throughout a database.

There are four distinct sections displayed on the first **Database Overview** tab of Space Analyst:

- Database Overview
- Tablespace Space Overview
- Tablespace Fragmentation Overview
- Tablespace Objects Overview

Proceed to Space Analyst for Oracle - Database Overview

Space Analyst for Oracle - Database Overview

The Database Overview displays:

- The total count of all tablespaces, datafiles and temporary files.
- The percentage breakdown of the various object types in a database. Positioning the mouse pointer over the graph causes it to display the actual percentage amounts for each object type.
- The amount of space in kilobytes that each object type consumes is also shown. This section helps you get a clear picture of exactly how much space the various object types in your database are taking up.

Tip: If you want to see a grid-style display of the information shown in the graph, position your mouse over the graph, click the right mouse button and select the 'toggle display' option.

Proceed to Space Analyst for Oracle - Tablespace Space Overview

Space Analyst for Oracle - Tablespace Space Overview

The Tablespace Space Overview section displays summary information regarding every tablespace in the database. When invoked in graph form, this section shows how much total space is currently allocated to each tablespace. When shown in grid form, this section shows not only the total space allocated for each tablespace, but also includes the used, free, and percent free amounts, along with an indicator that shows whether any of the tablespace's underlying data or temp files can automatically grow if necessary.

Tip: To see a grid-style display of the information shown in the graph, position your mouse over the graph, click the right mouse button and select the 'toggle display' option. To return to the graph, just perform the same actions over the grid.

When this section is in grid form, you can also perform a number of actions on a tablespace like open it in a DBArtisan editor, extract it's DDL, convert it to either local or dictionary-managed, and more. For example, to open a tablespace in one of DBArtisan's object editors, do the following:

1 In the grid, select a tablespace.

2 Right-click and select Open.

OR

Select a tablespace and click the Open toolbar button.

DBArtisan opens a Tablespaces Editor.

Proceed to Space Analyst for Oracle - Fragmentation Overview

Space Analyst for Oracle - Fragmentation Overview

The fragmentation overview provides a quick glance into how badly fragmented a tablespace is. Space Analyst computes a fragmentation index for every tablespace that can be used to quickly tell if fragmentation problems are plaguing a tablespace. A 100 percent reading indicates no fragmentation, while lower scores indicate the presence of fragmentation.

To see the percentage of fragmentation for each tablespace in the graph, do the following:

1 Position the pointer over each bar in the chart.

To see a grid-style display of the information shown in the graph, do the following:

1 Position the pointer over the graph.

2 Right-click and select Toggle Display.

Proceed to Space Analyst for Oracle - Tablespace Objects Overview

Space Analyst for Oracle - Tablespace Objects Overview

The tablespace objects overview provides a window into the types of objects that each tablespace contains. This view is useful because it helps you quickly understand if undesired objects are present in a tablespace.

Note: There is no graphical view available for this section.

Proceed to Session 3: Identifying Storage Structure Problems Using Space Analyst for Oracle (DBArtisan)

Session 3: Identifying Storage Structure Problems Using Space Analyst for Oracle (DBArtisan)

Space Analyst can help you quickly find storage structure problems that threaten the availability of your database.

Proceed to Pinpointing ‘out of space’ Problems Using Space Analyst for Oracle

Pinpointing ‘out of space’ Problems Using Space Analyst for Oracle

To quickly tell if any of the tablespaces in your database are about to run out of free space, do the following:

1 Select the **Database Overview** tab.

2 In the Tablespace Space Overview section, position the pointer over the graph.

3 Right-click and select Toggle Display.

4 Examine the Percent Free columns of each tablespace along with the Autoextend column. If a tablespace is nearing the end of free space and cannot automatically grow, take action to prevent an out-of-space error.

Proceed to Locating Fragmentation Issues Using Space Analyst for Oracle

Locating Fragmentation Issues Using Space Analyst for Oracle

There are a number of facilities within Space Analyst that helps you understand storage structure fragmentation.

To get summary information regarding tablespace fragmentation, do the following:

1 Select the **Database Overview** tab.

2 In the Tablespace Fragmentation section, position the pointer over the graph.

3 Right-click and select Toggle Display.

The tablespace fragmentation grid scores each tablespace in terms of fragmentation levels and also present data regarding the number of free chunks in a tablespace along with the largest free chunk size. Tablespaces with poor fragmentation ratings are highlighted in red. Note that you can sort the grid from worst to best tablespaces by clicking on the Fragmentation Index column header in the grid.

To get the same information for datafiles, do the following:

1 Select the **Data Files** tab.

The datafile fragmentation grid scores each datafile in terms of fragmentation levels and also present data regarding the number of free chunks in the datafile along with the largest free chunk size. Note that temp files are not shown in this display. Datafiles with poor fragmentation ratings are highlighted in red.

To determine the type of fragmentation for a tablespace (bubble or honeycomb), do the following:

1 Select a tablespace or datafile that appears to have fragmentation issues.

2 Right-click and select Map.

Space Analyst displays a graphical rendering of the tablespace on the **Tablespace Map** tab.

Bubble fragmentation can be confirmed by the presence of many free space fragments being found throughout the tablespace. Honeycomb fragmentation can be identified by seeing free space fragments adjacent to one another in the map.

Note: Large tablespace maps can be read easier if the “Full Screen” option is chosen. The standard view may be returned from full screen by clicking on the small full screen dialog box that appears at the upper left side of the display.

Note that object extent fragmentation problems can also be identified in Space Analyst’s tablespace map. To see if any table or index in a designated tablespace suffers from extent fragmentation, do the following:

1 Select the **Tablespace Map** tab.

2 Select a tablespace.

3 In the View By section, select the Extent Problems option.

Space Analyst reconstructs the tablespace map and accompanying left-hand object grid to reveal any objects that have extent problems. All objects with extent problems are color-coded in red for easy identification. Objects colored in green indicate that they doe not suffer from any extent problems.

Proceed to Using Analysis Reports in Space Analyst for Oracle

Using Analysis Reports in Space Analyst for Oracle

You can run HTML-formatted reports that pinpoint all major storage-related problems for a particular tablespace and the objects that are contained within it. To produce an Analysis Report, do the following:

1 On the toolbar, click Analysis Report.

Space Analyst opens the **Analysis Report** dialog.

2 In Report File Name and Report Title you can change the location and report title of the report.

3 In Category, select the types of space problems you would like to investigate. The default option is All.

4 In Tablespace, select the tablespace you would like to analyze.

5 Click Execute.

Space Analyst creates a frame-based Analysis report for the tablespace.

Proceed to Session 4: Pinpointing Object Performance Inefficiencies Using Space Analyst for Oracle (DBArtisan)

Session 4: Pinpointing Object Performance Inefficiencies Using Space Analyst for Oracle (DBArtisan)

Space Analyst can greatly accelerate the process of locating objects in your database that are degrading overall response times because of their current storage/space situation.

Proceed to Finding Space-related Table Problems With Space Analyst for Oracle

Finding Space-related Table Problems With Space Analyst for Oracle

To quickly determine if any tables in your database may be contributing to performance slowdowns, do the following:

1 In DBArtisan connect to the database you would like to check.

2 Open Space Analyst.

3 Select the **Tables** tab.

Space Analyst runs diagnostics against your database and graphically display counts of tables that are found to contain various types of space problems.

The graph at the top of the **Tables** tab shows the overall count of problem tables, but you can also view the details for each type of problem by using the Details grid at the bottom of the tab. For example, if Space Analyst identifies tables that have block inefficiencies, to find the actual tables and view details regarding the identified space problem, do the following in the Details grid:

1 In the Tablespace list, select All.

1 In the Table Owner list, select All.

2 In the Display list, select Tables with inefficient block usage.

3 Click Execute.

Space Analyst displays the actual tables that suffer from block inefficiencies along with their demographic details. To take corrective action on any object, select one or more tables and click the right mouse button to bring up a menu of possible actions.

Proceed to Finding Space-related Index Problems with Space Analyst for Oracle

Finding Space-related Index Problems with Space Analyst for Oracle

To quickly determine if any indexes in your database may be contributing to performance slowdowns by following these steps:

1 In DBArtisan connect to the database you would like to check.

2 Open Space Analyst.

3 Select the **Indexes** tab.

The graph at the top of the **Indexes** tab shows the overall counts of problem indexes, but you can also view the details for each type of problem by using the Details grid at the bottom of the tab. For example, if Space Analyst identifies indexes that have next extent problems, then to find the actual indexes and view details regarding the space problem, do the following in the Details grid:

1 In the Tablespace list, select All.

2 In the Index Owner list, select All.

3 In the Display list, select Indexes with NEXT_EXTENT space deficits.

4 Click Execute.

Space Analyst displays the actual indexes that suffer from next extent space problems along with their demographic details. If you wanted to take corrective action on any object, select one or more indexes and click the right mouse button to bring up a menu of possible actions.

Note: Another critical statistic in determining index problems is deleted leaf rows. To display deleted leaf row statistics in the Details grid, do the following:

1 On the Command toolbar, select Get Deleted Leaf Row Statistics.

Proceed to Session 5: Correcting Space-Related Problems Using Space Analyst for Oracle (DBArtisan)

Session 5: Correcting Space-Related Problems Using Space Analyst for Oracle (DBArtisan)

After you have identified space-related bottlenecks in your database with Space Analyst, you can use a number of different tools and facilities within the product to eliminate each storage problems.

Proceed to Correcting Storage Structure 'out of space' Problems for Space Analyst for Oracle

Correcting Storage Structure 'out of space' Problems for Space Analyst for Oracle

Space Analyst can help you quickly remedy tablespaces that are running out of room by:

- Resizing an existing tablespace to have more room.
- Adding a new datafile to an existing tablespace.
- Setting the Autoextend attribute for one or more of a tablespace's datafiles to allow for automatic growth.

To manage these functions, do the following:

1 Select the **Database Overview** tab.

2 In the Tablespace Space Overview section, right-click the graph, and then select Toggle Display.

3 Select the tablespace that is running low on free space.

4 Select the Open toolbar button or the Open shortcut menu option.

DBArtisan opens a Tablespace Editor.

In the Tablespace Editor, you can remedy the tablespace problem three ways:

- To resize an existing tablespace, open the tablespace's datafile in the editor (using the Edit button) and change the datafile's size to a new, larger size. Then close the editor and select the Alter toolbar button to make the change permanent to the database.
- To add a new datafile to the tablespace, select the Add button in the **Datafile** tab of the tablespace editor, and specify the new datafile's attributes. Once complete, close the Add Datafile dialog and select the Alter toolbar button to make the change permanent to the database.
- To set Autoextend for one or more of a tablespace's datafiles, open each datafile in the datafile editor by selecting the Edit button. Once in the Modify Datafile dialog, select the option for Auto Extend and specify other attributes as needed. When complete, close the dialog and select the Alter toolbar button to make the change permanent to the database.

Proceed to Correcting Storage Structure Fragmentation Issues Using Space Analyst for Oracle

Correcting Storage Structure Fragmentation Issues Using Space Analyst for Oracle

Space Analyst can help you quickly fix tablespaces that are suffering from fragmentation by:

- Coalescing adjacent free space fragments (honeycombs) in a tablespace
- Reorganizing a tablespace to eliminate free space bubbles
- Converting a dictionary-managed tablespace to locally-managed

To coalesce free space fragments in a tablespace, follow these steps:

1 On the **Database Overview** tab, select any tablespace that appears to be suffering from fragmentation problems.

2 Select the Open toolbar button or the Open shortcut menu option.

DBArtisan opens a Tablespaces Editor.

3 To view fragmentation details for the tablespace, click the **Space** tab of the Tablespaces Editor.

4 To combine adjacent free space pockets into singular free space chunks, click the Coalesce button.

Note: The fragmentation score of the tablespace does not change after a coalesce operation, then the tablespace likely suffers from bubble fragmentation. This can be verified by viewing the tablespace map for the tablespace.

To reorganize any tablespace suffering from bubble fragmentation, do the following:

1 On the toolbar, click the Open Wizard button.

Space Analyst opens the Reorganization Wizard.

2 Select the New Reorganization option, and then click Next to continue.

Space Analyst opens the next panel of the Reorganization Wizard.

3 Select Reorganize a Tablespace, and then click Next to continue.

Space Analyst opens the next panel of the Reorganization Wizard.

4 Select one or more tablespaces you would like to reorganize, and then click Next to continue.

Space Analyst opens the next panel of the Reorganization Wizard.

5 Select a scratch tablespace that temporarily holds the reorganized tablespace’s objects.

6 Select the option to perform a space analysis so you can be assured that the scratch tablespace has enough room to temporarily hold all of the reorganized tablespace’s objects.

7 If the selected scratch tablespace does not contain enough free space, select another tablespace to serve as the scratch tablespace.

8 Click Next to continue.

Space Analyst opens the next panel of the Reorganization Wizard.

9 You can choose to perform no customizations for the reorganization, or you can select various global or object-level options to customize your reorganization.

10 Click Finish.

Space Analyst builds a reorganization script to reorganize your selected tablespace. Note that while the script is being constructed, you can still perform other work in Space Analyst or DBArtisan. Just minimize the script build window and continue to do other work in DBArtisan. You can check back whenever you'd like to see if the script is ready.

After the reorganization script is complete, you can choose to view/run the script in a DBArtisan ISQL window, schedule the script for later execution, or immediately execute the reorganization.

If you would like to convert a dictionary-managed tablespace to locally-managed, you can do one of the following:

- Create a new locally-managed tablespace in DBArtisan and then use the Reorganization Wizard to permanently relocate all objects from the dictionary-managed tablespace to the new locally-managed tablespace. This is the cleanest method for converting a dictionary-managed tablespace to locally-managed.
- Choose the option in Space Analyst that allows a quick conversion of an existing dictionary-managed tablespace to a locally-managed tablespace. This option is only available for Oracle versions 8.1.6 or later. This option does not defragment an existing dictionary-managed tablespace, but converts it so that all space management is conducted through the tablespace's datafiles instead of the data dictionary and ensure that any new objects added to the tablespace conform to the tablespace's storage attributes.

Proceed to Correcting Space-Related Object Problems for Space Analyst for Oracle

Correcting Space-Related Object Problems for Space Analyst for Oracle

Space Analyst can help you quickly fix objects that are experiencing space problems by:

- Modifying various storage-related properties for an object.
- Reorganizing the object to eliminate space inefficiencies.

Storage properties that are affecting the availability of an object (for example, the next extent or maximum extent values for a table or index) can easily be altered by using the integration that exists between Space Analyst and DBArtisan. For example, to modify a table's next extent value, do the following:

1 On the **Tablespace Map** tab or the Table Details grid on the **Tables** tab, select a table you would like to alter.

2 Select the Open toolbar button or the Open shortcut menu option.

DBArtisan opens a Tablespaces Editor.

3 Select the **Storage** tab.

4 Change the numeric value for the table's next extent.

Note: You can not change next extent values for tables that exist in locally-managed tablespaces.

5 On the toolbar, click the Alter button.

DBArtisan displays the DDL that change the table's next extent value.

You can easily reorganize one or more objects with the Reorganization Wizard. To reorganize any table or index that in any grid, do the following:

1 Select the object(s).

2 On the toolbar, click the Reorganization Wizard button, or right-click and select Reorganize.

Space Analyst opens the Reorganization Wizard.

You can customize object reorganizations by using either the global or specific object customization options in the Reorganization Wizard. For example, you can perform online reorganizations of indexes or index-organized tables.

Space Analyst's Reorganization Wizard can also intelligently identify objects that need to be reorganized. For example, to reorganize tables in your database that have chained/migrated rows, do the following:

1 On the toolbar, click the Reorg Wizard button.

Space Analyst opens the Reorganization Wizard.

2 Select the New Reorganization option, and then click Next to continue.

Space Analyst opens the next panel of the Reorganization Wizard.

3 Select Repair tables with chained/migrate rows, and then click Next to continue.

Space Analyst runs diagnostics against your database to find any table or table partition that suffers from chained rows. Space Analyst displays the objects under either the Tables or Table Partition(s) node in the Reorg Wizard explorer tree. You can then select one or all table objects and continue with the reorganization process.

Space Analyst for Microsoft SQL Server Tutorial (DBArtisan)

Embarcadero Space Analyst provides sophisticated diagnostic capabilities to troubleshoot bottlenecks and performance inefficiencies that result from poor space management. Running inside the Embarcadero DBArtisan console, Space Analyst helps you identify and correct storage problems by supplying powerful analysis facilities and corrective utilities and that will satisfy advanced experts, yet are easy enough to use for database novices.

Space Analyst for Microsoft Product Benefits

Embarcadero Space Analyst allows the database administrator to quickly ascertain if space problems are threatening the availability or performance of their key databases. If storage problems are found, Space Analyst provides powerful but easy to use space management wizards and utilities that can accomplish simple or complex reorganizations of all or parts of a database.

About this Space Analyst for Microsoft Tutorial

This tutorial helps you get started with Embarcadero Space Analyst.

After completion of this tutorial, you will have the foundation you need to explore the many features and benefits of Space Analyst. You will have learned how to get a summary view of storage for your database as well as how to identify potential space-related threats. Finally, you'll use the reorganization wizard to correct any space problems that you find.

This tutorial is divided into 5 sections.

Session 1: Getting Started with Space Analyst

Session 2: Obtaining a Global View of Storage

Session 3: Identifying Storage Structure Problems

Session 4: Pinpointing Object Performance Inefficiencies

Session 5: Correcting Space-Related Problems

You can use this basic tutorial as a roadmap of product highlights.

After starting this tutorial, you can select Help from the menu to find many additional resources that complement and build on many of the activities shown in this brief tutorial.

For more information, see [DBATutorial.htm#AdditionalEvaluationResources Additional Evaluation Resources].

Session 1: Getting started with Space Analyst for Microsoft (DBArtisan)

Download and Install

You can obtain the latest version of Space Analyst by downloading DBArtisan from the Embarcadero Web site at http://www.embarcadero.com/downloads/downloaddbartisan.asp. Space Analyst runs fully contained within DBArtisan so you will need the DBArtisan product to use Space Analyst.

Click Download, and follow the steps indicated. Save the file on your computer and then double-click to launch the self-extracting file that will guide you through the installation process.

When you first install an evaluation copy of DBArtisan, you can use the tool for 14 days. After that time, a permanent license is needed.

Proceed to Space Analyst for Microsoft Overview

Space Analyst for Microsoft Overview

Once you are connected to an Microsoft SQL Server database, you can open Space Analyst from within DBArtisan two ways:

1 On the Analyst menu, select Space Analyst.

Note: You do not need to install anything for Space Analyst to work.

Proceed to Proceed to Session 2: Obtaining a Global View of Storage Using Space Analyst for Microsoft (DBArtisan)

Session 2: Obtaining a Global View of Storage Using Space Analyst for Microsoft (DBArtisan)

When first invoked, Space Analyst displays a complete overview of the storage situation of a particular SQL Server. The first tab of Space Analyst can be used to see all pertinent global statistics regarding storage structure usage throughout a server.

There are four distinct sections displayed on the first tab (labeled Storage Overview) of Space Analyst:

Server Overview

Database Overview

Log Overview

Server Disk Overview

Space Analyst for Microsoft - Server Overview

The total count of all databases, file groups, and files are shown in this section. You can also see the percentage breakdown of the database vs. log space on the server. Positioning the mouse pointer over the graph causes it to display the actual percentage amounts for storage type.

Tip: If you want to see a grid-style display of the information shown in most graphs, position your mouse over the graph, click the right mouse button and select the 'toggle display' option.

Proceed to Space Analyst for Microsoft - Database Overview

Space Analyst for Microsoft - Database Overview

Summary information regarding every database on the server is present in the database overview section. When invoked in graph form, this section shows how much total space is used by each database. When shown in grid form, this section shows not only the total space used for each database, but also includes the total allocated, free, and percent free amounts, along with an indicator that shows whether any of the database's underlying datafiles can automatically grow if necessary.

Tip: To see a grid-style display of the information shown in the graph, position your mouse over the graph, click the right mouse button and select the ‘toggle display’ option. To return to the graph, just perform the same actions over the grid.

When this section is in grid form, you can also perform a number of actions on a database like open it in a DBArtisan editor, extract its DDL, shrink it, and more. For example, if you want to open a database in DBArtisan's database manager, do the following:

1 In the grid, select a database.

2 Right-click and select Open.

OR

3 Select a database and click the Open toolbar button.

DBArtisan opens the Database Manager.

Proceed to Space Analyst for Microsoft - Log Overview

Space Analyst for Microsoft - Log Overview

The log overview provides a quick glance into database transaction log usage. When invoked in graph form, this section shows the percentage of used log space per database. When shown in grid form, this section shows not only the total percentage of transaction log space used for each database, but also includes the total allocated and free amounts (in MB), along with an indicator that shows whether any of the database's transaction logs underlying log files can automatically grow if necessary.

Proceed to Space Analyst for Microsoft - Server Disk Overview

Space Analyst for Microsoft - Server Disk Overview

The server disk overview provides a window into how much free space exists on the actual server machine. In graph form, this section shows the amount of free space (in MB) across each of the server's drives. When in grid form, it shows free space per drive, but also includes the amount of allocated SQL Server data and log space per drive.

Proceed to Session 3: Identifying Storage Structure Problems in Space Analyst for Microsoft (DBArtisan)

Session 3: Identifying Storage Structure Problems in Space Analyst for Microsoft (DBArtisan)

Space Analyst can help you quickly find storage structure problems that threaten the availability of your database.

Proceed to Pinpointing ‘out of space’ Problems Using Space Analyst for Microsoft

Pinpointing ‘out of space’ Problems Using Space Analyst for Microsoft

To quickly tell if any of the tablespaces in your database are about to run out of free space, do the following:

1 Select the **Storage Overview** tab.

2 In the Database Overview and Log Overview sections, position the pointer over the graph.

3 Right-click and select Toggle Display.

Examine the Percent DB Used columns of each database and transaction log along with the DB Can Grow column. If a database or transaction log is nearing the end of free space and cannot automatically grow, then action should be taken to prevent an out-of-space error.

You can also obtain graphical views of databases and pertinent storage property information by viewing the **Databases** tab of Space Analyst. You can visually get a handle on the amounts of table, index, and free space of your databases and quickly view critical storage property facts such as the number of databases that cannot automatically grow in size if needed.

All the graphs can be toggled to grids by using the right mouse menu option of Toggle Display. When in grid mode, very detailed information is available concerning database and transaction log space. Note that the graph is scrollable so you may have to scroll to see all your databases.

You can perform a number of actions on a database, such as shrinking it or invoking database-related DBCC operations by using either the right mouse menu or the Command toolbar button (the yellow arrow).

Proceed to Viewing Filegroup and File Details Using Space Analyst for Microsoft

Viewing Filegroup and File Details Using Space Analyst for Microsoft

You can also view space usage at the filegroup and file levels. To get summary and detailed information for filegroups and files, follow these steps:

1 Invoke Space Analyst and view the **File Groups/Files** tab.

The default presentation is a graphical overview of filegroups. Note that the graph is scrollable so you may have to scroll to see all your filegroups.

2 To view detailed information for filegroups, right-click and select Toggle Display to view the grid.

3 To view file information instead of filegroup metrics, select Files.

When viewing filegroup and file information, you can perform a number of actions on selected filegroups and files such as extracting their DDL or shrinking the parent database or selected file. All options are available by using either the shortcut menu or Command menu.

Proceed to Using Analysis Reports in Space Analyst for Microsoft

Using Analysis Reports in Space Analyst for Microsoft

You can run HTML-formatted reports that pinpoint all major storage-related problems for a particular tablespace and the objects that are contained within it. To produce an Analysis Report, do the following:

1 On the toolbar, click Analysis Report.

Space Analyst opens the **Analysis Report** dialog.

2 In Report File Name and Report Title you can change the location and report title of the report.

3 In Category, select the types of space problems you would like to investigate. The default option is All.

4 In Database, select the database you would like to analyze.

5 In File Group, select the filegroup level.

6 Click Execute.

Space Analyst creates a frame-based Analysis report for the database.

Proceed to Session 4: Pinpointing Object Performance Inefficiencies Using Space Analyst for Microsoft (DBArtisan)

Session 4: Pinpointing Object Performance Inefficiencies Using Space Analyst for Microsoft (DBArtisan)

Space Analyst can greatly accelerate the process of locating objects in your databases that are degrading overall response times because of their current inefficient storage usage.

Proceed to Finding Space-related Table Problems with Space Analyst for Microsoft

Finding Space-related Table Problems with Space Analyst for Microsoft

To quickly determine if any tables in your database may be contributing to performance slowdowns, do the following:

1 Open Space Analyst.

2 Select the **Objects** tab.

3 In Object Details, select the target object.

By default, the General Space button is selected. This view will show you a graphical layout of table used and free space for the database. Note that the graph is scrollable so you can scroll to see all the tables for your database.

The Object details grid in the lower area of the tab can be used to view detailed storage information for every table in the selected database.

To troubleshoot table storage issues in your database, select the Space Diagnostics radio button. This option will interrogate your database and will provide you with a graphical count of the number of tables that are experiencing specific types of storage inefficiencies. Space Analyst uses predefined thresholds (which you can customize by selecting the Threshold Options toolbar button) to determine how many and what objects are suffering from storage inefficiencies.

The table below describes the types of table-related space problems shown in the bar graph found on the Objects Tab:

Problem	Description
Scan Density	This is computed by dividing the optimum number of extent switches by the actual number of extent switches. This percentage can be misleading, however, if the object spans more than one database file and should not be considered in such an event. Values close to 100% are best.
Logical Fragmentation	This metric represents the percentage of pages that are out of physical order. High values (greater than 25%) for this metric may mean that index scan performance is not what it could be. Note that this statistic is meaningless for tables without clustered indexes (heap tables).
Extent Fragmentation	This highlights any gaps (lack of contiguous order) in extents and indicates the presence of external fragmentation. Percentages of 25-50% or more are not a favorable reading.
Average Page Density	This metric represents the fullness of the data or index pages, and is an indicator of internal fragmentation. Remember that the more full a data or index page is, the less I/O needed by SQL Server when performing scans of tables or indexes. High values are good here, with anything below 50% being suspect. Low values for this metric often indicate the presence of page splitting. Keep in mind, however, that internal fragmentation isn't necessarily bad in OLTP environments where large FILLFACTORs are often specified. Also understand that small objects (those will little or no data) will likely show low readings because they simply don't consume enough space in a database page. You can ignore these objects.
Forwarded Records	SQL Server will move rows in a table under certain conditions. One situation might arise when you update a row in a table that has a variable-length column to a larger size that will no longer fit on its original page. Another situation would be if SQL Server moves a row when the clustered index column changes. Forwarded records can reduce performance at times because additional I/O is involved to first obtain the record pointer to the relocated row, and then the row itself. But when does row forwarding become a problem? For example, just because a table has one hundred forwarded rows, does that mean a performance problem exists? The answer is: it depends. If the table has one hundred rows in it, then yes, a problem exists - 100% of the table suffers from forwarded rows. If the table has three million rows, then the forwarded row situation is likely not causing much fuss in the database.

The graph at the top of the **Objects** tab shows the overall count of problem tables, but you can also view the details for each type of problem by using the Details grid at the bottom of the tab. For example, if Space Analyst identifies tables that have forwarded records, then to find the actual tables and view details regarding the identified space problem, do the following:

1 In the File Group list, select All.

2 In the Owner list, select All.

3 In the Display list, select Tables with forwarded records problems.

4 Click Refresh.

You would then see the actual tables that suffer from forwarded records along with their demographic details. If you wanted to take corrective action on any object, you could highlight one or more tables and click the right mouse button to bring up a menu of possible actions.

Proceed to Session 5: Correcting Space-related Problems Using Space Analyst for Microsoft (DBArtisan)

Session 5: Correcting Space-related Problems Using Space Analyst for Microsoft (DBArtisan)

After you have identified space-related bottlenecks in your database with Space Analyst, you can use a number of different tools and facilities within the product to eliminate each storage problems.

Proceed to Correcting Storage Structure ‘out of space’ Problems Using Space Analyst for Microsoft

Correcting Storage Structure ‘out of space’ Problems Using Space Analyst for Microsoft

Space Analyst can help you quickly remedy databases and transaction logs that are running out of room by:

- Resizing an existing database or transaction log to have more room.
- Adding a new filegroup or file to an existing database or transaction log.
- Setting the Autogrowth attribute for one or more of a database's or transaction log's files to allow for automatic growth.

To manage these functions, do the following:

1 Select the Databases tab.

2 In the Database Summary section, right-click the graph, and then select Toggle Display.

3 Select the database that is running low on free space.

4 Select the Open toolbar button or the Open shortcut menu option.

DBArtisan opens the Database Manager.

In the Database Manager, you can remedy the database problem in one of three ways:

- To resize an existing database or transaction log, open one of the database's or log's files in the editor (using the Edit button) and change the file's size to a new, larger size. Then close the editor and select the Alter toolbar button to make the change permanent to the database.
- To add a new file or file group to the database or transaction log (note: you can't add new filegroups to a transaction log, only files), select the Add button in the editor, and specify the new filegroup or file's attributes. Once complete, close the Add Datafile dialog and select the Alter toolbar button to make the change permanent to the database.
- To set Autogrowth for one or more of a database's or transaction log's files, open each file in the file editor by selecting the file and then pressing the Edit button. Once in the Modify File dialog, select the option for Allow Growth and specify other attributes as needed. When complete, close the dialog and select the Alter toolbar button to make the change permanent to the database.

Proceed to Removing Wasted Space in Databases or Files Using Space Analyst for Microsoft

Removing Wasted Space in Databases or Files Using Space Analyst for Microsoft

Space Analyst can help you quickly eliminate unused space in a database or database file. The unused space can be released back to the operating system for other use. To remove unused space from a database, do the following:

1 To shrink an entire database, right-click a database in any of Space Analyst's grids, and then select Shrink Database.

DBArtisan opens the **Shrink File** dialog.

2 To shrink a specific file in a database, select the **Filegroups/Files** tab and then view files by selecting the Files radio button.

3 Right-click a file and select Shrink File.

DBArtisan opens the **Shrink File** dialog.

4 Select desired options.

5 Click Execute to make the change permanent.

Proceed to Correcting Space-related Object Problems for Space Analyst for Microsoft

Correcting Space-related Object Problems for Space Analyst for Microsoft

Once you identify specific objects that need storage corrections, you can use Space Analyst to help you quickly resolve your found issues. The Space Management wizard inside of Space Analyst can be used to build reorganization and space maintenance jobs that run right the first time. All jobs can either be immediately run or scheduled for later execution.

To create a standard database or object reorganization job, do the following:

1 Click the Open Wizard button.

2 Select New Space Management Task.

3 Click Next.

4 Select Create standard reorganization job.

5 Click Next.

6 Select the databases and objects you would like to reorganize from the Explorer tree on the left hand side of the wizard. You can reorganize full databases, selected schema accounts, or selected objects.

7 Click Next.

8 The next panel allows you to customize the reorganization. You can either set options at the global level (affects all objects) or specific object level. Using the options, you can relocate objects to different filegroups, perform online index rebuilds, and specify other customizations.

9 Click Next.

10 The next panel builds your reorganization script and allows you to preview the operation. Click Next when ready.

11 Click Next.

The final panel allows you to open the reorganization script in a SQL editor, immediately run it, or schedule it for later execution.

Proceed to Proactive Space Management Using Space Analyst for Microsoft

Proactive Space Management Using Space Analyst for Microsoft

If you have many databases to take care of, then you might want to relieve the burden of performing detailed storage analysis on every database, each of which may or may not require attention. Fortunately, Space Analyst makes it easy to create smart space management jobs that can automate this task for you.

You can create two specialized space management jobs that allow you to proactively eliminate space threats before they affect your database:

You can create a "reorganization analyst job" that allows you to specifically select various databases and/or objects and then set reorganization thresholds that determine when a reorganization will occur. Once constructed, you can schedule these jobs to run as often as you'd like. During execution, these jobs will interrogate your chosen database/objects and determine if a reorganization needs to occur by applying your customized reorganization thresholds. If any objects are found to need reorganization, they are dynamically reorganized using any customized options you specify.

You can build space maintenance jobs that will perform functions such as check the validity of your databases and filegroups, repair any found minor corruption, and eliminate unused space in your databases. You can also set the job up to notify you if any problems are found during job execution.

These "smart" space maintenance jobs allow you to truly "set and forget" the task of monitoring and reorganizing the space of your critical databases.

Space Analyst for Sybase ASE Tutorial (DBArtisan)

Embarcadero Space Analyst provides sophisticated diagnostic capabilities to troubleshoot bottlenecks and performance inefficiencies that result from poor space management. Running inside the Embarcadero DBArtisan console, Space Analyst helps you identify and correct storage problems by supplying powerful analysis facilities and corrective utilities and that will satisfy advanced experts, yet are easy enough to use for database novices.

Space Analyst for Sybase ASE Product Benefits

Embarcadero Space Analyst allows the database administrator to quickly ascertain if space problems are threatening the availability or performance of their key databases. If storage problems are found, Space Analyst provides powerful but easy to use space management wizards and utilities that can accomplish simple or complex reorganizations of all or parts of a database.

About this Space Analyst for Sybase ASE Tutorial

This tutorial helps you get started with Embarcadero Space Analyst.

After completion of this tutorial, you will have the foundation you need to explore the many features and benefits of Space Analyst. You will have learned how to get a summary view of storage for your database as well as how to identify potential space-related threats. Finally, you'll use the reorganization wizard to correct any space problems that you find.

This tutorial is divided into 4 sections.

Session 1: Getting Started With Space Analyst for Sybase ASE (DBArtisan)

Session 2: Obtaining a Global View of Storage Using Space Analyst for Sybase ASE (DBArtisan)

Session 3: Identifying Storage Structure Problems in Space Analyst for Sybase ASE (DBArtisan)

Session 4: Correcting Space-related Problems Using Space Analyst for Sybase ASE (DBArtisan)

You can use this basic tutorial as a roadmap of product highlights.

After starting this tutorial, you can select Help from the menu to find many additional resources that complement and build on many of the activities shown in this brief tutorial.

For more information, see [DBATutorial.htm#AdditionalEvaluationResources Additional Evaluation Resources].

Session 1: Getting Started With Space Analyst for Sybase ASE (DBArtisan)

Download and Install

You can obtain the latest version of Space Analyst by downloading DBArtisan from the Embarcadero Web site at http://www.embarcadero.com/downloads/downloaddbartisan.asp. Space Analyst runs fully contained within DBArtisan so you will need the DBArtisan product to use Space Analyst.

Click Download, and follow the steps indicated. Save the file on your computer and then double-click to launch the self-extracting file that will guide you through the installation process.

When you first install an evaluation copy of DBArtisan, you can use the tool for 14 days. After that time, a permanent license is needed.

Proceed to Space Analyst for Sybase ASE Overview

Space Analyst for Sybase ASE Overview

The graphic below illustrates all the main elements of the DBArtisan User Interface.

Once you are connected to an Sybase ASE database, you can open Space Analyst from within DBArtisan two ways:

1 On the Analyst menu, select Space Analyst.

Note: You do not need to install anything for Space Analyst to work.

Proceed to Session 2: Obtaining a Global View of Storage Using Space Analyst for Sybase ASE (DBArtisan)

Session 2: Obtaining a Global View of Storage Using Space Analyst for Sybase ASE (DBArtisan)

When first invoked, Space Analyst displays a complete overview of the storage situation of a particular server. The first tab of Space Analyst can be used to see all pertinent global statistics regarding storage structure usage throughout a server.

There are four distinct sections displayed on the **Storage Overview** tab of Space Analyst:

Server Overview

Log Overview

Database Overview

Device Overview

Space Analyst for Sybase ASE - Server Overview

The total count of all databases, file groups, and files are shown in this section. You can also see the percentage breakdown of the database vs. log space on the server. Positioning the mouse pointer over the graph causes it to display the actual percentage amounts for storage type.

Tip: If you want to see a grid-style display of the information shown in most graphs, position your mouse over the graph, click the right mouse button and select the 'toggle display' option.

Proceed to Space Analyst for Sybase ASE - Log Overview

Space Analyst for Sybase ASE - Log Overview

The log overview provides a quick glance into database transaction log usage. When invoked in graph form, this section shows the percentage of used log space per database. When shown in grid form, this section shows not only the total percentage of transaction log space used for each database, but also includes the total allocated and free amounts (in MB), along with an indicator that shows whether any of the database's transaction logs underlying log files can automatically grow if necessary.

Proceed to Space Analyst for Sybase ASE - Database Overview

Space Analyst for Sybase ASE - Database Overview

Summary information regarding every database on the server is present in the database overview section. When invoked in graph form, this section shows how much total space is used by each database. When shown in grid form, this section shows not only the total space used for each database, but also includes the total allocated, free, and percent free amounts, along with an indicator that shows whether any of the database's underlying datafiles can automatically grow if necessary.

Tip: To see a grid-style display of the information shown in the graph, position your mouse over the graph, click the right mouse button and select the ‘toggle display’ option. To return to the graph, just perform the same actions over the grid.

When this section is in grid form, you can also perform a number of actions on a database like open it in a DBArtisan editor, extract its DDL, shrink it, and more. For example, if you want to open a database in DBArtisan's database manager, do the following:

1 Select a database, right-click, and select Open.

OR

Select a database and click the Open toolbar button.

DBArtisan opens the Database Manager.

Proceed to Space Analyst for Sybase ASE - Device Overview

Space Analyst for Sybase ASE - Device Overview

The device overview section contains a graphical and grid control that works in the following way:

- When the graph is displayed, a bar chart is shown that lists the total space used (in MB) of all devices defined to the server.
- When the grid is displayed, detailed storage information for each database device is presented:
- Device name
- Physical name
- Total space
- Used space
- Free space
- Percent free
- Percent used

Proceed to Session 3: Identifying Storage Structure Problems in Space Analyst for Sybase ASE (DBArtisan)

Session 3: Identifying Storage Structure Problems in Space Analyst for Sybase ASE (DBArtisan)

Space Analyst can help you quickly find storage structure problems that threaten the availability of your database.

Proceed to Pinpointing ‘out of space’ Problems Using Space Analyst for Sybase ASE

Pinpointing ‘out of space’ Problems Using Space Analyst for Sybase ASE

To quickly tell if any of the tablespaces in your database are about to run out of free space, do the following:

1 Select the **Storage Overview** tab.

2 In the Database Overview and Log Overview sections, position the pointer over the graph.

3 Right-click and select Toggle Display.

Examine the Database Percent Free columns of each database and Log Percent Free for each transaction log. If a database or transaction log is nearing the end of free space and cannot automatically grow, then action should be taken to prevent an out-of-space error.

You can also obtain graphical views of databases and pertinent storage property information by viewing the **Databases** tab of Space Analyst. You can visually get a handle on the amounts of table, index, and free space of your databases and quickly view critical storage property facts such as the number of databases that cannot automatically grow in size if needed.

All the graphs can be toggled to grids by using the right mouse menu option of 'Toggle Display'. When in grid mode, very detailed information is available concerning database and transaction log space. Note that the graph is scrollable so you may have to scroll to see all your databases.

Proceed to Using Analysis Reports in Space Analyst for Sybase ASE

Using Analysis Reports in Space Analyst for Sybase ASE

You can run HTML-formatted reports that pinpoint all major storage-related problems for a particular tablespace and the objects that are contained within it. To produce an Analysis Report, do the following:

1 On the toolbar, click Analysis Report.

Space Analyst opens the **Analysis Report** dialog.

2 In Report File Name and Report Title you can change the location and report title of the report.

3 In Category, select the types of space problems you would like to investigate. The default option is All.

4 In Database, select the database you would like to analyze.

5 In File Group, select the filegroup level.

6 Click Execute.

Space Analyst creates a frame-based Analysis report for the database.

Proceed to Session 4: Correcting Space-related Problems Using Space Analyst for Sybase ASE (DBArtisan)

Session 4: Correcting Space-related Problems Using Space Analyst for Sybase ASE (DBArtisan)

After you have identified space-related bottlenecks in your database with Space Analyst, you can use a number of different tools and facilities within the product to eliminate each storage problems.

Proceed to Correcting Storage Structure ‘out of space’ Problems Using Space Analyst for Sybase ASE

Correcting Storage Structure ‘out of space’ Problems Using Space Analyst for Sybase ASE

Space Analyst can help you quickly remedy databases and transaction logs that are running out of room by:

- Resizing an existing database or transaction log to have more room.
- Adding a new filegroup or file to an existing database or transaction log.
- Setting the Autogrowth attribute for one or more of a database's or transaction log's files to allow for automatic growth.

To manage these functions, do the following:

1 Select the **Storage Overview** tab.

2 In the Database Overview or Log Overview section, right-click the graph, and then select Toggle Display.

3 Select the database that is running low on free space.

4 Select the Open toolbar button or the Open shortcut menu option.

DBArtisan opens the Database Manager.

In the Database Manager, you can remedy the database problem in one of three ways:

- To resize an existing database or transaction log, open one of the database's or log's files in the editor (using the Edit button) and change the file's size to a new, larger size. Then close the editor and select the Alter toolbar button to make the change permanent to the database.
- To add a new file or file group to the database or transaction log (note: you can't add new filegroups to a transaction log, only files), select the Add button in the editor, and specify the new filegroup or file's attributes. Once complete, close the Add Datafile dialog and select the Alter toolbar button to make the change permanent to the database.
- To set Autogrowth for one or more of a database's or transaction log's files, open each file in the file editor by selecting the file and then pressing the Edit button. Once in the Modify File dialog, select the option for Allow Growth and specify other attributes as needed.

When complete, close the dialog and select the Alter toolbar button to make the change permanent to the database.

Proceed to Correcting Space-related Object Problems for Space Analyst for Sybase ASE

Correcting Space-related Object Problems for Space Analyst for Sybase ASE

Once you identify specific objects that need storage corrections, you can use Space Analyst to help you quickly resolve your found issues. The Space Management wizard inside of Space Analyst can be used to build reorganization and space maintenance jobs that run right the first time. All jobs can either be immediately run or scheduled for later execution.

To create a standard database or object reorganization job, do the following:

1 Click the Open Wizard button.

2 Select New Space Management Task.

3 Click Next.

4 Select Create standard reorganization job.

5 Click Next.

6 Select the databases and objects you would like to reorganize from the Explorer tree on the left hand side of the wizard. You can reorganize full databases, selected schema accounts, or selected objects.

7 Click Next.

8 The next panel allows you to customize the reorganization. You can either set options at the global level (affects all objects) or specific object level.

9 Click Next.

10 The next panel builds your reorganization script and allows you to preview the operation. Click Next when ready.

11 Click Next.

The final panel allows you to open the reorganization script in a SQL editor, immediately run it, or schedule it for later execution.

Proceed to Proactive Space Management Using Space Analyst for Sybase ASE

Proactive Space Management Using Space Analyst for Sybase ASE

If you have many databases to take care of, then you might want to relieve the burden of performing detailed storage analysis on every database, each of which may or may not require attention. Fortunately, Space Analyst makes it easy to create smart space management jobs that can automate this task for you.

You can create two specialized space management jobs that allow you to proactively eliminate space threats before they affect your database:

You can create a "reorganization analyst job" that allows you to specifically select various databases and/or objects and then set reorganization thresholds that determine when a reorganization will occur. Once constructed, you can schedule these jobs to run as often as you'd like. During execution, these jobs will interrogate your chosen database/objects and determine if a reorganization needs to occur by applying your customized reorganization thresholds. If any objects are found to need reorganization, they are dynamically reorganized using any customized options you specify.

You can build space maintenance jobs that will perform functions such as check the validity of your databases and filegroups, repair any found minor corruption, and eliminate unused space in your databases. You can also set the job up to notify you if any problems are found during job execution.

These "smart" space maintenance jobs allow you to truly "set and forget" the task of monitoring and reorganizing the space of your critical databases.

Space Analyst for IBM DB2 for Linux, Unix, and Windows Tutorial (DBArtisan)

Embarcadero Space Analyst provides sophisticated diagnostic capabilities to troubleshoot bottlenecks and performance inefficiencies that result from poor space management. Running inside the Embarcadero DBArtisan console, Space Analyst helps you identify and correct storage problems by supplying powerful analysis facilities and corrective utilities and that will satisfy advanced experts, yet are easy enough to use for database novices.

Space Analyst for IBM DB2 for Linux, Unix, and Windows Product Benefits

Embarcadero Space Analyst allows the database administrator to quickly ascertain if space problems are threatening the availability or performance of their key databases. If storage problems are found, Space Analyst provides powerful but easy to use space management wizards and utilities that can accomplish simple or complex reorganizations of all or parts of a database.

About this Space Analyst for IBM DB2 for Linux, Unix, and Windows Tutorial

This tutorial helps you get started with Embarcadero Space Analyst.

After completion of this tutorial, you will have the foundation you need to explore the many features and benefits of Space Analyst. You will have learned how to get a summary view of storage for your database as well as how to identify potential space-related threats. Finally, you'll use the reorganization wizard to correct any space problems that you find.

This tutorial is divided into 5 sections.

Session 1: Getting Started with Space Analyst

Session 2: Obtaining a Global View of Storage

Session 3: Identifying Storage Structure Problems (DBArtisan)

Session 4: Pinpointing SQL server Performance Inefficiencies (DBArtisan)

Session 5: Correcting Space-related Problems (DBArtisan)

You can use this basic tutorial as a roadmap of product highlights.

After starting this tutorial, you can select Help from the menu to find many additional resources that complement and build on many of the activities shown in this brief tutorial.

For more information, see [DBATutorial.htm#AdditionalEvaluationResources Additional Evaluation Resources].

Session 1: Getting Started with Space Analyst for IBM DB2 for Linux, Unix, and Windows (DBArtisan)

Download and Install

You can obtain the latest version of Space Analyst by downloading DBArtisan from the Embarcadero Web site at http://www.embarcadero.com/downloads/downloaddbartisan.asp. Space Analyst runs fully contained within DBArtisan so you will need the DBArtisan product to use Space Analyst.

Click Download, and follow the steps indicated. Save the file on your computer and then double-click to launch the self-extracting file that will guide you through the installation process.

When you first install an evaluation copy of DBArtisan, you can use the tool for 14 days. After that time, a permanent license is needed.

Proceed to Space Analyst for IBM DB2 for Linux, Unix, and Windows Overview

Space Analyst for IBM DB2 for Linux, Unix, and Windows Overview

Once you are connected to an IBM DB2 for Linux, Unix, and Windows database, you can open Space Analyst from within DBArtisan two ways:

1 On the Analyst menu, select Space Analyst.

Note: You do not need to install anything for Space Analyst to work.

Proceed to Session 2: Obtaining a Global View of Storage Using Space Analyst for IBM DB2 for Linux, Unix, and Windows (DBArtisan)

Session 2: Obtaining a Global View of Storage Using Space Analyst for IBM DB2 for Linux, Unix, and Windows (DBArtisan)

When it opens, Space Analyst shows a complete overview of the storage situation of a particular database or database server. The first tab of Space Analyst displays all pertinent global statistics regarding storage structure usage throughout a database.

There are four distinct sections displayed on the first **Storage Overview** tab of Space Analyst:

- Storage Overview
- Tablespace Space Overview
- Log Overview
- Tablespace Objects Overview

Space Analyst for IBM DB2 for Linux, Unix, and Windows - Storage Overview

The breakdown of space usage of the allocated tablespace space, by index, table and temp data is shown in this section. You can also see the total number of tablespaces and the number SMS and DMS spaces here.

Note: Make sure your database object statistics are up to date. Space analyst uses these statistics in the various graphs and grids.

Proceed to Space Analyst for IBM DB2 for Linux, Unix, and Windows -Tablespace Space Overview

Space Analyst for IBM DB2 for Linux, Unix, and Windows - Tablespace Space Overview

The Tablespace Space Overview section displays summary information regarding every tablespace on the database. When invoked in graph form, this section shows how much total space is allocated to each tablespace. When shown in grid form, this section shows not only the total allocated for each tablespace, but also includes information such as used space, free space, state, number of containers etc.

Tip: To see a grid-style display of the information shown in the graph, position your mouse over the graph, click the right mouse button and select the 'toggle display' option. To return to the graph, just perform the same actions over the grid.

When this section is in grid form, you can also perform a number of actions on a tablespace like open it in a DBArtisan editor, extract its DDL, update its object statistics and more. For example, if you want to update the statistics on all tables in a tablespace, you would perform the following:

1 Select a tablespace in a grid.

2 Right-click the tablespace, and select Update Statistics.

OR

3 On the Space Analyst toolbar click the Update Statistics.

Proceed to Space Analyst for IBM DB2 for Linux, Unix, and Windows - Log Overview

Space Analyst for IBM DB2 for Linux, Unix, and Windows - Log Overview

The log overview provides a quick glance into database transaction log usage. This section shows the percentage of used and free log space for the database, the various database configuration parameters associated with the log space and the high watermarks for the log space usage.

Proceed to Space Analyst for IBM DB2 for Linux, Unix, and Windows -Tablespace Objects Overview

Space Analyst for IBM DB2 for Linux, Unix, and Windows - Tablespace Objects Overview

The tablespace objects overview section provides an overview of how much space is used by the tables and indexes on each tablespace.

Proceed to Session 3: Identifying Storage Structure Problems (DBArtisan)

Session 3: Identifying Storage Structure Problems (DBArtisan)

Space Analyst can help you quickly find storage structure problems that threaten the availability of your databases.

Proceed to Pinpointing 'out of space' Problems

Pinpointing 'out of space' Problems

You can quickly tell if your transaction logs are about to run out of free space by following these steps:

1 Open Space Analyst.

2 View the Database and Log Overview sections on the **Storage Overview** tab.

3 Examine the free and used space metrics in the pie chart. If the transaction log is nearing the end of free space, then action should be taken to prevent a log full error.

You can also obtain graphical views of tablespaces and pertinent storage property information by viewing the Tablespaces tab of Space Analyst. You can visually get a handle on the amount of used and free space of your tablespaces and quickly view critical storage property facts such as the number of containers for each tablespace, the space allocated for each container, the usage of tablespace space on each partition in a multi-partition database.

All the graphs can be toggled to grids by using the right mouse menu option of 'Toggle Display'. When in grid mode, very detailed information is available concerning the selected tablespace.

Note: You can perform further corrective actions on a tablespace, by opening it in the tablespace editor by using either the right mouse menu or the Command toolbar button (the yellow arrow).

Proceed to Using Analysis Reports

Using Analysis Reports

You can run HTML-formatted reports that pinpoint all major storage-related problems for a particular database and the objects that are contained within it. To produce an Analysis Report, follow these steps:

1 On the Space Analyst toolbar, select Analysis Report.

2 If you want to change the location and report title of the report, edit the predefined entries for the Report File Name and Report Title.

3 Select the types of space problems you would like identified on the report by using the Category list.

4 Click Execute.

Space Analyst creates a frame-based Analysis report for the database you indicated.

Proceed to Session 4: Pinpointing SQL server Performance Inefficiencies (DBArtisan)

Session 4: Pinpointing SQL server Performance Inefficiencies (DBArtisan)

Space Analyst can greatly accelerate the process of locating objects in your database that are degrading overall response times because of their current inefficient storage usage.

Proceed to Finding Space-related Table Problems

Finding Space-related Table Problems

You can quickly tell if any tables in a tablespace that may be contributing to performance slowdowns by following these steps:

1 Open the Space Analyst and select the **Tables** tab.

2 Select the tablespace you would like to interrogate in the Tablespace list.

3 Click the Refresh button.

You with see a graphical count of the number of tables that are experiencing specific types of storage inefficiencies in the upper area of the tab. The table reorganization candidate details grid in the lower area of the tab can be used to view detailed storage information for every table in the selected tablespace. Space Analyst uses predefined thresholds (which you can customize by selecting the Threshold Options toolbar button) to determine how many and what tables are suffering from storage inefficiencies.

The types of table-related space problems shown in the bar graph found on the objects tab include:

- Overflow Problems -Overflows occur in tables when overflows occur when rows are updated with more bytes than the original row (VARCHAR fields etc.). Overflows also occurs when the new columns are added to existing tables. Since the overflow data will not be stored contiguously with the other data, it causes access inefficiencies.
- Space Utilization Problems - Normally, DB2 reclaims any space freed up by deletes, in subsequent insert/update operations. However, if a DB2 table has the 'APPEND' mode set to on and a large amount of data is deleted from the table DB2 will not reclaim freed space. Also when some types of large load operations are performed, the table may be left with a large amount of unused space.
- Page Utilization Problems - As the number of empty pages increases, the need for table reorganization also increases. Empty pages waste space and reduce the table scan efficiency, since empty pages are read into the buffer pool for a table scan wasting bufferpool memory.
- Index Contention - Unless the user explicitly separates the index and table tablespaces, DB2 tables and indexes are placed on the same tablespace. This can become a bottleneck with heavily accessed tables using index scans since data and index accesses will compete for the IO device resources.

The graph at the top of the Tables tab shows the overall count of problem tables, but you can also view the details for each type of problem by using the Details grid at the bottom of the tab. For example, if Space Analyst identifies tables that have overflow records, then to find the actual tables and view details regarding the identified space problem, you would perform the following:

1 Select All from the Tablespace list in the Details grid.

2 Select All from the Table Schema list in the Details grid.

3 Select Tables with Overflow Problems from the Display list in the Details grid.

4 Click Refresh.

You would then see the actual tables that suffer from overflowed records along with their demographic details. If you wanted to take corrective action on any object, you could highlight one or more tables and click the right mouse button to bring up a menu of possible actions.

Proceed to Finding Space-related Index Problems (DBArtisan)

Finding Space-related Index Problems (DBArtisan)

You can quickly tell if any indexes in your database may be contributing to performance slowdowns by following the exact same steps as outline above for tables with one exception: You want to select the Indexes tab instead of the Tables tab. The overview graph and index details sections of the Indexes tab work in the exact same manner for indexes as they do for tables.

Proceed to Session 5: Correcting Space-related Problems (DBArtisan)

Session 5: Correcting Space-related Problems (DBArtisan)

Once you have identified space-related bottlenecks in your database with Space Analyst, you can use a number of different tools and facilities within the product to eliminate each storage headache.

Proceed to Correcting Tablespace 'out of space' Problems

Correcting Tablespace 'out of space' Problems

Space Analyst can help you quickly remedy tablespaces that are running out of room by:

- Resizing existing tablespace containers to have more room.
- Adding new containers file to an existing tablespace.
- Moving Objects from a near full tablespace to less utilized tablespace

Resizing and adding of tablespace containers can be managed by performing the following steps:

1 Open Space Analyst and select the **Database Overview** tab.

2 Select the graph in Tablespace Overview section and change to the grid display by selecting the right mouse menu option toggle display.

3 Select the DMS tablespace that is running low on free space and either select the Open toolbar option or the Open right mouse menu option.

This opens the Tablespace in DBArtisan's tablespace editor. Once you are in the editor, you can remedy the tablespace problem in one of two ways:

- To resize an existing container, select the container from the grid and click the Edit button and extend/ resize the container to a new, larger size. Then close the editor and select the Alter toolbar button to make the change permanent to the tablespace.
- To add a new container to the tablespace, select the Add button in the editor, and specify the new container attributes. Once complete, close the Add Container dialog and select the Alter toolbar button to make the change permanent to the database.

Proceed to Correcting Space-related Object Problems

Correcting Space-related Object Problems

Once you identify specific objects that need storage corrections, you can use Space Analyst to help you quickly resolve your found issues. The Space Management wizard inside of Space Analyst can be used to build reorganization and space maintenance jobs that run right the first time. All jobs can either be immediately run or scheduled for later execution.

To create a standard database or object reorganization job, follow these steps:

1 On the toolbar, click the Space Management.

2 Select New Space Management Task.

3 Click Next.

4 Select Create standard reorganization job.

5 Click Next.

6 Select the objects you would like to reorganize from the Explorer tree on the left hand side of the wizard. Note that you can reorganize full tablespaces, selected schemas or selected tables.

7 Click Next.

8 The next few panels let you customize the reorganization. You can choose to either set options at the global level (affects all objects) or specific object level. Using the options, you can reorganize tables and indexes, perform LOB and Long data reorganizations and specify other customizations.

9 The second to last panel builds your reorganization script and allows you to preview the operation.

10 Click Next.

11 The final panel allows you to execute the reorganization script immediately, or schedule it for later execution.

Proceed to Proactive Space Management

Proactive Space Management

If you have many tablespace/schemas to take care of, then you might want to relieve the burden of performing detailed storage analysis on every tablespace or schema, each of which may or may not require attention. Fortunately, Space Analyst makes it easy to create smart space management jobs that can automate this task for you.

You can create a "reorganization analyst job" that allows you to specifically select various tablespaces/ schemas or objects and then set reorganization thresholds that determine when reorganization will occur. Once constructed, you can schedule these jobs to run as often as you'd like. During execution, these jobs will interrogate your chosen database/objects and determine if a reorganization needs to occur by applying your customized reorganization thresholds. If any objects are found to need reorganization, they are dynamically reorganized using any customized options you specify. Optionally, this job can also automatically update the statistics for the selected objects before and after reorganization and rebind any packages that depend on the reorganized objects.

These "autonomic" space maintenance jobs allow you to truly "set and forget" the task of monitoring and reorganizing the space of your critical databases.

Space Analyst Expert Guide (DBArtisan)

IBM DB2 fragmentation can be broken down into two basic types:

- External Fragmentation - this situation exists when indexes have a logical order, based on their key value, which does not match the actual physical order inside the database file that houses the index. When an ordered scan is performed on an index, I/O performance is enhanced when the physical order of the database pages matches the logical order of the index. This is because the disk head can scan in one direction instead of moving back and forth to obtain the needed information.
- Internal Fragmentation - this situation exists when tables and indexes aren't utilizing space as efficiently as they should. The amount of wasted space in the database pages artificially inflates the size of the table or index and causes more I/O to be performed than would otherwise be needed if the object were compacted.

Keep in mind that fragmentation is not a performance factor at all once data reaches the IBM DB2 memory caches.

Related Topics

Environments That Benefit From Fragmentation Elimination

Diagnosing Fragmentation

"I've Reorganized, But It Didn't Help!"

Forwarded Rows

Environments That Benefit From Fragmentation Elimination (DBArtisan)

One important point must be understood regarding fragmentation: not every situation benefits from fragmentation removal. Before you invest a lot of time and effort into diagnosing and attempting the removal of fragmentation, you first need to see if jumping through hoops will actually improve your database's availability and performance.

At the highest level, the environments that benefit the most from fragmentation removal are read-intensive databases where large indexes are being frequently scanned. There are a couple of reasons why this is the case.

In most OLTP environments, data retrieval tends to be mostly selective, which negates most of the bad effects of external fragmentation. OLTP environments also often benefit from internal fragmentation because it's smart to leave room on index pages that are being added to and updated frequently. Free index page space helps avoid the dreaded page split where a page is divided in two due to the fact that incoming clustered index data can't logically fit on a needed page, so IBM DB2 makes room by creating two index pages out of one. Administrators oftentimes specify a low FILLFACTOR for their indexes to create internal fragmentation and hopefully avoid page splits.

Data warehousing or databases with many resource-intensive/scanning queries are another matter. These environments will likely benefit from fragmentation removal. One reason is that the indexes tend to be larger, and therefore are not cached by IBM DB2 as easily as small indexes. Remember that fragmentation has no effect on performance once the data is nestled safely in IBM DB2's memory caches.

Another reason why these environments benefit from fragmentation removal is the effect fragmentation has on IBM DB2's read ahead manager. The read ahead manager helps queries that perform large scans by scanning index pages (and data pages) that it believes will be read and placing them into memory before they are actually needed. Naturally, this process can reduce overall scan times because data read in memory is many times faster than when that same data must be retrieved from disk. However, fragmentation affects how well the read ahead manager works. The read ahead manager will dynamically adjust the size of I/O it performs based on the actual physical order of the data and index pages on the server. When little external fragmentation is present, the read ahead manager can work very efficiently because it can read larger blocks at a time, whereas excessive fragmentation causes the read ahead manager to read smaller blocks of data. The end result when fragmentation is present is less overall throughput.

SQL execution patterns are something else to examine. Queries that perform large index scans are the ones that should see performance improvements when fragmentation is removed in a database. In DBArtisan, you can visually set I/O statistics on in the ISQL facility and examine queries to determine the amount of physical and logical read activity. Queries that show physical and read ahead I/O may see their response times drop when fragmentation has been removed.

Diagnosing Fragmentation (DBArtisan)

If you have an environment that will benefit from reducing fragmentation, how do you go about recognizing it? There are a number of metrics and factors to consider before you can intelligently start creating reorganization plans.

When diagnosing fragmentation, pay particular attention to the following metrics:

- Extent Fragmentation - this highlights any gaps (lack of contiguous order) in extents and indicates the presence of external fragmentation. Percentages of 25-50% or more are not a favorable reading.
- Logical Fragmentation - this metric represents the percentage of pages that are out of physical order. High values (greater than 25%) for this metric may mean that index scan performance is not what it could be. Note that this statistic is meaningless for tables without clustered indexes (heap tables).
- Average Page Density - this metric represents the fullness of the data or index pages, and is an indicator of internal fragmentation. Remember that the more full a data or index page is, the less I/O needed by IBM DB2 when performing scans of tables or indexes. High values are good here, with anything below 50% being suspect. Low values for this metric often indicate the presence of page splitting. Keep in mind, however, that internal fragmentation isn't necessarily bad in OLTP environments where large FILLFACTORs are often specified. Also understand that small objects (those will little or no data) will likely show low readings because they simply don't consume enough space in a database page. You can ignore these objects.
- Extent Switches - in a perfect world, this will equal # of object extents - 1. Higher values (values many times higher than an object's extent count) can indicate external fragmentation.
- Scan Density - this is computed by dividing the optimum number of extent switches by the actual number of extent switches. This percentage can be misleading, however, if the object spans more than one database file and should not be considered in such an event. Values close to 100% are best.

As you can see, you can't just blindly diagnose fragmentation in IBM DB2, but instead have to evaluate a number of factors.

"I've Reorganized, But It Didn't Help!" (DBArtisan)

Sometimes, a DBA will reorganize a table or index and no measurable benefits are noticed afterwards. The same or worse fragmentation metrics are observed for the objects in question. What could have gone wrong?

When attempting the reorganization of either tables or indexes, you have to understand that only objects that are of a significant size will show improvement from a reorganization. IBM DB2 for Linux, Unix, and Windows has stated that objects with less than 1,000 pages will oftentimes offer no performance improvements whatsoever from being reorganized.

Other points to keep in mind include:

- Tables without indexes receive zero benefit (naturally) from running either a DBCC DBREINDEX or a DBCC INDEXDEFRAG. The only way to reorganize a table without indexes is to create a clustered index on the table and then immediately drop it.
- Heap tables should not be diagnosed as having fragmentation through the logical scan fragmentation measure. This metric is meaningless for such tables.
- As mentioned above, small objects seldom benefit from reorganizations. For example, the page density for a table with only 10 rows in it will likely not be high, and reorganizing it won't change things one bit. You need to keep such things in mind when determining whether to reorganize an object or not. Space Analyst keeps you informed of not only reorganization metrics but object metrics (extent size, row counts, etc.) as well so you can make smart reorganization choices.

- Lack of contiguous free space can result in all fragmentation not being eliminated. Without large amounts of contiguous free space, reorganization operations may have to reuse other areas of space within the database files. The end result is oftentimes an index being reorganized, but still having an amount of logical scan fragmentation.

Forwarded Rows (DBArtisan)

IBM DB2 will move rows in a table under certain conditions. One situation might arise when you update a row in a table that has a variable-length column to a larger size that will no longer fit on its original page. Another situation would be if IBM DB2 moves a row when the clustered index column changes. When IBM DB2 creates a forwarding pointer, the pointer remains in place unless one of two things happens. The first is when a row shrinks enough to move back to its original location. The second is when the entire database shrinks. When a database file shrinks, IBM DB2 will reassign the row identifiers, which are used as the row locators, so the shrink process never generates forwarded rows.

Forwarded records can reduce performance at times because additional I/O is involved to first obtain the record pointer to the relocated row, and then the row itself. But when does row forwarding become a problem? For example, just because a table has one hundred forwarded rows, does that mean a performance problem exists? The answer is: it depends. If the table has one hundred rows in it, then yes, a problem exists - 100% of the table suffers from forwarded rows. If the table has three million rows, then the forwarded row situation is likely not causing much fuss in the database.

Parameters, Arguments, Options, and Variables Appendix (DBArtisan)

This appendix provides nothing more than a listing of the server initialization parameters, servers and options supported by DBArtisan.

Note: It is beyond the scope of this document to provide detailed documentation on the parameters and options listed. Refer to the documentation set of the relevant DBMS platform for details. For guidance, see Accessing Third Party Documentation.

See the following topics for listings:

- Supported DB2 LUW System Configuration Parameters (DBArtisan)
- Supported MySQL Server System Variables (DBArtisan)
- Supported Oracle Initialization Parameters (DBArtisan)
- Supported SQL Server Configuration Options (DBArtisan)
- Supported Sybase ASE Configuration Parameters (DBArtisan)
- Supported Sybase IQ Configuration Parameters (DBArtisan)
- Data Pump Import/Export Parameters that can be Added Manually (DBArtisan)

Supported DB2 LUW System Configuration Parameters (DBArtisan)

The following parameters are available from the Explorer's **Database Configuration** node. For information on working with these parameters, see Working with Configuration/initialization Parameters (DBArtisan).

- app_ctl_heap_sz
- appgroup_mem_sz
- applheapsz
- archretrydelay
- auto_maint
- auto_db_backup
- auto_tbl_maint
- auto_runstats
- auto_stmt_stats
- auto_stats_prof
- auto_prof_upd
- auto_reorg
- autorestart
- avg_appls
- blk_log_dsk_fu
- catalogcache_sz
- chngpgs_thresh
- database_memory
- dbheap
- dft_degree
- dft_extent_sz
- dft_loadrec_ses
- dft_mttb_types
- dft_prefetch_sz
- dft_queryopt
- dft_refresh_age
- dft_sqlmathwarn
- discover_db
- dlchktime
- dyn_query_mgmt
- failarchpath
- groupheap_ratio
- hadr_local_host
- hadr_local_svc
- hadr_remote_host
- hadr_remote_inst
- hadr_remote_svc
- hadr_syncmode
- hadr_timeout
- hadr_peer_window
- indexrec Index
- locklist Maximum storage
- locktimeout Lock timeout
- logarchmeth1
- logarchmeth2
- logarchopt1
- logarchopt2
- logbufsz
- logfilsiz
- logindexbuild
- logprimary
- logretain
- logsecond
- max_log
- maxappls
- maxfilop
- maxlocks
- min_dec_div_3
- mincommit
- mirrorlogpath
- newlogpath
- num_db_backups
- num_freqvalues
- num_iocleaners
- num_ioservers
- num_log_span
- num_quantiles
- numarchretry
- overflowlogpath
- pckcachesz
- rec_his_retentn
- self_tuning_mem
- seqdetect
- sheapthres_shr
- softmax
- sortheap
- stat_heap_sz
- stmtheap
- trackmod
- tsm_mgmtclass
- tsm_nodename
- tsm_owner
- tsm_password
- userexit

util_heap_sz	vendoropt	backup_pending
codepage	codeset	collate_info
country	database_consistent	database_level
hadr_db_role	log_retain_status	loghead
logpath	multipage_alloc	numsegs
pagesize	release	restore_pending
rollfwd_pending	territory Database	user_exit_status
db_mem_thresh	db_collname	enable_xmlchar
restrict_access	appl_memory	wlm_collect_int
decflt_rounding		

Supported MySQL Server System Variables (DBArtisan)

The following variables are available from the Explorer's **Configuration** node. For information on working with these variables, see Working with Configuration/initialization Parameters (DBArtisan).

autocommit
back_log
basedir
big-tables
binlog_cache_size
bulk_insert_buffer_size
character_set_client
character_set_connection
character_set_database
character_set_results
character-set-server
character_set_system
character-sets-dir
collation_connection
collation_database
collation-server
concurrent_insert
connect_timeout
datadir
date_format
datetime_format
default_week_format
delay_key_write
delayed_insert_limit
delayed_insert_timeout
delayed_queue_size
expire_logs_days
flush
flush_time
foreign_key_checks
ft_boolean_syntax
ft_max_word_len
ft_min_word_len
ft_query_expansion_limit
ft_stopword_file
group_concat_max_len
have_archive
have_bdb
have_blackhole_engine
have_compress
have_crypt
have_csv
have_example_engine
have_geometry
have_innodb
have_isam
have_merge_engine
have_ndbcluster
have_openssl
have_query_cache
have_raid
have_rtree_keys
have_symlink
identity
init_connect
init_file
init_slave
innodb_additional_mem_pool_size
innodb_autoextend_increment
innodb_buffer_pool_awe_mem_mb
innodb_buffer_pool_size
innodb_data_file_path
innodb_data_home_dir
innodb_fast_shutdown
innodb_file_io_threads
innodb_file_per_table
innodb_flush_log_at_trx_commit
innodb_flush_method
innodb_force_recovery
innodb_lock_wait_timeout
innodb_locks_unsafe_for_binlog
innodb_log_arch_dir
innodb_log_archive
innodb_log_buffer_size
innodb_log_file_size
innodb_log_files_in_group
innodb_log_group_home_dir
innodb_max_dirty_pages_pct
innodb_max_purge_lag
innodb_mirrored_log_groups
innodb_open_file

innodb_table_locks	innodb_thread_concurrency	insert_id
interactive_timeout	join_buffer_size	key_buffer_size
key_cache_age_threshold	key_cache_block_size	key_cache_division_limit
language	large_files_support	last_insert_id
lc_time_names	license	local_infile
log	log_bin	log_error
log_slave_updates	log_slow_queries	log_update
log_warnings	long_query_time	low_priority_updates
lower_case_file_system	lower_case_table_names	max_allowed_packet
max_binlog_cache_size	max_binlog_size	max_connect_errors
max_connections	max_delayed_threads,max_error_count	max_heap_table_size
max_insert_delayed_threads	max_join_size	max_length_for_sort_data
max_prepared_stmt_count	max_relay_log_size	max_seeks_for_key
max_sort_length	max_tmp_tables	max_user_connections
max_write_lock_count	myisam_data_pointer_size	myisam_max_extra_sort_file_size
myisam_max_sort_file_size	myisam_recover_option	myisam_repair_threads
myisam_sort_buffer_size	myisam_stats_method	named_pipe
net_buffer_length	net_read_timeout	net_retry_count
net_write_timeout	new	old_passwords
open_files_limit	pid_file	port
preload_buffer_size	prepared_stmt_count	protocol_version
query_alloc_block_size	query_cache_limit	query_cache_min_res_unit
query_cache_size	query_cache_type	query_cache_wlock_invalidate
query_prealloc_size	range_alloc_block_size	read_buffer_size
read_only	read_rnd_buffer_size	relay_log_purge
relay_log_space_limit	rpl_recovery_rank	secure_auth
shared_memory	shared_memory_base_name	server_id

skip_external_locking
skip_networking
skip_show_database
slave_net_timeout
slave_transaction_retries
slow_launch_time
sort_buffer_size
sql_auto_is_null
sql_big_selects
sql_buffer_result
sql_log_bin
sql_log_off
sql_log_update
sql_mode
sql_notes
sql_quote_show_create
sql_safe_updates
sql_select_limit
sql_warnings
storage_engine
sync_binlog
sync_frm
sync_replication
sync_replication_slave_id
sync_replication_timeout
system_time_zone
table_cache
table_type
thread_cache_size
thread_stack
timestamp
time_format
time_zone
tmp_table_size
tmpdir
transaction_alloc_block_size
transaction_prealloc_size
tx_isolation
unique_checks
version
version_comment
version_compile_machine
version_compile_os
wait_timeout

Supported Oracle Initialization Parameters (DBArtisan)

The following parameters are available from the Explorer's **Configuration** node. For information on working with these parameters, see Working with Configuration/initialization Parameters (DBArtisan).

O7_DICTIONARY_ACCESSIBILITY
active_instance_count
aq_tm_processes
archive_lag_target
asm_diskgroups
asm_diskstring
asm_power_limit
asm_preferred_read_failure_groups
audit_file_dest
audit_sys_operations
audit_trail
background_core_dump
background_dump_dest
backup_tape_io_slaves
bitmap_merge_area_size
blank_trimming
buffer_pool_keep
buffer_pool_recycle
circuits
client_result_cache_lag
client_result_cache_size
cluster_database
cluster_database_instances
cluster_interconnects
commit_logging
commit_point_strength
commit_wait
commit_write
compatible
control_file_record_keep_time
control_files
control_management_pack_access
core_dump_dest
cpu_count
create_bitmap_area_size
create_stored_outlines
cursor_sharing
cursor_space_for_time
db_16k_cache_size
db_2k_cache_size
db_32k_cache_size
db_4k_cache_size
db_8k_cache_size
db_block_buffers
db_block_checking
db_block_checksum
db_block_size
db_cache_advice
db_cache_size
db_create_file_dest
db_create_online_log_dest_1
db_create_online_log_dest_2
db_create_online_log_dest_3
db_create_online_log_dest_4
db_create_online_log_dest_5
db_domain
db_file_multiblock_read_count
db_file_name_convert
db_files
db_flashback_retention_target
db_keep_cache_size
db_lost_write_protect
db_name
db_recovery_file_dest
db_recovery_file_dest_size
db_recycle_cache_size
db_securefile
db_ultra_safe
db_unique_name
db_writer_processes
dbwr_io_slaves
ddl_lock_timeout
dg_broker_config_file1
dg_broker_config_file2
dg_broker_start
diagnostic_dest
disk_asynch_io
dispatchers
distributed_lock_timeout
dml_locks
drs_start

enable_ddl_logging
event
fal_client
fal_server
fast_start_io_target
fast_start_mttr_target
fast_start_parallel_rollback
file_mapping
fileio_network_adapters
filesystemio_options
fixed_date
gc_files_to_locks
gcs_server_processes
global_context_pool_size
global_names
global_txn_processes
hash_area_size
hi_shared_memory_address
hs_autoregister
ifile
instance_groups
instance_name
instance_number
instance_type
java_jit_enabled
java_max_sessionspace_size
java_pool_size
java_soft_sessionspace_limit
job_queue_processes
large_pool_size
ldap_directory_access
ldap_directory_sysauth
license_max_sessions
license_max_users
license_sessions_warning
local_listener
lock_name_space
lock_sga
log_archive_config
log_archive_dest
log_archive_dest_1
log_archive_dest_10
log_archive_dest_2
log_archive_dest_3
log_archive_dest_4
log_archive_dest_5
log_archive_dest_6
log_archive_dest_7
log_archive_dest_8
log_archive_dest_9
log_archive_dest_state_1
log_archive_dest_state_10
log_archive_dest_state_2
log_archive_dest_state_3
log_archive_dest_state_4
log_archive_dest_state_5
log_archive_dest_state_6
log_archive_dest_state_7
log_archive_dest_state_8
log_archive_dest_state_9
log_archive_duplex_dest
log_archive_format
log_archive_local_first
log_archive_max_processes
log_archive_min_succeed_dest
log_archive_start
log_archive_trace
log_buffer
log_checkpoint_interval
log_checkpoint_timeout
log_checkpoints_to_alert
log_file_name_convert
max_commit_propagation_delay
max_dispatchers
max_dump_file_size
max_enabled_roles
max_shared_servers
memory_max_target
memory_target
nls_calendar
nls_comp
nls_currency
nls_date_format
nls_date_language

nls_dual_currency
nls_iso_currency
nls_language
nls_length_semantics
nls_nchar_conv_excp
nls_numeric_characters
nls_sort
nls_territory
nls_time_format
nls_time_tz_format
nls_timestamp_format
nls_timestamp_tz_format
object_cache_max_size_percent
object_cache_optimal_size
olap_page_pool_size
open_cursors
open_links
open_links_per_instance
optimizer_capture_sql_plan_baselines
optimizer_dynamic_sampling
optimizer_features_enable
optimizer_index_caching
optimizer_index_cost_adj
optimizer_mode
optimizer_secure_view_merging
optimizer_use_invisible_indexes
optimizer_use_pending_statistics
optimizer_use_sql_plan_baselines
os_authent_prefix
os_roles
parallel_adaptive_multi_user
parallel_automatic_tuning
parallel_execution_message_size
parallel_instance_group
parallel_io_cap_enabled
parallel_max_servers
parallel_min_percent
parallel_min_servers
parallel_server
parallel_server_instances
parallel_threads_per_cpu
pga_aggregate_target
plscope_settings
plsql_ccflags
plsql_code_type
plsql_debug
plsql_native_library_dir
plsql_native_library_subdir_count
plsql_optimize_level
plsql_v2_compatibility
plsql_warnings
pre_page_sga
processes
query_rewrite_enabled
query_rewrite_integrity
rdbms_server_dn
read_only_open_delayed
recovery_parallelism
recyclebin
redo_transport_user
remote_dependencies_mode
remote_listener
remote_login_passwordfile
remote_os_authent
remote_os_roles
replication_dependency_tracking
resource_limit
resource_manager_cpu_allocation
resource_manager_plan
result_cache_max_result
result_cache_max_size
result_cache_mode
result_cache_remote_expiration
resumable_timeout
rollback_segments
sec_case_sensitive_logon
sec_max_failed_login_attempts
sec_protocol_error_further_action
sec_protocol_error_trace_action
sec_return_server_release_banner
serial_reuse

service_names
session_cached_cursors
session_max_open_files
sessions
sga_max_size
sga_target
shadow_core_dump
shared_memory_address
shared_pool_reserved_size
shared_pool_size
shared_server_sessions
shared_servers
skip_unusable_indexes
smtp_out_server
sort_area_retained_size
sort_area_size
spfile
sql92_security
sql_trace
sql_version
sqltune_category
standby_archive_dest
standby_file_management
star_transformation_enabled
statistics_level
streams_pool_size
tape_asynch_io
thread
timed_os_statistics
timed_statistics
trace_enabled
tracefile_identifier
transactions
transactions_per_rollback_segment
undo_management
undo_retention
undo_tablespace
use_indirect_data_buffers
user_dump_dest
utl_file_dir
workarea_size_policy
xml_db_events

Supported SQL Server Configuration Options (DBArtisan)

The following options are available from the Explorer's **Configuration** node. For information on working with these options, see Working with Configuration/initialization Parameters (DBArtisan).

access check cache bucket count
access check cache quota
Ad Hoc Distributed Queries
affinity I/O mask
affinity mask
Agent XPs
allow updates
awe enabled
backup compression default
blocked process threshold
c2 audit mode
clr enabled
common criteria compliance enabled
cost threshold for parallelism
cross db ownership chaining
cursor threshold
Database Mail XPs 0 0 0 1 Yes
default full-text language
default language
default trace enabled
disallow results from triggers
EKM provider enabled
filestream access level
fill factor (%)
ft crawl bandwidth (max)
ft crawl bandwidth (min)
ft notify bandwidth (max)
ft notify bandwidth (min)
index create memory (KB)
in-doubt xact resolution
lightweight pooling 0 0 0 1 No
locks 0 0 5000 2147483647 No
max degree of parallelism
max full-text crawl range
max server memory (MB)
max text repl size (B)
max worker threads
media retention
min memory per query (KB)
min server memory (MB
nested triggers
network packet size (B)
Ole Automation Procedures
open objects
optimize for ad hoc workloads
PH timeout (s)
precompute rank
priority boost
query governor cost limit
query wait (s)
recovery interval (min)
remote access
remote admin connections
remote login timeout (s)
remote proc trans
remote query timeout (s)
Replication XPs
scan for startup procs
server trigger recursion
set working set size
show advanced options
SMO and DMO XPs
SQL Mail XPs 0 0 0 1
transform noise words 0 0 0 1
two digit year cutoff
user connections
user options
xp_cmdshell

Supported Sybase ASE Configuration Parameters (DBArtisan)

The following parameters are available from the Explorer's **Configuration** node. For information on working with these parameters, see Working with Configuration/initialization Parameters (DBArtisan).

SQL Perfmon Integration
SQL batch capture
abstract plan cache
abstract plan dump
abstract plan load
abstract plan replace
additional network memory
allocate max shared memory
allow backward scans
allow nested triggers
allow procedure grouping
allow remote access
allow resource limits
allow sendmsg
allow sql server async i/o
allow updates to system tables
audit queue size
auditing
average cap size
'caps per ccb
check password for digit
cis bulk insert array size
cis bulk insert batch size
cis connect timeout
cis cursor rows
cis packet size
cis rpc handling
configuration file
cpu accounting flush interval
cpu grace time'
curread change w/ open cursors
current audit table
deadlock checking period
deadlock pipe active
deadlock pipe max messages
deadlock retries
default character set id
default database size
default exp_row_size percent
default fill factor percent
default language id
default network packet size
default sortorder id
default unicode sortorder
default xml sortorder
disable character set conversions
disable disk mirroring
disk i/o structures
dtm detach timeout period
dtm lock timeout period
dump on conditions
dynamic allocation on demand
enable DTM
enable HA
enable cis
enable enterprise java beans
enable file access
enable full-text search
enable housekeeper GC
enable java
enable job scheduler
enable ldap user auth
enable metrics capture
enable monitoring
enable pam user auth
enable real time messaging
enable rep agent threads
enable row level access
enable semantic partitioning
enable snmp
enable sort-merge join and JTC
enable ssl
enable surrogate processing
enable unicode conversions
enable unicode normalization
enable webservices
enable xact coordination
enable xml
errorlog pipe active
errorlog pipe max messages
esp execution priority
esp execution stacksize
esp unload dll
event buffers per engine

event log computer name
event logging
executable codesize + overhead
extended cache size
global async prefetch limit
global cache partition number
heap memory per user
histogram tuning factor
housekeeper free write percent
i/o accounting flush interval
i/o batch size
i/o polling process count
identity burning set factor
identity grab size
job scheduler interval
job scheduler tasks
license information
lock address spinlock ratio
lock hashtable size
lock scheme
lock shared memory
lock spinlock ratio
lock table spinlock ratio
lock wait period
log audit logon failure
log audit logon success
max SQL text monitored
max async i/os per engine
max async i/os per server
max cis remote connections
max concurrently recovered db
max memory
max native threads per engine
max network packet size
max number network listeners
max online engines
max parallel degree
max repartition degree
max resource granularity
max scan parallel degree
maximum dump conditions
maximum failed logins
maximum job output
memory alignment boundary
memory per worker process
messaging memory
minimum password length
msg confidentiality reqd
msg integrity reqd
number of alarms
number of aux scan descriptors
number of ccbs
number of checkpoint tasks
number of devices
number of dtx participants
number of dump threads
number of engines at startup
number of histogram steps
number of index trips
number of java sockets
number of large i/o buffers
number of locks
number of mailboxes
number of messages
number of oam trips
number of open databases
number of open indexes
number of open objects
number of open partitions
number of pre-allocated extents
number of remote connections
number of remote logins
number of remote sites
number of sort buffers
number of user connections
number of worker processes
o/s file descriptors
object lockwait timing
open index hash spinlock ratio
open index spinlock ratio
open object spinlock ratio
optimization goal
optimization timeout limit
page lock promotion HWM

page lock promotion LWM
page lock promotion PCT
page utilization percent
partition groups
partition spinlock ratio
per object statistics active
percent database for history
percent database for output
percent history free
percent output free
performance monitoring option
permission cache entries
plan text pipe active
plan text pipe max messages
print deadlock information
print recovery information
procedure cache size
process wait events
read committed with lock
recovery interval in minutes
remote server pre-read packets
row lock promotion HWM
row lock promotion LWM
row lock promotion PCT
rtm thread idle wait period
runnable process search count
sampling percent
secure default login
select on syscomments.text
send doneinproc tokens
shared memory starting address
size of auto identity column
size of global fixed heap
size of process object heap
size of shared class heap
size of unilib cache
sql server clock tick length
sql text pipe active
sql text pipe max messages
stack guard size
stack size
start mail session
statement cache size
statement pipe active
statement pipe max messages
statement statistics active
strict dtm enforcement
suspend audit when device full
syb_sendmsg port number
sysstatistics flush interval
systemwide password expiration
tape retention in days
tcp no delay
text prefetch size
time slice
total data cache size
total logical memory
total physical memory
txn to pss ratio
unified login required
upgrade version
use security services
user log cache size
user log cache spinlock ratio
wait event timing
xact coordination interval
xp_cmdshell context

Supported Sybase IQ Configuration Parameters (DBArtisan)

The following parameters are available from the Explorer's **Configuration** node. For information on working with these parameters, see Working with Configuration/initialization Parameters (DBArtisan).

Alias
AuditingTypes
BlankPadding
BlobArenas
CacheHits
CacheRead
CacheReadIndInt
CacheReadIndLeaf
CacheReadTable
Capabilities
CaseSensitive
CaseSensitivePasswords
CharSet
CheckpointLogBitmapPagesWritten
CheckpointLogBitmapSize
CheckpointLogCommitToDisk
CheckpointLogPageInUse
CheckpointLogPagesRelocated
CheckpointLogPagesWritten
CheckpointLogSavePreimage
CheckpointLogSize
CheckpointLogWrites
CheckpointUrgency
[Checksum]
Chkpt
ChkptFlush
ChkptPage
ClusteredIndexes
Collation
CommitFile
CompressedBTrees
Compression
ConnCount
ConnsDisabled
CurrentRedoPos
CurrIO
CurrRead
CurrWrite
DBFileFragments
DiskRead
DiskReadIndInt
DiskReadIndLeaf
DiskReadTable
DiskWrite
DriveType
Encryption
ExprCacheAbandons
ExprCacheDropsToReadOnly
ExprCacheEvicts
ExprCacheHits
ExprCacheInserts
ExprCacheLookups
ExprCacheResumesOfReadWrite
ExprCacheStarts
ExtendDB
ExtendTempWrite
File
FileSize
FileVersion
FreePageBitMaps
FreePages
FullCompare
GetData
GlobalDBId
HashForcedPartitions
HashRowsFiltered
HashRowsPartitioned
HashWorkTables
HistogramHashFix
Histograms
IdleCheck
IdleChkpt
IdleChkTime
IdleWrite
IndAdd
IndexStatistics
IndLookup
IOToRecover
IQStore
JavaHeapSize
JavaNSSize
JDKVersion
Language
LargeProcedureIDs

LockTablePages
LogFileFragments
LogFreeCommit
LogName
LogWrite
LTMGeneration
LTMTrunc
MapPages
MaxIO
MaxRead
MaxWrite
MultiByteCharSet
Name
NamedConstraints
PageRelocations
PageSize
PreserveSource
ProcedurePages
ProcedureProfiling
QueryBypassed
QueryCachedPlans
QueryCachePages
QueryJHToJNLOptUsed
QueryLowMemoryStrategy
QueryOptimized
QueryReused
QueryRowsBufferFetch
QueryRowsMaterialized
[ReadOnly]
RecoveryUrgency
RecursiveIterations
RecursiveIterationsHash
RecursiveIterationsNested
RecursiveJNLMisses
RecursiveJNLProbes
RelocatableHeapPages
RemoteTrunc
RollbackLogPages
SeparateCheckpointLog
SeparateForeignKeys
SortMergePasses
SortRowsMaterialized
SortRunsWritten
SortSortedRuns
SortWorkTables
StringHistogramsFix
SyncTrunc
TableBitMaps
TablesQualTriggers
TempFileName
TempTablePages
TransactionsSpanLogs
TriggerPages
[UniqueIdentifier]
VariableHashSize
ViewPages

Data Pump Import/Export Parameters that can be Added Manually (DBArtisan)

Pages of the Data Pump Import/Export Wizards let you provide a minimal set of **impdp.exe** and **expdp.exe** parameter values. The **Advanced Parameters** tabs let you provide additional *name***=***value* pairs. The following table lists the parameters available on that page.

DATA_OPTIONS	ENCRYPTION_PASSWORD	ESTIMATE
EXCLUDE	FLASHBACK_SCN	FLASHBACK_TIME
HELP	INCLUDE	NETWORK_LINK
PARALLEL	PARFILE	PARTITION_OPTIONS
QUERY	REMAP_DATA	REMAP_DATAFILE
REMAP_TABLE	SOURCE_EDITION	SQLFILE
STATUS	STREAMS_CONFIGURATION	TARGET_EDITION
TRANSFORM_DATAFILES	TRANSFORM_FULL_CHECK	TRANSFORM_TABLESPACES
TRANSPORTABLE	VERSION	

For context information, see Data Pump Export (DBArtisan - ORCL10g ^) and Data Pump Import Wizard (DBArtisan - ORCL10g ^).