



*What's New in Sybase Adaptive Server Enterprise?*

**Adaptive Server Enterprise**

*Version 12*

**Document ID:** 37429-01-1200-01

**Last revised:** October 1999

Copyright © 1989-1999 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase database management software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase, the Sybase logo, ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Adaptive Server IQ, Adaptive Warehouse, AnswerBase, Anywhere Studio, Application Manager, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-FORMS, APT-Translator, APT-Library, Backup Server, ClearConnect, Client-Library, Client Services, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DB-Library, dbQueue, Developers Workbench, Direct Connect Anywhere, DirectConnect, Distribution Director, E-Anywhere, E-Whatever, Embedded SQL, EMS, Enterprise Application Server, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, EWA, Gateway Manager, ImpactNow, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InternetBuilder, iScript, Jaguar CTS, jConnect for JDBC, KnowledgeBase, MainframeConnect, Maintenance Express, MAP, MDI Access Server, MDI Database Gateway, media.splash, MetaWorks, MySupport, Net-Gateway, Net-Library, NetImpact, ObjectConnect, ObjectCycle, OmniConnect, OmniSQL Access Module, OmniSQL Toolkit, Open Client, Open ClientConnect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, PB-Gen, PC APT Execute, PC DB-Net, PC Net Library, Power++, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, PowerJ, PowerScript, PowerSite, PowerSocket, Powersoft, PowerStage, PowerStudio, PowerTips, Powersoft Portfolio, Powersoft Professional, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, Report Workbench, Report-Execute, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Resource Manager, RW-DisplayLib, RW-Library, S Designer, S-Designer, SDF, Secure SQL Server, Secure SQL Toolset, Security Guardian, SKILS, smart.partners, smart.parts, smart.script, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL SMART, SQL Toolset, SQL Server/CFT, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQLJ, STEP, SupportNow, Sybase Central, Sybase Client/Server Interfaces, Sybase Financial Server, Sybase Gateways, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase User Workbench, SybaseWare, Syber Financial, SyberAssist, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, Transact-SQL, Translation Toolkit, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, Viewer, Visual Components, VisualSpeller, VisualWriter, VQL, WarehouseArchitect, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server and XP Server are trademarks of Sybase, Inc. 9/99

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., 6475 Christie Avenue, Emeryville, CA 94608.

# Contents

<b>CHAPTER 1</b>	<b>New Features in Adaptive Server Version 12</b> .....	<b>1</b>
	New Features in Version 12.....	2
	Sybase's Failover for A High Availability System.....	3
	Rebuilding Indexes.....	5
	New Backup Server Features.....	6
	High Performance Backup and Restore.....	6
	Non-Rewinding Tapes.....	6
	UNIX File System Support.....	7
	Modifying an Existing Table's Schema with alter table.....	8
	Suspending Database Updates with quiesce database.....	9
	Taking Adaptive Server Engines Offline.....	10
	User Settable Process Information.....	11
	Identity Number Gap for Tables.....	12
	Diagnostic Database.....	13
	Java in Adaptive Server Enterprise.....	14
	Capabilities of Java in Adaptive Server.....	14
	Number of Tables in a Query.....	16
	Query Processing and Optimization Enhancements.....	17
	Merge Joins.....	17
	Query Costing Improvements.....	17
	ANSI Joins.....	18
	Dynamic Execution of Transact-SQL.....	19
	text and image Datatype Enhancements.....	20
	Abstract Plans.....	21
	Disabling Triggers.....	22
	Cache Partitions.....	23
	Distributed Transaction Management.....	24
	Enhancements to Component Integration Services (CIS).....	25
	Network Security.....	26
	Performance Considerations.....	26
	Security Features Supported in Adaptive Server 12.....	26
	User-Defined Login Security.....	28
	Increased Maximum Number of Users and Logins.....	28
	Setting and Changing the Maximum Login Attempts.....	29
	Locking and Unlocking Logins and Roles.....	29

	Displaying Password Information .....	30
	Checking Passwords for At Least One Character .....	30
	Setting and Changing Minimum Password Length .....	30
	Setting the Expiration Interval for a Password .....	31
	Concrete Identification .....	32
<b>CHAPTER 2</b>	<b>System Changes in Adaptive Server Version 12 .....</b>	<b>33</b>
	System Changes in Version 12 .....	34
	Changes to Configuration Parameters .....	34
	Changes to Transact-SQL Commands .....	35
	New and Changed System Procedures .....	37
	New Utility Programs .....	39
	Changes to Databases and System Tables .....	39
	New Reserved Words .....	41
	Changes to the Documentation .....	42
	Changes That May Affect Existing Applications .....	43
	New Transact-SQL Keywords for 12 .....	43
	dsync On By Default for UNIX File Devices .....	43
<b>CHAPTER 3</b>	<b>New Features in Adaptive Server Version 11.9.2 .....</b>	<b>45</b>
	New Locking Schemes .....	46
	Changes to Statistics and Query Optimization .....	47
	Enhancements to the create index Command .....	48
	Changes and Additions to Transact-SQL Syntax .....	49
	New readpast Concurrency Option .....	49
	New lock table Command .....	49
	Specifying a Wait Time for Locks .....	49
	Repeatable Read Transaction Isolation .....	50
	Configurable Database Recovery Order .....	51
	Fault Verification for dbcc checkstorage Faults .....	52
	License Use Monitor .....	53
	Task-to-Engine Affinity .....	54
	Dynamic SQL Performance Improvements .....	55
	Direct Updates Through Joins .....	56
	Component Integration Services Changes .....	57
	Character Set Changes .....	58
	Changes That May Affect Existing Applications .....	59
	Effects of Changing to Data-Only-Locking .....	59
	Query Optimization Changes and Forced Query Plans .....	60
	Ordering of Results with Data-Only Locked Tables .....	61
	Delay of Cursor Compilation Until Cursor Open .....	62

---

<b>CHAPTER 4</b>	<b>New Features in Adaptive Server Release 11.5 .....</b>	<b>65</b>
	New Features in Release 11.5 .....	66
	Asynchronous Prefetch .....	67
	Auditing Enhancements .....	69
	New System Procedures .....	69
	case Expression .....	72
	Component Integration Services .....	73
	New Commands .....	74
	New System Procedures .....	74
	Modified System Procedures .....	75
	create index Enhancements .....	76
	dbcc Enhancements .....	77
	Advantages of Using dbcc checkstorage .....	77
	System and Stored Procedures for Creating dbccdb .....	77
	Stored Procedures for Maintaining dbccdb .....	78
	Stored Procedures for Generating Reports on dbccdb .....	78
	Descending Index Scan Optimization .....	80
	Directory Services .....	81
	Engine Affinity and Execution Precedence .....	82
	New System Procedures .....	82
	Expanded Error Logging .....	84
	New System Procedures .....	84
	Modified System Procedures .....	84
	Extended Stored Procedures (ESPs) .....	86
	XP Server .....	86
	System-Defined ESPs .....	87
	New System Procedures for ESPs .....	87
	Supporting Commands .....	88
	Increased Referential Integrity Limits .....	89
	Metadata Cache Management .....	90
	Monitor Access to SQL Batch Text .....	91
	Allocating Memory for Batch Text .....	91
	New Columns in sysprocesses .....	91
	Network-Based Security .....	93
	Security Services .....	93
	Parallel Bulk Copy .....	95
	Parallel Queries and Enhanced Partitioning .....	96
	Point-in-Time Recovery .....	97
	Proxy Authorization .....	98
	Recovery Fault Isolation .....	99
	New System Procedures .....	99
	Side Effects of Recovery Fault Isolation .....	100
	Relaxed LRU Cache Replacement Policy .....	101
	Resource Limits .....	102

New System Procedures.....	102
Supporting Commands.....	103
Source Text Verification and Encryption .....	104
Sybase Central.....	105
sp_sysmon .....	106
Two-Phase Commit Enhancements.....	107
spt_comittab Moved to a New System Database (sybssystemdb)	
107	
Commit Identifier Changed to a Random Value.....	107
User-Defined Roles.....	109
Windows NT Event Log Integration.....	110
Supporting System ESPs and System Procedures.....	110
Windows NT MAPI Support .....	111
Supporting System ESPs and System Procedures.....	111
Windows NT Performance Monitor Integration .....	112
Supporting System Procedure .....	112

**CHAPTER 5**

<b>New Features in SQL Server Release 11.0.....</b>	<b>113</b>
New Features in Release 11.0.....	114
User-Defined Caches.....	114
Data Storage Changes.....	117
Transaction Log Changes .....	120
Isolation Level 0 .....	121
Lock Manager Changes .....	122
Housekeeper Task .....	124
SQL Server Configuration .....	125
Lock Escalation .....	125
Multiple Network Engines.....	126
Improvements to showplan .....	126
Query and Data Modification Changes .....	127
Upgrading Database Dumps .....	128
Tape Device Determination by Backup Server .....	129
IDENTITY Column Changes .....	129
New text and image Global Variables .....	130
System Changes in Release 11.0.....	131
New online database Command .....	131
Changes to Existing Commands .....	131
New set options.....	132
New System Procedures.....	132
Changes to System Procedures.....	133
New System Tables .....	134
Changes to Existing System Tables.....	134
Changes That May Affect Existing Applications.....	136
New Transact-SQL Keywords in Release 11.0.....	136

Changes to SQL Server Configuration .....	136
Subquery Changes.....	140
Changes to showplan Output in Release 11.0 .....	144
New Caching Strategies May Affect Performance .....	145
Upgrading Database Dumps .....	145
Partitions and Physical Data Placement .....	146





# About This Book

<b>Audience</b>	This manual is intended for customers who are upgrading to Adaptive Server™ version 12.
<b>How to use this book</b>	<p><i>What's New in Adaptive Server Enterprise?</i> is an introduction to the new Sybase® Adaptive Server Enterprise features and the commands, system procedures, system tables, and documentation that support them.</p> <p>This document also covers the new features added and system changes made in releases 11.5 and 11.9.2. This is provided for users who are upgrading to Adaptive Server version 12 from earlier releases.</p> <p>This manual consists of the following chapters:</p> <ul style="list-style-type: none"><li>• Chapter 1, “New Features in Adaptive Server Version 12” describes the new features added to Adaptive Server version 12.</li><li>• Chapter 2, “System Changes in Adaptive Server Version 12” describes the Adaptive Server system changes made to support the new version 12 features. It also discusses how these additions and changes might affect your existing applications.</li><li>• Chapter 3, “New Features in Adaptive Server Version 11.9.2” describes the features added and system changes made in version 11.9.2.</li><li>• Chapter 4, “New Features in Adaptive Server Release 11.5” describes the features added and system changes made in release 11.5. Also discussed is how these additions and changes might affect your existing applications.</li><li>• Chapter 5, “New Features in SQL Server Release 11.0” describes the features added and system changes made in release 11.0. Also discussed is how these additions and changes might affect your existing applications.</li></ul>
<b>Related documents</b>	<p>The following documents comprise the Sybase Adaptive Server Enterprise documentation:</p> <ul style="list-style-type: none"><li>• The <i>Release Bulletin</i> for your platform – contains last-minute information that was too late to be included in the books.</li></ul>

---

A more recent version of the Release Bulletin may be available on the World Wide Web. To check for critical product or document information that was added after the release of the product CD, use the Sybase Technical Library.

- The Adaptive Server installation documentation for your platform – describes installation, upgrade, and configuration procedures for all Adaptive Server and related Sybase products.
- *What's New in Adaptive Server Enterprise?* – describes the new features in Adaptive Server version 12, the system changes added to support those features, and the changes that may affect your existing applications.
- *Transact-SQL User's Guide* – documents Transact-SQL, Sybase's enhanced version of the relational database language. This manual serves as a textbook for beginning users of the database management system. This manual also contains descriptions of the *pubs2* and *pubs3* sample databases.
- *System Administration Guide* – provides in-depth information about administering servers and databases. This manual includes instructions and guidelines for managing physical resources, security, user and system databases, and specifying character conversion, international language, and sort order settings.
- *Adaptive Server Reference Manual* – contains detailed information about all Transact-SQL commands, functions, procedures, and datatypes. This manual also contains a list of the Transact-SQL reserved words and definitions of system tables.
- *Performance and Tuning Guide* – explains how to tune Adaptive Server for maximum performance. This manual includes information about database design issues that affect performance, query optimization, how to tune Adaptive Server for very large databases, disk and cache issues, and the effects of locking and cursors on performance.
- The *Utility Programs* manual for your platform – documents the Adaptive Server utility programs, such as **isql** and **bcp**, which are executed at the operating system level.
- *Error Messages and Troubleshooting Guide* – explains how to resolve frequently occurring error messages and describes solutions to system problems frequently encountered by users.
- *Component Integration Services User's Guide* – explains how to use the Adaptive Server Component Integration Services feature to connect remote Sybase and non-Sybase databases.

- *Java in Adaptive Server Enterprise* – describes how to install and use Java classes as datatypes and user-defined functions in the Adaptive Server database.
- *Using Sybase Failover in a High Availability System* – provides instructions for using Sybase's Failover to configure an Adaptive Server as a companion server in a high availability system.
- *Using Adaptive Server Distributed Transaction Management Features* – explains how to configure, use, and troubleshoot Adaptive Server DTM Features in distributed transaction processing environments.
- *XA Interface Integration Guide for CICS, Encina, and TUXEDO* – provides instructions for using Sybase's DTM XA Interface with X/Open XA transaction managers.
- *Adaptive Server Glossary* – defines technical terms used in the Adaptive Server documentation.

**Other sources of information**

Use the Sybase Technical Library CD and the Technical Library Product Manuals web site to learn more about your product:

- Technical Library CD contains product manuals and technical documents and is included with your software. The DynaText browser (included on the Technical Library CD) allows you to access technical information about your product in an easy-to-use format.

Refer to the *Technical Library Installation Guide* in your documentation package for instructions on installing and starting Technical Library.

- Technical Library Product Manuals web site is an HTML version of the Technical Library CD that you can access using a standard web browser. In addition to product manuals, you'll find links to the Technical Documents web site (formerly known as Tech Info Library), the Solved Cases page, and Sybase/Powersoft newsgroups.

To access the Technical Library Product Manuals web site, go to Product Manuals at <http://sybooks.sybase.com>.

**Sybase certifications on the web**

Technical documentation at the Sybase web site is updated frequently.

v **For the latest information on product certifications and/or the EBF Rollups:**

- 1 Point your web browser to Technical Documents at <http://techinfo.sybase.com>.
- 2 In the Browse section, click on What's Hot.

- 
- 3 Select links to Certification Reports and EBF Rollups, as well as links to Technical Newsletters, online manuals, and so on.
- v **If you are a registered SupportPlus user:**
- 1 Point your web browser to Technical Documents at <http://techinfo.sybase.com>.
  - 2 In the Browse section, click on What's Hot.
  - 3 Click on EBF Rollups.  
You can research EBFs using Technical Documents, and you can download EBFs using Electronic Software Distribution (ESD).
  - 4 Follow the instructions associated with the SupportPlus<sup>SM</sup> Online Services entries.
- v **If you are not a registered SupportPlus user, and you want to become one:**
- You can register by following the instructions on the Web.
- To use SupportPlus, you need:
- 1 A Web browser that supports the Secure Sockets Layer (SSL), such as Netscape Navigator 1.2 or later
  - 2 An active support license
  - 3 A named technical support contact
  - 4 Your user ID and password
- v **Whether or not you are a registered SupportPlus user:**
- You may use Sybase's Technical Documents. Certification Reports are among the features documented at this site.
- 1 Point your web browser to Technical Documents at <http://techinfo.sybase.com>
  - 2 In the Browse section, click on What's Hot.
  - 3 Click on the topic that interests you.

## Conventions

The following style conventions are used in this manual:

- In a sample screen display, commands you should enter exactly as shown are given in:

`this font`

- In a sample screen display, words which you should replace with the appropriate value for your installation are shown in:

*this font*

- In the regular text of this document, the names of files and directories appear in italics:

*/usr/w/sybase*

- The names of programs, utilities, procedures, and commands appear in bold type:

**bcp**

- Commands for both the C and Bourne shells are provided in this document when they differ. The initialization file for the C shell is called *.cshrc*. The initialization file for the Bourne shell is called *.profile*. If you are using a different shell, such as the Korn shell, refer to your shell-specific documentation for the correct command syntax.

The conventions for syntax statements in this manual are as follows:

**Table 1: SQL syntax conventions**

Key	Definition
command	Command names, command option names, utility names, utility flags, and other keywords are in bold.
<i>variable</i>	Variables, or words that stand for values that you fill in, are in <i>italics</i> .
{ }	Curly braces indicate that you choose at least one of the enclosed options. Do not include braces in your option.
[ ]	Brackets mean choosing one or more of the enclosed options is optional. Do not include brackets in your option.
( )	Parentheses are to be typed as part of the command.
	The vertical bar means you can select only one of the options shown.
,	The comma means you can choose as many of the options shown as you like, separating your choices with commas to be typed as part of the command.

### If you need help

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.



## New Features in Adaptive Server Version 12

This chapter describes the new features and system changes introduced with Sybase Adaptive Server Enterprise version 12.

In addition to improving overall manageability and performance, Adaptive Server version 12 introduces several new licensed features for specialized use. These include:

- Java in Adaptive Server Enterprise
- Sybase Failover for High Availability Environments
- Distributed Transaction Management
- Advanced Security

Before using any of these features, you must purchase the feature's license and configure the feature using the SySAM asset management system (see your *Installation Guide* for detailed instructions). To purchase one or more licensed features, contact your sales representative.

## New Features in Version 12

The new features in Adaptive Server version 12 are:

<b>Category</b>	<b>Name</b>
Availability and Manageability	“Rebuilding Indexes” on page 5
	“High Performance Backup and Restore” on page 6
	“Non-Rewinding Tapes” on page 6
	“UNIX File System Support” on page 7
	“Modifying an Existing Table’s Schema with alter table” on page 8
	“Suspending Database Updates with quiesce database” on page 9
	“Taking Adaptive Server Engines Offline” on page 10
	“User Settable Process Information” on page 11
	“Identity Number Gap for Tables” on page 12
	“Diagnostic Database” on page 13
Performance and Productivity	“Java in Adaptive Server Enterprise” on page 14
	“Number of Tables in a Query” on page 16
	“Query Processing and Optimization Enhancements” on page 17
	“ANSI Joins” on page 18
	“Dynamic Execution of Transact-SQL” on page 19
	“text and image Datatype Enhancements” on page 20
	“Abstract Plans” on page 21
	“Disabling Triggers” on page 22
Distributed Processing	“Cache Partitions” on page 23
	“Distributed Transaction Management” on page 24
	“Enhancements to Component Integration Services (CIS)” on page 25
Security	“Network Security” on page 26
	“User-Defined Login Security” on page 28
	“Concrete Identification” on page 32



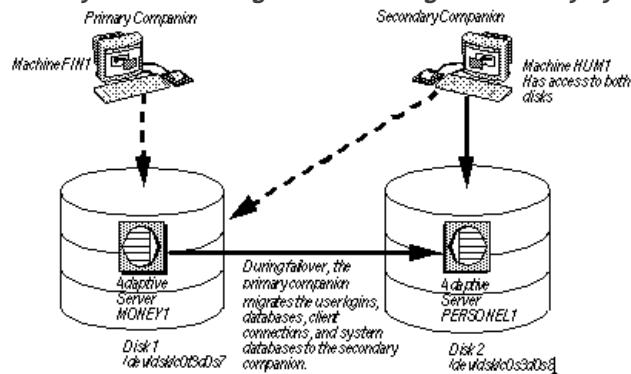
## Sybase's Failover for A High Availability System

A high availability cluster includes two machines that are configured so that, if one machine (or application) is brought down, the second machine assumes the workload of both machines. Each of these machines is called one **node** of the high availability cluster. A high availability cluster is typically used in an environment that must always be available, for example, a banking system to which clients must connect continuously, 365 days a year.

Sybase's Failover product enables Adaptive Server to work in a high availability cluster in an active-active configuration. That is, both the node in the cluster include Adaptive Servers managing independent workloads, and are capable of taking over each other's workload in the event of a failure. The Adaptive Server that takes over the workload is called a **secondary companion**, and the Adaptive Server that fails is called the **primary companion**. Together they are **companion servers**. This movement from one node to another is called **failover**. After the primary companion is ready to resume its workload, it is moved back to its original node. This movement is called **failback**. Clients connected to the failed Adaptive Server automatically reestablish their network connections via the second machine.

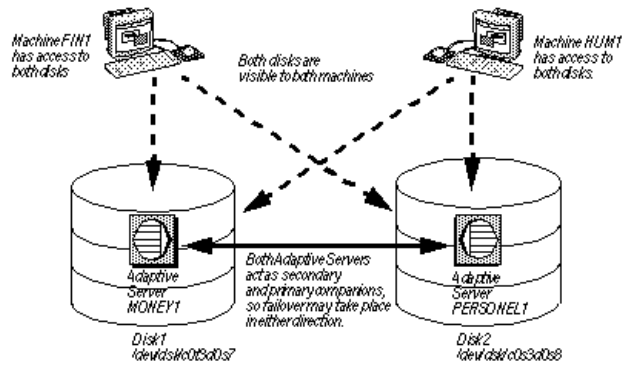
You can configure Sybase's Failover for either an *asymmetric* or a *symmetric* setup. In an asymmetric setup the primary Adaptive Server performs the work during day-to-day operations, while the secondary Adaptive Server is prepared to take over the work during a system failure or a scheduled maintenance. The secondary companion is an independent Adaptive Server, and can have its own applications running. Figure 1-1 describes an asymmetric setup:

**Figure 1-1: Asymmetric configuration in a high availability system**



A symmetric setup consists of two fully functional Adaptive Servers running on separate machines, with their own system devices, system databases, user databases, and user logins. However, in the case of failover, either of the Adaptive Servers acts as a primary or secondary companion for the other Adaptive Server. Figure 1-2 describes a symmetric setup:

**Figure 1-2: Symmetric configuration in a high availability system**



For information about running Sybase's Failover in a high availability system, see *Using Sybase Failover in A High Availability System*.

## Rebuilding Indexes

In Adaptive Server 12, you can use the **reorg rebuild** command to rebuild individual indexes while the table itself is accessible for read and update activities. For details, see the *System Administration Guide* and the **reorg** description in the *Adaptive Server Reference Manual*.

## New Backup Server Features

The following features are new for Backup server:

- High Performance Backup and Restore
- Non-Rewinding Tapes

### High Performance Backup and Restore

When you start Backup Server with the **backupserver** utility command, you can use the **-m** parameter to improve the performance of the dump and load commands by configuring more shared memory for the Backup Server. The **-m** parameter specifies the maximum amount of shared memory used by the Backup Server. You must also configure your operating system to ensure that this amount of shared memory is available to the Backup Server. After a dump or load operation is completed, its shared memory segments are released.

---

**Note** Configuring more shared memory improves dump and load performance only if the performance limits of the hardware setup have not been reached. Increasing the value of **-m** may not result in improved performance when dumping to a slow tape device such as QIC, but it can improve performance significantly when dumping to a faster device such as DLT.

---

For details about using Backup Server, see the Utility Programs manual for your platform and the *System Administration Guide*.

### Non-Rewinding Tapes

In previous versions of Adaptive Server, Backup Server rewound the tape before each dump operation. In version 12, the tape is positioned at the end of valid dump data, which saves time when you want to perform multiple dump operations. For details, see the *System Administration Guide*.

## UNIX File System Support

With previous versions of Adaptive Server, Sybase recommended not placing database devices on UNIX operating system files. The integrity of data on such devices could not be ensured, because writes to the file were buffered by the UNIX file system. Adaptive Server could not know when an update was reflected on the physical media, and therefore could not recover databases on the device under certain system failure scenarios. For this reason, UNIX raw partitions were required to ensure the integrity of database devices.

Adaptive Server version 12 introduces a new **dsync** setting for database device files, which controls whether or not writes to those files are buffered. When the **dsync** setting is on, Adaptive Server opens a database device file using the UNIX **dsync** flag. The **dsync** flag ensures that writes to the device file occur directly on the physical storage media, and Adaptive Server can recover data on the device in the event of a system failure.

For more information about the **dsync** flag, see the *Reference Manual* entries for **disk init**, **disk reinit**, **sp\_deviceattr**, or **sp\_helpdevice**.

## Modifying an Existing Table's Schema with *alter table*

Adaptive Server 12 includes new syntax for **alter table** that allows you to add, drop, or modify a table's columns. **alter table...modify** allows you to modify a column's datatype, length, locking scheme, or default value. **alter table** also allows you to add, drop, or modify an IDENTITY column and to add a **null** or **non-null** column.

**alter table** includes the following syntax for modifying a table's schema:

```
alter table table_name
[add column_name datatype [identity | null |
    not null] [, column_name datatype [identity |null | not null]]]
[drop column_name [, column_name]
    [modify column_name {[data_type]
    [[null] | [not null]]}
    [, column_name datatype [null | not null]]]
```

Where *table\_name* is the table you are altering, and *datatype* is the datatype of the altered column. The following changes the datatype of the *type* column in the *titles* table from char(12) to varchar(20) and makes it nullable:

```
alter table titles
    modify type varchar(20) null
```

## Suspending Database Updates with *quiesce database*

Adaptive Server version 12 provides the new **quiesce database** command to temporarily suspend updates to databases. You may want to suspend database updates to use an external database backup utility, or to separate a database from its mirror for reporting purposes.

For more information about this feature, see the **quiesce database** pages in the *Reference Manual*.

## Taking Adaptive Server Engines Offline

You can dynamically change the number of engines in use by Adaptive Server with the **dbcc engine** command to take an engine offline or bring an engine online. This allows a System Administrator to reconfigure CPU resources as processing requirements fluctuate over time.

For information on using **dbcc engine (offline)**, see Chapter 16, “Managing Multiprocessor Servers,” in the *System Administration Guide*.



## User Settable Process Information

The **set** command includes options that allow you to assign each client an individual name, host name, and application name. This is useful for differentiating among clients in a system where many clients connect to Adaptive Server using the same name, host name, or application name. The partial syntax for the **set** command is:

```
set [clientname client_name | clienthostname host_name |  
clientapplname application_name]
```

Where *client\_name* is the name you are assigning the client, *host\_name* is the name of the host from which the client is connecting, and *application\_name* is the application that is connecting to Adaptive Server. These parameters are stored in the *clientname*, *clienthostname*, *clientapplname* columns of the *sysprocesses* table.

For more information, see the “Managing Adaptive Server Logins and Database Users” chapter of the *System Administration Guide*.

## Identity Number Gap for Tables

The *identity* column contains a unique ID number, generated by Adaptive Server, for each row in a table. Because of the way the server generates ID numbers by default, you may have large gaps in the ID numbers. The **identity\_gap** parameter gives you control over ID numbers, and potential gaps in them, for a specific table.

This parameter can be used when creating a table with **create table** or **select into**. To change the identity gap, use **identity\_gap** with **sp\_chgattribute**. To see the identity gap setting for a table, use **sp\_help**.

For details about using **identity\_gap**, see the *System Administration Guide* and the command and system procedure descriptions in the *Adaptive Server Reference Manual*.

## Diagnostic Database

Sybase Technical Support may create the *sybdiagdb* database on your system for debugging purposes. This database holds diagnostic configuration data. It should not be used by customers.

The following **configname** options have been added to **sp\_helpconfig** for use with the *sybdiagdb* database:

- *number of ccbs* is the number of configurable action point control blocks available to aid debugging.
- *caps per ccb* is the maximum number of configurable action points that can be configured at any one time within one configurable action point.
- *average cap size* is the estimated number of bytes of memory required to store the information associated with a typical configurable action point.

---

**Note** The new options can be used by Sybase Technical Support, for diagnostic purposes only, with the *sybdiagdb* database.

---

## Java in Adaptive Server Enterprise

Adaptive Server version 12 provides a runtime environment for Java, which means that you can execute Java code in the server. Adaptive Server's Java capabilities provide you with powerful new ways of managing and storing both data and logic, using a language that is both portable and widely available.

- You can use the Java programming language as an integral part of Transact-SQL.
- You can reuse Java code in the different layers of your application—client, middle-tier, or server—and use them wherever makes most sense to you.
- Java provides a more powerful language than stored procedures for building logic into the database.
- Java classes become rich, user-defined datatypes.
- Methods of Java classes provide new functions accessible from SQL.
- Java can be used in the database without jeopardizing the integrity, security, and robustness of the database. Using Java does not alter the behavior of existing SQL statements or other aspects of non-Java relational database behavior.

### Capabilities of Java in Adaptive Server

Java in Adaptive Server provides you with:

- Java user-defined functions (UDFs)
- Java classes as datatypes in SQL

### Java User-Defined Functions

You can install Java classes in the Adaptive Server database, and then invoke the methods of those classes, both from within the SQL system and from client systems.

The methods of an object-oriented language correspond to the functions of a procedural language. You can invoke Java methods as UDFs in, for example, **select** lists and **where** clauses. You can use methods from other sources as well as methods you have created and tested.

## Java Classes as Datatypes

You can install pure Java classes in a SQL system, and then use those classes in a natural manner as datatypes in SQL. This capability adds a full object-oriented datatype extension mechanism to SQL. The objects that you create and store with this facility are readily transferable to any Java-enabled environment, either in another SQL system or stand-alone Java environment.

This capability of using Java classes in the database has two different but complementary uses:

- It provides a type extension mechanism for SQL, which you can use for data that is created and processed in SQL.
- It provides a persistent data capability for Java, which you can use to store data in SQL that is created and processed (mainly) in Java. Java in Adaptive Server provides a distinct advantage over traditional SQL facilities: You do not need to map Java objects into scalar SQL data types or store Java objects as untyped binary strings.

## Number of Tables in a Query

The number of user tables and worktables that can be used for a query has been increased from earlier versions of Adaptive Server. In previous versions, a query and any views used in a query could reference at most 16 user tables, and the number of worktables that could be created was 12. The limits are now 50 user tables and 14 worktables. Limits on the number of referential integrity checks allowed (192) and the number of subqueries allowed in a query (16) have not changed.

You may need to reset the configuration parameter number of aux scan descriptors if you increase the number of tables referenced in queries.

For information on how queries are optimized when large numbers of tables are included, see Chapter 7, “Access Methods and Query Costing for Joins and Subqueries,” in the *Performance and Tuning Guide*.

For information on resetting scan descriptors, see number of aux scan descriptors in Chapter 17, “Setting Configuration Parameters,” in the *System Administration Guide*

## Query Processing and Optimization Enhancements

Adaptive Server version 12 introduces several query processing enhancements.

### Merge Joins

Earlier versions of Adaptive Server perform all join queries using nested-loop joins. Version 12 adds merge joins as a query execution method for equijoins. Merge joins can produce dramatic improvements in query execution times, especially for joins involving several very large tables. By default, merge joins are not enabled. They can be enabled server-wide or for an individual session.

For information on configuring merge joins, see **enable sort-merge joins and JTC** in Chapter 17, “Setting Configuration Parameters,” in the *System Administration Guide*. For information on merge joins and query processing, see Chapter 7, “Access Methods and Query Costing for Joins and Subqueries,” in the *Performance and Tuning Guide*.

### Query Costing Improvements

The following changes help improve the accuracy of optimizer costing and provide additional paths for query execution:

- Transitive closure can be applied to joins
- Predicate factoring and transformation can improve costing for queries using **or**
- Special like optimization for leading wildcards in strings

These features are described in Chapter 5, “Understanding the Query Optimizer,” in the *Performance and Tuning Guide*.

## ANSI Joins

Previous versions of Adaptive Server only provided syntax for a Transact-SQL join, which included the `=*` and `*=` symbols for specifying an right or a left join, respectively. Adaptive Server 12 includes the ANSI syntax for joining tables or views, which allows you to write either:

- Inner joins, in which the joined table includes only the rows of the inner and outer tables that meet the conditions of the **on** clause. The result set of a query that includes an inner join does not include any null supplied rows for the rows of the outer table that do not meet the conditions of the **on** clause. The syntax for an ANSI inner join is:

```
select select_list
from table1 inner join table2
on join_condition
```

For example:

```
select au_id, titles.title_id, title, price
from titleauthor inner join titles
on titleauthor.title_id = titles.title_id
and price > 15
```

- *Outer joins*, in which the joined table includes all the rows from the outer table whether or not they meet the conditions of the **on** clause. If a row does not meet the conditions of the **on** clause, values from the inner table are stored in the joined table as null values. The **where** clause of an ANSI outer join restricts the rows that are included in the query result. ANSI syntax also allows you to write nested outer joins. The syntax for an ANSI outer join is:

```
select select_list
from table1 {left | right} [outer] join
table2
on predicate
[join restriction]
```

For example:

```
select au_fname, au_lname, pub_name
from authors left join publishers
on authors.city = publishers.city
```

Sybase recommends that applications use ANSI outer joins because they unambiguously specify whether the **on** or **where** clause contains the predicate, which is ambiguous when you use Transact-SQL outer joins.

For more information see *The Transact-SQL User's Guide*.



## Dynamic Execution of Transact-SQL

Adaptive Server version 12 provides an extension to the **execute** command that enables users to define Transact-SQL commands dynamically at execution time. Applications and procedures can use the new **execute()** syntax in cases where table names, column names, and so forth are not known until the application or procedure executes.

For more information about this feature, see the **execute** pages in the *Reference Manual*.

## ***text and image* Datatype Enhancements**

Adaptive Server version 12 improves the storage format for text and image datatypes. Changes to the storage format are transparent to end-user applications, but they enable Adaptive Server to perform random access when querying data. This improves query performance over previous versions, which had to access text and image data sequentially.

The new storage format for text and image datatypes also enables Adaptive Server to asynchronously prefetch the data into an existing buffer pool. You can fine-tune the prefetch settings for text and image data using the new **sp\_configure** parameter, **text prefetch size**. See the *System Administration Guide* for more information about **text prefetch size**.

## Abstract Plans

Adaptive Server can capture query text and save an abstract plan for a query in a new system table called *sysqueryplans*. Using a rapid hashing method, incoming SQL queries can be compared to stored query text, and if a match is found, the saved abstract plan is used to execute the query.

An abstract plan describes the execution plan for a query using a language created for that purpose. This language contains operators to specify the choices and actions that can be generated by the optimizer. For example, to specify an index scan on the *titles* table, using the index *title\_id\_ix*, the abstract plan says:

```
( i_scan title_id_ix titles)
```

Abstract plans provide a means for System Administrators and performance tuners to protect the overall performance of a server from changes to query plans. changes in query plans can arise due to:

- Adaptive Server software upgrades that affect optimizer choices
- New Adaptive Server features that change query plans
- Changing tuning options such as the parallel degree, table partitioning, or indexing

Other uses include:

- The ability to look for specific types of plans, such as table scans or reformatting
- The ability to determine which plans use particular indexes
- The ability to specify full or partial plans for poorly-performing queries.

For information about using abstract plans, see Chapter 21, “Introduction to Abstract Plans,” in the *Performance and Tuning Guide*.

## Disabling Triggers

Loading a database from a previous dump causes any triggers defined in the database to fire, which can increase the amount of time required to load the database. Adaptive Server 12 introduces the **disable trigger** option of the **alter table** command to disable any triggers in a database before you load the database, reducing the time required to load the database. Use **alter table... enable trigger** to re-enable the triggers after the load database is complete. **alter table... disable trigger** uses the following syntax:

```
alter table [database_name.[owner_name].]table_name  
{enable | disable} trigger [trigger_name]
```

## Cache Partitions

In a multiengine environment, each task that needs to access the data cache holds a spinlock on the cache while it accesses the cache. With a large number of engines and a high transaction rate, contention for buffer cache spinlocks can slow performance. Adaptive Server version 12 allows you to divide each cache into partitions, each with its own spinlock. See “Configuring Data Caches” in the *System Administration Guide* for information on how to configure cache partitions. See “Memory Use and Performance” and “Monitoring Performance with sp\_sysmon” in the *Performance and Tuning Guide* for information on how to check cache spinlock contention and other considerations when using cache partitions.

## Distributed Transaction Management

Adaptive Server version 12 introduces several distributed transaction management features to:

- Bring Adaptive Server into full compliance with the X/Open XA protocol when acting as a resource manager, without requiring additional services such as XA-Server
- Provide support for distributed transactions coordinated by Microsoft Distributed Transaction Coordinator (MSDTC)
- Ensure consistent commit or rollback for all transactions that update Adaptive Server version 12 data via remote procedure calls (RPCs) and Component Integration Services (CIS)
- Provide the framework to support additional distributed transaction management protocols in the future

These capabilities are available as a separately-licensed Adaptive Server feature. See *Using Adaptive Server Distributed Transaction Management Features* for more information.

## Enhancements to Component Integration Services (CIS)

Adaptive Server version 12 includes these new Component Integration Services (CIS) features:

- New **dbcc** commands – **dbcc cis showcaps** and **dbcc cis setcaps** allow examination and modification of server capabilities.
- Proxy databases – A proxy database is created containing proxy tables. Metadata for proxy tables is imported automatically from the remote location which contains the actual tables. This metadata is then used to create proxy tables within the proxy database.
- Synchronizing proxy tables – Proxy tables in a proxy database can be resynchronized using the **alter database** command.
- Java in the database
- Support for ANSI joins
- 50-table join limit
- New global variables
- Support for remote server logins
- Two-phase commit transaction management
- Support for **execute immediate**
- New set commands
- System proxy databases in support of high availability clusters
- **create table** syntax
- **create existing table** syntax
- **create proxy\_table**
- RPC's as proxy tables

See the *Component Integration Services User's Guide* for more information.

## Network Security

Network security for Adaptive Server is achieved through the security control layer (SCL). The security features supported by Adaptive Server are provided by one or more of the security drivers that work with a particular security mechanism.

Releases of Adaptive Server earlier than 12 could obtain network service information only by reading the interfaces file in the Sybase release directory. Adaptive Server 12 can access network service information, such as the host name and port number, from sources other than the interfaces file. Adaptive Server sends and receives this configuration information through the directory control layer (DCL), which defines a generic directory service for applications to access the network service information.

The drivers for specific service providers are called directory drivers. If no directory drivers are installed, the DCL obtains network services information from the interfaces file. Directory drivers are platform-specific. For example, the DCE directory driver is implemented on several UNIX platforms, but the NT Registry driver is implemented only on the NT platform.

## Performance Considerations

Using the DCL should have little or no impact on performance. You may notice some change in performance when the server is started and when connections are created. On listening ports there is no performance impact after the server is booted. On server-to-server connections, there is no performance impact after the site handler connection is established. The performance of the DCL depends on the performance of the directory driver.

## Security Features Supported in Adaptive Server 12

Depending upon the security mechanism you choose, Adaptive Server allows you to use one or more of these security services:

- *Unified login* – Adaptive Server receives a user's previously authenticated credential (login and password) from the client application when the client makes a request of the server. Adaptive Server then verifies that credential with the security mechanism, so the user supplies a login name and password only once to gain access to Adaptive Server.



- *Integrity* – Adaptive Server uses a message integrity algorithm negotiated at login time with the client application to verify that data communications from the client have not been tampered with. Adaptive Server “signs” all outgoing data communications so the client can verify their integrity.

For details on setting up your system to use security features, see Chapter 10, “Using Network-Based Security,” in the *System Administration Guide*.

## User-Defined Login Security

User-defined login security (UDLS) gives you more control over security-related features of Adaptive Server. In Adaptive Server 12, the System Security Officer can:

- Add more user logins and roles than was possible in earlier versions
- Specify the maximum allowable number of times an invalid password can be entered for a login or role before that login or role is automatically locked
- Lock and unlock roles manually
- Ensure that all user passwords have at least one digit
- Specify the minimum password length required server-wide or for a specific login or role
- Display all security-related information for logins and roles
- Associate a password expiration value with a specified login or role
- Auditing is part of the Adaptive Server security features, but it was not updated for version 12.0, so there are no documentation changes for auditing.

The features of UDLS are described in detail in the *System Administration Guide* and the *Adaptive Server Reference Manual*.

## Increased Maximum Number of Users and Logins

The maximum number of logins and the maximum number of users per-database was increased. You may notice the following new behaviors as a result of these changes:

- Negative values may be used for user IDs (*uid*).
- The server user ID (*suid*) associated with a group or a role in *sysusers* is no longer equal to the negation of their user ID (*uid*). In Adaptive Server 12.0, every *suid* associated with a group or a role in *sysusers* is set to -2 (INVALID\_SUID).

## Setting and Changing the Maximum Login Attempts

Setting the maximum number of login attempts allowed provides protection against “brute-force” or dictionary-based attempts to guess passwords. A System Security Officer can specify a maximum number of consecutive login attempts allowed, after which the login or role is automatically locked. The number of allowable failed login attempts can be set for the entire server or for individual logins and roles. Individual settings override the server-wide setting.

The number of failed logins is stored in the `logincount` column in `master.syslogins`. A successful login resets the number of failed logins to 0.

You can use the following commands or system procedures to set or change the maximum number of login attempts:

- **create role**
- **alter role**
- **sp\_addlogin**
- **sp\_modifylogin**
- **sp\_configure**

## Locking and Unlocking Logins and Roles

A login or role can be locked when:

- Its password expires, or
- The maximum number of failed login attempts occur, or
- The System Security Officer locks the login or role manually.

You can use the following commands or system procedure to lock or unlock logins and roles:

- **alter role**
- **dataserver -u**
- **sp\_locklogin**

## Displaying Password Information

You use the **sp\_displaylogin** system procedure to display password information for logins and roles.

## Checking Passwords for At Least One Character

The System Security Officer can tell the server to check for at least one character or digit in a password, using the server-wide configuration parameter, **check password for digit**. If set, this parameter does not affect existing passwords. By default, checking for digits is off.

## Setting and Changing Minimum Password Length

In previous releases, the minimum password length was a nonconfigurable, hard-coded value of six characters. In Adaptive Server 12.0, you can customize passwords to fit your needs such as using four-digit personal identification numbers (PINs) or anonymous logins with NULL passwords.

The System Security Officer can specify:

- A globally enforced minimum password length
- A per-login or per-role minimum password length

The per-login or per-role value overrides the server-wide value. Setting a minimum password length affects only new passwords created after setting the value. It does not affect existing passwords.

You can use the following commands or system procedure to set the minimum password length:

- **create role**
- **alter role**
- **sp\_addlogin**
- **sp\_modifylogin**
- **sp\_configure**

## Setting the Expiration Interval for a Password

System Administrators and System Security Officers can:

- Specify the expiration interval for a login password at creation
- Change the expiration interval for a login password
- Specify the expiration interval for a role password at creation
- Change the expiration interval for a role password

When a password for a login or role is about to expire, a warning message asks the user to contact the System Security Officer.

Password expiration did not affect roles in releases prior to Adaptive Server 12.0. Therefore, in Adaptive Server 12.0 password expiration is deactivated for any existing user-defined role passwords. During the upgrade all user-defined role passwords are stamped as having a password interval of 0.

Use **sp\_addlogin** to set the password expiration interval for a new login. Use **sp\_modifylogin** to change the password expiration interval for an existing login, add a password expiration interval to a login that did not have one, or remove a password expiration interval.

## Concrete Identification

Concrete identification enables Adaptive Server to verify chains of ownership between procedures, views, and triggers and the objects they reference in other databases. Adaptive Server identifies users during a session by login name or server user ID (*suid*). This identification applies to all databases in the server. When the user creates an object, the server associates both the owner's database user ID (*uid*) and the creator's login name with the object in the *sysobjects* table. This information concretely identifies the object as belonging to that user, which allows the server to recognize when permissions on the object can be granted implicitly.

For more information about concrete identification, see the *System Administration Guide*.

## **System Changes in Adaptive Server Version 12**

This chapter describes the system changes introduced in Adaptive Server version 12.

Topics covered are:

- “System Changes in Version 12” on page 34
- “Changes That May Affect Existing Applications” on page 43

## System Changes in Version 12

This section provides a summary of the version 12 changes, as follows:

- “Changes to Configuration Parameters” on page 34
- “Changes to Transact-SQL Commands” on page 35
- “New and Changed System Procedures” on page 37
- “Changes to Databases and System Tables” on page 39
- “New Reserved Words” on page 41
- “Changes to the Documentation” on page 42

### Changes to Configuration Parameters

This section describes the new and changed configuration parameters in version 12.

#### New Configuration Parameters

Table 2-1 summarizes the new configuration parameters:

**Table 2-1: New configuration parameters**

<b>Parameter</b>	<b>Function</b>
check password for digit	Sets the server to check for at least one character or digit in a password.
dtm detach timeout period	Sets the maximum amount of time that detached transactions remain active in Adaptive Server.
dtm lock timeout period	Sets the maximum amount of time, in seconds, that a distributed transaction branch will wait for lock resources to become available.
enable dtm	Enables or disables the Distributed Transaction Management feature.
enable HA	Enables Adaptive Server to be configured for Sybase’s Failover in a high availability system.
enable java	Enables or disables Java on the server.
enable xact coordination	Enables or disables Adaptive Server distributed transaction coordination services.
maximum failed logins	Sets the server-wide maximum number of login attempts for logins and roles.



Parameter	Function
minimum password length	Specifies a server-wide value for minimum password length for both logins and roles.
number of dtx participants	Sets the total number of remote transactions that the Adaptive Server transaction coordination service can propagate and coordinate at one time.
size of global fixed heap	Specifies the memory space for internal data structures.
size of process object fixed heap	Specifies the memory space for the Java VM for Java objects referenced during a session.
size of shared class heap	Specifies the shared memory space for all Java classes called into the Java VM.
strict dtm enforcement	Determines whether or not Adaptive Server transaction coordination services will strictly enforce the ACID properties of distributed transactions.
text prefetch size	Limits the number of pages of text and image data that can be prefetched into an existing buffer pool.
txn to pss ratio	Determines the total number of transaction descriptors available to the server.
xact coordination level	Defines the length of time between attempts to resolve transaction branches that were propagated to remote servers.

## Changes to Configuration Parameters

Table 2-2 summarizes the changes to existing configuration parameters.

*Table 2-2: Changed configuration parameters*

Parameter	Change
enable cis	The default value has been changed from 0 (off) to 1 (on).

## Changes to Transact-SQL Commands

This section describes the new and changed Transact-SQL commands in version 12.

### New Transact-SQL Commands

Table 2-3 summarizes the new Transact-SQL commands.

**Table 2-3: New commands**

Command	Function
quiesce database	Suspends and resumes updates to a specified list of databases.
join	Indicates the join uses ANSI syntax for either an inner or outer join.
remove java	Removes one or more Java classes, packages, or JARs from the database.

### Changed Transact-SQL Commands

Table 2-4 summarizes the changes to existing Transact-SQL commands:

**Table 2-4: Changed commands**

Command	Change
alter role	Includes new lock, unlock, and all overrides options to change the lock status of a role.  Includes new passwd expiration, max failed_logins, and min passwd length options to change the password expiration interval, the minimum password length, and the maximum number of failed logins allowed for a specified role.
alter table	Includes new off row and in row options to specify storage for Java-SQL columns.
create role	Includes new lock, unlock, and all overrides options to set the lock status of a role.  Includes new passwd expiration, min passwd length, max failed_logins options to set the password expiration interval, the minimum password length, and the maximum number of failed logins allowed for a specified role.
create table	Includes new off row and in row options to specify storage for Java-SQL columns.
dbcc	Includes new commands for heuristically completing distributed transactions.  Includes new rebuild_text command for rebuilding or creating internal data structures for text or image data. <b>Adaptive Server</b> uses these data structures to perform random access and asynchronous prefetch for queries against text or image data.
disk init	Includes the new dsync option to control buffered writes to devices.
disk reinit	Includes the new dsync option to control buffered writes to devices.

Command	Change
execute	Includes the new execute() syntax to dynamically execute Transact-SQL commands.

### New and Changed *set* Command Options

The following table summarizes the new and changed options for the **set** command:

*Table 2-5: New and changed set command options*

set Option	Description
clientappname	Assigns an application an individual name for the current session.
clienthostname	Assigns a host an individual name for the current session.
clientname	Assigns a client an individual name for the current session.
strict_dtm_enforcement	Determines whether the Adaptive Server will propagate transactions to servers that do not support Adaptive Server transaction coordination services
transactional_rpc	Determines whether or not Adaptive Server coordinates remote procedure calls and updates to remote server data.

### New Transact-SQL Functions

Table 2-6 summarizes the new Transact-SQL functions.

*Table 2-6: New Transact-SQL functions*

Function	Description
license_enabled	Returns 1 if a feature's license is enabled, 0 if the license is not enabled, or null if you specify an invalid license name.

### New and Changed System Procedures

This section describes the new system procedures added and changes made to existing system procedures.

## New System Procedures

Table 2-7 summarizes the new system procedures:

**Table 2-7: New system procedures**

System Procedure	Function
sp_deviceattr	Changes the dsync setting of an existing database device file.
sp_transactions	Reports detailed information about active transactions.
sp_companion	Performs cluster operations such as configuring Adaptive Server as a secondary companion in a high availability system and moving a companion server from one failover mode to another. sp_companion is run from the secondary companion
sp_ha_admin	Performs administrative tasks that address second point of failures on Adaptive Servers configured with Sybase's Failover in a high availability system.
sp_helpjava	Displays information about Java classes and JARs installed in the database.

## Changed System Procedures

Table 2-8 summarizes the changes made to existing system procedures.

**Table 2-8: Changed system procedures**

Procedure	Change
sp_addlogin	Specifies the password expiration interval, the minimum password length, and the maximum number of failed logins allowed for a specified login at creation.
sp_displaylogin	Displays the login security related parameters configured for a login. The syntax has not changed; however, the output shows the following additional information: <ul style="list-style-type: none"> <li>• Whether the account is locked</li> <li>• The date of the last password change</li> <li>• The password expiration interval</li> <li>• Whether the password has expired</li> <li>• The minimum length of the password</li> <li>• The maximum number of failed logins allowed before the login is locked</li> <li>• The current number of failed logins</li> </ul>

Procedure	Change
sp_displayroles	Displays the login security-related parameters configured for a role. The output has changed as described for sp_displaylogin.
sp_helpdevice	Output displays the new dsync setting in the “description” column.
sp_lock	Output displays the new lock owner ID (loid) field.
sp_modifylogin	Use to change password expiration interval, the minimum password length, and the maximum number of failed logins allowed for a specified existing login.
sp_monitorconfig	Displays additional usage statistics for transaction descriptors and DTX participants.
sp_who	Output displays the new block_xloid field, which indicates the lock owner ID of the blocking transaction. The older <i>blocked</i> field in the output is renamed to <i>blk_spid</i> .

## New Utility Programs

Table 2-9 describes the new utility programs added to Adaptive Server version 12.

**Table 2-9: New utility programs**

System Procedure	Function
extractjava (UNIX)	Copies a retained JAR and the Java classes it contains from Adaptive Server to a client file.
extrjava (Windows NT)	
installjava (UNIX)	Installs a JAR and the Java classes it contains from a client file to Adaptive Server.
instjava (Windows NT)	

## Changes to Databases and System Tables

This section describes the changes made to databases and system tables to support version 12 features.

### New Databases

Table 2-10 lists the databases that are new in this release:

**Table 2-10: New databases**

Database	Function
<i>sybdiagdb</i>	Holds diagnostic configuration data for debugging purposes. It is for Sybase Technical Support personnel only and should not be used by customers.

## New System Tables

Table 2-11 lists the system tables that are new in this release:

**Table 2-11: New system tables**

Table	Function
<i>syscolumns</i>	Adds the <i>xtype</i> and <i>xbid</i> columns. These columns are used only for Adaptive Servers enabled for Java.
<i>syscoordinations</i>	( <i>sybssystemdb</i> database only.) Contains information about remote Adaptive Servers participating in distributed transactions (remote participants) and their coordination states.
<i>sysdevices</i>	<i>status</i> column utilizes a new control bit 0x4000 (16384 Decimal) to control the <i>dsync</i> setting for a database device file.
<i>sysjars</i>	Contains one row for each Java archive file retained in the database. This table is only used for Adaptive Servers enabled for Java.
<i>syslocks</i>	Adds the <i>loid</i> column.
<i>sysprocesses</i>	Adds the <i>block_xloid</i> column. The <i>blocked</i> column is renamed to <i>block_xloid</i> .
<i>sysqueryplans</i>	Stores query text and abstract plans for a query.
<i>syssessions</i>	Contains one row for each client that connects to Adaptive Server with the failover property (for example, <i>isql -Q</i> ). This table is only used for Adaptive Servers that are configured for Sybase's Failover in a high availability system.
<i>systransactions</i>	( <i>master</i> database only.) Contains information about Adaptive Server transactions
<i>sysxtypes</i>	Contains one row for each extended, Java-SQL datatype. This table is only used for Adaptive Servers enabled for Java.

## Changed System Tables

Table 2-12 lists the system tables that have been changed in this release:

**Table 2-12: Changed system tables**

Table	Change
<i>sysattributes</i>	Stores information for the configuration parameters that set or change the password expiration interval, maximum failed login attempts, and the minimum password length.
<i>sysconfigures</i>	Stores the server-wide values set with the configuration parameters minimum password length and maximum failed logins for logins and roles.
<i>syslogins</i>	The new column, <i>logincount</i> , indicates the number of failed logins. A successful login automatically resets the number of failed logins to 0. The datatype of this column is int.
<i>syservroles</i>	The following columns have been added: <ul style="list-style-type: none"> <li>• <i>pwdate</i> – records the date and time when the password of a role was last changed. The datatype of this column is datetime.</li> <li>• <i>status</i> – indicates the status of the role. The datatype of this column is smallint. The status is shown as a numeric value that corresponds to the combination of expiration and lock status for the role. The values are: <ul style="list-style-type: none"> <li>0 – role is not expired or locked</li> <li>2 – role is not expired but is locked</li> <li>4 – role is expired but is not locked</li> <li>6 – role is expired and locked</li> </ul> </li> </ul>
<i>sysusers</i>	In previous releases, the <i>sysusers</i> system table in each database had a unique clustered index on <i>suid</i> . However, in <b>Adaptive Server 12.0</b> the uniqueness of <i>sysusers.suid</i> does not exist, so the clustered index has been dropped.

For more information about system tables, see the Adaptive Server Reference Manual.

## New Reserved Words

The *Adaptive Server Reference Manual* lists all of the reserved words for version 12. Reserved words cannot be used as object names or column names.

You must change all database names that are new reserved words before you can upgrade from an earlier release of the server. You can change table, view, and column names or use delimited identifiers. Once you upgrade to version 12, you cannot use database objects whose names are new reserved words until you modify your procedures, SQL scripts, and applications.

## Changes to the Documentation

Adaptive Server version 12 includes several new books that describe separately-licensed Adaptive Server features such as Java, High Availability, and Distributed Transaction Management. See “Related documents” on page ix for a brief description of these books.

Information from the following two books has been moved into the *System Administration Guide* and *Performance and Tuning Guide*:

- *Security Features User’s Guide*
- *Security Administration Guide*

Information from *Configuring Adaptive Server for UNIX* has been moved into the *Installation Guide*.

The **Adaptive Server** documentation set no longer includes the following documents:

- *Adaptive Server Enterprise Getting Started*
- *Introducing Adaptive Server Enterprise*
- *Master Index to Adaptive Server Documentation*
- *Navigating Adaptive Server Enterprise Documentation*
- Online *sybsyntax* database and **sp\_syntax** procedure
- *Quick Reference Card*



## Changes That May Affect Existing Applications

This section describes system changes introduced by version 12 that may affect your applications if you are upgrading from an earlier release.

### New Transact-SQL Keywords for 12

See “New Reserved Words” on page 41.

### *dsync* On By Default for UNIX File Devices

Beginning with version 12, Adaptive Server initializes UNIX file system devices using the *dsync* option to ensure recovery. If you require earlier behavior or performance characteristics, change the **dsync** setting using **sp\_deviceattr**. See **disk init** and **sp\_deviceattr** in the *Adaptive Server Reference Manual* for more information.



## New Features in Adaptive Server Version 11.9.2

This chapter describes the new functionality in version 11.9.2. Topics include:

- “New Locking Schemes” on page 46
- “Changes to Statistics and Query Optimization” on page 47
- “Enhancements to the create index Command” on page 48
- “Changes and Additions to Transact-SQL Syntax” on page 49
- “Configurable Database Recovery Order” on page 51
- “Fault Verification for dbcc checkstorage Faults” on page 52
- “License Use Monitor” on page 53
- “Task-to-Engine Affinity” on page 54
- “Dynamic SQL Performance Improvements” on page 55
- “Direct Updates Through Joins” on page 56
- “Component Integration Services Changes” on page 57
- “Character Set Changes” on page 58
- “Changes That May Affect Existing Applications” on page 59

## New Locking Schemes

Adaptive Server version 11.9.2 provides two new locking schemes to improve the concurrency and performance of Adaptive Server:

- Datapages locking
- Datarows locking, also known as row-level locking

Together, these new locking schemes are referred to as data-only locking.

The pre-11.9.2 locking scheme continues to be supported; it is called **allpages locking**, and it is the default locking scheme when you first install or upgrade to version 11.9.2. A System Administrator can specify any locking scheme as the server-wide default.

Users can specify a locking scheme for a newly created table, using the **create table** command, and can change the locking scheme for an existing table to any other locking scheme using the **alter table** command.

Some of the changes made to support these new locking schemes include:

- Additional types of locks
- Changes to table and index structures for tables using the new locking schemes
- New configuration parameters and changes to existing configuration parameters
- Additions and changes to Transact-SQL command syntax, including:
  - Additions to **create table** and **alter table** to allow specifying the locking scheme
  - A new **reorg** command to manage space in tables that use the new locking schemes
  - Changes to **select into** syntax to allow specifying the locking scheme on the created table
- Changes to system procedures to allow reporting on and configuring of new functionality

---

**Note** The number of locks available to all processes on the server is limited by the configuration parameter **number of locks**. Changing to data-only locking affects the number of locks required during query processing.

---

## Changes to Statistics and Query Optimization

Adaptive Server version 11.9.2 increases the flexibility of the statistics used by the query optimizer. Optimizer statistics are now kept in two system tables, `systabstats` and `sysstatistics`. A new utility, **optdiag**, can extract statistics and allows editing of statistics. **optdiag** makes many of these statistics viewable for the first time.

Statistics are now kept on a per-column basis, rather than a per-index basis. Query costing in the optimizer has been enhanced to use column-level statistics for search arguments and joins, even if indexes do not exist on the column.

The **update statistics** command now supports storing statistics for unindexed columns. New **update index statistics** syntax facilitates creating statistics on all columns used in an index, and **update all statistics** now generates statistics for all columns in a table. A new command, **delete statistics**, can be used to drop column-level statistics, since **drop index** no longer removes statistics.

New statistics are kept to support the new locking schemes. Additional statistics track the clustering of data rows and data pages in physical storage and improve the cost estimates made by the optimizer. Some queries on tables using new locking schemes are optimized differently than queries on tables using the old locking scheme.

## Enhancements to the *create index* Command

There are two enhancements to **create index**:

- The **create index** command allows the user to specify ascending or descending order for each column in the index. In earlier versions, all indexes were created in ascending order for all columns. Scans that needed to read the data in reverse order could scan the index backward, but if the required order was a mix of ascending and descending order on the keys, the query needed to perform a sort step. Performance can be improved by matching the index ordering to the ordering used by most queries.
- Indexes can include as many as 31 key columns, an increase from 16 in previous versions. The maximum total number of bytes allowed for index keys is 600.

## Changes and Additions to Transact-SQL Syntax

Adaptive Server version 11.9.2 provides new commands and options that affect locking:

- **readpast** option
- **lock table** command
- Server-level and session level options that specify how long tasks wait for locks
- Support for repeatable reads transaction isolation level (level 2) for data-only-locked tables

### New *readpast* Concurrency Option

**select**, **delete**, **update**, **readtext**, and **writetext** commands now include the *readpast* option. This option allows the command to skip all pages or rows that are not immediately accessible due to incompatible locks.

### New **lock table** Command

A new command, **lock table**, explicitly locks entire tables in either shared or exclusive mode.

### Specifying a Wait Time for Locks

In earlier versions of Adaptive Server and SQL Server, all commands that cannot immediately acquire a lock block, or wait, until the required lock is available. In version 11.9.2, the maximum length of time that a task waits can be specified on a server-wide or session-level basis. Processes that fail to acquire the lock within the specified waiting period fail with an error message.

- For information on the new configuration parameter, **lock wait period**, see the *System Administration Guide*.
- For information on the new **set** option, **set lock wait**, see the *Adaptive Server Reference Manual*.

## **Repeatable Read Transaction Isolation**

Version 11.9.2 adds support for repeatable read transactions (isolation level 2).



## **Configurable Database Recovery Order**

You can now specify the order in which user databases are recovered when you boot the server. This allows your most critical databases to be recovered first.

## Fault Verification for *dbcc checkstorage* Faults

The **dbcc checkstorage** command, introduced in Adaptive Server version 11.5, checks the consistency of database objects without blocking access by other tasks. Since user activity can produce soft faults, **dbcc checkstorage** attempts to resolve the fault by a second check; if the fault persists, it is recorded in the dbccdb database. Since heavy database use during **dbcc checkstorage** operations can lead to lengthy fault reports, the **dbcc checkverify** command introduced in version 11.9.2 rechecks these faults, while holding a lock on the object, in order to eliminate the soft faults from the list of faults.

## **License Use Monitor**

System Administrators can determine whether the usage on their Adaptive Server system exceeds license agreements by using the license monitor feature. Once a day, it records the maximum number of users in a system table in the master database.

## Task-to-Engine Affinity

For servers running multiple engines, task-to-engine affinity improves performance by reducing the number of times the task context has to be switched between engines. The only visible effects of this are during system tuning. When there are fewer tasks running than there are engines, you may see some engines showing high utilization, while others are idle for the entire interval. For example, in earlier versions, while a single task runs a **select \*** query, you would generally see each of 6 engines performing about 16% of the work. In version 11.9.2, **sp\_sysmon** might report:

```
Engine Busy Utilization
Engine 0                97.2 %
Engine 1                 0.0 %
Engine 2                 0.0 %
Engine 3                 0.0 %
Engine 4                 0.0 %
Engine 5                 0.0 %
-----
Summary                 Total    97.2 %           Average  16.2 %
```

For more information, see the *Performance and Tuning Guide*.

## **Dynamic SQL Performance Improvements**

In pre-11.9.2 releases, the simultaneous use of Dynamic SQL by a large number of users performs poorly, largely due to contention on system tables in tempdb. In version 11.9.2, the contention problems have been eliminated, and the performance of the prepare and deallocate statements has been significantly improved. There are no user interface changes. All changes are internal, so applications do not need to be modified.

## **Direct Updates Through Joins**

In version 11.9.2, many restrictions on when direct updates can be performed have been removed.

## Component Integration Services Changes

Component Integration Services for Adaptive Server version 11.9.2 adds the following enhancements for remote table access:

- Enhances the performance of queries that include remote tables
- Simplifies the mapping of local proxy tables to remote tables
- Adds new server classes for Adaptive Server Enterprise, Adaptive Server Anywhere™, and Adaptive Server IQ™
- Adds a new trace flag that, when enabled, makes any query that is part of a **declare cursor** command, and that references proxy tables, read-only by default

## **Character Set Changes**

Adaptive Server version 11.9.2 includes support for the European currency symbol, or “Euro”. Adaptive Server can also perform character set conversions that cause a change in the data length.



## Changes That May Affect Existing Applications

Changes in 11.9.2 that may affect user applications and system administration are:

- Changes to statistics and query optimization can cause changes to query plans, even for tables where the locking scheme is not changed.
- More logging is performed than in previous versions.
- For standby or report-processing servers, new syntax is required to dump the transaction log for the production server and bring the database online. It is no longer possible to load a dump that contains open transactions, bring the database online, and then load a subsequent dump of the log.
- The **update statistics** command may run more slowly, because it now performs a table scan as well as scanning indexes. Concurrency is improved, because **update statistics** with a table name or index name operates at transaction isolation level 0 on data-only-locked tables, and does not acquire table locks on allpages-locked tables.
- The **update all statistics** command now generates statistics for all columns in a table. It performs a complete table scan for each column that is not the leading column in an index. If your system runs scripts that include **update all statistics**, you may want to replace **update all statistics** commands with **update statistics** and **update partition statistics** commands.

## Effects of Changing to Data-Only-Locking

Changing tables to data-only locking may have these effects:

- The number of locks required for your applications may increase, and you may need to change the **number of locks** configuration parameter.
- On data-only-locked tables with clustered indexes, the default behavior of disallowing duplicate rows and the **ignore\_dup\_row** option are not enforced during inserts and updates. This changes the behavior of commands that insert duplicate rows and may change the behavior of **create clustered index** if the table contains duplicate rows.
- On data-only-locked tables, the **sorted\_data** option to **create index** can be used only immediately following a bulk copy operation that copies into an existing table. Use of the **sorted\_data** option is prohibited once additional page allocation operations have been made for the table.

- Bulk copy into data-only-locked tables requires the version of **bcp** and the bulk copy libraries shipped with Adaptive Server version 11.9.2. Older versions of **bcp** and the bulk-copy libraries can still be used to copy into allpages-locked tables. See the *Release Bulletin* for the required version number.
- When using parallel sort for data-only-locked tables, the number of worker processes must be configured to equal or exceed the number of partitions, even for empty tables. The database option **select into/bulkcopy/pllsort** must also be enabled.

## Query Optimization Changes and Forced Query Plans

Changes in query optimization provide greater accuracy than in previous versions in costing some queries, especially those performing nonclustered index scans. In version 11.9.2, you can create statistics on unindexed columns, which can further improve query optimization. You should review any queries that are using forced query plans, especially index or join order forcing, to determine whether forcing is still needed or whether the optimizer can provide an improved plan.

### Index Forcing and Data-Only-Locked Tables

The choice of index to use for a query can be forced with **index** clause and an index name, as in this command:

```
select title_id, type, price
from titles (index title_id_ix)
```

Adaptive Server accepts an index ID in place of the keyword **index** and the index name. The following statement forces the use of a clustered index on an allpages-locked table:

```
select title_id, type, price
from titles (1)
```

When you convert a table with a clustered index to data-only locking, the index ID of the clustered index changes. If you execute a query that specifies index ID 1, the optimizer still uses the clustered index on the table. You should carefully check all query plans that force index ID 1.

---

**Note** Sybase strongly recommends the use of index names, rather than index IDs, for forcing index selection.

---

### Performance After Loading Pre-11.9 Databases

When pre-11.9 databases are loaded into an 11.9.2 Adaptive Server, the upgrade process inserts rows into the statistics tables, based on available statistics. The upgrade process does not scan the tables and indexes to generate the new types of statistics that can be used by the improved query optimizer. For statistics that are new in this version, the upgrade process inserts computed or magic values.

If you load a pre-11.9 database, run **update statistics** on the tables as soon as possible. If you experience performance problems after loading a pre-11.9 database, run **update statistics** for the tables in the application before you attempt other solutions.

### Ordering of Results with Data-Only Locked Tables

Clustered indexes on data-only-locked tables may not return rows in clustered key order if there is no **order by** clause. Bulk copy is not guaranteed to copy out a table in clustered key order.

Queries on unpartitioned allpages-locked tables with clustered indexes return rows in the clustered key order if the query does not include a sort (that is, if there are no clauses such as **order by** or **distinct**). Bulk copy also copies the rows out of allpages-locked tables in clustered key order for both partitioned and unpartitioned tables.

## Delay of Cursor Compilation Until Cursor Open

In pre-11.9.2 versions of SQL Server and Adaptive Server, a cursor is optimized and compiled when it is declared. For cursors with local variables or parameters, this can result in a suboptimal query plan, since default values are used to estimate the number of rows to be returned. The problem of poor plans is most common with client cursors, sent using Open Client™ Client-Library™ or Embedded SQL™.

In version 11.9.2, query optimization is delayed until the cursor is opened. The values of the variables are known at cursor open time, and query optimization may be considerably improved.

## Visible Effects of Delayed Cursor Optimization

Besides improved query plans, the only user-visible changes are as follows:

- If you are using showplan, the plan is no longer displayed at the declare cursor statement, but at the open statement.
- The query is normalized at the declare cursor statement, so error messages about syntax errors, missing tables, and so forth, are reported at that time.
- Some error messages do not appear until the query is compiled and optimized, including those for cursors that require unique indexes. If error messages result at cursor open time (such as the lack of a needed index), the cursor must be deallocated and redeclared. You cannot just correct the problem and reopen the cursor.
- In pre-11.9.2 versions, changing isolation levels between cursor declare time and cursor open time generated an error message when the cursor was opened. In version 11.9.2, the following sequence of commands does not generate an error, and the cursor is compiled at transaction isolation level 0:

```
*start at isolation level 1 */
declare curs1 cursor for
select t.title_id, au_lname
from titles t , authors a, titleauthor ta
where t.title_id = ta.title_id
and a.au_id = ta.au_id
go
set transaction isolation level 0
go
open curs1
go
fetch curs1
```

go

---

**Note** Delayed optimization does not apply to cursors that are declared in a stored procedure (called server cursors). Server cursors are compiled and optimized when they are declared.

---



## New Features in Adaptive Server Release 11.5

This chapter describes the new features and system changes introduced in Sybase Adaptive Server Enterprise release 11.5.

In addition to providing many new server features, Adaptive Server 11.5 provides a more comprehensive product by incorporating previous products and tools:

- SQL Server Manager™ has been replaced by Sybase Central™  
For more information, see “Sybase Central” on page 105.
- OmniConnect™ has been replaced by Component Integration Services

Many of the OmniConnect release 10.5 features have been incorporated into the Component Integration Services (CIS) feature of Adaptive Server 11.5. For more information on CIS, see “Component Integration Services” on page 73.

## New Features in Release 11.5

The new features in Adaptive Server release 11.5 are:

<b>Name</b>
“Asynchronous Prefetch” on page 67
“Auditing Enhancements” on page 69
“case Expression” on page 72
“Component Integration Services” on page 73
“create index Enhancements” on page 76
“dbcc Enhancements” on page 77
“Descending Index Scan Optimization” on page 80
“Directory Services” on page 81
“Engine Affinity and Execution Precedence” on page 82
“Expanded Error Logging” on page 84
“Extended Stored Procedures (ESPs)” on page 86
“Increased Referential Integrity Limits” on page 89
“Metadata Cache Management” on page 90
“Monitor Access to SQL Batch Text” on page 91
“Parallel Bulk Copy” on page 95
“Parallel Queries and Enhanced Partitioning” on page 96
“Point-in-Time Recovery” on page 97
“Proxy Authorization” on page 98
“Recovery Fault Isolation” on page 99
“Relaxed LRU Cache Replacement Policy” on page 101
“Resource Limits” on page 102
“Source Text Verification and Encryption” on page 104
“Sybase Central” on page 105
“sp_sysmon” on page 106
“Two-Phase Commit Enhancements” on page 107
“User-Defined Roles” on page 109
“Windows NT Event Log Integration” on page 110
“Windows NT MAPI Support” on page 111
“Windows NT Performance Monitor Integration” on page 112



## Asynchronous Prefetch

Asynchronous prefetch improves performance by anticipating the pages that will be required for certain well-defined classes of database activities whose access patterns are predictable. The I/O requests for these pages are issued before the query needs them so that most pages are in cache by the time query processing needs to access the page.

Asynchronous prefetch can improve performance for:

- Sequential scans, such as table scans, clustered index scans, and covered nonclustered index scans
- The **update statistics** command
- Accesses via nonclustered indexes
- Some **dbcc** checks
- Recovery

Asynchronous prefetch can improve the performance of queries that access large numbers of pages, such as decision support system (DSS) applications, as long as the I/O subsystems on the machine are not saturated.

Asynchronous prefetch cannot help (or may help only slightly) when the I/O subsystem is already saturated or when Adaptive Server is CPU-bound. It can be used in some online transaction processing (OLTP) applications, but to a much lesser degree, since the queries in OLTP applications usually perform fewer I/O operations.

---

**Note** The default setting for **global async prefetch limit** is 10, meaning that asynchronous prefetch is enabled when you install or upgrade to release 11.5, and that all pools can use up to 10 percent of the buffers for asynchronous reads.

---

Table 4-1 shows where to find additional information on asynchronous prefetch.

**Table 4-1: Additional information on asynchronous prefetch**

<b>For information On</b>	<b>See</b>
How queries can benefit from asynchronous prefetch	Chapter 18, "Tuning Asynchronous Prefetch," in the <i>Performance and Tuning Guide</i>
Configuring pool limits	Chapter 9, "Configuring Data Caches," in the System Administration Guide

<b>For information On</b>	<b>See</b>
Configuring server-wide limits	Chapter 11, "Setting Configuration Parameters," in the System Administration Guide
Tuning asynchronous prefetch limits	Chapter 24, "Monitoring Performance with sp_sysmon," in the Performance and Tuning Guide

## Auditing Enhancements

The auditing system provides the following enhancements:

- Support of multiple audit tables. This capability allows you to archive and process audit data with no manual intervention and no loss of audit records.
- A new, streamlined, user interface for the System Security Officer to control auditing. In the new interface, which replaces the current auditing interface, a single system procedure, **sp\_audit**, provides the capability to set all auditing options.
- The addition of events that can be audited, including:
  - Creation of objects
  - Binding and unbinding of rules, defaults, and messages
  - All actions by individual users
  - All actions performed with a particular role active
  - A set of server-wide, security-relevant events

## New System Procedures

Use the system procedures listed in Table 4-2 to manage the auditing process.

**Table 4-2: New system procedures for auditing**

System Procedure	Description
sp_addauditrecord	Adds user-defined audit records (comments) into the audit trail. Users can add these records only if a System Security Officer enables ad hoc auditing with sp_audit.
sp_audit	Enables and disables all auditing options. This is the only system procedure required to establish the events to be audited.
sp_configure	
"auditing"	Enables or disables auditing for Adaptive Server. The parameter takes effect immediately upon execution of sp_configure. Auditing occurs only when this parameter is enabled.
"audit queue size"	Establishes the size of the audit queue. Because this parameter affects memory allocation, it does not take effect until Adaptive Server is restarted.
"current audit table"	Sets the current audit table. This takes effect immediately upon execution of sp_configure.
"suspend auditing when full"	Controls the behavior of the audit process when an audit device becomes full. This takes effect immediately upon execution of sp_configure.
sp_displayaudit	Displays the active auditing options.

**Note** The system procedures **sp\_auditoption**, **sp\_auditdatabase**, **sp\_auditobject**, **sp\_auditsproc**, and **sp\_auditlogin** are obsolete. Use **sp\_audit** for setting the options that you set with those system procedures in pre-11.5 releases.

Table 4-3 shows the system procedures that were used to set auditing options in pre-11.5 releases and the types of auditing options that replace them in Adaptive Server release 11.5. The auditing options in the groups mentioned in the second column of Table 4-3 are described in the discussion of **sp\_audit** in the Adaptive Server Reference Manual.

**Table 4-3: Comparison of pre-11.5 and 11.5 options**

Option set in pre-11.5 releases	Option to set In 11.5
sp_auditoption	“global” auditing options
sp_auditdatabase	“database-specific” auditing options
sp_auditobject	“object-specific” auditing options
sp_auditsproc	“exec” auditing options

Option set in pre-11.5 releases	Option to set In 11.5
sp_auditlogin	“user-specific” auditing options

For example, in pre-11.5 releases, you used the following command to audit the actions of deleting, updating and inserting information in the *authors* table of the *pubs2* database:

```
sp_auditobject "authors", "pubs2", "both", "dui"
```

In release 11.5, these actions are covered by the **delete**, **update**, and **insert** options in the “object-specific” group. You use the following commands to achieve the same results as you did using the preceding, obsolete commands:

```
sp_audit "delete", "all", "authors", "on"  
sp_audit "update", "all", "authors", "on"  
sp_audit "insert", "all", "authors", "on"
```

To install auditing, see the Adaptive Server configuration guide for your platform. For more information about auditing, see Chapter 8, “Auditing,” in the *Security Administration Guide*.

## case Expression

Using **case** expression simplifies standard SQL expressions by allowing a **when...then** construct instead of an **if...else** construct. A **case** expression is more concise than an **if...else** statement and Adaptive Server's response time is faster, since the optimizer processes fewer lines of SQL code. Also, queries written using **case** expressions are often easier to read and understand than those written in standard SQL. Because of the concise nature of **case** expressions, they are allowed anywhere value expressions are used.

A **case** expression is written using the following syntax:

```
case
    [when search_condition1 then result1]
    [when search_conditionn then resultn]
end
```

For more information about the syntax of **case** expressions, see the *Adaptive Server Reference Manual*.

For more information and examples of **case** expressions, see the *Transact-SQL User's Guide*.

## Component Integration Services

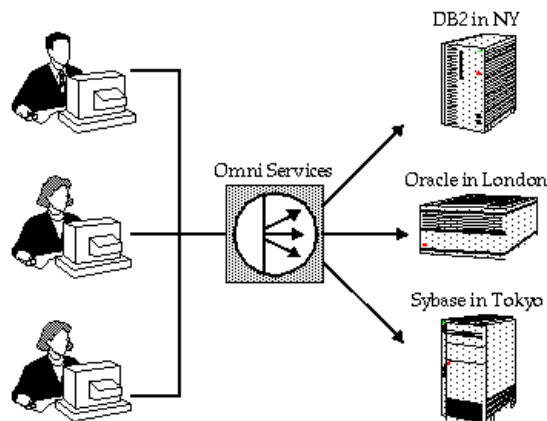
The Component Integration Services (CIS) feature allows you to connect to remote Sybase and non-Sybase databases. It presents a uniform view of enterprise data to client applications and provides location transparency to the data sources that you access.

When you access both Sybase and non-Sybase databases on different servers, you can use the host data files, tables, triggers, views, remote procedure calls, and most commands in database systems such as Adaptive Server, Informix, DB2, and Oracle.

Component Integration Services can be used by anyone who needs to access multiple data sources or legacy data. It can also be used by anyone who needs to migrate data from one server to another.

A single server is often used to access data on multiple external servers. Component Integration Services manages the data regardless of the location of the external servers. Data management is transparent to the client application.

**Figure 4-1: CIS connects to multiple vendor databases**



Using Component Integration Services, you can:

- Access tables in remote servers as if the tables were local.
- Perform joins between tables in multiple, remote, and heterogeneous servers. For example, you can join tables between Adaptive Server and an Oracle DBMS, between Adaptive Server and Record Management System (RMS) files in OpenVMS, and between tables in multiple Adaptive Servers.

- Use the **select into** command to transfer the contents of one table into a new table on any supported remote server.
- Provide applications such as PowerBuilder®, Microsoft Access, and DataEase with transparent access to heterogeneous data.
- Maintain referential integrity across heterogeneous data sources.
- Access native remote server capabilities using the Component Integration Services passthrough mode.

For more information on using Component Integration Services, see the Component Integration Services User's Guide.

## New Commands

Table 4-4 lists the new commands that support the Component Integration Services feature:

*Table 4-4: Commands for CIS*

Command	Description
connect to...disconnect	Connects to the specified server and disconnects the connected server.
create existing table	Creates a proxy table, then retrieves and stores metadata from a remote table, and places the data into the proxy table.
set cis_rpc_handling	Determines whether Component Integration Services handles outbound remote procedure call (RPC) requests by default.
set transactional_rpc	Controls the handling of remote procedure calls.

## New System Procedures

Table 4-5 lists the new system procedures that support the Component Integration Services feature.

*Table 4-5: New system procedures for CIS*

System Procedure	Description
sp_addexternlogin	Creates an alternate login account and password to use when communicating with a remote server through CIS.



<b>System Procedure</b>	<b>Description</b>
sp_addobjectdef	Specifies the mapping between a local table and an external storage location. sp_addobjectdef replaces the sp_addtabledef command.
sp_autoconnect	Defines a passthrough connection to a remote server for a specific user, which allows the named user to enter passthrough mode automatically at login.
sp_configure "enable cis"	Enables or disables CIS. Default is 0 (disabled).
sp_defaultloc	Defines a default storage location for objects in a local database.
sp_dropexternlogin	Drops the definition of a remote login previously defined by sp_addexternlogin.
sp_dropobjectdef	Deletes the external storage mapping provided for a local object.
sp_passthru	Allows the user to pass a SQL command buffer to a remote server.
sp_remotesql	Establishes a connection to a remote server, passes a query buffer to the remote server from the client, and relays the results back to the client.

## Modified System Procedures

The following system procedures have been modified to support the Component Integration Services feature:

- **sp\_addserver**
- **sp\_dropserver**
- **sp\_help**
- **sp\_helpdb**
- **sp\_helpserver**
- **sp\_serveroption**

## ***create index* Enhancements**

In earlier releases of SQL Server™, you could use the **create clustered index...with sorted\_data** to skip the sort step on data that was already in sorted order. However, **create clustered index** always copied the data pages to a new location on the data devices.

In release 11.5, **create clustered index...with sorted\_data** does not sort or copy data pages except when certain optional clauses are specified to **create index**. If these clauses are specified for partitioned tables, a parallel sort must be performed.

See Chapter 23, “Maintenance Activities and Performance,” in the Performance and Tuning Guide for more information.

## ***dbcc* Enhancements**

**dbcc** has been enhanced to improve its performance in parallel environments and to expand the way in which it detects, records, and reports errors. The **dbcc checkstorage** command allows you to check a database that is in use, with little or no impact on performance. It stores the results of the check in the *dbccdb* database. When the check is complete, you can use the new **dbcc** stored procedures to generate reports based on the data that is collected.

### **Advantages of Using *dbcc checkstorage***

The **dbcc checkstorage** command:

- Uses a named cache so that running it on the target database does not interfere with concurrent use of that database
- Scales linearly with the aggregate I/O throughput
- Does not lock tables or pages for extended periods, which allows **dbcc checkstorage** to locate errors while allowing concurrent update activity
- Combines many of the checks provided by the other **dbcc** commands
- Provides a detailed description of space usage in the target database
- Separates the functions of checking and reporting, which allows custom evaluation and report generation
- Records **dbcc checkstorage** activity and results in the *dbccdb* database, which allows trend analysis and a source of accurate diagnostic information for support

See Chapter 18, “Checking Database Consistency,” in the System Administration Guide for more information on **dbcc**.

### **System and Stored Procedures for Creating *dbccdb***

Before you can use **dbcc checkstorage**, you must set up the *dbccdb* database. Table 4-6 lists the system and stored procedures for creating *dbccdb*.

**sp\_plan\_dbccdb** is the only *system* procedure. It resides in *sybsystemprocs*.

**sp\_dbcc\_createws** and **sp\_dbcc\_updateconfig** are **dbcc** *stored* procedures. They reside in *dbccdb*.

**Table 4-6: System and stored procedures for creating dbccdb**

Procedure	Description
sp_dbcc_createws	Creates a workspace of the specified type and size on the specified segment and database.
sp_dbcc_updateconfig	Updates the configuration parameters for the specified database.
sp_plan_dbccdb	Recommends values for the size of <i>dbccdb</i> , suitable devices, sizes for the <i>scan</i> and <i>text</i> workspaces, cache size, and the number of worker processes for optimal operation.

## Stored Procedures for Maintaining *dbccdb*

Table 4-7 lists the **dbcc** stored procedures for maintaining *dbccdb*.

**Table 4-7: Stored procedures for maintaining dbccdb**

Stored Procedure	Description
sp_dbcc_alterws	Changes the size of the specified workspace to the specified value, and initializes the workspace.
sp_dbcc_deletedb	Deletes all information on the specified database from <i>dbccdb</i> .
sp_dbcc_deletehistory	Deletes results of dbcc checkstorage operations on the specified database from <i>dbccdb</i> .
sp_dbcc_evaluatedb	Recommends values for configuration parameters, using the results from previous dbcc operations.
sp_dbcc_runcheck	Runs dbcc checkstorage on the specified database and generates a summary report or the report you specify.

## Stored Procedures for Generating Reports on *dbccdb*

Table 4-8 lists the **dbcc** stored procedures for generating reports on *dbccdb*.

**Table 4-8: Stored procedures for generating reports on dbccdb**

Stored Procedure	Description
sp_dbcc_configreport	Reports the configuration parameters for the specified database.
sp_dbcc_differentialreport	Compares the results of dbcc checkstorage operations completed on <i>date1</i> and <i>date2</i> for <i>dbname..object_name</i> .

<b>Stored Procedure</b>	<b>Description</b>
sp_dbcc_faultreport	Reports faults that were recorded for <i>dbname..object_name</i> on or before the specified date.
sp_dbcc_fullreport	Reports configuration, statistics, and fault information for <i>database..object_name</i> on or before the specified date.
sp_dbcc_statisticsreport	Reports statistics information from the <i>dbcc_counters</i> table generated by the dbcc checkstorage operation on or before the specified date.
sp_dbcc_summaryreport	Reports all dbcc checkstorage operations completed for <i>dbname</i> on or before the specified date.

## Descending Index Scan Optimization

Descending index scan optimization is a performance enhancement that can improve the performance of queries that use the **desc** keyword in **order by** queries to return result sets in descending order. In earlier releases of SQL Server, descending result sets required a worktable and a sort. In Adaptive Server 11.5, the optimizer can choose to use the index and avoid the sort step, if the strategy reduces the query cost.

Descending index scans can speed the use of both clustered and nonclustered index access, reduce the *tempdb* space required for temporary tables, save the CPU time required for sorts, and shorten the time that locks are held, if descending scans use **holdlock** or transaction isolation level 3. However, there can be an increased chance of deadlocking in some applications.

See Chapter 11, “Setting Configuration Parameters,” in the System Administration Guide for information on the **allow backward scans** configuration parameter, which can be used to disable this optimization if deadlocking is a problem. Descending scans are enabled by default.

This feature does not change the syntax for the **order by** clause; it only changes the way that **order by** clauses with the **desc** keyword can be optimized.

## Directory Services

Before Adaptive Server and a client application can communicate with each other, each must know where the other resides on the network. Prior to Adaptive Server 11.5, the only location for storing network service information (server name, service type, and network protocol, device, and address) was the SQL Server interfaces file.

Adaptive Server 11.5 provides an alternative to using the traditional interfaces file by supporting the ability to connect to network service information through third-party directory service providers.

The advantage of using a directory service is that when your network is changed (for example, when a new server is added or a server address is changed), you do not need to update multiple interfaces files. You need to update only the directory service being used by the servers and clients.

Table 4-9 shows the directory service provider and the platform it supports in Adaptive Server 11.5.

*Table 4-9: Directory service provider*

<b>Directory Service Provider</b>	<b>Platform</b>
Windows Registry	Microsoft Windows NT

If Adaptive Server is not configured to use a directory service provider, it defaults to the Sybase interfaces file for obtaining network service information.

For information on how to configure Adaptive Server for directory services, see the Adaptive Server configuration guide for your platform.

## Engine Affinity and Execution Precedence

The new Logical Process Manager in release 11.5 gives you a measure of control over the order in which Adaptive Server executes requests from connections and sessions. Two new capabilities, engine affinity and execution precedence, offer system procedures that let you tell Adaptive Server the performance trade-offs you want to make among individual connections and sessions in mixed work-load environments.

For example, you can rank a client application, login, or stored procedure as preferred with respect to others using system procedures to assign *execution attributes*. Adaptive Server will consider your execution assignments when it places the entity in one of three priority run queues. In addition, you can use *engine affinity* to make suggestions about how to partition Adaptive Server engine resources among connections and sessions.

In a multiple-application environment, establishing execution precedence can improve performance for selected applications and “on the fly” for some running Adaptive Server processes.

You establish which client applications, logins, and stored procedures receive execution precedence when you create a hierarchy among *execution objects* (client applications, logins, and stored procedures) using a new set of Adaptive Server system procedures (see Table 4-10). Adaptive Server uses the hierarchy as a guideline for distributing processing resources among execution objects.

For more information, see Chapter 22, “Distributing Engine Resources Between Tasks,” in the Performance and Tuning Guide.

## New System Procedures

You implement and manage execution hierarchy for client applications, logins, and stored procedures using the five categories of system procedures listed in Table 4-10.

**Table 4-10: New system procedures for managing execution precedence**

Category	System Procedure	Description
User-defined execution class	<ul style="list-style-type: none"> <li>sp_addexclass</li> <li>sp_dropexclass</li> </ul>	Creates or drops a user-defined class with custom attributes or changes the attributes of an existing class.
Execution class binding	<ul style="list-style-type: none"> <li>sp_bindexclass</li> <li>sp_unbindexclass</li> </ul>	Binds or unbinds both predefined and user-defined classes to client applications and logins.



<b>Category</b>	<b>System Procedure</b>	<b>Description</b>
"On the fly"	<ul style="list-style-type: none"><li>• sp_setpsex</li><li>• sp_clearpsex</li></ul>	Sets or clears attributes of an active session only.
Engines	<ul style="list-style-type: none"><li>• sp_addengine</li><li>• sp_dropengine</li></ul>	Adds engines to or drops engines from engine groups; creates and drops engine groups.
Reporting	<ul style="list-style-type: none"><li>• sp_showcontrolinfo</li><li>• sp_showexeclass</li><li>• sp_showpsex</li></ul>	Reports on engine group assignments, application bindings, and execution class attributes.

## Expanded Error Logging

Users can now exercise control over whether certain messages are always logged to the Adaptive Server error log.

By setting the **log audit logon success** or **log audit logon failure** configuration parameter, you can enable or disable the logging of successful or unsuccessful Adaptive Server logins.

You can also specify that a particular message, identified by its message identifier, should be logged, using the **with\_log** parameter to the **sp\_addmessage** or **sp\_altermessage** system procedure. See the Adaptive Server Reference Manual for information about these system procedures.

These features affect logging to the Adaptive Server error log on all platforms, as well as logging to the Windows NT Event Log on Windows NT servers, if **event logging** is enabled.

## New System Procedures

Table 4-11 lists the new system procedures that support the expanded error logging feature.

*Table 4-11: New system procedures for expanded error logging*

System Procedure	Description
sp_altermessage	Enables and disables the logging of a system-defined or user-defined message in the Adaptive Server error log.
sp_configure	
"log audit logon success"	Specifies whether to log successful Adaptive Server logins to the Adaptive Server error log and, on Windows NT servers, to the Windows NT Event Log, if event logging is enabled.
"log audit logon failure"	Specifies whether to log unsuccessful Adaptive Server logins to the Adaptive Server error log and, on Windows NT servers, to the Windows NT Event Log, if event logging is enabled.

## Modified System Procedures

The following system procedures have been modified to support the expanded error logging feature by using the **with\_log** parameter:

- **sp\_addmessage**
- **sp\_addserver**

## **Extended Stored Procedures (ESPs)**

This release supports both user-defined and system-defined extended stored procedures (ESPs).

ESPs provide a method for calling procedural language functions from within Adaptive Server. The procedural language in which the functions are written must be capable of calling C language functions and manipulating C data types.

ESPs allow Adaptive Server to perform a task outside Adaptive Server in response to an event occurring within Adaptive Server. For example, a procedural task outside the relational database management system (RDBMS), such as selling a security, could be invoked by a trigger on a database table that fires when the value of the security reaches a certain value. Or an email notification or a network-wide broadcast could be sent in response to an event occurring within the RDBMS.

The interface to ESPs is similar to the interface to system procedures and user-defined stored procedures. The difference is that an ESP executes procedural language code, rather than Transact-SQL statements.

## **XP Server**

Extended stored procedures are implemented by an Open Server™ application called XP Server, which runs on the same machine as Adaptive Server. Running ESPs in a separate process protects Adaptive Server from failing because of faulty ESP code. Adaptive Server and XP Server communicate through remote procedure calls (RPCs).

The function that implements the ESP is compiled and linked into a dynamic link library (DLL) or a shared library. Adaptive Server looks in the system tables for the function that has the same name as the requested ESP and passes the function name and the DLL name to XP Server.

XP Server is responsible for:

- Loading the DLL, if it is not already loaded.
- Invoking the function that implements the ESP.
- Passing the function's return status, output parameters, and results back to Adaptive Server.

Adaptive Server passes the results of the function to the client.

Adaptive Server starts XP Server on the first ESP request and shuts down XP Server when Adaptive Server shuts down.

The installation or configuration program for your platform adds the default `<hostname>_XP` entry to the interfaces file and the `sys.servers` table. Additional XP Server entries must be added to the interfaces file using `dsedit`. Additional XP Server entries to the `sys.servers` table must be manually added. There is no `run_server` file associated with XP Server.

See the Transact-SQL User's Guide for detailed information about using and developing your own ESPs.

To develop your own ESPs, you need to purchase the Open Server product.

## System-Defined ESPs

Table 4-12 lists the system-defined ESPs.

**Table 4-12: System-defined ESPs**

System Procedure	Description
<code>xp_cmdshell</code>	Executes a native operating system command on the host system running Adaptive Server.
<code>xp_enumgroups</code>	Displays groups for a specific Windows NT domain. (Windows NT only)

In addition, several Windows NT-specific ESPs provide Windows NT features, such as event logging and Adaptive Server mail support.

See Chapter 5, "System Extended Stored Procedures," in the *Adaptive Server Reference Manual* for a complete list of system ESPs.

## New System Procedures for ESPs

Table 4-13 lists the system procedures that support ESPs.

**Table 4-13: New system procedures for ESP**

System Procedure	Description
<code>sp_addextendedproc</code>	Creates an ESP.
<code>sp_configure</code>	
"esp execution priority"	Sets the priority of the XP Server thread for ESP execution.

<b>System Procedure</b>	<b>Description</b>
"esp execution stacksize"	Sets the size of the stack, in bytes, to allocate for ESP execution.
"esp unload dll"	Specifies whether DLLs that support ESPs should automatically be unloaded from XP Server memory after the ESP call has completed.
"xp_cmdshell_context"	Sets the security context for the operating system command to be executed using the xp_cmdshell system ESP.
sp_dropextendedproc	Drops an ESP.
sp_freedll	Unloads a DLL that was loaded into XP Server memory to support the execution of an ESP.
sp_helpextendedproc	Displays ESPs in the current database with their associated DLL files.

## **Supporting Commands**

The following Transact-SQL commands have been enhanced to support ESPs:

- **create procedure**
- **drop procedure**
- **execute**

## Increased Referential Integrity Limits

The referential integrity relaxed limit feature increases the maximum number of table references allowed in a query from 16 to 192. The new limit eliminates the need to write special triggers to compensate for the previous 16-table reference limitation.

When a table has references to other tables, Adaptive Server scans the referenced tables when the table is being modified. Adaptive Server uses *scan descriptors* to manage the table scans. A scan descriptor is an internal data structure that manages the scan of the tables being queried. To make efficient use of Adaptive Server resources, you can configure Adaptive Server to use an appropriate number of scan descriptors for your system.

For more information on using referential integrity, see the *Transact-SQL User's Guide*.

For information on configuring scan descriptors, see the **number of aux scan descriptors** configuration parameter in Chapter 11, "Setting Configuration Parameters," of the *System Administration Guide*.

## Metadata Cache Management

Adaptive Server allows you to manage individual metadata caches for:

- User indexes
- Objects such as procedures, triggers, views, rules, and defaults
- Databases

Managing individual metadata caches for these objects is beneficial for a server that contains a large number of user indexes and objects and where there is high concurrency among users.

A *metadata cache* is a reserved area of memory for tracking information for indexes, objects, or databases. When you configure a metadata cache for indexes, objects, or databases, Adaptive Server accesses the information directly from the in-memory structure that describes them in the *sysindexes*, *sysobjects*, or *sysdatabases* row.

This method improves performance because Adaptive Server bypasses expensive calls that require disk access. Configuring metadata caches also improves performance by reducing synchronization and spinlock contention when Adaptive Server has to retrieve index, object, or database information at run time.

For more information on configuring metadata caches, see “Metadata Caches” on page 11-61 in the System Administration Guide.



## Monitor Access to SQL Batch Text

The monitor access to SQL batch text feature allows a user with the System Administrator role to display the SQL batch text of any client command that is running on Adaptive Server. Although you configure Adaptive Server to save the SQL batch text, the batches are viewable only through Adaptive Server Monitor's Server.

Monitor access includes **sp\_showplan**, which allows you to view the query plan for the query that is currently running. Monitor Server is not required to retrieve the query plans.

Both monitor access and **sp\_showplan** are particularly useful for determining why a client session is hanging or using large amounts of CPU time or physical I/O.

## Allocating Memory for Batch Text

Monitor access to SQL batch text is turned off by default when Adaptive Server is installed, so you must configure Adaptive Server to allocate memory for this feature. Consider the following to help you determine how much memory to allocate per user:

- SQL batches exceeding the allocated amount of memory are truncated without warning. Therefore, if you do not allocate enough memory for the batch statements, the text you want to view might be the section of the batch that is truncated.
- The more memory you allocate for SQL text from shared memory, the less chance there is that the problem statement will be truncated from the batch copied to shared memory. However, the memory you allocate for SQL text reduces the amount of memory for data and procedure caches.

## New Columns in *sysprocesses*

The procedure you are viewing may be nested in a batch of SQL text. To aid in the viewing of nested procedure, new columns have been added to the *sysprocesses* table to specify the object ID, current statement number and line number of the current statement of the procedure that is currently running.

Table 4-14 shows the new columns added to the *sysprocesses* table.

**Table 4-14: New columns in sysprocesses**

<b>Column Name</b>	<b>Datatype</b>	<b>Specifies</b>
<i>id</i>	integer	The object ID of the running procedure; if 0, no procedure is running.
<i>linenum</i>	integer	The line number of the current statement within the running stored procedure, or the line number of the current SQL batch statement, if no procedure is running.
<i>stmtnum</i>	integer	The current statement number within the running procedure, or the SQL batch statement number, if no procedure is running.

For more information, see “Configuring Adaptive Server to Save SQL Batch Text” on page 4-17 of the System Administration Guide.

## Network-Based Security

Adaptive Server provides network-based security services that enable you to authenticate users and protect data that is transmitted among machines on a network.

In a distributed client/server computing environment, intruders can view or tamper with confidential data. In Adaptive Server 11.5, you can use security services provided by third-party providers. These security services allow you to:

- Authenticate users, clients, and servers by ensuring that they are who they say they are
- Provide data confidentiality with encryption by ensuring that the data cannot be read by an intruder
- Provide data integrity by detecting when data tampering has occurred

The network-based security mechanism supported by Adaptive Server is Windows NT LAN Manager.

## Security Services

Adaptive Server allows you to use one or more of these network-based security services:

- Unified login – authenticates users *once* without requiring them to supply a name and password every time they log into Adaptive Server.
- Message confidentiality – encrypts data over the network.
- Mutual authentication – verifies the identity of the client and the server.

---

**Note** This service must be requested by the client and cannot be required by Adaptive Server.

---

- Message integrity – verifies that data communications have not been modified.
- Replay detection – verifies that data has not been intercepted by an intruder.
- Out-of-sequence check – verifies the order of data communications.
- Message origin checks – verifies the origin of the message.

- Remote procedure security - establishes mutual authentication, message confidentiality, and message integrity for remote procedure communications.

---

**Note** The security mechanism you are using may not support some of these services.

---

For more information about network-based security, see the Security Administration Guide.

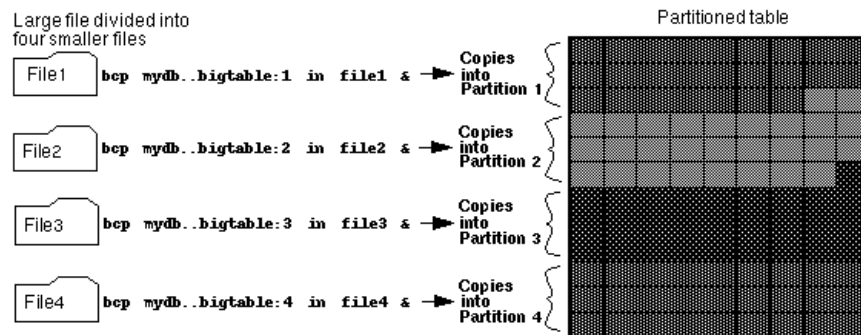
## Parallel Bulk Copy

Parallel bulk copy allows you to copy data in parallel to Adaptive Server from multiple files. Parallel bulk copy substantially increases performance during **bcp** sessions because large bulk copy jobs can be split into multiple sessions and run concurrently.

Before you copy data into your database using parallel bulk copy, you must first partition the table that will contain the data. You can then issue multiple **bcp** sessions, with each session copying its data from a different source file into a different partition. If you are copying data from one, large operating system file, you divide this file into smaller files and then start multiple parallel bulk copy sessions.

Figure 4-2 illustrates how parallel bulk copy works.

*Figure 4-2: Copying data into a partitioned table using parallel bcp*



You can also use the **-F** and **-L** flags to designate the first and last row of the same host file for each of the multiple parallel bulk copy sessions.

For more information about parallel bulk copy, see **bcp** in *the Utility Programs manual for your platform*.

## Parallel Queries and Enhanced Partitioning

When you configure Adaptive Server for parallel query processing, the optimizer evaluates each query to determine whether it is eligible for parallel execution. If it is, the query is divided into components that process simultaneously. The results are combined and delivered to the client in a shorter period of time than it would take to process the query serially as a single component.

For the same throughput, processing a query in parallel requires more work from Adaptive Server and additional resources than processing a query in serial. It also involves evaluating more complex trade-offs to achieve optimal performance. Fully enabled parallel query processing requires multiple processes, engines, and partitions, resulting in increased overhead for Adaptive Server, additional CPU requirements, and increased disk I/O.

You can configure various levels of parallelism, each providing a performance gain and requiring corresponding trade-offs in physical resources. Chapter 13, “Introduction to Parallel Query Processing,” in the Performance and Tuning Guide introduces the Adaptive Server parallel query processing model and concepts. It also discusses the trade-offs between resources and performance gains for different levels of parallelism.

For More Information

In the Performance and Tuning Guide, see:

- Chapter 5, “Locking in Adaptive Server,” for information on how Adaptive Server supports locking for parallel query execution.
- Chapter 9, “Understanding Query Plans,” for information on new **showplan** messages added for parallel query execution.
- Chapter 14, “Parallel Query Optimization,” for details on how the Adaptive Server optimizer determines eligibility for parallel execution.
- Chapter 15, “Parallel Sorting,” to understand parallel sorting topics.
- Chapter 17, “Controlling Physical Data Placement,” for information on partitioned tables, creating clustered indexes on partitioned tables, and parallel processing.

In the System Administration Guide, see:

- Chapter 11, “Setting Configuration Parameters,” for details on how you can configure various levels of parallelism, each providing a performance gain and requiring corresponding trade-offs in physical resources.

## Point-in-Time Recovery

Point-in-time recovery allows you to recover a database by rolling it forward to a particular time in its transaction log. You do this by using the **until\_time** option of the **load transaction** command.

This feature is useful if, for example, a user inadvertently drops an important table from a database; you can use the **until\_time** option to recover the changes to that database up to a time just before the table was dropped.

You can use point-in-time recovery for any database that can be loaded and dumped. However, you cannot use it for databases in which the data and logs are on the same device, such as the *master* database or for any database, such as *tempdb*, whose log has been truncated since the last **dump database**.

For more information on point-in-time recovery, see “Loading a Transaction Log to a Point in Time” on page 21-44 in the *System Administration Guide* and the **load transaction** command in the *Adaptive Server Reference Manual*.

## Proxy Authorization

Adaptive Server provides the proxy authorization capability, which allows one user to assume the identity of another user on a server-wide basis. A System Security Officer can grant the ability to assume the security context of another user to selected logins. If a login has permission to use proxy authorization, the login can impersonate any other login in Adaptive Server.

---

**Warning!** The ability to assume another user's identity is extremely powerful and must be strictly limited. A user with this permission could assume the identity of the "sa" login, which would give the user unlimited power within Adaptive Server. Plan to limit this permission to trusted administrators and applications, and to audit their server activity.

---

A System Security Officer or a System Administrator might want to assume the permissions of another user to make sure the permissions are correct for a user or to perform maintenance on a user's database objects.

An application server can use proxy authorization to login to Adaptive Server with a generic login, which the application server uses to execute procedures and commands for several users.

A System Security Officer uses the **grant set proxy** or **grant set session authorization** command to give a user permission to use proxy authorization. The user with this permission can then execute either **set proxy** or **set session authorization** to become another user. A user executing **set proxy** or **set session authorization** operates with both the login and server user ID of the user being impersonated. The login and server user ID are active across the entire server in all databases.

---

**Note** **set proxy** and **set session authorization** are identical in function and can be used interchangeably. The only difference between them is that **set session authorization** is SQL92 compatible, and **set proxy** is a Transact-SQL extension.

---

For more information about proxy authorization, see the Security Administration Guide.



## Recovery Fault Isolation

Prior to this release, when recovery detected corruption in a database, it would make the entire database inaccessible. The database remained unavailable to users until the suspect pages had been repaired or removed from the database.

With recovery fault isolation, a System Administrator has the option of isolating the corrupt pages marked suspect by recovery while the rest of the database is brought online and made available to users.

Recovery fault isolation provides a System Administrator the ability to:

- Specify whether an entire database or just the suspect pages become inaccessible when recovery detects corruption
- Specify whether a database with suspect pages is brought online in **read\_only** mode
- List databases that have suspect pages
- List suspect pages by page ID, index ID, and object name
- Bring suspect pages online for the System Administrator while they are being repaired
- Bring suspect pages online for all database users after they have been repaired

The ability to isolate the suspect pages while bringing the rest of the database online provides a greater degree of flexibility in dealing with data corruption. Problems can be diagnosed, and sometimes corrected, while most of the database is accessible to users. The extent of the damage can be assessed and emergency repairs can be scheduled for a convenient time.

A System Administrator can force offline pages online temporarily, for System Administrator use only, to aid in the analysis and repair of the suspect pages.

Recovery fault isolation applies only to user databases. System databases are always taken entirely offline if they have any suspect pages.

## New System Procedures

Table 4-15 lists the system procedures that support recovery fault isolation:

*Table 4-15: New system procedures for recovery fault isolation*

<b>System Procedure</b>	<b>Description</b>
sp_forceonline_db	Provides access to all the suspect pages in a database.

System Procedure	Description
sp_forceonline_page	Provides access to an individual suspect page.
sp_listsuspect_db	Lists all databases that have pages marked suspect by recovery.
sp_listsuspect_page	Lists all pages in a database that are marked suspect by recovery.
sp_setsuspect_granularity	Displays and sets the recovery fault isolation mode. This mode governs whether recovery marks an entire database or only the corrupt pages suspect.
sp_setsuspect_threshold	Sets the maximum number of suspect pages that recovery will allow in a database before marking the entire database suspect.

## Side Effects of Recovery Fault Isolation

- Transactions that attempt to access offline data directly or indirectly will fail.
- You cannot use **dump database** on a database that has offline pages. This prevents users from dumping a corrupt database.
- You cannot use **dump transaction with no\_log** or **dump transaction with truncate\_only** on a database that has offline pages.

For more information on recovery fault isolation, see “Fault Isolation During Recovery” on page 20-7 in the System Administration Guide.

## Relaxed LRU Cache Replacement Policy

Relaxed LRU (least recently used) cache replacement policy is a performance feature designed to reduce overhead in named caches where little or no buffer replacement is occurring.

Relaxed LRU replacement policy removes the cache linkage overhead required by caches maintained on an MRU/LRU (most recently used/least recently used) basis. Relaxed LRU replacement policy is meant to be used when both of the following conditions are true:

- There is little or no replacement of buffers in the cache, and
- The data is not updated or is updated rarely.

In addition to saving the overhead to maintain the MRU/LRU chain, relaxed LRU replacement policy can improve CPU performance on multiple-CPU servers.

See Chapter 9, “Configuring Data Caches,” in the System Administration Guide for information on configuring caches to use LRU cache replacement policy.

See Chapter 16, “Memory Use and Performance,” in the Performance and Tuning Guide for information on how to select caches where relaxed LRU replacement policy can improve performance.

## Resource Limits

Adaptive Server provides resource limits to help System Administrators prevent queries and transactions from monopolizing server resources. A *resource limit* is a set of parameters specified by a System Administrator to prevent an individual login or application from:

- Exceeding estimated or actual I/O costs, as determined by the optimizer
- Returning more than a set number of rows
- Exceeding a given elapsed time

The set of parameters for a resource limit includes the time of day to enforce the limit and the type of action to take. For example, you can prevent huge reports from running during critical times of the day or kill a session whose query produces unwanted **Cartesian products**.

To specify times of the day or week, you can create named time ranges. A *time range* is a contiguous block of time within a single day across one or more contiguous days of the week. It is defined by its starting and ending periods. Contiguous days of the week are allowed to wrap around the end of one week to the beginning of the next.

Adaptive Server includes one predefined time range, the “at all times” range, which covers the period of midnight to midnight, Monday through Sunday. You can create, modify, and drop additional time ranges as necessary for resource limits.

For more information on resource limits, see Chapter 12, “Limiting Access to Server Resources,” in the System Administration Guide.

## New System Procedures

The system procedures in Table 4-16 support the resource limits feature.

**Table 4-16: New system procedures for resource limits**

System Procedure	Description
sp_add_resource_limit	Creates a resource limit.
sp_add_time_range	Creates a named time range.
sp_configure	

<b>System Procedure</b>	<b>Description</b>
"allow resource limits"	Signals the server to allocate internal memory for time ranges, resource limits, and internal server alarms. It also signals the server to internally assign applicable ranges and limits to user sessions. When this option is set to 1 (on), <code>sp_configure</code> displays the optimizer's cost estimate for a query. The default is 0 (off).
<code>sp_drop_resource_limit</code>	Drops a resource limit.
<code>sp_drop_time_range</code>	Drops a named time range.
<code>sp_help_resource_limit</code>	Displays resource limits in the current server.
<code>sp_modify_resource_limit</code>	Modifies a resource limit.
<code>sp_modify_time_range</code>	Modifies a named time range.

## Supporting Commands

The following commands support the resource limits feature:

- **set showplan**
- **set statistics io**
- **statistics time**
- **select @@rowcount**

## Source Text Verification and Encryption

In releases prior to Adaptive Server release 11.5, the **source text of compiled objects** was saved in *syscomments* only so that it could be returned to a user who executed **sp\_helptext**. Since this was the only reason for saving the source text, users often deleted it to save disk space and to remove confidential information from this public area.

In Adaptive Server release 11.5, you should not remove the source text from *syscomments*. Deleting the source text from *syscomments* will cause problems when upgrading to the next release of Adaptive Server. The upgrade process in future releases of Adaptive Server will require that the source text of compiled objects be present in *syscomments* at the time of the upgrade. Before upgrading to the next release of Adaptive Server, you need to restore missing source text to *syscomments*.

You can use the **sp\_checksource** system procedure to verify that source text is present in *syscomments* for each compiled object. To prevent users from seeing the source text in *syscomments*, you can encrypt the text with the **sp\_hidetext** system procedure.

For more information, see “Compiled Objects” on page 1-3 in the Transact-SQL User’s Guide. Also see **sp\_checksource** and **sp\_hidetext** in the Adaptive Server Reference Manual.

## Sybase Central

With the release of Adaptive Server, Sybase no longer supplies SQL Server Manager, which has been replaced by Sybase Central™, Sybase's common management interface, and the Adaptive Server Enterprise plug-in for Sybase Central.

The Adaptive Server plug-in for Sybase Central allows you to manage Adaptive Server installations using the Sybase Central graphical management tool. You can manage Adaptive Server running on any platform from a PC that is running Windows 95 or Windows NT.

To learn how to use Sybase Central to manage Adaptive Server, see *Managing and Monitoring Sybase Adaptive Server Enterprise*.

## ***sp\_sysmon***

**sp\_sysmon** supports the new performance features in release 11.5, including the addition of these new sections in the **sp\_sysmon** report:

- Application Queue Management
- Asynchronous Prefetch Usage
- Metadata Cache Management
- Monitor Access to Executing SQL
- Parallel Query Management
- Worker Process Management

The syntax for **sp\_sysmon** has been changed:

- You can specify the interval using the form “hh:mm:ss”.
- You can use the **begin\_sample** argument to start **sp\_sysmon** which continues to monitor until you use **end\_sample** to print the report.
- You can print only one section of the report at a time. This example specifies an interval of 15 minutes and prints only the Data Cache Management section:

```
sp_sysmon "00:15:00, dcache
```

See Chapter 24, “Monitoring Performance with **sp\_sysmon**,” in the Performance and Tuning Guide for more information.



## Two-Phase Commit Enhancements

The two-phase commit protocol allows client applications to coordinate transaction updates across two or more Adaptive Servers and treats those transactions as a single transaction. Two-phase commit guarantees that either all or none of the databases in the participating Adaptive Servers are updated.

The enhancements to two-phase commit are:

- Improved space and log management – The *spt\_committab* table has been moved from the *master* database to a new system database, *sybssystemdb*.
- Improved performance – You can use a random value for the commit identifier—*commid*.

### ***spt\_committab* Moved to a New System Database (*sybssystemdb*)**

The *spt\_committab* table in Adaptive Server stores information and tracks the completion status of each two-phase commit transaction. All activities made against *spt\_committab* are logged. Prior to 11.5, *spt\_committab* resided in the *master* database. However, the logging activity against *spt\_committab* for two-phase commit transactions can fill up the master transaction log, causing frequent space management problems.

In release 11.5, the *spt\_committab* table resides in a new system database called *sybssystemdb*, instead of the *master* database. Providing *spt\_committab* its own database makes space management and recovery of two-phase commit transactions easier, for two reasons:

- The **dump tran** command can be used against *sybssystemdb*.
- Unlike the *master* database, the *sybssystemdb* database is extensible.

### **Commit Identifier Changed to a Random Value**

The value of the commit identifier, *commid* in the *spt\_committab* table, becomes a random value (between 1 and 2,147,483,647 ( $2^{31}-1$ )), instead of a monotonically increasing value. This reduces page lock contention on the row where the maximum *commid* resides when several distributed transactions try to insert an entry into the *spt\_committab* table. *spt\_committab* continues to have a unique, clustered index.

For information on how to configure Adaptive Server for two-phase commit functionality, see the Adaptive Server configuration documentation for your platform.

## User-Defined Roles

The user-defined roles feature allows a System Security Officer to simplify the task of granting and preventing access to data.

The System Security Officer can create roles, request object owners to grant privileges to each role, and grant the user-defined roles to individual employees, based on the functions they perform in the organization. The System Security Officer can also revoke user-defined roles granted to the employee.

For more information on user-defined roles, see the *Security Administration Guide*.

## Windows NT Event Log Integration

This release has expanded support for logging Adaptive Server errors in the Windows NT Event Log. This is in addition to automatic logging in the Adaptive Server error log. This option is available on Windows NT servers only.

By default, Windows NT Event Logging of Adaptive Server messages is enabled, but you can disable it by setting the **event logging** configuration parameter to 0. You can also specify a remote Windows NT computer to receive event logging by setting the **event log computer name** configuration parameter to the name of the remote computer.

This feature provides an extended stored procedure that you can use to log user-defined events to the Windows NT Event Log from within Adaptive Server. See **sp\_logdevice** in the Adaptive Server Reference Manual.

## Supporting System ESPs and System Procedures

The following system ESP and **sp\_configure** parameters support the Windows NT Event Log Integration feature.

*Table 4-17: System support for Windows NT Event Log Integration*

Procedure	Description
xp_logevent	A system-defined ESP that provides for logging a user-defined event in the Windows NT Event Log.
sp_configure	
"event log computer name"	Specifies the name of the Windows NT PC that logs Adaptive Server messages in its Windows NT Event Log.
"event logging"	Enables and disables the logging of Adaptive Server messages in the Windows NT Event Log.

## Windows NT MAPI Support

Sybase Adaptive Server Enterprise for Windows NT can send and receive email messages using Microsoft's Message Application Programming Interface (MAPI). This facility, called Sybmail, is currently available only on Windows NT.

Adaptive Server can send outgoing messages that consist of text or query results. Adaptive Server can receive queries as incoming messages.

To use Sybmail, Adaptive Server must have a Windows NT Mail account and a Windows NT Mail profile. Sybmail must also have a login in Adaptive Server.

See *Configuring Adaptive Server for Windows NT* for information on setting up and using Sybmail.

## Supporting System ESPs and System Procedures

The following system ESPs and system procedures support the Windows NT MAPI feature:

*Table 4-18: System support for Windows NT MAPI*

Procedure	Description
sp_configure	
"start mail session"	Enables and disables the automatic initiation of a Adaptive Server mail session when Adaptive Server is started.
sp_processmail	Reads, sends, and deletes messages in the Adaptive Server message inbox. Available on Windows NT only.
xp_deletemail	Deletes a message from the Adaptive Server message inbox.
xp_findnextmsg	Retrieves the message identifier of the next message in the Adaptive Server message inbox.
xp_readmail	Reads a message from the Adaptive Server message inbox.
xp_sendmail	Sends a message to the specified recipients using the MAPI interface.
xp_startmail	Starts an Adaptive Server mail session.
xp_stopmail	Stops an Adaptive Server mail session.

For more information on the system-defined ESPs, see Chapter 5, "System Extended Stored Procedures," in the Adaptive Server Reference Manual.

## Windows NT Performance Monitor Integration

You can enable and disable the ability to monitor Adaptive Server statistics from the Windows NT Performance Monitor by setting the **SQL Perfmon Integration** configuration parameter. This parameter applies to Windows NT servers only.

Adaptive Server must be registered as an NT Service to support performance monitor integration. This occurs automatically when you start Adaptive Server using Sybase Central, the Services Manager, or the Services option in the Control Panel or when you have configured Windows NT to start Adaptive Server as an automatic service.

For more information on Windows NT Performance Monitor, see Chapter 9, “Administrative Tasks and Performance and Tuning,” in Configuring Adaptive Server for Windows NT.

### Supporting System Procedure

The following **sp\_configure** configuration parameter supports the Windows NT Performance Monitor Integration feature:

*Table 4-19: System support for Windows NT Performance Monitor*

Procedure	Description
sp_configure	
"SQL Perfmon Integration"	Enables and disables the ability to monitor Adaptive Server statistics from the Windows NT Performance Monitor.

## **New Features in SQL Server Release 11.0**

This chapter describes the features and system changes introduced in SQL Server release 11.0. If you are upgrading to release 11.5 from a pre-11.0 release, these features will be new to you.

Topics covered are:

- “New Features in Release 11.0” on page 114
- “System Changes in Release 11.0” on page 131
- “Changes That May Affect Existing Applications” on page 136

## New Features in Release 11.0

The new features in release 11.0 are as follows:

- “User-Defined Caches” on page 114
- “Data Storage Changes” on page 117
- “Transaction Log Changes” on page 120
- “Isolation Level 0” on page 121
- “Lock Manager Changes” on page 122
- “Housekeeper Task” on page 124
- “SQL Server Configuration” on page 125
- “Lock Escalation” on page 125
- “Multiple Network Engines” on page 126
- “Improvements to showplan” on page 126
- “Query and Data Modification Changes” on page 127
- “Upgrading Database Dumps” on page 128
- “Tape Device Determination by Backup Server” on page 129
- “IDENTITY Column Changes” on page 129
- “New text and image Global Variables” on page 130

### User-Defined Caches

Release 11.0 allows System Administrators to split the SQL Server data cache into separate named data caches and to bind databases or database objects to those caches. Also, System Administrators can create pools within caches that perform large I/O to disk, improving performance for many queries.

### Configuring Caches

Named data caches are created with the new system procedure **sp\_cacheconfig**. Memory pools for large I/O are configured within data caches with **sp\_poolconfig**. Configuring caches and pools can also be done by editing a configuration file.



Entities are bound to caches with **sp\_bindcache** and unbound with **sp\_unbindcache** or **sp\_unbindcache\_all**. The system procedure **sp\_helpcache** provides information about caches and cache bindings. **sp\_cacheconfig** also provides information about caches and the pools within caches.

See Chapter 9, “Configuring Data Caches,” in the *System Administration Guide* for instructions on configuring named caches and binding database objects to them.

### Binding Objects to User-Defined Caches

In general, you can increase performance by binding the following objects to their own named caches:

- *sysindexes* table and its index

Bind these two objects to the same named cache. This is the most frequently used table and its index. However, you do not have to make its named cache very large.

- *syslogs* table

This helps reduce the contention on the buffer manager spinlocks. You should also configure a 4K memory pool in its named cache to benefit from the larger log I/O size, as defined by **sp\_logiosize**.

- *tempdb* database

This helps performance if many temporary or work tables are generated by your applications.

In addition to the above objects, the most frequently used tables and indexes (based on your database and application designs) are good candidates for binding to their own caches. However, if you have only one table with no index, binding that table to a named cache will not improve performance.

---

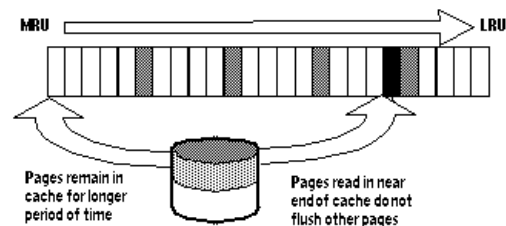
**Note** Having too many named caches without increasing the total memory available to your server is generally not a good idea. Each additional named cache reduces the amount of memory available in the default cache.

---

## Cache Strategies

SQL Server release 11.0 provides new optimization strategies and new commands to control the use of those strategies for objects, sessions, and queries. Figure 5-1 shows the two cache strategies.

**Figure 5-1: Cache strategy choices**



SQL Server can:

- Read pages in at the start of the MRU (most recently used) chain in order to keep the pages in cache so that they can be accessed many times without performing additional I/O
- Read pages in near the end of the cache so that they will not force other pages out of the cache

See Chapter 16, “Memory Use and Performance,” in the *Performance and Tuning Guide* for performance information on cache strategies.

## Large I/O

When caches are created, all space is assigned to a 2K pool. This pool can be split to allow the cache to perform large I/O, reading up to eight data pages at once. Since most I/O time is spent performing queueing, seeking, and positioning operations, large I/O can greatly increase performance for queries that scan entire tables or ranges of tables.

SQL Server can also perform large I/O on the transaction log.

## Changing the Log I/O Size

Using the new **sp\_logiosize** system procedure, you can change the log I/O size used by SQL Server. This can improve performance by reducing the number of times that SQL Server writes transaction log pages to disk. The value you specify for **sp\_logiosize** must correspond to an existing memory pool configured for the cache used by the database’s transaction log.

By default, SQL Server defines the log I/O size of a database as 4K.

If the transaction log for a database is bound to the default cache or a user-defined cache that does not contain a 4K memory pool, SQL Server sets the log I/O size to 2K (a 2K memory pool is always present in any cache). For most workloads, a log I/O size of 4K performs much better than one of 2K, so each cache used by a transaction log should have a 4K memory pool.

## New Query Tuning Options

New query tuning options support the capabilities of named caches and provide additional control when tuning queries:

- The **set forceplan on** command forces SQL Server to join the tables in a query in the order in which they are listed in the **from** clause.
- The **set table count** command allows you to specify the number of tables that are considered at once as joins are optimized.
- The new system procedure **sp\_cachestrategy** disables and reenables cache strategies and large I/O for individual objects.
- New clauses for **select**, **delete**, and **insert** commands allow you to specify the index, cache strategy, or I/O size for a query.
- The **set prefetch** command allows you to enable or disable large I/O for a session.

See Chapter 10, “Advanced Optimizing Techniques,” in the *Performance and Tuning Guide* for information on query processing options that control caching strategies and I/O size.

## Data Storage Changes

This section describes changes to the way SQL Server stores data.

### Maximum Number of Rows per Page

You can now configure the maximum number of rows stored on a data page or index leaf page. The **max\_rows\_per\_page** value specified in a **create table**, **create index**, or **alter table** command restricts the number of rows allowed on a data page, a clustered index leaf page, or a nonclustered index leaf page. This reduces lock contention and improves concurrency for frequently accessed tables.

The **max\_rows\_per\_page** value applies to the data pages of an unindexed table or the leaf pages of an index. The value of **max\_rows\_per\_page** is stored in the *maxrowsperpage* column of *sysindexes*. (The *maxrowsperpage* column was named *rowpage* in previous releases of SQL Server.)

Unlike **fillfactor**, which is not maintained after creating a table or index, SQL Server retains the **max\_rows\_per\_page** value when adding or deleting rows. For information about how to use **max\_rows\_per\_page** to reduce lock contention and improve concurrency for your server, see the *Performance and Tuning Guide*.

## Page Allocation

SQL Server 10.0 and earlier releases searched the OAM page chain for unused pages before allocating new extents to objects. Release 11.0 provides a system-wide configuration parameter, **page utilization percent**, that allows SQL Server to allocate a new extent to an object without searching the OAM page chain, depending on the ratio of used and unused pages in the table to the table's allocated extents.

## Partitioned Tables

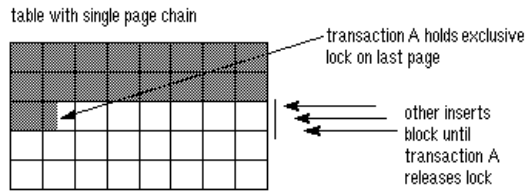
---

**Note** 11.5 Note: Release 11.5 adds the ability to scan partitioned tables in parallel and to create clustered indexes.

---

By default, SQL Server stores a **heap table**'s data in one double-linked chain of database pages. When a transaction inserts a row into the table, it holds an exclusive page lock on the last page of the page chain while inserting the row. As multiple transactions attempt to insert rows into the same table at the same time, performance problems can occur. Because only one transaction at a time can obtain an exclusive lock on the last page, other, concurrent insert transactions block, as shown in Figure 5-2.

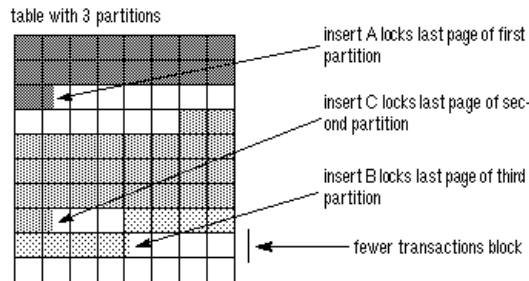
**Figure 5-2: Page contention during inserts**



In release 11.0, SQL Server provides the ability to partition heap tables. A **partition** is simply another term for a page chain. Partitioning a table creates multiple page chains (partitions) for the table and, therefore, multiple last pages for insert operations.

When a transaction inserts data into a partitioned table, SQL Server randomly assigns the transaction to one of the table's partitions. Concurrent inserts are less likely to block, since multiple last pages are available for inserts, as shown in Figure 5-3.

**Figure 5-3: Addressing page contention with partitions**



If a table's segment spans several physical disks, SQL Server distributes the table's partitions across those disks when you create the partitions. This can improve I/O performance when SQL Server writes the table's cached data to disk, since the I/O is distributed over several devices.

SQL Server release 11.0 manages partitioned tables transparently to users and applications. Partitioned tables appear exactly like unpartitioned tables, except when accessed via the **dbcc checktable** and **dbcc checkdb** commands or when viewed with the new **sp\_helppartition** procedure.

See Chapter 17, “Controlling Physical Data Placement,” of the *Performance and Tuning Guide* for information about how to create and administer partitioned tables.

## Transaction Log Changes

This section describes the changes made to the transaction log for release 11.0.

### User Log Caches

There is one user log cache for each configured user connection. SQL Server uses these user log caches to buffer the user transaction log records, which reduces the contention at the end of the transaction log. When a user log cache becomes full or when another event occurs (such as when the transaction completes), SQL Server “flushes” all log records from the user log cache to the database transaction log. You can configure the size of all user log caches for your server using the **user log cache size** parameter of **sp\_configure**; the default size is 2048 bytes.

Since more than one process can access the contents of a user log cache when determining a possible “flush,” SQL Server uses spinlocks to protect the user log caches. A spinlock is an internal locking mechanism that prevents a second process from accessing the data structure being used by another process. SQL Server uses the **user log cache spinlock ratio** parameter of **sp\_configure** to specify the ratio of user log caches per user log cache spinlock; the default is 20 user log caches for each spinlock.

For information about configuring **user log cache size** and **user log cache spinlock ratio**, see Chapter 11, “Setting Configuration Parameters,” in the *System Administration Guide*.

### *syslogshold* Table

You can query the new *syslogshold* system table to determine the oldest active transaction in each database. *syslogshold* exists in the *master* database, and each row in the table represents either:

- The oldest active transaction in a database or
- The Replication Server® truncation point for the database’s log.

A database may have no rows in *syslogshold*, a row representing one of the above, or two rows representing both of the above. For information about how Replication Server truncation points affects the truncation of a database's transaction log, see the Replication Server documentation.

Querying *syslogshold* can help you when the transaction log becomes too full, even with frequent log dumps. The **dump transaction** command truncates the log by removing all pages from the beginning of the log up to the page that precedes the page containing an uncommitted transaction record (the oldest active transaction). The longer this active transaction remains uncommitted, the less space is available in the transaction log, since **dump transaction** cannot truncate additional pages.

For information about how to query *syslogshold* to determine the oldest active transaction that is holding up your transaction dumps, see Chapter 21, "Backing Up and Restoring User Databases," in the *System Administration Guide*.

## Isolation Level 0

You can specify isolation level 0 for the queries in transactions along with the isolation levels 1 and 3 supported in release 10.0. Isolation level 0 prevents other transactions from changing data that has already been modified by an uncommitted transaction. The other transactions are blocked from modifying that data until the transaction commits. However, other transactions can still read the uncommitted data (known as *dirty reads*).

Queries executing at isolation level 0 do not acquire any read locks for their scans, so they do not block other transactions from writing to the same data (and vice versa). Applications that are not impacted by dirty reads may see better concurrency and reduced deadlocks when accessing the same data using isolation level 0. However, transactions that require data consistency should *not* use isolation level 0.

You can specify isolation level 0 for the transactions of a session as follows:

```
set transaction isolation level 0
```

Since this command makes all queries execute at isolation level 0, you should be wary of using it for any transaction that requires data consistency. Instead, you can selectively choose isolation level 0 for a query in a transaction using the **at isolation** clause as follows:

```
select *  
from titles
```

at isolation read uncommitted

For information about transactions and isolation levels, see Chapter 18, “Transactions: Maintaining Data Consistency and Recovery,” in the *Transact-SQL User’s Guide*.

## Lock Manager Changes

This section describes the changes to locking behavior.

### Table, Page, and Address Lock Tables

SQL Server manages the acquiring and releasing of table locks by using an internal hash table with 101 rows (known as *hash buckets*) and of page and address locks by using internal hash tables with 1031 rows each. In release 11.0, these tables use one or more spinlocks to serialize access between processes that are running on different engines. A spinlock is an internal locking mechanism that prevents a second process from accessing the resource currently being used by another process. All processes trying to access the resource must wait (or *spin*) until the lock is released.

For SQL Servers running with multiple engines, you can set the ratio of spinlocks protecting each hash table using the **table lock spinlock ratio**, **page lock spinlock ratio**, and **address lock spinlock ratio** configuration parameters. The table locks hash table default ratio is 20 rows for each spinlock, and the page or address locks hash table default ratio is 100 rows for each spinlock. A SQL Server configured with only one engine has only one spinlock for each hash table, regardless of the values specified for these parameters.

For information about configuring the **table lock spinlock ratio**, **page lock spinlock ratio**, and **address lock spinlock ratio** parameters, see Chapter 11, “Setting Configuration Parameters,” in the *System Administration Guide*.

### Deadlock Checking

SQL Server performs deadlock checking after a minimum period of time for any process that is waiting (or sleeping) for a lock to be released. By default, this minimum period of time is 500 ms. Previous releases of SQL Server perform this deadlock check at the time the process begins to wait for a lock. This deadlock checking is a time--consuming overhead for applications that wait without a deadlock.



You can change the minimum amount of time (in milliseconds) that a process must wait before it initiates a deadlock check using the **deadlock checking period** configuration parameter. If you expect your applications to deadlock infrequently, you can delay deadlock checking even further and reduce the overhead cost. However, configuring **deadlock checking period** to a higher value produces longer delays before deadlocks are detected.

For information about deadlock checking, see Chapter 5, “Locking in Adaptive Server,” in the *Performance and Tuning Guide*.

### New Engine Freelock Lists

In release 11.0, when a process that is running on a multi-engine SQL Server requests a lock, it looks for one in its engine’s freelock list. If the engine freelock list is out of locks, SQL Server moves a certain number of locks from its global freelock list to the engine freelock list. For single-engine SQL Servers, the entire global freelock list is moved to the engine freelock list at server start-up time.

After an engine completes a process, all locks held by that process are released and returned to that engine’s freelock list. This reduces the contention of each engine accessing the global freelock list. However, if the number of locks released to the engine exceed the maximum number of locks allowed in the engine’s freelock list, SQL Server moves a number of locks to the global freelock list. This replenishes the number of locks that are available to other engines from the global list.

You can configure the maximum number of locks available to the engine freelock lists as a percentage of the total number of locks available to your server with the **max engine freelocks** configuration parameter. You can also configure the number of locks transferred back and forth between the engine and global freelock lists using the **freelock transfer block size** parameter. For information about these parameters, see Chapter 11, “Setting Configuration Parameters,” in the *System Administration Guide*.

## Increasing the Engine Freelock Lists

Each engine's freelock list is a portion of the global freelock list allocated to the engines, defined by the **max engines freelocks** configuration parameter as a percentage of the total number of locks. By default, SQL Server assigns 10 percent of the total number of locks to all the engines' freelock lists; the remainder stays in the global freelock list. For example, if your server is configured with 5000 locks, 500 locks are distributed evenly between each engine freelock list, and 4500 remain in the global freelock list.

For most SQL Server configurations, you can double the percentage of locks assigned to the engine freelock lists and improve the performance of your applications.

## Housekeeper Task

When SQL Server has no user tasks to process, a housekeeper task automatically begins writing dirty buffers from cache to disk. Because these writes are done during the server's idle cycles, they are known as *free writes*.

The benefits of the housekeeper task are:

- Improved CPU utilization
- Decreased need for buffer washing during transaction processing
- Faster checkpoints
- Shorter recovery time

In applications that repeatedly update the same database page, the housekeeper task may initiate some database writes unnecessarily. System Administrators can use the **housekeeper free write percentage** configuration parameter to disable the housekeeper task or to control its side effects.

For more information about the housekeeper task, see Chapter 21, "How Adaptive Server Uses Engines and CPUs," in the *Performance and Tuning Guide*.

## SQL Server Configuration

In previous releases, SQL Server was configured using **sp\_configure**, **reconfigure**, and a number of undocumented **dbcc tune** and **buildmaster -y** parameters. In order to provide a single entry point for all SQL Server configuration (except for Buffer Manager), **sp\_configure** has been redesigned to incorporate the configuration parameters previously implemented with **dbcc tune** and **buildmaster -y**. In addition, a number of new configuration parameters have been implemented.

For information about which old configuration names have new names, see “New Names for Existing Configuration Parameters” on page 137.

The **sp\_configure** interface has been redesigned to include these new features:

- Display levels, which allow the user to select one of three views of the configuration parameters:
  - Basic level, appropriate for general tuning of SQL Server
  - Intermediate level, somewhat more inclusive than the Basic level
  - Comprehensive level, appropriate for the most detailed level of SQL Server tuning
- Parameter hierarchy, which organizes the configuration parameters according to the area of SQL Server behavior to which they pertain.
- Configuration files, which allow users to replicate specific configurations, validate configurations before setting them, and create multiple configurations that can be easily switched.

For information about SQL Server configuration, see Chapter 11, “Setting Configuration Parameters,” in the *System Administration Guide*.

## Lock Escalation

In previous releases of SQL Server, once a statement accumulated more than 200 page locks on a table, SQL Server tried to issue a table lock on that object. If the table lock succeeded, the page locks were no longer necessary and were released.

The **lock promotion HWM**, **lock promotion LWM**, and **lock promotion PCT** configuration parameters and the **sp\_setpglockpromote** system procedure configure the number of page locks that SQL Server acquires on a table before it attempts to escalate to a table lock on a server-wide, per database, and per table basis.

For more information, see Chapter 5, “Locking in Adaptive Server,” in the *Performance and Tuning Guide*.

## Multiple Network Engines

If your symmetric multiprocessing (SMP) system supports network affinity migration, each SQL Server engine handles the network I/O for its connections in release 11.0. During login, SQL Server migrates the client connection task from engine 0 to the engine that is servicing the smallest number of connections. The client’s tasks run network I/O on that engine until the connection is terminated.

By distributing the network I/O among its engines, SQL Server can handle more user connections. The per-process limit on the maximum number of open file descriptors no longer limits the number of connections. Adding more engines linearly increases the maximum number of file descriptors. For more information about how multiple network engines work in an SMP environment, see Chapter 10, “Managing Multiprocessor Servers,” in the *System Administration Guide*.

## Improvements to *showplan*

The output for the **set showplan** command has been improved as follows:

- Indents and delimiters have been added to help readability.
- Line numbers and statement numbers help debug long batches and procedures.
- Additional messages include the keys used on indexes and messages that provide more information about access methods.
- Subquery types, nesting levels, and other subquery information help check subquery performance.
- Messages have been added for the new release 11.0 features to display the I/O size and the caching strategy.

In the following example, there is an index named *title\_ix* on the *titles* table:

```
select title_id, price
       from titles
       where title = "Computers and Privacy"
```

This is the pre-11.0 output from **showplan**:

```
STEP 1
The type of query is SELECT.
FROM TABLE
titles
Nested iteration
Index : title_ix
```

This is the release 11.0 output from **showplan**:

```
QUERY PLAN FOR STATEMENT 1 (at line 1).
```

```
STEP 1
The type of query is SELECT.

FROM TABLE
titles
Nested iteration.
Index : title_ix
Ascending scan.
Positioning by key.
Keys are:
title
Using I/O Size 2 Kbytes.
With LRU Buffer Replacement Strategy.
```

All **showplan** messages are documented in Chapter 9, “Understanding Query Plans,” of the *Performance and Tuning Guide*.

## Query and Data Modification Changes

This section describes the changes made to subquery processing and update behavior.

### Subquery Changes

The behavior of certain subqueries has changed. See “Changes That May Affect Existing Applications” on page 136 for detailed information about these changes and how they may affect your existing applications.

## Update Changes

Update strategies for SQL Server have improved. More updates are now made in place or directly, with fewer writes to the transaction log. See the *Performance and Tuning Guide* for more information on performance and update operations.

## Upgrading Database Dumps

When a SQL Server installation is upgraded to a new release, all databases associated with that server are automatically upgraded.

As a result, the database and transaction log dumps created with any previous SQL Server must be upgraded before they can be used with the current SQL Server.

SQL Server release 11.0 provides an automatic upgrade mechanism, on a per-database basis, for upgrading a database or transaction log dump from any SQL Server release 10.0 to the current SQL Server, thus making the dump compatible for use. This mechanism is entirely internal to SQL Server release 11.0 and requires no external programs. It provides the flexibility of upgrading individual dumps as needed.

The following tasks are not supported by this automatic upgrade functionality:

- Loading a SQL Server release 10.0 *master* database onto release 11.0.
- Installing new or modified stored procedures. Continue to use **installmaster**.
- Loading and upgrading dumps prior to SQL Server release 10.0.

Associated with this feature is a new **online database** command. This command marks a database available for public use after a normal load sequence, and if needed, upgrades a loaded database and/or transaction log dumps to the current version of SQL Server.

For more information on upgrading database dumps, see Chapter 21, “Backing Up and Restoring User Databases,” in the *System Administration Guide* and the **load database** and **dump database** commands in the *Reference Manual*.

For more information on the **online database** command, see the *Reference Manual*.

## Tape Device Determination by Backup Server

To become less device dependent and more flexible, Backup Server™ provides a method for determining the tape device characteristics (tape positioning for read, close, append, I/O size, file marks, and ability to overwrite a tape mark) for a dump operation.

When you issue a **dump database** or **dump transaction** command, Backup Server checks to see whether the device type of the specified dump device is known (supplied and supported internally) by SQL Server. If the device is not a known type, Backup Server checks the tape configuration file (default location is *\$SYBASE/backup\_tape.cfg*) for the device configuration. If the configuration is found, the **dump** command proceeds.

If the configuration is not found in the tape device configuration file, the **dump** command fails with the following error message:

```
Device not found in configuration file. INIT needs to
be specified to configure the device.
```

This means the device needs to be configured. Issue the **dump database** or **dump transaction** with **init** qualifier to configure the device. Using operating system calls, Backup Server attempts to determine the device's characteristics, and if successful, stores the device characteristics in the tape configuration file.

If Backup Server cannot determine the dump device characteristics, it defaults to one dump per tape. The device can not be used if the configuration fails to write at least one dump file.

Tape configuration by Backup Server applies only to UNIX platforms.

For more information on tape device determination and the tape configuration file, see Chapter 21, "Backing Up and Restoring User Databases," in the *System Administration Guide*.

## IDENTITY Column Changes

The **IDENTITY** column feature allows you to create a column with system-generated values that uniquely identify each row in a table. Release 11.0 provides the following enhancements to **IDENTITY** columns:

- For tables with no unique indexes, the **identity in nonunique index** database option allows you to create a unique index by including an **IDENTITY** column in the index key. Unique indexes are required for isolation level 0 reads and updatable cursors.

- The **identity grab size** configuration parameter allows each SQL Server process in a multiprocessor environment to reserve a specified number of IDENTITY column values when you add rows to tables that contain IDENTITY columns.
- The **size of auto identity** configuration parameter sets the precision of the IDENTITY columns that SQL Server generates for tables created in databases where the **auto identity** database option is set to **true**.

For more information about the enhancements to IDENTITY columns, see Chapter 7, “Creating Databases and Tables,” in the *Transact-SQL User’s Guide*.

## New *text* and *image* Global Variables

The following five global variables have been added to release 11.0 to facilitate inserting or updating *text* and *image* data through Open Client™ applications:

- @@*textcolid*
- @@*textdbid*
- @@*textobjid*
- @@*textptr*
- @@*textts*

For information about these global variables, see Chapter 13, “Using Batches and Control-of-Flow Language,” in the *Transact-SQL User’s Guide*. For information about how to use these variables in a procedure for Open Client applications, see the *Client-Library/C Reference Manual*.



## System Changes in Release 11.0

This section provides a summary of release 11.0 changes. The changes are:

- New online database Command
- Changes to Existing Commands
- New set options
- New System Procedures
- Changes to System Procedures
- New System Tables
- Changes to Existing System Tables

### New *online database* Command

Table 5-1 describes the new **online database** command:

*Table 5-1: New online database command*

Name	Function
online database	Makes a database available for public use after a normal load sequence and, if needed, upgrades a loaded database and transaction log dumps to the current version of SQL Server.

### Changes to Existing Commands

Table 5-2 summarizes the changes to existing commands:

*Table 5-2: Changes to existing commands*

Name	Change
alter table	The new partition clause allows you to create additional page chains for a table with no clustered index. The new unpartition clause allows you to concatenate all page chains for a partitioned table.
buildmaster	The buildmaster -r flag is no longer supported.
delete	New clauses allow specifying the index, cache strategy, and I/O size for a query.

Name	Change
reconfigure	The reconfigure command is no longer required to activate changes made with sp_configure. reconfigure is now non-operational. Scripts using reconfigure will still run, but should be changed at your earliest convenience, as reconfigure may not be supported in future releases.
select	New clauses allow specifying the index, cache strategy, and I/O size for a query. The new at isolation clause allows you to specify the isolation level for the query.
set showplan	Changes to output.
set transaction isolation level	New isolation level option: 0
update	New clauses allow you to specify the index, cache strategy, and I/O size for a query.

## New set options

Table 5-3 summarizes the new **set** options:

**Table 5-3: New set options**

Option	Function
forceplan	Forces the optimizer to perform joins in the order in which tables are named in the from clause of a query.
prefetch	Disables or enables large I/O (data prefetch) for a session.
statistics subquerycache	Prints statistics about the use of the subquery cache during the execution of a subquery.
table count <i>integer</i>	Specifies the number of tables that are to be optimized at the same time in a join query.

## New System Procedures

The following new system procedures provide some of the functionality for the features described earlier:

**Table 5-4: New system procedures**

Procedure	Function
sp_bindcache	Binds databases, tables, indexes, or text or image chains to data caches.

Procedure	Function
sp_cacheconfig	Configures named data caches and provides information about caches.
sp_cachestrategy	Enables and disables caching strategies and large I/O for specific tables and indexes.
sp_chgattribute	Changes the max_rows_per_page value for future space allocations of a table or index.
sp_displaylevel	Sets and displays a user's display level. The display level determines which SQL Server configuration parameters are displayed in sp_configure output.
sp_droplockpromote	Removes lock promotion values from a table or database.
sp_helpcache	Provides information about cache overhead requirements and provides information about caches and cache bindings.
sp_helppartition	Lists the first page and the control page for each partition of a partitioned table.
sp_logiosize	Changes the log I/O size used by SQL Server to a different memory pool when doing I/O for the transaction log of the current database.
sp_poolconfig	Configures pools within named caches to enable large I/O.
sp_procqmode	Reports the subquery processing mode of an object.
sp_setpglockpromote	Sets or changes the lock promotion thresholds for a database, table, or for SQL Server.
sp_unbindcache	Unbinds a specific database, table, index, or text or image object from a data cache.
sp_unbindcache_all	Unbinds all objects bound to a cache.

## Changes to System Procedures

The following system procedures have been changed in this release:

**Table 5-5: Changes to system procedures**

Procedure	Change
sp_dboption	Has a new database option, identity in nonunique index, that allows you to create a unique index by including an IDENTITY column in the index key.
sp_configure	Includes new parameters and new optional subcommands, and displays new output.

## New System Tables

The following system tables are new in this release:

**Table 5-6: New system tables**

Table	Function
<i>sysattributes</i>	Defines attributes for objects such as databases, tables, indexes, users, logins, and procedures.
<i>syslogshold</i>	Stores information about the oldest active transaction and the Replication Server truncation point for each database
<i>syspartitions</i>	Stores internal information about table partitions.

## Changes to Existing System Tables

Table 5-7 lists system tables that have been changed in this release:

**Table 5-7: Changes to system tables**

Table	Change
<i>sysindexes</i>	Changed column: <i>rowpage</i> to <i>maxrowsperpage</i> The <i>root</i> column entry for a table becomes obsolete when the table is partitioned. Root pages for individual partition are derived from information in the new <i>syspartitions</i> table.
<i>sysconfigures</i>	Added new columns: <ul style="list-style-type: none"><li>• <i>name</i> contains the configuration parameter name</li><li>• <i>parent</i> contains the group to which the parameter belongs</li><li>• <i>value2</i> is used for configuration parameters whose values are character strings</li><li>• <i>value3</i> is not currently used</li></ul>

<b>Table</b>	<b>Change</b>
<i>syscurconfigs</i>	Added new columns: <ul style="list-style-type: none"><li>• <i>value2</i> is used for configuration parameters whose values are character strings</li><li>• <i>defvalue</i> contains default values for configuration parameters</li><li>• <i>minimum_value</i> contains the minimum legal value for a configuration parameter</li><li>• <i>maximum_value</i> contains the maximum legal value for a configuration parameter</li><li>• <i>memory_used</i> contains the per-unit amount of memory used by a configuration parameter</li><li>• <i>display_level</i> contains the display level associated with a configuration parameter</li><li>• <i>datatype</i> and <i>message_num</i> are for internal use</li></ul>
<i>sysdatabases</i>	Added new column: <ul style="list-style-type: none"><li>• <i>offline</i> status bit is used by the upgrading database dumps feature</li></ul>

## Changes That May Affect Existing Applications

This section describes the system changes introduced in release 11.0 that may affect your applications if you are upgrading to release 11.5 from a pre-11.0 release. Topics covered are as follows:

- New Transact-SQL Keywords in Release 11.0
- Changes to SQL Server Configuration
- Subquery Changes
- Changes to showplan Output in Release 11.0
- New Caching Strategies May Affect Performance
- Upgrading Database Dumps
- Partitions and Physical Data Placement

### New Transact-SQL Keywords in Release 11.0

The following keywords are new reserved words in SQL Server 11.0. They cannot be used as object names or column names.

- **errordata**
- **max\_rows\_per\_page**
- **online**
- **partition**
- **unpartition**

You must change all database names that are new reserved words before you can upgrade an earlier version of the server. You can change table, view, and column names or use delimited identifiers. Once you upgrade, you cannot use database objects whose names are new reserved words until you modify your procedures, SQL scripts, and applications.

### Changes to SQL Server Configuration

The following changes to the SQL Server configuration interface will affect existing applications.

## The *reconfigure* Command

In previous releases, changes made with **sp\_configure** had to be activated with the **reconfigure** command. This is no longer required. **reconfigure** is no longer operational. Existing scripts should continue to run, but should be changed at your earliest convenience because **reconfigure** may not be supported in future releases.

## *buildmaster -r* No Longer Supported

The **buildmaster -r** flag, which was used to rewrite the configuration block with default values for SQL Server configuration parameters, is no longer supported.

To run SQL Server with its built-in default values:

- 1 Rename, move, or delete your configuration file.
- 2 Shut down and restart SQL Server.

When you do this, SQL Server uses the built-in default configuration values, and creates a configuration file *server\_name.cfg* in the directory from which SQL Server was started. However, these values do not get written to *sysconfigures*. In order to have them written to *sysconfigures*, you need to shut down and restart SQL Server once again.

## New Names for Existing Configuration Parameters

The following SQL Server configuration parameters (formerly called “configuration variables”) have new names:

**Table 5-8: New configuration parameter names for release 11.0**

Old Name	New Name
T1204 (trace flag)	print deadlock information
T1603 (trace flag)	allow sql server async i/o
T1610 (trace flag)	tcp no delay
T1611 (trace flag)	lock shared memory
allow updates	allow updates to system tables
calignment	memory alignment boundary
cckrate	sql server clock tick length
cfgcprot	permission cache entries
cguardsz	stack guard size
cindextrips	number of index trips

## Changes That May Affect Existing Applications

---

<b>Old Name</b>	<b>New Name</b>
cmaxnetworks	max number network listeners
cmaxschedules	i/o polling process count
cnalarm	number of alarms
cnblkio	disk i/o structures
cnlanginfo	number of languages in cache
cnmaxaio_engine	max async i/os per engine
cnmaxaio_server	max async i/os per server
cnmbox	number of mailboxes
cnmsg	number of messages
coamtrips	number of oam trips
cpreallocext	number of pre-allocated extents
cpu flush	cpu accounting flush interval
cschedspins	runnable process search count
csortbufsize	number of sort buffers
csortpgcount	sort page count (Removed in release11.5.)
ctimemax	cpu grace time
database size	default database size
default language	default language id
devices	number of devices
extent i/o buffers	number of extent i/o buffers (Removed in 11.5.)
fillfactor	default fill factor percent
i/o flush	i/o accounting flush interval
language in cache	number of languages in cache
locks	number of locks
maximum network packet size	max network packet size
memory	total memory
mrstart	shared memory starting address
nested trigger	allow nested triggers
open databases	number of open databases
open objects	number of open objects
password expiration interval	systemwide password expiration
pre-read packets	remote server pre-read packets
procedure cache	procedure cache percent
recovery flags	print recovery information
recovery interval	recovery interval in minutes
remote access	allow remote access
remote connections	number of remote connections



Old Name	New Name
remote logins	number of remote logins
remote sites	number of remote sites
sql server code size	executable code size
tape retention	tape retention in days
user connections	number of user connections

### New Configuration Parameters

Release 11.0 introduced a number of new configuration parameters listed in Table 5-9. See Chapter 11, “Setting Configuration Parameters,” in the *System Administration Guide* for a complete description of these parameters.

**Table 5-9: New SQL Server configuration parameters for release 11.0**

address lock spinlock ratio
configuration file
deadlock checking period
deadlock retries
event buffers per engine
freelock transfer block size
housekeeper free write percent
identity grab size
lock promotion HWM
lock promotion LWM
lock promotion PCT
max engine freelocks
o/s async i/o enabled
o/s file descriptors
page lock spinlock ratio
page utilization percent
partition groups
partition spinlock ratio
size of auto identity column
table lock spinlock ratio
total data cache size
user lock cache size
user lock cache spinlock ratio

### ***New deadlock checking period Parameter***

Previous releases of SQL Server performed deadlock checking when the process began to wait for a lock. In release 11.0, SQL Server performs deadlock checking for a process only after it has waited a minimum of 500 ms. for a lock to be released. To return to the previous way of deadlock checking, set **deadlock checking period** to 0.

### ***New page utilization percent Parameter***

The default behavior of previous releases of SQL Server is to search the OAM page chain for unused pages before allocating a new extent. To keep this behavior, set the **page utilization percent** to 100.

### **Compiled Object Sizes Have Grown**

The size of compiled objects is larger in System 11, and was significantly larger in System 10. You may need to resize your procedure cache to maintain the same performance.

### **SQL Server Code Size Has Grown**

More memory is used for the kernel and other internal structures. You may need to add more memory to maintain the same performance as your previous release.

## **Subquery Changes**

Several problems relating to subqueries have been fixed in SQL Server release 11.0. This section describes old and new behavior in detail. Examine applications that use subqueries to determine if your application relies on behavior that has been corrected.

Subquery processing in previous releases of SQL Server used “inside-out” processing. Release 11.0 subqueries are processed “outside-in” for greater efficiency. Upgrading to release 11.0 does not automatically change the processing mode of the subquery.

After upgrading to release 11.0, you must drop and re-create objects to take advantage of the new processing style. Use **sp\_procmode** to identify objects that include subqueries. After upgrading to SQL Server release 11.0, you cannot create an object that uses release 10.0 processing.

It is best to test procedures that contain subqueries and make any changes that are necessary for your application before upgrading your production system.

### Different Results

Subqueries in objects created in SQL Server release 11.0 may return different results than subqueries in objects created on an earlier SQL Server release. To test performance after upgrade, create the objects in SQL Server release 11.0, test the results and performance, and make any changes that are necessary for your application.

### Determining Query Processing Mode

A new system procedure, **sp\_procqmode**, reports on the subquery processing mode of an object.

### Duplicates in Subquery Results

As of release 10.0, subqueries using **in** or **any** predicates no longer return duplicates.

**set dup\_in\_subquery on** was provided as an upgrade path to SQL Server release 10.0. It is no longer supported. If your application depends on the duplicate rows in the result set, rewrite the subquery as a join.

For example:

```
select a from s where b in
      (select c from t)
```

would become:

```
select a from s, b where s.b = t.c
```

Procedures created in SQL Server release 10.0 that use **dup\_in\_subquery** will continue to run until the procedure is dropped and re-created in release 11.0.

### Changes in Subquery Restrictions

The following subquery restrictions have been removed:

- You can use **distinct** with **group by** in a subquery.

The **distinct** keyword suppressed duplicate rows in the output of pre-release 10.0 subqueries. This keyword is no longer necessary, and is ignored, in subqueries.

- You can reference a correlated variable in the select list of a subquery. For example, the following expression subquery is now supported:

```
select t1.c1 from t1
       where t1.c2 = (select t2.c2 + t1.c3 from t2)
```

The following subquery restriction has been added:

- There is a maximum of 16 subqueries within a single side of a union.

## Handling NULL Results

Prior to release 11.0, a correlated expression subquery in the **set** clause of an update returned 0 instead of NULL when there were no matching rows. Release 11.0 correctly returns NULL when there are no matching rows, and an error is raised if the column does not permit nulls.

If you have applications that depend on the pre-release 11.0 behavior, you will need to rewrite them.

For example, the following trigger tries to update a column that does not permit NULL values:

```
update t1
       set c1 = (select max(c1)
                from inserted where t1.c2 = inserted.c2)
```

The correct trigger is:

```
update t1
       set c1 = (select isnull(max(c1), 0)
                from inserted
                where t1.c2 = inserted.c2)
```

or:

```
update t1
       set c1 = (select max(c1)
                from inserted
                where t1.c2 = inserted.c2)
       where exists (select * from inserted
                    where t1.c2 = inserted.c2)
```

The **where** clause updates table *t1.c1* to 0, if the subquery does not return any correlation values from the outer table *t1*.

Another example of this is the *totalsales\_trig* in the *pubs2* sample database. In previous versions, the trigger read as follows:

```
create trigger totalsales_trig
```

```

on salesdetail
for insert, update, delete
as
/* Save processing:  return if there are no rows affected */
if @@rowcount = 0
    begin
        return
    end

/* add all the new values */
/* use isnull:  a null value in the titles table means
**          "no sales yet" not "sales unknown"
*/
update titles
    set total_sales = isnull(total_sales, 0) +
        (select sum(qty)
         from inserted
         where titles.title_id = inserted.title_id)

/* remove all values being deleted or updated */
update titles
    set total_sales = isnull(total_sales, 0) -
        (select sum(qty)
         from deleted
         where titles.title_id = deleted.title_id)

```

**sum(qty)** is NULL if no row is returned from the table, so when a statement changes the *total\_sales* column, the trigger changes to NULL all the rows in *titles* that do not qualify.

To guarantee that the subquery in the expression for the update returns a non-null value, the corrected trigger is:

```

create trigger totalsales_trig
on salesdetail
for insert, update, delete
as
/* Save processing:  return if there are no rows affected */
if @@rowcount = 0
    begin
        return
    end

/* add all the new values */
/* use isnull:  a null value in the titles table means
**          "no sales yet" not "sales unknown"
*/

```

```
update titles
  set total_sales = isnull(total_sales, 0) +
    (select sum(qty)
     from inserted
     where titles.title_id = inserted.title_id)
  where title_id in (select title_id from inserted)

/* remove all values being deleted or updated */
update titles
  set total_sales = isnull(total_sales, 0) -
    (select sum(qty)
     from deleted
     where titles.title_id = deleted.title_id)
  where title_id in (select title_id from deleted)
```

## Improved Performance

Most subqueries will perform better after being dropped and re-created.

However, an expression subquery containing an aggregate where the outer table is very large and has few duplicate correlation values, and the inner table is small, may not perform as well in release 11.0. For example:

```
select * from huge_table where x=
  (select sum(a) from tiny_table
   where b = huge_table.y)
```

The workaround is:

```
select huge_table.y, s=sum(a)
  into #t
  from huge_table, tiny_table
  where b=huge_table.y
  group by huge_table.y

select huge_table.*
  from huge_table, #t
  where x=#t.x
  and huge_table.y=#t.y
```

## Changes to *showplan* Output in Release 11.0

The output for the **set** option **showplan** has changed. If you have applications that use this generated output, they need to be changed to use the new output.

See Chapter 9, “Understanding Query Plans” in the *Performance and Tuning Guide*.

## New Caching Strategies May Affect Performance

The optimizer may now choose a new caching strategy called “fetch and discard” strategy. Standard caching strategy reads pages in at the head of the MRU/LRU (most recently used/least recently used) chain in cache. This flushes other pages out of cache. Fetch-and- discard (or MRU) strategy reads pages into cache much closer to the end of the cache, so it does not flush other pages. But these pages remain in cache a much shorter time, so subsequent user queries that might get a “cache hit” by finding these pages in cache under the standard strategy will be less likely to find the page in cache.

When the optimizer estimates that a particular query will read a significant number of pages, and only need those pages once during a query, it may choose fetch and discard strategy. If your disk I/O increases in release 11.0, check the I/O strategy for queries using **showplan** to see the cache strategy in use. If the queries are using fetch- and-discard (MRU) strategy, you can add a clause to **select**, **update**, and **delete** commands to specify standard caching.

See Chapter 10, “Advanced Optimizing Techniques” in the *Performance and Tuning Guide*.

## Upgrading Database Dumps

If you use scripts to perform database loads using the **load database** and **load transaction** commands, the scripts will fail unless you include the **online database** command at the end of your load sequence. You should also remove the following **sp\_dboption** options from your scripts: **no chkpt on recovery**, **dbo use only**, and **read only**. These options are no longer needed with release 11.0, as **load database** sets the database status to offline, thus making it inaccessible by users.

For more information on upgrading database dumps, see Chapter 21, “Backing Up and Restoring User Databases,” in the *System Administration Guide* and the **load database** and **dump database** commands in the *Reference Manual*.

## Partitions and Physical Data Placement

---

**Note 11.5** Note: Release 11.5 adds the ability to create clustered indexes on partitioned tables and to access the tables in parallel. See the *Performance and Tuning Guide* for more information.

---

In release 10.0, a heap table's data was always inserted at the end of a single page chain. This meant that a heap table (for example, a history table) would physically store its newest entries at the end of the page chain, and its oldest entries at the beginning. A simple **select** statement or cursor scan against such a table could return rows in roughly the same order in which they were inserted.

In release 11.0 you can create partitioned tables that have multiple page chains. Inserts into partitioned tables can occur at the ends of many separate page chains. Data in such a table *is not* physically stored in sequential order. If you want to view rows from a partitioned table in sequential order, use the **order by** clause in your **select** statement.

See the *Performance and Tuning Guide* for more information about partitions.



# Index

## Symbols

@@textcolid global variable 130  
@@textdbid global variable 130  
@@textobjid global variable 130  
@@textptr global variable 130  
@@textts global variable 130

## A

Abstract plans for queries 21  
Address locks 122  
alter role command 36  
alter table command 36  
    enabling or disabling triggers 22  
    modifying an existing tables schema 8  
alter table command 36  
ANSI joins 18  
Application execution precedence  
    system procedures 82  
Assigning names to clients 11  
at isolation clause 121  
Auditing  
    system procedures for 69  
Authentication 93  
auto identity  
    sp\_dboption 130

## B

Backup Server  
    dump operations 6  
    high performance backup and restore 6  
    non-rewinding tapes 6  
Backup Server utility  
    tape device determination 128  
bcp  
    copying data in parallel 95

Bourne shell xiii  
buildmaster -y utility 125

## C

C shell xiii  
Cache strategies 116  
case expression 72  
    syntax 72  
Changing identity gaps 12  
check password for digit configuration parameter 34  
Checkpoints  
    housekeeper task and 124  
clientapplname command 37  
clienthostname command 37  
clientname command 37  
Clients, assigning names to 11  
Component Integration Services 73  
Component Integration Services (CIS) 24, 25  
configname options for sp\_helpconfig 13  
Configuration changes 125, 136  
Configuring caches 114  
Consting for queries 17  
Corruption 99  
CPU utilization  
    housekeeper task and 124  
create role command 36  
create table command 12, 36  
create table command 36  
Creating objects 32  
.cshrc file xiii  
Cursors  
    compilation 62  
    optimization 62  
    partitioned tables and 146

**D**

Data caches 114  
Data storage, `max_rows_per_page` and 117  
Database  
  for debugging 13  
  integrity, ensuring 7  
  object owners, tracking 32  
  suspending 9  
Datatypes  
  enhancements for 20  
  java classes as 15  
`dbcc checkdb` 119  
`dbcc checktable` 119  
`dbcc command` 36  
`dbcc engine command` 10  
`dbcc tune` 125  
Deadlock checking 122  
`delete` command, tracking creator of 32  
`delete`, new clauses for 117  
Direct updates 128  
Directory control layer (DCL) 26  
Directory Services 81  
Disabling triggers 22  
`disk init` command 7, 36  
`disk reinit` command 7, 36  
Distributed transaction management 24  
`dsync` command 7  
`dtm detach timeout period` configuration parameter 34  
`dtm lock timeout period` configuration parameter 34  
`dump database` command 100  
Dump operations and Backup Server 6  
`dump transaction` command 100, 121  
Dumps  
  Upgrading database dumps 128, 145  
Duplicate subquery results 141

**E**

Email messages 111  
`enable cis` configuration parameter 35  
`enable dtm` configuration parameter 34  
`enable HA` configuration parameter 34  
`enable java` configuration parameter 34  
`enable sort-merge joins and JTC` configuration parameter

17

`enable xact coordination` configuration parameter 34  
Enabling triggers 22  
Engine affinity 82  
Engine freelock lists 123  
Engines offline 10  
Enhancements  
  for datatypes 20  
  for queries 17  
Error log 84  
ESPs 86  
`execute` command 19, 37  
Execution  
  attributes 82  
  objects 82  
  precedence 82  
  system procedures for 82  
Execution of Transact-SQL commands 19  
Extended stored procedures. See ESPs 86  
`extractjava` (UNIX) utility 39  
`extrjava` (NT) utility 39

**F**

Failover, using in a high availability system 3–4, 34  
Files  
  `.cshrc` xiii  
  `.profile` xiii  
`fillfactor`, compared to `max_rows_per_page` 118  
Freelock lists 123  
Functions, java 14

**G**

Global freelock lists 123  
Global variables  
  `@@textcolid` 130  
  `@@textdbid` 130  
  `@@textobjid` 130  
  `@@textptr` 130  
  `@@textts` 130  
Granting permission for proxy authorization 98

**H**

- Hash tables 122
- Heap tables 118
- High availability, using Sybase's Failover 3–4
- High availability, using Sybase's Failover 34
- High performance backup and restore 6
- Housekeeper task
  - free writes and 124

**I**

- IDENTITY columns
  - changes for 11.0 129
- identity gap parameter 12
- Identity gaps, changing 12
- identity in nonunique index
  - sp\_dboption 129
- Image datatypes 20
- image global variables 130
- Impersonating another user 98
- Indexes, rebuilding 5
- In-place updates 128
- insert command, tracking creator of 32
- insert, new clauses for 117
- installjava (UNIX) utility 39
- instjava (NT)) utility 39
- Integrity
  - database 7
  - message 27
- Integrity of messages 93
- Interfaces file 26
- Isolation levels 121

**J**

- Java
  - capabilities 14
  - classes as datatypes 15
  - datatypes 15
  - in Adaptive Server Enterprise 14–15
  - user-defined functions 14
- join 36
- join command 36
- Joins

- merged 17
- using ANSI syntax 18

**K**

- Keywords
  - new for release 11.0 136
  - new for release 11.5 43
- Korn shell xiii

**L**

- Large I/O 116
- Licensed feature 3, 14, 24
- Limits on tables in queries 16
- load transaction command 97
- Lock escalation 125
- Lock manager 122
- log audit logon failure configuration parameter 84
- log audit logon success configuration parameter 84
- Log I/O size 116
- Logical Process Manager 82
- Login security, user-defined 28
- Logins
  - server 32
  - unified 26

**M**

- Managing distributed transactions 24
- max\_rows\_per\_page
  - compared to fillfactor 118
- maximum failed logins configuration parameter 34
- Memory pools 116
- Merge joins 17
- Message confidentiality 93
- Message integrity 27, 93
- Message origin checks 93
- Metadata caches 90
- Microsoft Distributed Transaction Coordinator (MSDTC) 24
- minimum password length configuration parameter 35
- Monitor Access to Executing SQL 91

configuring amount of text saved in memory 91  
Monitoring  
    from Windows NT Performance Monitor 112  
Multiple network engines 126

## N

Named data caches 114  
Named time ranges 102  
Network I/O engines 126  
Network security 26  
Network-based security  
    overview 93  
    security services and the server 93  
Non-rewinding tapes for Backup Server 6  
number of aux scan descriptors configuration parameter  
    89  
number of dtx participants configuration parameter 35

## O

OAM pages 118  
Objects, creating 32  
Oldest active transactions 120  
online database command  
    usage 128, 131, 145  
Optimization for queries 17  
order by clause 146

## P

Page allocation 118  
Page locks 122  
page utilization percent 140  
Parallel bulk copy  
    and partitioned tables 95  
    and the -F and -L flags 95  
    copying in data in parallel 95  
Parallel query processing 96  
Parameter hierarchy 125  
Parameters  
    name changes 137  
Partitioned tables 118

data placement and 146  
Performance  
    application tradeoffs 82  
Performance of network security 26  
Plans, query 21  
Point-in-time recovery 97  
Pools 116  
Procedural language functions  
    calling 86  
    .profile file xiii  
Proxy authorization  
    granting permission for 98  
    overview 98  
    value for applications 98  
    value for users 98

## Q

Queries  
    abstract plans for 21  
    enhancements to 17  
    improvements in costing 17  
    limiting with `sp_add_resource_limit` 102  
    number of tables in 16  
Query changes 127  
Query processing  
    mode 141  
    parallel 96  
Query tuning options 117  
quiesce database command 9, 36

## R

Raw partitions 7  
Rebuilding indexes 5  
reconfigure command 137  
Recovery 97, 99  
Recovery time, housekeeper task and 124  
Referential integrity  
    enhancements 89  
Remote procedures  
    security for 93  
Remote servers, connecting with Component Integration  
    Services 73

- remove java command 36
  - remove java command 36
  - reorg rebuild command 5
  - Replay detection 93
  - Reserved words
    - new for release 11.0 136
    - new for version 12.0 43
  - Resource limits 102
  - Rows per page 117
- S**
- Security
    - network 26
    - user-defined login 28
  - Security services
    - message confidentiality 93
    - message integrity 93
    - message origin checks 93
    - mutual authentication 93
    - out-of-sequence check 93
    - remote procedure security 94
    - replay detection 93
    - supported by the server 93
    - unified login 93
  - Segments
    - partitions and 119
  - select command 18
    - new clauses 117, 121
  - select into command 12
  - Sequence checking 93
  - Server user name and ID 32
  - set command 11
    - clientapplname 37
    - clienthostname 37
    - clientname 37
  - set dup\_in\_subquery 141
  - set forceplan 117
  - set options overview 132
  - set prefetch 117
  - set proxy command 98
  - set session authorization command 98
  - set showplan command 126
  - set table count 117
  - set transaction isolation level 121
  - Shells xiii
  - showplan improvements 126
  - size of global fixed heap configuration parameter 35
  - size of global fixed heap configuration parameter 35
  - size of process object fixed heap configuration parameter 35
  - size of process object fixed heap configuration parameter 35
  - size of shared class heap configuration parameter 35
  - size of shared class heap configuration parameter 35
  - SMP systems 126
  - sp\_addlogin system procedure 38
  - sp\_addmessage system procedure 84
  - sp\_altermessage system procedure 84
  - sp\_bindcache 115, 132
  - sp\_cacheconfig 114, 133
  - sp\_cachestrategy 117, 133
  - sp\_chgattribute 133
  - sp\_chgattribute system procedure 12
  - sp\_companion system procedure 38
  - sp\_configure 125, 133, 140
    - address lock spinlock ratio 122
    - deadlock checking period 123, 140
    - freelock transfer block size 123
    - identity grab size 130
    - lock promotion 125
    - max engine freelocks 123
    - page lock spinlock ratio 122
    - page utilization percent 118
    - table lock spinlock ratio 122
    - user log cache size 120
    - user log cache spinlock ratio 120
  - sp\_configure command 20
  - sp\_dboption 133
    - auto identity 130
    - identity in nonunique index 129
  - sp\_deviceattr system procedure 7, 38
  - sp\_displaylevel 133
  - sp\_displaylogin system procedure 38
  - sp\_displayroles system procedure 39
  - sp\_droplockpromote 133
  - sp\_ha\_admin system procedure 38
  - sp\_help system procedure 12
  - sp\_helppartition 119, 133
  - sp\_helpcache 115, 133
  - sp\_helpconfig system procedure 13

sp\_helpdevice system procedure 7, 39  
 sp\_helpjava system procedure 38  
 sp\_helpjava system procedure 38  
 sp\_lock system procedure 39  
 sp\_logiosize 116, 133  
 sp\_modifylogin system procedure 39  
 sp\_monitorconfig system procedure 39  
 sp\_poolconfig system procedure 114, 133  
 sp\_procqmode 133, 141  
 sp\_setpglockpromote 125, 133  
 sp\_transactions system procedure 38  
 sp\_unbindcache 115, 133  
 sp\_unbindcache\_all 133  
 sp\_who system procedure 39  
 Spinlocks 120, 122  
 SQL Perfmon Integration configuration parameter 112  
 status column, sysserverroles 41  
 strict dtm enforcement configuration parameter 35  
 strict\_dtm\_enforcement command 37  
 Subqueries 140, 142  
     NULL results 142  
     restrictions 141  
 Subquery changes 127  
 suid (server user ID), object creator and 32  
 Support for UNIX file systems 7  
 Suspect pages 99, 100  
 Suspending databases 9  
 sybdiagdb database 13  
 Sybmail 111  
 Syntax conventions xiii  
 sysattributes system table 41, 134  
 syscolumns system table 40  
 syscolumns system table 40  
 sysconfigures system table 41, 134  
 syscoordinations system table 40  
 syscurconfigures system table 135  
 sysdatabases system table 135  
 sysdevices system table 40  
 sysserverroles system table 41  
 sysindexes system table 134  
 sysjars system table 40  
 sysjars system table 40  
 syslocks system table 40  
 syslogins system table 41  
 syslogshold system table 120, 134  
 syspartition system table 134

sysprocesses system table 40  
 syssessions system table 40  
 sysserverroles table 41  
 System procedures  
     sp\_bindcache 132  
     sp\_cacheconfig 133  
     sp\_cachestrategy 133  
     sp\_chgattribute 133  
     sp\_configure 125, 133  
     sp\_dboption 133  
     sp\_displaylevel 133  
     sp\_droplockpromote 133  
     sp\_helppartition 133  
     sp\_helpcache 133  
     sp\_logiosize 116, 133  
     sp\_poolconfig 133  
     sp\_procqmode 133, 141  
     sp\_setpglockpromote 125, 133  
     sp\_unbindcache 133  
     sp\_unbindcache\_all 133  
 System tables 40, 134  
 systransactions system table 40  
 sysusers system table 41  
 sysxtypes system table 40  
 sysxtypes system table 40

## T

Table locks 122  
 Table scans  
     partitioned tables and 146  
 Tables, number of, in a query 16  
 Taking engines offline 10  
 Tape device determination 129  
 Text datatypes 20  
 text global variables 130  
 text prefetch size configuration parameter 20, 35  
 Time ranges 102  
 titles system table 21  
 Transaction log 116, 120  
 transactional\_rpc command 37  
 Transactions  
     limiting with sp\_add\_resource\_limit 102  
     Two-Phase Commit 107  
 Transactions, managing distributed 24

Transact-SQL, dynamic execution of 19  
Triggers  
    disabling 22  
    enabling 22  
    NULL results 142  
Truncation points 120  
Two-phase commit 106  
txn to pss ratio configuration parameter 35

## U

Unified login 26, 93  
UNIX file system support 7  
Update changes in subqueries 128  
update command tracking creator of 32  
User log caches 120  
User-defined  
    functions for java 14  
    login security 28  
User-defined caches 114

## W

Windows NT Event Log 84  
Windows NT Mail account 111  
Windows NT Mail profile 111  
Windows NT Performance Monitor 112

## X

X/Open XA protocol 24  
xact coordination level configuration parameter 35  
XP Server 86  
    priority 87  
xp\_cmdshell configuration parameter 88











