



Product Documentation

Embarcadero® DB Change Manager™ XE and DB Change Manager™ 5.5

User Guide

Versions XE and 5.5
Last Revised May 14, 2010

© 2010 Embarcadero Technologies, Inc. Embarcadero, the Embarcadero Technologies logos, and all other Embarcadero Technologies product or service names are trademarks or registered trademarks of Embarcadero Technologies, Inc. All other trademarks are property of their respective owners.

Embarcadero Technologies, Inc. is a leading provider of award-winning tools for application developers and database professionals so they can design systems right, build them faster and run them better, regardless of their platform or programming language. Ninety of the Fortune 100 and an active community of more than three million users worldwide rely on Embarcadero products to increase productivity, reduce costs, simplify change management and compliance and accelerate innovation. The company's flagship tools include: Embarcadero® Change Manager™, CodeGear™ RAD Studio, DBArtisan®, Delphi®, ER/Studio®, JBuilder® and Rapid SQL®. Founded in 1993, Embarcadero is headquartered in San Francisco, with offices located around the world. Embarcadero is online at www.embarcadero.com.

CORPORATE HEADQUARTERS
100 CALIFORNIA STREET
12TH FLOOR
SAN FRANCISCO, CALIFORNIA
94111 USA

EMEA HEADQUARTERS
YORK HOUSE
18 YORK ROAD
MAIDENHEAD, BERKSHIRE
SL6 1SF, UNITED KINGDOM

ASIA-PACIFIC HEADQUARTERS
L7. 313 LA TROBE STREET
MELBOURNE VIC 3000
AUSTRALIA

Contents

Introducing DB Change Manager	11
Key Features of DB Change Manager	11
Data Source Comparison and Synchronization	12
Schema Comparison and Synchronization	12
Database Configuration Management	12
Reports, Notifications, and Change History	12
Data Masking	13
Source Control System Integration	13
Technical Requirements	13
Database Support	14
Installing and Upgrading DB Change Manager	14
Key Concepts of DB Change Manager	14
Understanding Jobs and Scripts in DB Change Manager	14
Understanding The DB Change Manager User Interface	15
Additional Product Information	17
What You Can Do With DB Change Manager	19
Commonly-Performed Tasks	19
Comparing and Synchronizing Data	19
Comparing and Synchronizing Schemas	20
Managing Database Configurations	20
Creating Masked Data for Development Environments	20
Automating and Scripting Jobs	20
What is New in DB Change Manager	21
Registering and Organizing Data Sources	23
Registering Data Sources on Startup	23
Scanning for Data Sources and Registering Them	24
Registering a New Data Source One at a Time	26
Grouping Data Sources	27
Importing and Exporting Data Sources	28

Creating a Job Report	31
Scheduling a Job	35
Setting Job Notifications	39
Email Notifications	39
File System Notifications for Log Files	39
System Tray Notifications	40
Send Notification and Attachment Options	41
Filtering Data Source Objects	45
Comparing Data	49
Create a Data Comparison Job	50
Describe and Save the Job	50
Specify the Source and Target Data Sources	51
Specify Execution, Mapping, Comparison, and Resolution Options	52
Understand the Data Source Mapping Process	53
Customize Database Mapping	53
Inspect the Databases	54
Customize Table Mapping	55
Reconfigure the Data Source Mapping	56
Set Job Notification Options	57
Run the Data Comparison Job	57
Review Comparison Results	58
Synchronizing Data Repositories	59
Select Data to Resolve	59
Run the SQL Synchronization Script	60
Comparing Database Configuration Settings	63
Create or Open a Configuration Comparison Job	64
Describe and Save the Job	64
Specify Configuration Comparison Source and Targets	65
Refine Property Comparisons	67
Specify Configuration Comparison Job Options	67
Set Configuration Comparison Job Notifications	67
Run the Configuration Comparison Job	68
View the Results of the Configuration Comparison	68

Create a Job Report.	69
Synchronize Configurations.	69
Creating a Standard for Configuration Settings	71
Create or Open a Configuration Standard	71
Describe and Save the Standard	72
Add Standard Sources	72
Define Configuration Standard Properties	73
Use a Configuration Standard in Configuration Comparison Jobs	74
Creating a Configuration Archive	77
Create or Open a Configuration Archive Job	77
Describe the Configuration Archive Job	78
Specify a Configuration Archive Source.	78
Refine Configuration Archive Properties	78
Set Configuration Archive Job Notifications	79
Specify Configuration Archive Job Options	79
Create a Version and Save the Archive Job.	80
Create an Archive Job Report.	81
Use an Archive in a Configuration Comparison Job.	81
Comparing Schemas	83
Create or Open a Schema Comparison Job	84
Describe and Save the Schema Comparison Job.	84
Specify a Schema Comparison Source and Targets	85
Cross-Platform Schema Comparisons Supported by DB Change Manager.	85
Add Multiple Schema Comparison Job Targets	86
Refine Schema Comparisons	86
Customize Schema Mapping	87
Specify Schema Comparison Job Options	88
Select Schema Properties to Ignore During a Comparison	90
Set Schema Comparison Job Notifications	91
Run the Schema Comparison Job	91
View Individual Schema Comparison Results.	93
Create a Schema Comparison Job Report	94
Clone a Schema Comparison Job	94

Creating a Schema Archive	97
Create or Open a Schema Archive Job	98
Describe and Save the Schema Archive Job	98
Specify a Schema Archive Job Source	98
Refine Schema Archive Properties	99
Set Schema Archive Job Notifications	100
Specify Schema Archive DDL Options	100
Create a Version and Save the Schema Archive Job	100
Review the History of Archived Objects	102
Create a Job Report.	102
Use an Archive in Schema Comparison Jobs	103
Synchronizing Schemas	105
Resolve Schema via a Generated SQL Script	105
Execute a SQL Synchronization Script	106
Configure an SQL Session	108
Masking Data	111
Create a New Masking Job	111
Specify a Masking Job Name	111
Specify Source and Target Data Sources.	112
Set to Automatically Synchronize	113
Specify Mapping Options	114
Specify Masking Rules for Individual Columns	114
Run the Database Masking Job	116
See the Masked Data.	117
Automating Batch Scripts	119
Generating Command Line Scripts	119
Using the changec Command Line Program.	120
Database Management System Reference	123
DBMS Operations Permissions	123
Permissions Required for DB2 for LUW	123
Permissions Required for Firebird and InterBase	124
Permissions Required for SQL Server	124
Permissions Required for Oracle	125
Permissions Required for Sybase	125

Version Information Reference	126
Firebird and InterBase Version Information	126
IBM DB2 for LUW Version Information	126
Oracle Version Information	127
SQL Server Version Information	127
Sybase Version Information	127
Preferences Reference	129
DB Change Manager Preferences	130
CM/Config > Archive Options	131
CM/Config > Comparison Options	131
CM/Data > Comparison Options	131
Execution Options	131
Mapping Options	132
Comparison Options	132
Resolution Options	133
CM/Data > Data Result Options	133
CM/Data > General	133
CM/Data > Mapping Validation	134
CM/Data > Script Generation Options	136
CM/Schema > Archive Options	136
CM/Schema > Comparison Options	137
CM/Schema > General	140
CM/Schema > Mapping Options	141
Command Line	141
Compliance	141
Label Decorations	142
Notification > Email Notification	142
Notification > Email Notification > Advance Settings	143
Notification > Email Notification > Contacts	143
Notification > Email Notification > Templates	143
Notification > File System Notifier	144
Notification > Popup Notification	144
Notification > Popup Notification > Advanced Settings	145
Projects	145
Reports	145
Script Execution Options	147
System Tray	147

Task Scheduler	147
Data Sources	148
Eclipse SQL Development Preferences	148
SQL Development > SQL Editor	149
SQL Development > SQL Editor > Code Assist	149
SQL Development > SQL Editor > Code Formatter	149
SQL Development > SQL Editor > Results Viewer	150
SQL Development > SQL Editor > Syntax Coloring	150
SQL Development > SQL Filters	150
User Interface Reference	151
The DB Change Manager Workbench	152
Welcome Page	154
Workspace	154
Views	155
Job Explorer	157
Viewing Change History	157
Data Source Explorer	158
Project Explorer	160
Navigator View	161
Compliance Explorer	162
SQL Errors View	162
Data Diff View	163
Extract DDL Scripts	164
SQL Log	164
SQL Errors Log	164
Search View	165
Editors	168
Data Comparison Job Editor	169
Configuration Comparison Job Editor	170
Configuration Standard Editor	170
Configuration Archive Job Editor	171
Schema Comparison Job Editor	172
Schema Archive Job Editor	173
SQL Editor	175
SQL Session Options	176

Eclipse Plugin Extensions Reference	179
Creating and Implementing a Notification Plugin	179
Creating and Implementing a Custom Configuration Plugin	191

INTRODUCING DB CHANGE MANAGER

Embarcadero® DB Change Manager™ helps database administrators and developers to:

- Simplify and automate database change management.
- Streamline development cycles.
- Ensure availability, performance and compliance.

DB Change Manager provides comparison, synchronization, data masking, and auditing capabilities that let you track and report on changes, roll out new releases, and pinpoint performance problems resulting from changes at the data, schema and database configuration levels.

By comparing a live database to a schema or configuration “snapshot” administrators can quickly identify changes and correct problems. By monitoring configuration settings, administrators can ensure compliance with regulatory policies and performance standards, and maintain overall database performance and availability.

- [Key Features of DB Change Manager](#)
- [Technical Requirements](#)
- [Database Support](#)
- [Key Concepts of DB Change Manager](#)
- [Additional Product Information](#)

KEY FEATURES OF DB CHANGE MANAGER

DB Change Manager provides features to help you manage databases from development to deployment and maintenance cycles. These features include:

- [Data Source Comparison and Synchronization](#)
- [Schema Comparison and Synchronization](#)
- [Database Configuration Management](#)
- [Reports, Notifications, and Change History](#)
- [Data Masking](#)
- [Source Control System Integration](#)

DATA SOURCE COMPARISON AND SYNCHRONIZATION

DB Change Manager can **analyze and compare** the data in two different data sources. It generates a SQL script that can **update one data source so that it matches** another. You can modify the SQL and run it as you wish. You can even define a comparison and synchronization process, called a **job**, and set it to run automatically. These features help you better manage common scenarios, such as:

- Keeping primary and backup data in sync
- Keep up with data changes during the development and testing cycles, and later on to deployment environments
- Manage data migrations during upgrade activities

SCHEMA COMPARISON AND SYNCHRONIZATION

DB Change Manager can **capture and compare** the database schema in your enterprise. Over the lifecycle of a single schema, you can capture a baseline or **archive schema** and analyze changes through time. In addition, DB Change Manager generates a SQL script that can **update a target schema** so that it matches a standard. You can modify the SQL and execute it as needed.

For more information, see the video demos [Schema Compare with Auto-Synchronization](#) and [Cross-DBMS Schema Compare](#). The demos were made for the 5.1 release and is still relevant.

DATABASE CONFIGURATION MANAGEMENT

In addition to schema, there are configuration settings that can vary across an enterprise. DB Change Manager can **capture the configuration settings** of a data source. You can capture a baseline or **archive configuration** and analyze changes through time. You can create a **configuration standard** that defines the norms for your enterprise and against which you can compare your databases.

For more information, see the video demo [Monitoring for Change](#). The demo was made for the 5.1 release and is still relevant.

REPORTS, NOTIFICATIONS, AND CHANGE HISTORY

DB Change Manager supports your activities with **notifications** when processes execute, **reports** that capture key information, and **change history** tracking.

For more information, see the video demo [Monitoring for Change](#). The demo was made for the 5.1 release and is still relevant.

DATA MASKING

There are specific scenarios when you need to hide the true values of the data in a database. For example, during development you need to protect personal or proprietary data, yet you still need to work with data that accurately represents the source. By combining data synchronization with data masking features, such as **randomization** or **shuffling**, you can duplicate source data and use the masked copies in non-production or non-secure environments.

For more information, see the video demo [Data Masking](#). The demo was made for the 5.1 release and is still relevant.

SOURCE CONTROL SYSTEM INTEGRATION

You can integrate DB Change Manager with source control systems such as Concurrent Versions System (CVS), Visual Source Safe (VSS), or Eclipse plugins. The **Navigator** in DB Change Manager is the interface for accessing files under source control. Job files and archive versions are stored as XML files and synchronization scripts are stored as text files. You perform all source code control actions using the source control system.

For more information, see the video demo [Developer Versioning Database and Code](#). The demo was made for the 5.1 release and is still relevant.

TECHNICAL REQUIREMENTS

Before installing DB Change Manager, verify that your environment meets the following requirements:

- 1.2 Ghz processor
- 1 GB of RAM
- 500 MB of hard disk space (more space is recommended for large data source comparisons)
- High resolution monitor (1024 x 768), 16-bit display (or greater)
- Operating systems:
 - Microsoft Windows 7 (x86-32, Win32), Windows 2008 (x86-32, Win32), Windows XP (x86-32, Win32), Windows Vista (x86-32, Win32), or Windows Server 2003
 - Red Hat Enterprise Linux 5.0 (x86-32), GTK 2, or SuSe Linux Enterprise Server (x86) GTK+ 2.x
- If you choose to install the Eclipse plugin version, you need:
 - Eclipse version 3.3 or newer
 - Sun Java Standard Edition 5.0 Update 11 or newer

DATABASE SUPPORT

DB Change Manager 5.5 supports the following database platforms:

- IBM DB2 LUW 8.0 - 9.5
- Oracle 8i - 11g
- Microsoft SQL Server 2000, 2005, and 2008
- Sybase ASE 12.5 - 15, 15.0.1, and 15.0.2
- Firebird® 1.5, 2.0
- InterBase® 2007, 2009

DB Change Manager 5.5 supports data sources that use the Unicode format.

INSTALLING AND UPGRADING DB CHANGE MANAGER

To install DB Change Manager 5.5, run the installer and follow the prompts provided by the **Installation Wizard**. When needed, DB Change Manager provides an upgrade process for moving jobs and other information from an existing DB Change Manager installation to the new version. See the DB Change Manager *Quick Start* documentation at http://docs.embarcadero.com/products/change_manager for more information.

You may have acquired an Embarcadero ToolCloud, XE, or an All-Access edition of DB Change Manager, in either full installation or InstantOn mode. For information about these options, see: http://docs.embarcadero.com/products/all_access or <http://docs.embarcadero.com/products/x>.

KEY CONCEPTS OF DB CHANGE MANAGER

To use DB Change Manager most effectively, first learn about how it represents and stores reusable processes that help you manage your data, and the features of its user interface.

- [Understanding Jobs and Scripts in DB Change Manager](#)
- [Understanding The DB Change Manager User Interface](#)

UNDERSTANDING JOBS AND SCRIPTS IN DB CHANGE MANAGER

DB Change Manager uses the concept of a job to organize reusable processes. A job uses objects of the following types:

- **Data source:** A data source is a data repository or database.

- **Configuration standard:** A standard is a customized set of properties with threshold operators that help you create optimal configurations against which you can compare other data source configurations.
- **Schema or configuration archive:** An archive is a snapshot of a configuration or schema. You can use this archive to compare against other schemas or configurations.

There are two types of jobs in DB Change Manager:

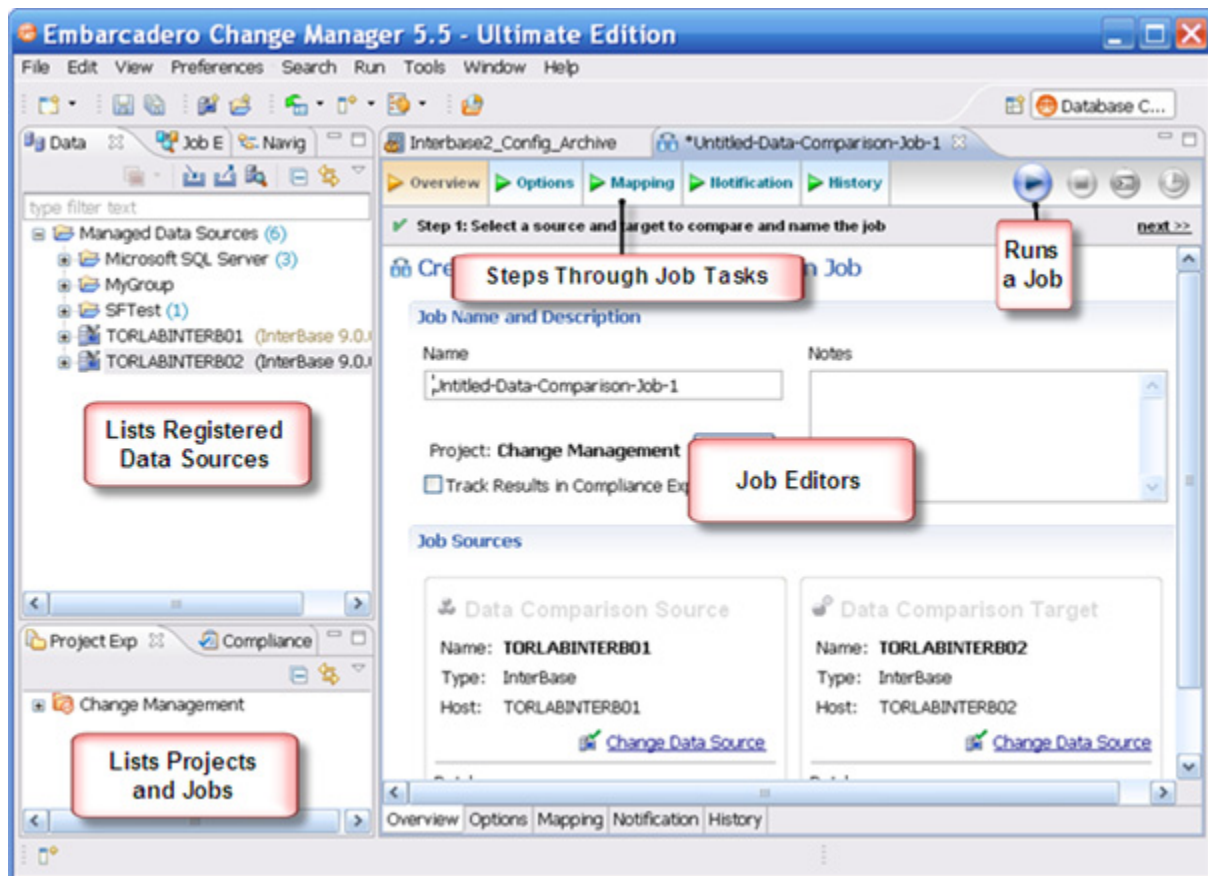
- **Object creation jobs:** Some jobs create objects that are used in other jobs. You use a job to create configuration standards and archives.
- **Comparison jobs:** A comparison job processes **source** and **target** objects and provides information describing their similarities and differences. This comparison is then used to generate SQL scripts. A comparison job can be of the following types:
 - **Data comparison:** A data comparison job looks at the data repositories of two data sources and determines their differences. You can use this information to determine whether to update a target data source.
 - **Data masking:** A data masking job is a special kind of data comparison job that creates a copy of sensitive data which masks the true values of specified fields.
 - **Configuration comparison:** A configuration comparison job determines whether the configuration settings between a source and a target are the same. This information helps you quickly diagnose problems.
 - **Schema comparison:** A schema comparison job lets you compare the schemas of different data sources or archives. It helps you manage schema changes through development, testing, and production cycles.

Once it has run a comparison job, DB Change Manager can generate an **SQL synchronization script** that will update target data, configurations, or schemas. You can modify a script before you run it. For example, you can use a DB Change Manager script to synchronize data between development and testing environments. Once you are comfortable with the job's specifications, you can let it automatically run the script and synchronize the data.

You can group jobs and scripts into projects, which are visible in the **Project** explorer and in the **Navigator** view. The **Navigator** view provides a way to interact with files in a source control system.

UNDERSTANDING THE DB CHANGE MANAGER USER INTERFACE

The **Workbench** is the DB Change Manager development environment. It provides you with an interface to create, manage, and navigate comparison jobs through interaction with views, editors, and menus. If you are familiar with Eclipse you will recognize many elements of the interface.



The **Welcome Page** is the first screen you see when you initially launch the application. The screen provides links to information that may be of value to new users.

Views help you navigate a hierarchy of information, open editors, or display the properties of various application elements. For example, the **Data Source Explorer** view provides a tree of all data sources in the environment and the comparison jobs associated with each. You can launch these jobs directly, modify the connection properties or data sources, or create and edit configuration archives from the view.

Editors help you perform various tasks such as running a job. For example, you use the **Schema Comparison Job** editor to define and execute schema comparison jobs.

The **Menu Bar** and **Command Toolbar** options provide various functions, such as launching views and editors, navigating, and setting preferences.

For more information about the interface, see the [User Interface Reference](#).

ADDITIONAL PRODUCT INFORMATION

Embarcadero Technologies web sites are an excellent source for additional product information, including white papers, articles, FAQs, discussion groups. Visit these sites:

- The DB Change Manager home page at <http://www.embarcadero.com/products/change-manager> or <http://www.embarcadero.com/products/change-manager-xe>.
- [The Embarcadero Technologies Documentation site](#)
- [The online DB Change Manager Demos](#)

WHAT YOU CAN DO WITH DB CHANGE MANAGER

This section describes what you can do with DB Change Manager, and links to more detailed information about how to perform the tasks. In addition, DB Change Manager includes step-by-step Eclipse-style cheat sheets to help you learn the tasks.

- [Commonly-Performed Tasks](#)
- [Comparing and Synchronizing Data](#)
- [Comparing and Synchronizing Schemas](#)
- [Managing Database Configurations](#)
- [Creating Masked Data for Development Environments](#)
- [Automating and Scripting Jobs](#)

COMMONLY-PERFORMED TASKS

These tasks describe basic functions and help you use DB Change Manager most effectively. In addition, you can customize the interface using preferences. See the [Preferences Reference](#) for details.

- [Registering and Organizing Data Sources](#)
- [Creating a Job Report](#)
- [Scheduling a Job](#)
- [Filtering Data Source Objects](#)
- [Searching for Database Objects and Synchronization Scripts](#)
- [Viewing Change History](#)

COMPARING AND SYNCHRONIZING DATA

Data Comparison jobs help you manage data changes and synchronize changes between databases.

- [Comparing Data](#)
- [Synchronizing Data Repositories](#)

COMPARING AND SYNCHRONIZING SCHEMAS

In DB Change Manager you can select schema objects and types, compare them across data sources, and synchronize them to each other. You can also create a snapshot or archive of the schema objects, to use in a comparison or to save for future reference. In addition, you can clone schema comparison jobs to make it easier to reuse existing jobs.

- [Comparing Schemas](#), including how to [Clone a Schema Comparison Job](#)
- [Synchronizing Schemas](#)
- [Creating a Schema Archive](#)

MANAGING DATABASE CONFIGURATIONS

DB Change Manager helps you keep track of database settings and version information such as patch levels. You can define database configuration values and ranges, and then run a configuration comparison job to check for compliance against those parameters.

- [Comparing Database Configuration Settings](#)
- [Creating a Standard for Configuration Settings](#)
- [Creating a Configuration Archive](#)

CREATING MASKED DATA FOR DEVELOPMENT ENVIRONMENTS

Protecting personally identifiable information while having accurate data for testing can be difficult. DB Change Manager lets you create a masked copy of production data for use during development and testing. This way the data is functionally correct and at the same time does not compromise privacy or security.

- [Masking Data](#)

AUTOMATING AND SCRIPTING JOBS

You may need some jobs to run on a regular basis without using Windows Task Scheduler (as described in [Scheduling a Job](#).) The Script Wizard lets you define scripts that run jobs, and batch files to run multiple scripts.

- [Scheduling a Job](#)
- [Generating Command Line Scripts](#)

WHAT IS NEW IN DB CHANGE MANAGER

General changes in this release include:

- The product name “Change Manager” is now “DB Change Manager”.
- There is now an XE version for use with Embarcadero ToolCloud.
- DB Change Manager now runs on Microsoft Windows 7 (x86-32, Win32).

DB Change Manager XE and 5.5 contain usability improvements including:

- Support for Windows Task Scheduler to schedule comparison, configuration, and schema jobs; see [Scheduling a Job](#)
- Ability to clone schema comparison jobs; see [Clone a Schema Comparison Job](#)
- Ability to switch workspaces between versions of DB Change Manager; see [Workspace](#)
- New toolbars for **Data Sources**, and **Layout & Preferences**; see [Command Toolbar icons](#)
- New **Preference** and **View** main menu items, and an updated **File** menu
- Elapsed and remaining time estimates for in-progress jobs in the **Results** tab; see [Run the Data Comparison Job](#)
- Datasource target names are links that open the individual results of schema comparison jobs; see [Run the Schema Comparison Job](#)
- Individual archived objects now have history, which is useful for comparisons; see [Review the History of Archived Objects](#)
- Ability to ignore selected objects and properties in a comparison job; see [Select Schema Properties to Ignore During a Comparison](#)
- Schema comparison jobs can now automatically include dependent objects for a table; see [CM/Schema > Comparison Options > Include Objects Options](#)
- For Oracle, the mapping page can show all users, including those without schemas in the target; see [Show Users \(Oracle\)](#)
- Cross-platform comparison and synchronization between Firebird and InterBase; see [Cross-Platform Schema Comparisons Supported by DB Change Manager](#)
- For SQL Server and Sybase databases:
 - `USE <DATABASE NAME>` statements added to scripts generated by DB Change Manager
 - Option to `SET QUOTED IDENTIFIER ON` and `OFF`; see [Set quoted identifier ON/OFF](#)
- New, easier to understand icons for **Compliance** range preferences
- Easier to use **Email Notification** preferences; see [Notification > Email Notification](#)
- Reports now include the date an archive was created

WHAT IS NEW IN DB CHANGE MANAGER >

- Quality updates for SQL scripts and schemas

DB Change Manager XE and 5.5 contain performance enhancements including:

- Faster schema archive jobs and reports
- Option to turn off DDL in schema archive and comparison reports
- Faster automatic data synchronization jobs

REGISTERING AND ORGANIZING DATA SOURCES

DB Change Manager can register data sources by scanning your systems on startup or as needed. Also, instead of scanning, you can provide the information for one data source at a time.

Once registered, the data source appears in the **Data Source Explorer** view. If you have created more than one workspace, they all share the same data source catalog. You can organize data sources into groups and subgroups.

Once a data source has been registered, the connection parameters are stored locally. In some cases, a user ID and password are required to connect to a registered data source. DB Change Manager can encrypt and save user IDs and passwords to connect automatically.

You can import an existing data source catalog from other Embarcadero products, or from third-party DBMS sources. You can also export a data source to an XML file. These topics include:

- [Registering Data Sources on Startup](#)
- [Scanning for Data Sources and Registering Them](#)
- [Registering a New Data Source One at a Time](#)
- [Grouping Data Sources](#)
- [Importing and Exporting Data Sources](#)

REGISTERING DATA SOURCES ON STARTUP

When DB Change Manager first starts, it prompts you to discover data source catalogs that have been created by any previously installed Embarcadero products (DBArtisan, Rapid SQL, DB Optimizer), or other instances of DB Change Manager.

Additionally, the system can scan your machine for the client software of all supported third-party DBMS platforms (TOAD or Eclipse Data Tools Platform) and add them to the data source catalog.

If you do not want to do this on startup, at a later time you can scan for data sources, or add them one at a time.

SCANNING FOR DATA SOURCES AND REGISTERING THEM

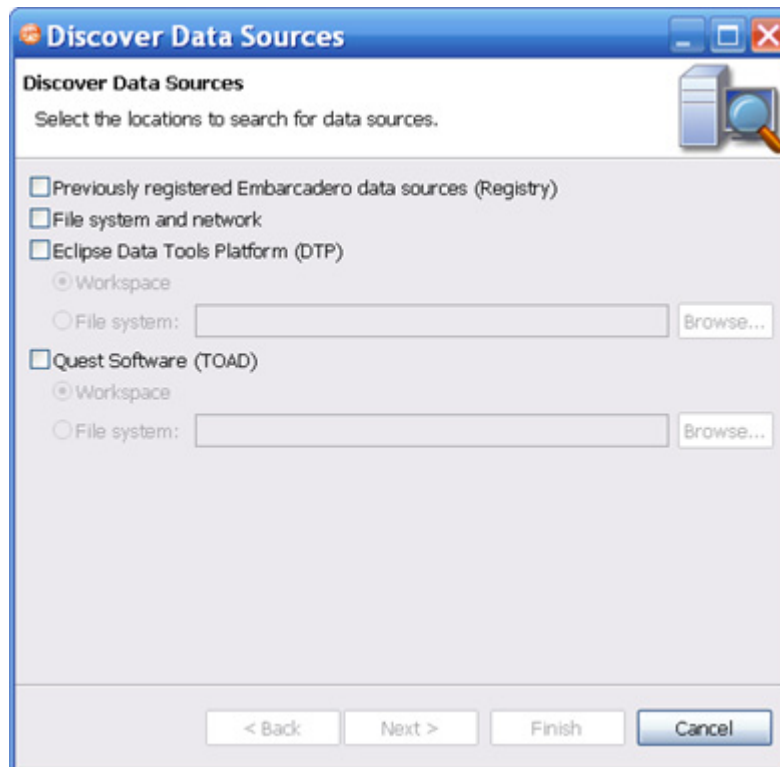
DB Change Manager can register data sources by scanning your systems.

To initiate a scan for data sources

- 1 Click the **Discover Data Sources** icon at the top of **Data Source Explorer**.



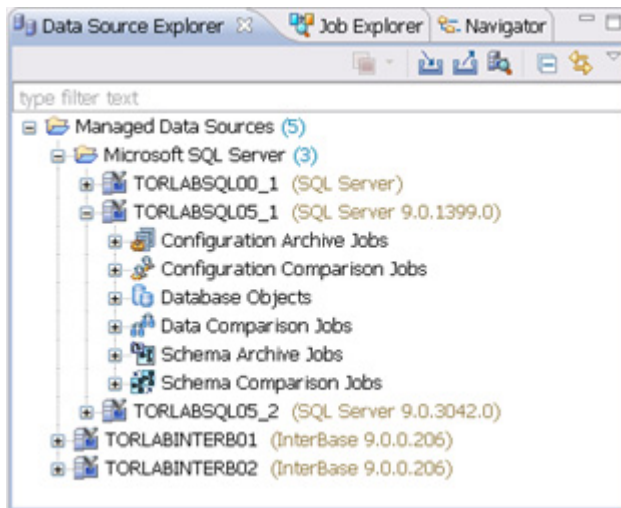
- 2 In the **Discover Data Sources** dialog, choose the type of data sources to return in the search results.



- 3 Click **Next**. The search returns the data sources it discovers.



- 4 Choose the data sources you want to add to the DB Change Manager environment and click Finish. **Data Source Explorer** automatically populates with the new data source selections.

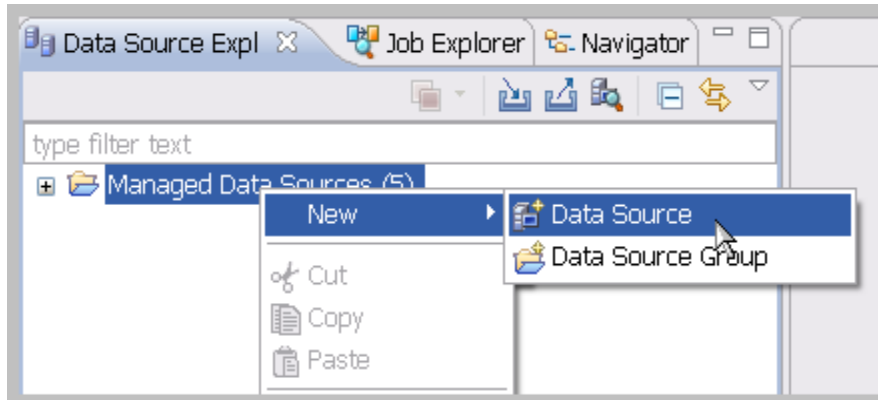


REGISTERING A NEW DATA SOURCE ONE AT A TIME

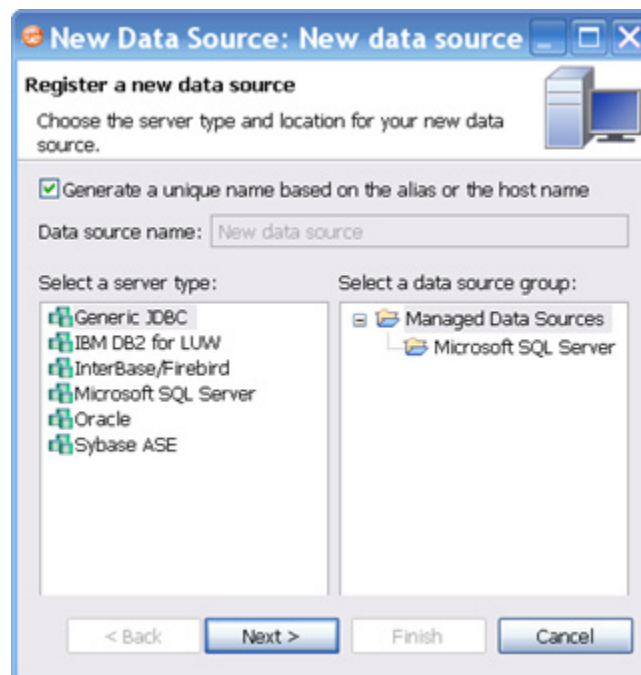
If a data source is not present in **Data Source Explorer**, you can add it.

To register a new data source

- 1 In **Data Source Explorer**, right-click the Managed Data Sources folder and select **New > Data Source**.



- 2 In the **New Data Source** wizard, select the data source server platform.



- 3 If you have previously set up a group want to add this data source to the group, select the group. Otherwise leave the default selection, **Managed Data Sources**.
- 4 Click **Next**.
- 5 Enter the connectivity and login information.

- Click **Test Connection** to confirm that the configuration information is correct.



- Click **Finish**. **Data Source Explorer** lists the new data source.

GROUPING DATA SOURCES

You can organized data sources in **Data Source Explorer** into groups and subgroups, which appear as folders. Groups help you manage a large number of data sources.

To create a data source group

- In **Data Source Explorer**, right-click the **Managed Data Sources** folder and select **New > Data Source Group**.
- Enter a meaningful name for the new group in the **Name** field.
- Click **OK**.

Data Source Explorer lists the new group.

You can drag and drop data sources into the group folder, or add new data sources to the folder.

To create a data source subgroup

- 1 In **Data Source Explorer**, right-click the group folder and select **New > Data Source Group**.
- 2 Enter a meaningful name for the new group in the **Name** field.
- 3 Click **OK**.

IMPORTING AND EXPORTING DATA SOURCES

You can import an existing data source catalog from other Embarcadero products, or from third-party DBMS sources, including:

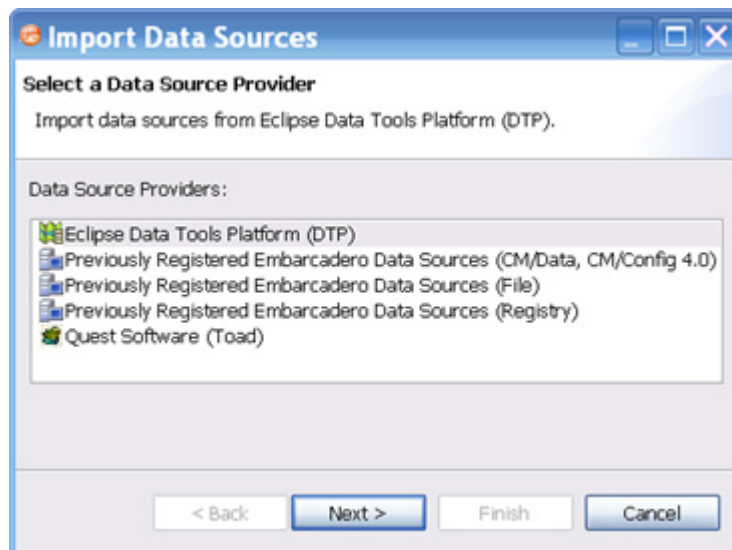
- Eclipse Data Tools Platform (DTP)
- Previously Registered Embarcadero Data Sources (File)
- Previously Registered Embarcadero Data Sources (Registry)
- Quest Software (TOAD)

Conversely, you can export your current data source catalog to a file, which can then be imported into other instances of DB Change Manager using the **Previously Registered Embarcadero Data Sources (File)** option.

NOTE: If you experience problems importing data sources from other Embarcadero products, you can import the old data sources using your Windows registry by selecting **File > Import... > Embarcadero > Data Sources > Previously Registered Embarcadero Data Sources (Registry)**.

To import data sources

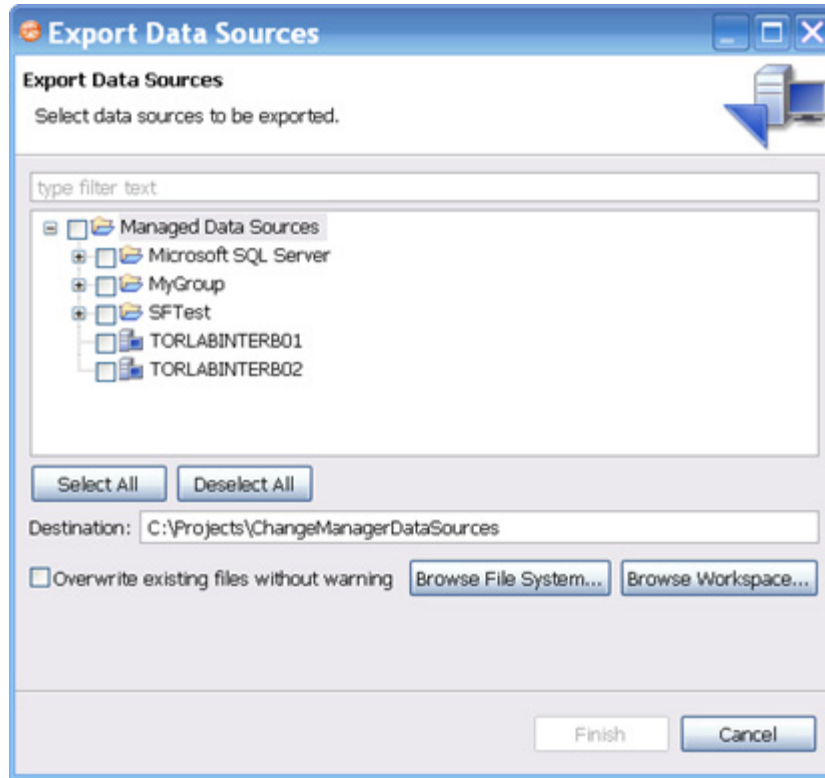
Select **File > Import > Data Sources**.



- 4 Choose a source from which you want to import the data sources.
- 5 Click **Next**.
- 6 Specify the details about the location of the import source.
- 7 Click **Finish**. **Data Source Explorer** is automatically populated with the new data sources.

To export data sources

- 1 Select **File > Export > Data Sources**.



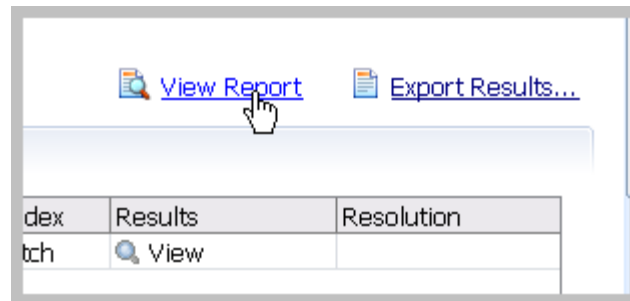
- 2 Browse the list and select a data source to export.
- 3 Type or use the **Browse File System** or **Browse Workspace** buttons to enter the directory path and file name.
- 4 Click **Finish**. The data source is are exported as an XML file. You can import this file to other instances of DB Change Manager.

CREATING A JOB REPORT

After you run a job, you can see its default report, and you can export a report in various formats.

To view or print a default job report

- On the **Results** tab, click the **View Report** button.



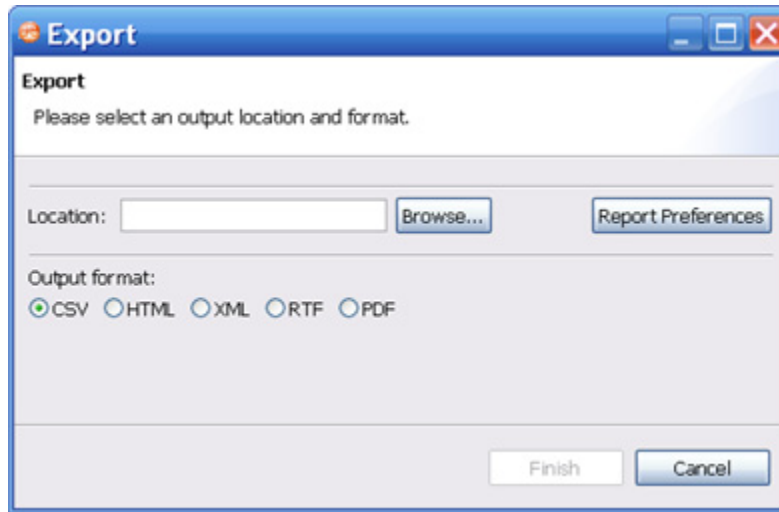
The report opens in a separate window and can be read on screen, sent to a printer, and saved to disk.

To save a report in a particular file format

- 1 On the **Results** tab, click the **Export Results** button.



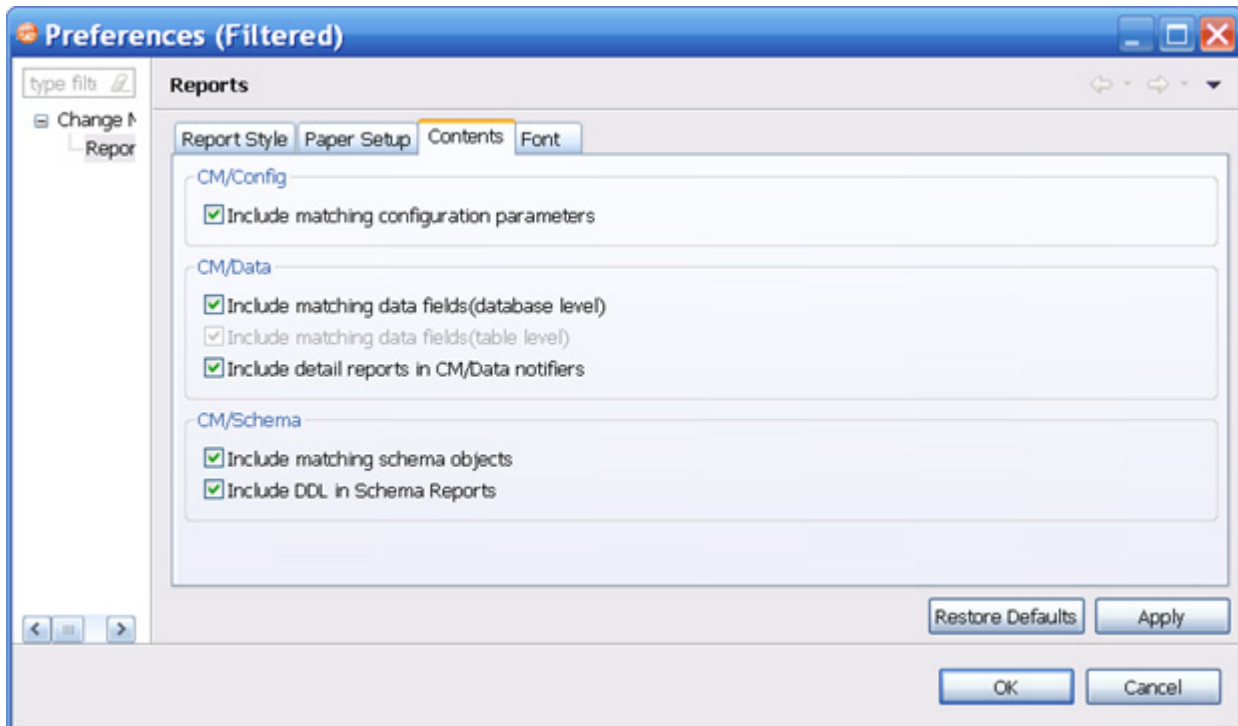
- 2 In the **Export** dialog, select a directory location where the report will be saved.



- 3 Click an output format.
- 4 Click **Report Preferences**.
- 5 In the **Report Preferences** tabs, select the visual properties of the report including style, paper size, and font.

For details about the preferences, see [Reports](#).

- 6 In the **Contents** tab, specify the criteria to include in the report.



For configuration comparison jobs:

- Select **Include matching configuration parameters** to include configuration parameters and values that matched when the job was run.

For data comparison jobs:

- Select **Include matching data fields (database level)** to include tables and/or views that matched in a data comparison job. If this option is not checked, only comparison tables and/or views that did not match will be included.
- Select **Include matching data fields (table level)**, after deselecting the database level option, to include data fields that matched in rows, for which one value did not match. If this option is not checked, only comparison key values and values that did not match will be included.
- Select **Include detail reports in CM/Data notifiers** to include information for all data fields selected in a data comparison job.

For schema comparison jobs:

- Select **Include matching schema objects** to include information about the objects that matched in a schema comparison job.
- Select **Include DDL in Schema Reports** to include information about the DDL.

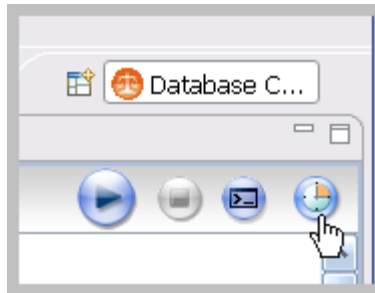
TIP: A schema report runs faster if you do not generate the DDL in the report.

SCHEDULING A JOB

You can use Windows Task Scheduler to run jobs on a regular schedule.

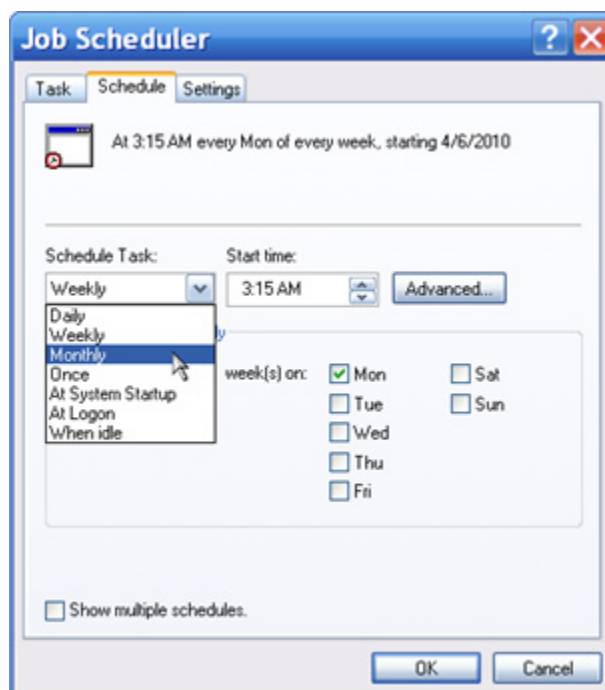
To schedule a job

- 1 In a job editor, click the **Schedule Job** button.



TIP: You can also open the **Job Scheduler** from the **Tools > Schedule Job** menu, or right-click on any job and select **Schedule Job**.

- 2 In the informational dialog, click **OK**.
- 3 In the **Schedule** tab, specify when and how often you want the job to run.



- 4 Click **OK**.
- 5 In the dialog, verify your user name and type your Windows password.

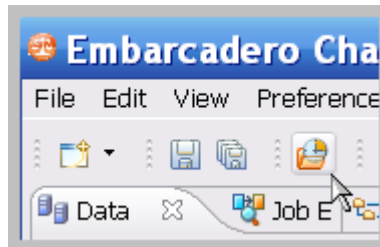
- 6 Click **OK**.

The **Job Scheduler** gives you additional options on the **Settings** tab.



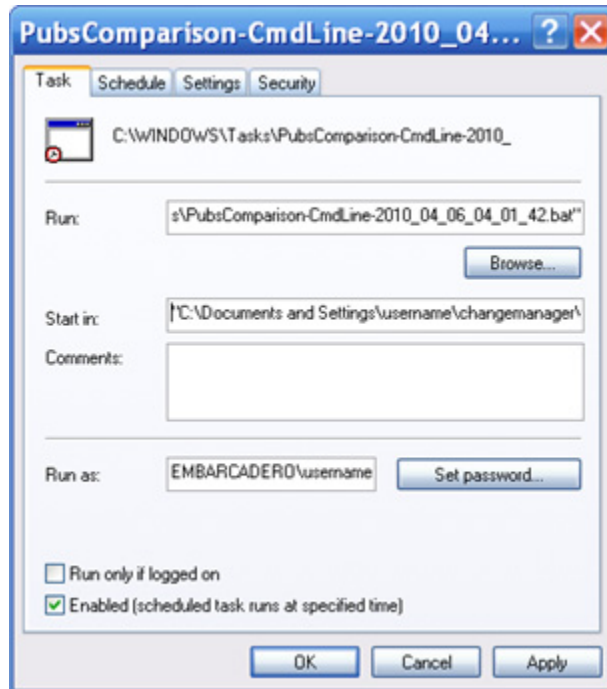
To modify an existing job schedule

- 1 In the toolbar, click the **Open Scheduled Tasks** button.



- 2 In the **Scheduled Tasks** list, double-click the job you want to change.

- 3 In the **Job Scheduler** window, view the existing settings and change them as needed.



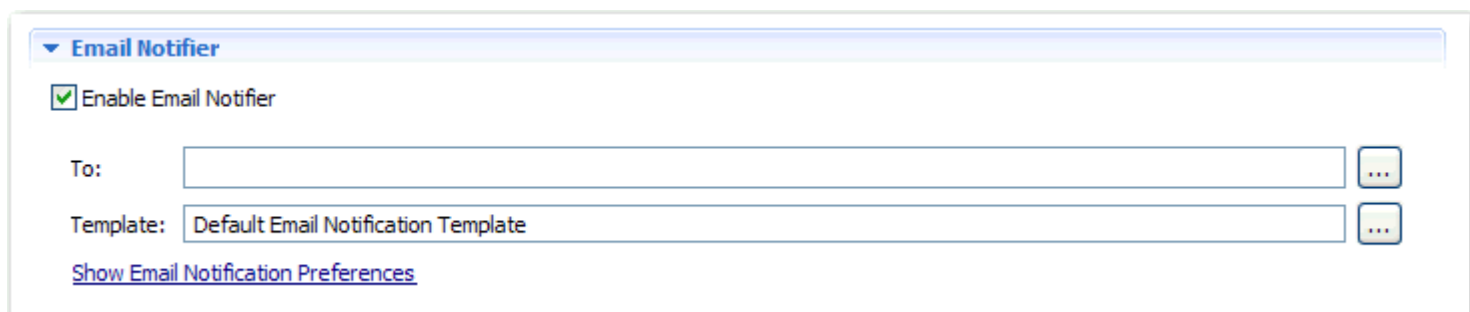
SETTING JOB NOTIFICATIONS

When a job completes, DB Change Manager provides various ways of notifying you, including:

- [Email Notifications](#)
- [File System Notifications for Log Files](#)
- [System Tray Notifications](#)
- [Send Notification and Attachment Options](#)

EMAIL NOTIFICATIONS

You can configure notification emails, with optional reports, to be sent when a job runs. You can specify the email addresses, the email message format, the conditions under which a notification email will be sent.



The screenshot shows a configuration window titled "Email Notifier". It features a checked checkbox labeled "Enable Email Notifier". Below this are two input fields: "To:" and "Template:". The "To:" field is empty, and the "Template:" field contains the text "Default Email Notification Template". Both fields have a three-dot menu icon to their right. At the bottom of the window, there is a link that says "Show Email Notification Preferences".

To set up email notifications

- 1 Click **Enable Email Notifier** to enable email notification. When you select this check box, a list of notification parameters appear.
- 2 Enter the email addresses to which the notification will be sent in the **To** field.
- 3 Select an email format template in the **Template** field.
- 4 Specify the [Send Notification and Attachment Options](#).

FILE SYSTEM NOTIFICATIONS FOR LOG FILES

You can configure logging options in the **File System Notifier**, to specify when the log file is generated and where it is saved.

File System Notifier

Enable File System Notifier

Report Directory:

Log Directory:

Script Directory:

[Show File System Notification Preferences](#)

To set up file system notifications

- 1 Click **Enable File System Notifier** to enable reports and logs.
- 2 Browse to and select a directory location for each of the output file in the **Report Directory**, **Log Directory**, and **Script Directory** fields.
- 3 If you want to change the directory locations at the application level rather than just the job level, click **Show File System Notification Preferences**.
- 4 Specify the [Send Notification and Attachment Options](#).

SYSTEM TRAY NOTIFICATIONS

When you are running jobs in the background, DB Change Manager can send Windows system tray notifications to your desktop.

System Tray Notifier

Enable System Tray Notifier

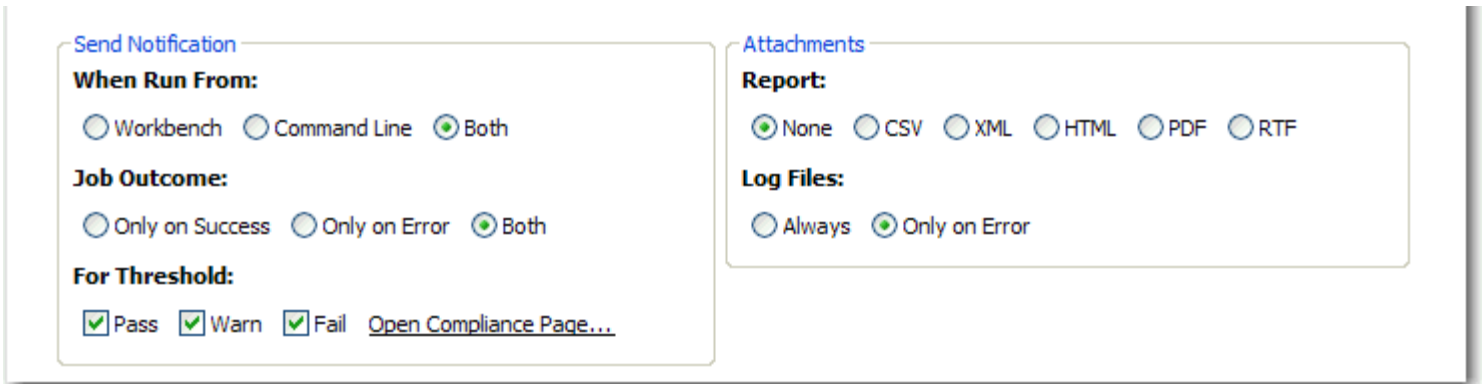
System tray is available only in minimized mode when 'Show only when minimized to tray' is checked. [Show System Tray Notification Preferences](#)

To set up system tray notifications

- 1 Click **Enable System Tray Notifier**.
- 2 Click **Show System Tray Notifications** to specify the appearance and behavior of the specific pop-up windows.
- 3 Specify the [Send Notification and Attachment Options](#).

SEND NOTIFICATION AND ATTACHMENT OPTIONS

Email, file system, and system tray notifications have options that control when and how notifications are sent.



Send Notification

When Run From:

Workbench Command Line Both

Job Outcome:

Only on Success Only on Error Both

For Threshold:

Pass Warn Fail [Open Compliance Page...](#)

Attachments

Report:

None CSV XML HTML PDF RTF

Log Files:

Always Only on Error

To specify when notifications are sent

- Select the **Send Notification** options, including:
 - The **When Run From** options specify that email notifications will be sent depending on how the job is run. By default notifications are sent either way.
 - The **Job Outcome** options specify that email notifications will be sent depending on whether the job is successful or not, or either way. .
 - The **For Threshold** options enable you to specify the threshold level for which email notifications are sent, based on job results.

You can also click **Open Compliance Page** to specify the percentage value of each of the threshold warning levels. These values are based on the percent match results of a given comparison job.

To specify the format of notification files

- Select the **Attachments** options, including:
 - The file format of an attached report that will display job results with the notification email.
 - The **Log Files** specifies whether the DB Change Manager log file will include all logs or only when there are errors.

FILTERING DATA SOURCE OBJECTS

You can specify a filter on data sources and their objects to control how they are viewed in **Data Source Explorer**. If you have data sources that contain large numbers of database objects, you can apply filters to view only the ones you need.

There are two types of data source filters:

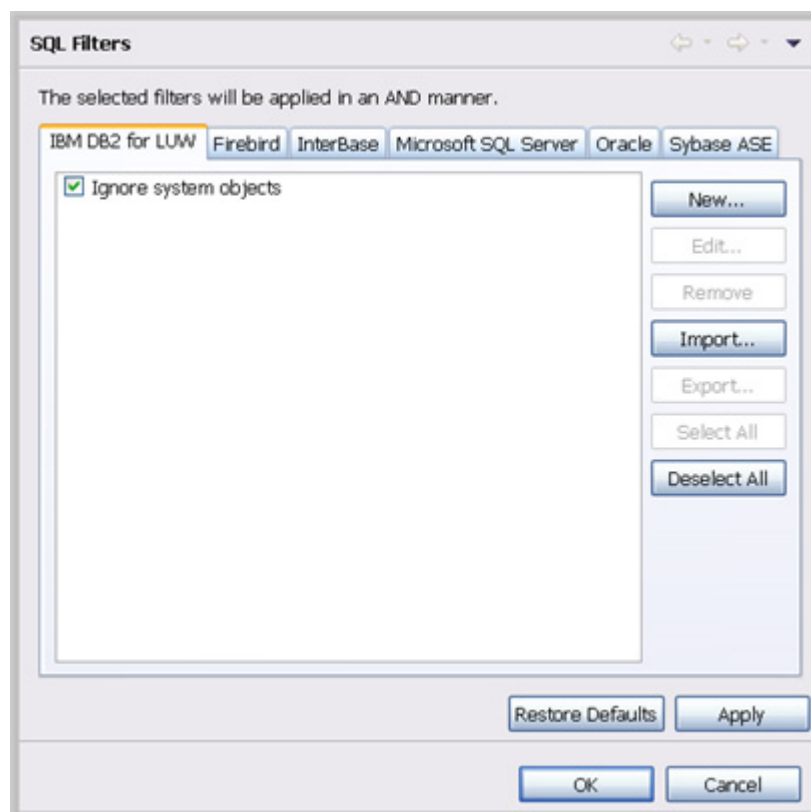
- Global filters affect all registered data sources.
- Data source specific filters affect only the specified data source.

You can create one or more filter templates and apply them to one or more data sources. You can activate and deactivate filter templates as needed. You can use multiple filter templates on global or specific data sources.

To define a global filter

- 1 From the Main menu, select **Preferences > SQL Filters**.

The **SQL Filters** pane displays global filter templates in tabs organized by platform. You can create and apply a global filter to a database platform such as SQL Server, and all databases associated with that platform will use the filter. By default all filters ignore system objects.

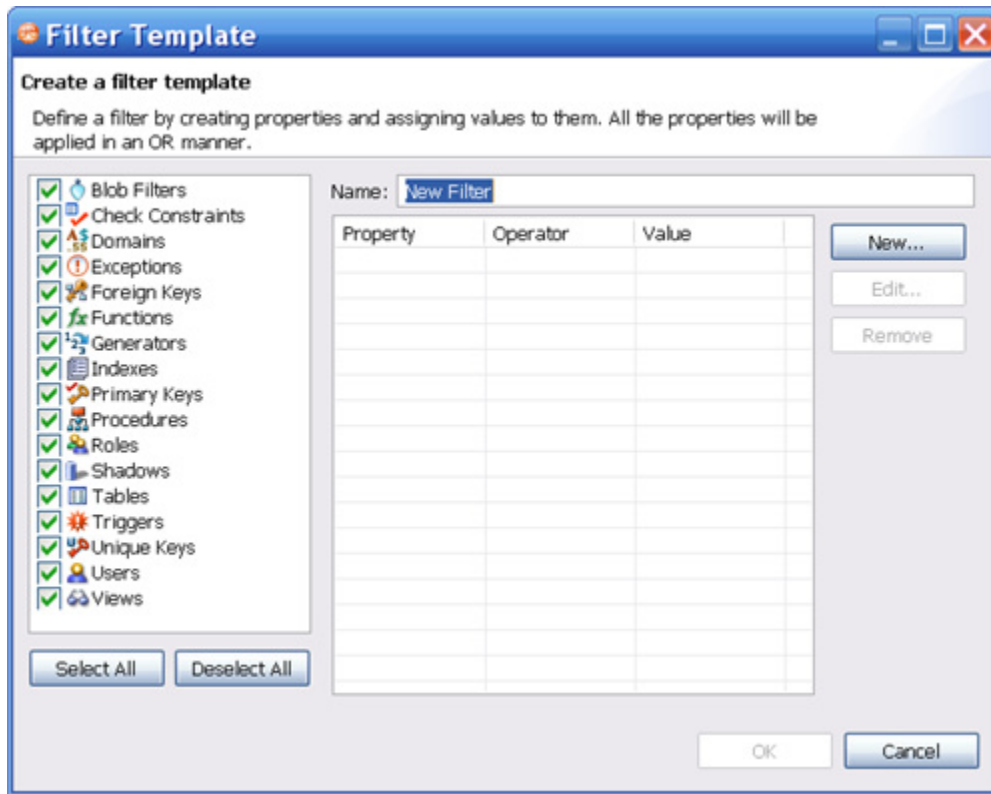


- 2 Click **New**. The **Filter Template** dialog appears.
- 3 See [To create a filter template](#).

To define a data source specific filter

- 1 In **Data Source Explorer**, right-click the data source and select **Properties**.
- 2 Select the **SQL Filter** node.
- 3 Select **Enable data source specific settings**. The other controls on the dialog become enabled.
- 4 Click **New**. The **Filter Template** dialog appears.
- 5 See [To create a filter template](#).

To create a filter template



- 1 Specify the parameters of the filter template:
 - In the **Name** field, enter the name of the filter that will appear in the **SQL Filter** list.
 - The **Database Type** pane on the left lists data source objects. Deselect the data source objects that you do not want to see in **Data Source Explorer**.
 - Click **New** to add filter parameters for data source objects properties. The **New SQL Filter Predicate** dialog appears.
 - Use the **Property** and **Operator** fields to supply the filter criteria. **Property** specifies if the value is a name, schema, or specific. **Operator** specifies the matching criteria: Equals, Not Equals, Like, Not Like, In, or Not In.
 - In the **Value** field, enter the full or partial syntax of the property or properties you want the template to filter.
- 2 Click **OK**.
- 3 Repeat the previous step to add more filter parameters.
- 4 When you have finished defining the filter template, click **OK**.

The **SQL Filters** pane lists the new template.

COMPARING DATA

You can create a data comparison job to compare the data of two registered data sources, specify configuration adjustments and refine mapping parameters. After running the job, DB Change Manager can generate SQL scripts which you can run to synchronize any discrepancies. You can run the scripts to update data sources immediately, or store the scripts for later use.

DB Change Manager displays data comparison job results at row level, and it can automatically update mismatches or add missing rows at the table level.

The following tasks step you through creating and running a data comparison job:

- 1 [Create a Data Comparison Job](#)
- 2 [Describe and Save the Job](#)
- 3 [Specify the Source and Target Data Sources](#)
- 4 [Specify Execution, Mapping, Comparison, and Resolution Options](#)
- 5 [Understand the Data Source Mapping Process](#)
- 6 [Customize Database Mapping](#)
- 7 [Inspect the Databases](#)
- 8 Optionally [Reconfigure the Data Source Mapping](#)
- 9 Optionally request email, logs, or system tray notifications; see [Setting Job Notifications](#)
- 10 [Run the Data Comparison Job](#)
- 11 [Review Comparison Results](#)
- 12 Optionally generate a job report; see [Creating a Job Report](#)

After comparing the data, you can synchronize it to match; see [Synchronizing Data Repositories](#).

CREATE A DATA COMPARISON JOB

You create, modify, and run a Data Comparison job in the **Data Comparison Job** editor. For more details about the editor, see [Data Comparison Job Editor](#) in the [User Interface Reference](#).

To create a new data comparison job

- Select **File > New > Data Comparison Job**.

The **Data Comparison Job** editor opens.

To open an existing data comparison job

- Select **File > Open Job**.
- Select a data comparison job from the list.
- Click **OK**.

The **Data Comparison Job** editor opens.

DESCRIBE AND SAVE THE JOB

The **Job Name** and **Description** pane contains more than just the **Name** field. It shows where the job is stored in **Project Explorer**. When you click the **Details** button, you can see the project's directory path.

If the job will be used to monitor compliance against company standards or policies, you can add it to the [Compliance Explorer](#) list. This list lets you quickly see all the jobs that have recently passed or failed the compliance criteria you set.

Step 1: Select a source and target to compare and name the job

Create or Modify a Data Comparison Job

Job Name and Description

Name: Compare SQL Server Data

Notes

Project: Change Management

Track Results in Compliance Explorer

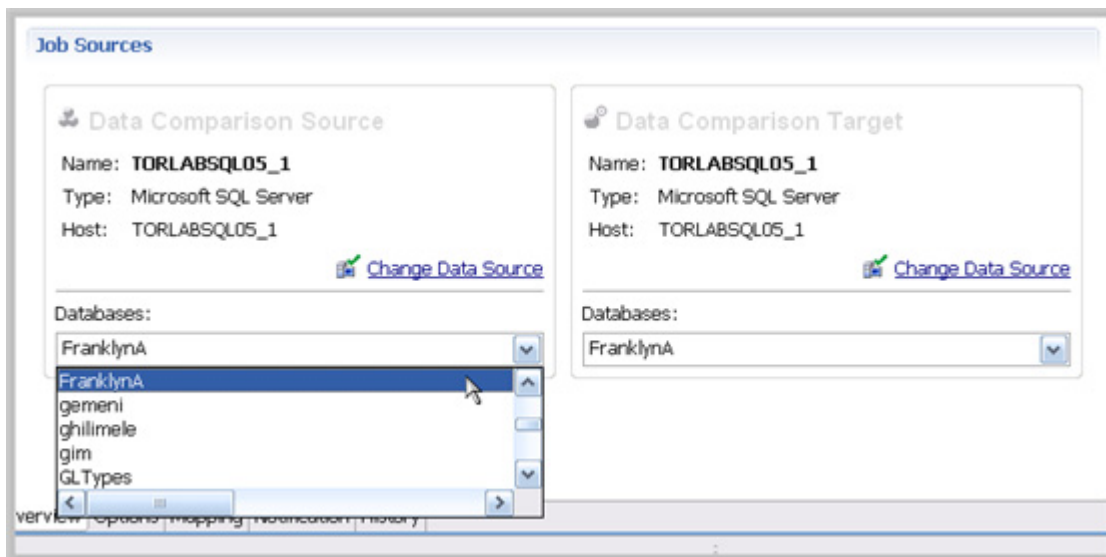
Details...

To describe and save a job

- 1 Type a new name for the job in the **Name** field. Enter a meaningful name that will quickly and uniquely identify the job in other views and dialogs.
- 2 Add relevant information in the **Notes** field.
- 3 Click **Track Results in Compliance Explorer** if you want the job to be listed in the [Compliance Explorer](#). Compliance is used most commonly for configuration comparisons.
- 4 Click **File > Save** or **File > Save All**, or highlight the editor window and press **Ctrl+S**.

SPECIFY THE SOURCE AND TARGET DATA SOURCES

The **Data Comparison Source** and **Data Comparison Target** boxes identify what data sources are compared when the job runs. When a data source is added to a job, its name, database type, and host are displayed. The **Databases** drop down menu lets you select from a list of the databases on the source.



To add a data source to a job

- 1 Click and drag a data source from **Data Source Explorer** to the **Comparison Source** or **Comparison Target** box, or click **Select Data Source** in the appropriate box and choose a data source from the dialog.

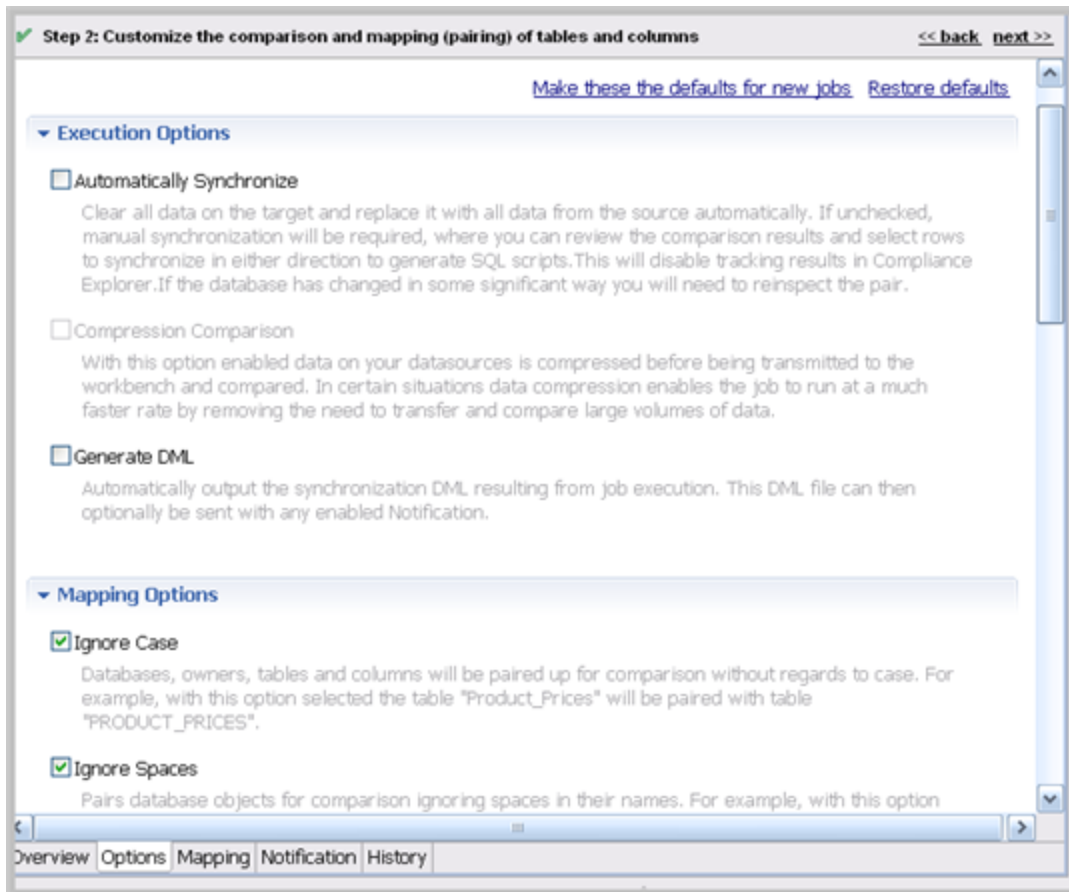
NOTE: You can change a data source selection by clicking **Change Data Source** from the appropriate box, or by dragging a different data source to the box from **Data Source Explorer**.

- 2 Select the databases, or tables and views (depending on the type of database), from the **Databases** pulldown menu.

- 3 After you name the job and specify the data sources, click **Options**.

SPECIFY EXECUTION, MAPPING, COMPARISON, AND RESOLUTION OPTIONS

You can set various options that control how a job is processed on the **Options** tab.



To set data comparison job options

- 1 Scroll down to see all the options and their descriptions, and select or deselect them as needed.

If an option is not available for the data source, the check box will be greyed out.

The options are also described in the [DB Change Manager Preferences](#) section:

- [Execution Options](#)
- [Mapping Options](#)
- [Comparison Options](#)

- [Resolution Options](#)

2 When you are satisfied with the options, click **Mapping**.

To save options as defaults for future jobs

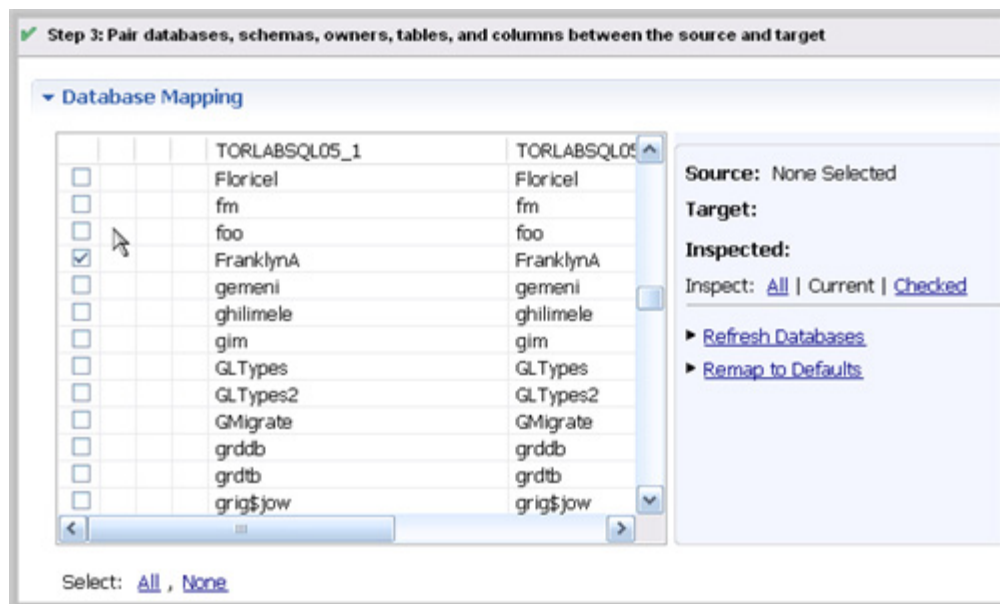
- 1 Set the options.
- 2 Click **Make these the defaults for new jobs**.

UNDERSTAND THE DATA SOURCE MAPPING PROCESS

Mapping is the process of pairing data source elements between the source and target of a job. These elements (databases, tables, and rows) are compared when the job runs to determine matches and mismatches between the two sources. Mapping details are shown on the **Mapping** tab of the editor.

CUSTOMIZE DATABASE MAPPING

When you first click into the **Mapping** tab, the **Database Mapping** table lists the two data sources and lists all of their databases. If you previously selected a database in the **Overview** tab, it will be selected. You can scroll right to see the **Last Inspected** column, which indicates that the pair has not yet been analyzed, or inspected, by DB Change Manager. The right side of the pane provides similar information.



To select databases to compare

- Use the check boxes to select the databases you want to compare.
- The **Select All** or **None** option is helpful if there are many databases in the list.

To change the default target database

- 1 In the database row, select the target.
- 2 From the the dropdown menu, select a new database.

INSPECT THE DATABASES

Before you can see the default, initial table mappings, you run an inspection process.

To run the database comparison inspection

- On the right side of the **Database Mapping** pane, click one of the following **Inspect** options:
 - **All** inspects all pairs in the table. This is not recommended if you have a large number of databases.
 - **Current** inspects the current object mappings.
 - **Checked** inspects only those pairs selected in the table.

TIP: Alternatively, you can select a row and in the **Table Mapping** pane click **Yes, retrieve the objects in this pair**.

Once inspected, icons in the the table that indicate how much of the selected objects matched. Hover over the icon to see more information.

<input type="checkbox"/>		too	too
<input checked="" type="checkbox"/>		FranklynA	FranklynA
<input type="checkbox"/>		gemeni	gemeni

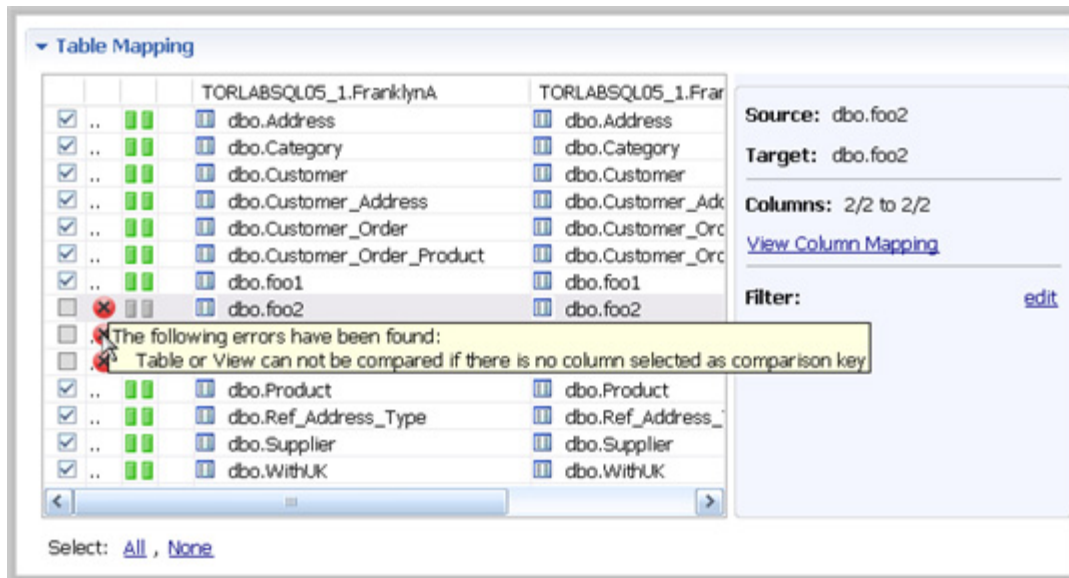
To restore the Database Mapping table to its original settings

- In the **Database Mappings** pane, click **Remap to Defaults**.

All settings revert to the original object mappings saved with the job, based on the options you selected in the **Mapping** tab.

CUSTOMIZE TABLE MAPPING

The **Table Mapping** pane lists the paired tables. The icons indicate whether mappings were complete, and if not, error icons indicate what problems were encountered, if any.

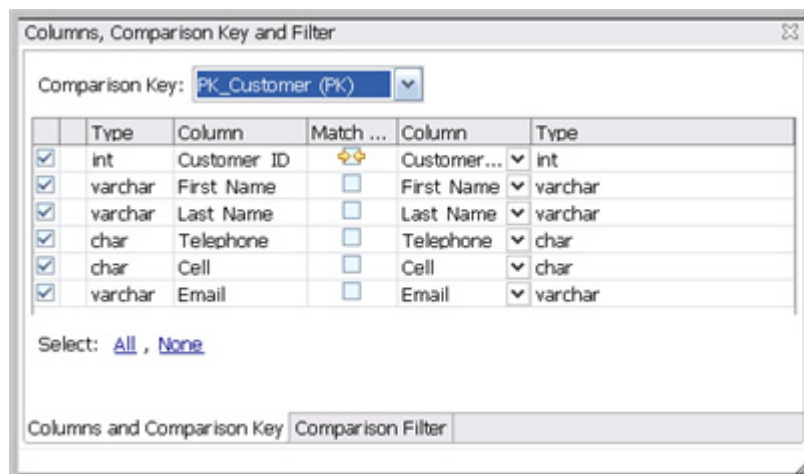


To change the mapping to a new target column

- 1 In the table row, select the target.
- 2 From the the dropdown menu, select a new target column.

To view or update the column mappings of a pair of tables

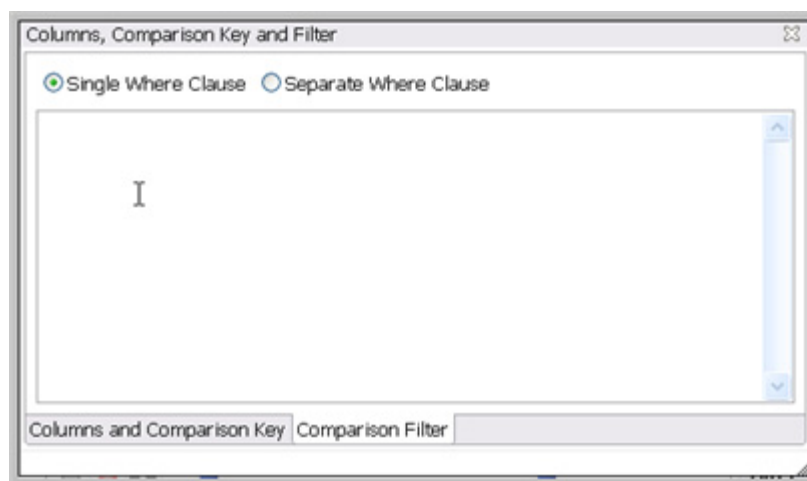
- 1 Select a row by highlighting it with the cursor. (The check box is irrelevant for this task.)
- 2 On the right side of the pane, click **View Column Mapping**.



In the **Columns and Comparison Key** pane, you can:

- Select or deselect columns to use in the comparison job.
- Change or add a comparison key if multiple keys are available. In the **Comparison Filter** pane, you can:
- Write a WHERE clauses to use during the comparison.
- Create a different WHERE clause per table in the pair. Select **Separate Where Clause** and an additional text box is added for a second WHERE clause.

NOTE: When entering selective WHERE clauses, do not include the WHERE keyword; for example, use "AGE>12" instead of "WHERE AGE>12".



RECONFIGURE THE DATA SOURCE MAPPING

After configuring and running a comparison job, you may discover that you need to set different mapping options. You can reconfigure the way DB Change Manager maps data source elements and then run the job again.

To reconfigure the automatic mapping process

- 1 In the **Options** tab, select or deselect **Mapping Options**. The options are also described in the [DB Change Manager Preferences](#) section [Mapping Options](#).

TIP: If you change the mapping options in **Preferences**, they will become the defaults for all new jobs.

- 2 In the **Mapping** tab, click **Refresh Databases and Owners** to update the automatic comparison.

SET JOB NOTIFICATION OPTIONS

When a job completes, DB Change Manager provides various ways of notifying you. See [Setting Job Notifications](#) for details.

RUN THE DATA COMPARISON JOB

After you have specified the options and mappings for a data comparison job, you can run it.

To run a job

- 1 Click **File > Save** to save the job before you run it.
- 2 In the upper right of the **Data Comparison Job** editor, click the **Run** icon.

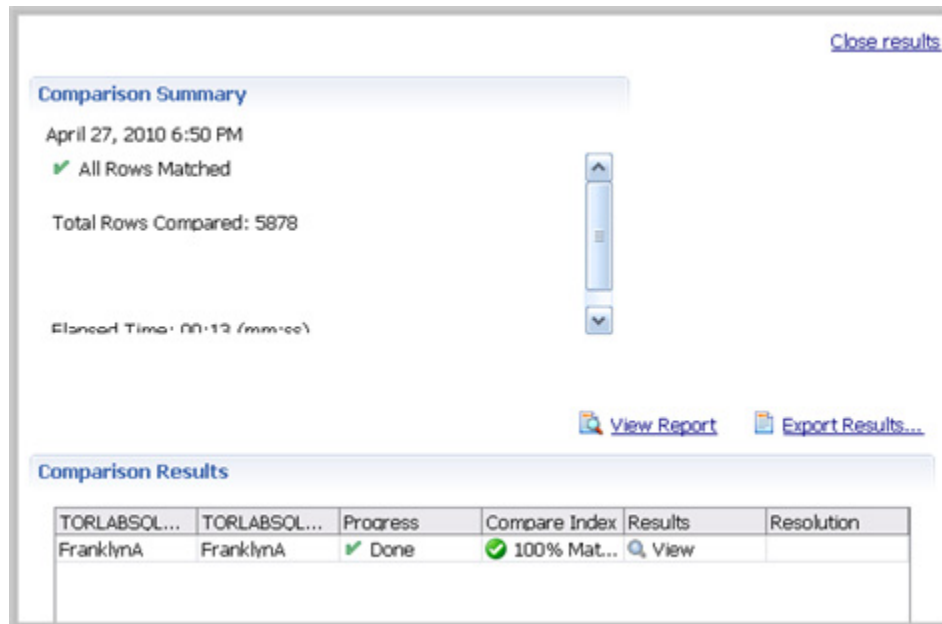


While the process runs, the **Results** tab appears and displays the **Comparison Summary** pane. This information includes:

- Source and target data source names
- When the job started
- Estimated remaining time for the job to run
- Elapsed time of the job
- The names of objects currently being compared
- A progress bar showing percentage completion

REVIEW COMPARISON RESULTS

After a job run completes, the **Results** tab shows the **Comparison Results** table.



The screenshot displays the 'Comparison Results' interface. At the top right is a 'Close results' link. Below it is a 'Comparison Summary' section with the following details:

- Date and Time: April 27, 2010 6:50 PM
- Status: ✓ All Rows Matched
- Total Rows Compared: 5878
- Elapsed Time: 00:12 (mm:ss)

Below the summary are two buttons: 'View Report' and 'Export Results...'. The main section is a table titled 'Comparison Results' with the following data:

TORLABSOL...	TORLABSOL...	Progress	Compare Index	Results	Resolution
FranklynA	FranklynA	✓ Done	✓ 100% Mat...	View	

The **Comparison Results** table lists:

- The source and target databases
- **Progress** status; either **Done**, **Failed**, or, if no pairs were specified, **Nothing to Compare**
- The **Compare Index** percentage of matches
- The **Results** column **View** icon links to details about the comparison job

After running a data comparison job, you may decide to synchronize one to the other. For more information, see [Synchronizing Data Repositories](#).

SYNCHRONIZING DATA REPOSITORIES

After a data comparison job completes, you can use synchronization commands to match the data between repositories. You specify how you want the data to be resolved, and DB Change Manager generates UPDATE, INSERT, and DELETE statements for the relevant databases. Then you run the scripts to synchronize the data.

- [Select Data to Resolve](#)
- [Run the SQL Synchronization Script](#)

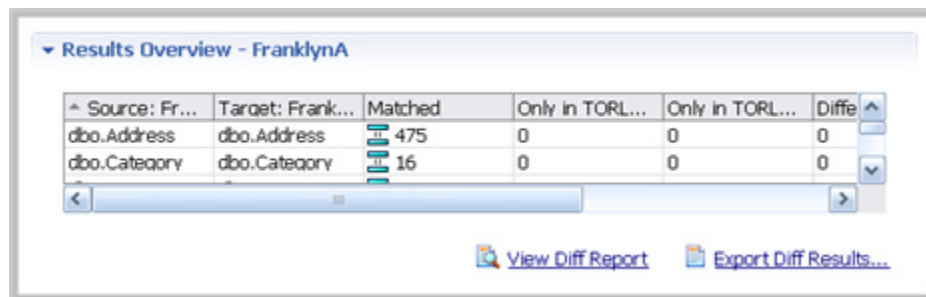
SELECT DATA TO RESOLVE

If data is inconsistent between the sources, you can synchronize them.

To view columns and rows to resolve

- On the **Results** tab in the **Comparison Results** table, click **View** in the row you want to resolve. (See [Review Comparison Results](#) for context.)

The **Database Results** tab appears, and the **Results Overview** table displays the pairings in the selected tables.



The screenshot shows a window titled "Results Overview - FranklynA". It contains a table with the following data:

Source: Fr...	Target: Frank...	Matched	Only in TORL...	Only in TORL...	Diffe
dbo.Address	dbo.Address	475	0	0	0
dbo.Category	dbo.Category	16	0	0	0

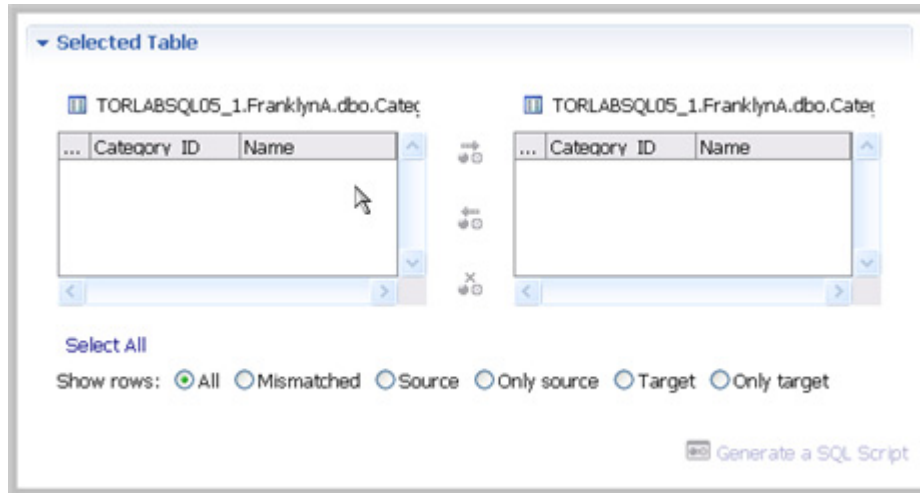
Below the table are two buttons: "View Diff Report" and "Export Diff Results...".

The **Results Overview - Activation** pane includes:

- The source and target database columns
- The number of matched rows between the source and target.
- The rows that exist only in the source database.
- The rows that exist only in the target database.
- The number of differences between the two databases.

To view data in the columns

- 1 Select a row from the **Results Overview** table.
- 2 Examine the row information in the **Selected Table** pane.

**To specify how the data synchronization is resolved**

- 1 Under the **Selected Table** pane, click **All**, **Mismatched**, **Source**, **Only Source**, **Target**, or **Only Target**.
- 2 In the **Data Diff** view, choose a **Row Synchronization** option:
 - **Make Target Match Source for Selected Row**
 - **Make Source Match Target for Selected Row**
 - **Do Not Resolve Selected Row**

To generate an SQL script

- 1 At the bottom right of the **Database Results**, click **Generate an SQL Script**.
- 2 In the **SQL Editor**, review the script and edit it as needed. For more information, see [SQL Editor](#).

RUN THE SQL SYNCHRONIZATION SCRIPT

You can run synchronization scripts in the **SQL Editor**. For more information, see [SQL Editor](#).

To run the SQL synchronization script

- In the **SQL Editor**, click the **Execute SQL** icon. DB Change Manager executes the code on the data source you specified. Results are displayed in the same tab or in a new tab.

After running a comparison job, you can create a job report using the **View Report** command on the **Results** tab. See [Creating a Job Report](#) for more information.

COMPARING DATABASE CONFIGURATION SETTINGS

Configuration comparison jobs compare the configuration settings between two registered data sources. They help you ensure consistency between the settings of two data sources in order to meet guidelines for data storage and retention.

Configuration comparison job results are presented as an overall “percentage matched” between two data sources. The results list individual configuration properties and details. The job shows:

- The total number of properties that match
- The total number of properties that do not match
- The properties available on only one data source
- The overall total number of properties compared in the job

You can create a configuration standard, which is a customized list of configuration parameters, values, and comparison operators, to compare against a data source. For instructions, see: [Creating a Standard for Configuration Settings](#).

You can also compare a data source or standard against an archive, which is a saved version of the configuration properties of a registered data source. For instructions, see: [Creating a Configuration Archive](#).

The following tasks step you through creating and running a configuration comparison job:

- 1 [Create or Open a Configuration Comparison Job](#)
- 2 [Describe and Save the Job](#)
- 3 [Specify Configuration Comparison Source and Targets](#)
- 4 [Specify Configuration Comparison Job Options](#)
- 5 [Refine Property Comparisons](#)
- 6 Optionally request email, logs, or system tray notifications; see [Setting Job Notifications](#)
- 7 [Run the Configuration Comparison Job](#)
- 8 [View the Results of the Configuration Comparison](#)
- 9 Optionally generate a job report; see [Creating a Job Report](#)
- 10 Optionally [Synchronize Configurations](#)

All comparison jobs, archives, and standards provide version information with regards to each data source platform. For more information, see the [Version Information Reference](#).

CREATE OR OPEN A CONFIGURATION COMPARISON JOB

You create, modify, and run a configuration comparison job in the **Configuration Comparison Job** editor. For more details about the editor, see [Configuration Comparison Job Editor](#) in the [User Interface Reference](#).

To create a new configuration comparison job:

- Select **File > New > Configuration Comparison Job**.

The **Configuration Comparison Job** editor opens.

To open an existing configuration comparison job:

- Select **File > Open Job**.
- Select a configuration comparison job from the list.
- Click **OK**.

The **Configuration Comparison Job** editor opens.

DESCRIBE AND SAVE THE JOB

The Job Name and Description pane contains more than just the Name field. It shows where the job is stored in Project Explorer. When you click the Details button, you can see the project's file directory path.

If the job will be used to monitor compliance against company standards or policies, you can add it to the [Compliance Explorer](#) list. This list lets you quickly see all the jobs that have recently passed or failed the compliance criteria you set.

To describe and save a job

- 1 Type a new name for the job in the **Name** field. Enter a meaningful name that will quickly and uniquely identify the job in other views and dialogs.
- 2 Add relevant information in the **Notes** field.
- 3 Click **Track Results in Compliance Explorer** if you want the job to be listed in the [Compliance Explorer](#).
- 4 Click **File > Save** or **File > Save All**, or highlight the editor window and press **Ctrl+S**.

SPECIFY CONFIGURATION COMPARISON SOURCE AND TARGETS

The **Job Sources** section of the **Configuration Comparison Job** editor specifies the data sources, archives, and standards to be compared. You can compare:

- Two data sources
- A data source against an archive
- A standard against a data source
- A standard against an archive
- An archive against a data source
- Two archives

You can also compare a single source against multiple targets.

To add a data source to a job

- 1 Click and drag a data source from **Data Source Explorer** to the **Comparison Source** or **Comparison Target** box, or click **Select Data Source** in the appropriate box and choose a data source from the dialog.
- 2 Select **Use Live Sources** to compare the current properties of the data source when the job is executed.

NOTE: You can change a data source selection by clicking **Change Data Source** from the appropriate box, or by dragging a different data source to the box from **Data Source Explorer**.

To add a configuration standard to the job

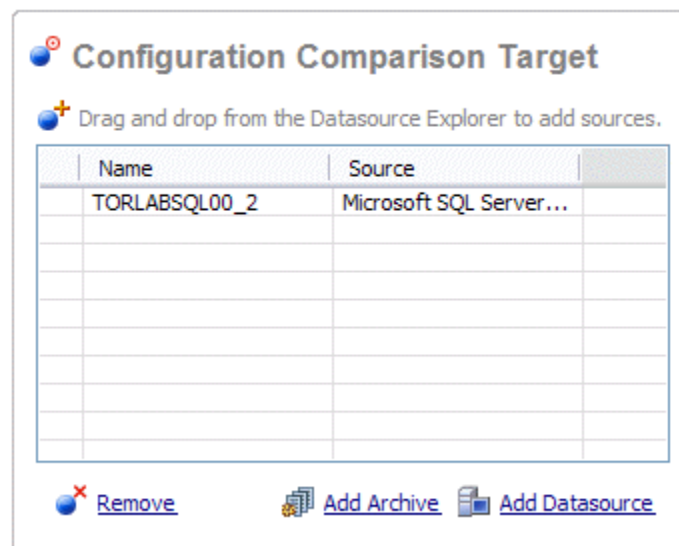
- 1 If you have not already, create a configuration standard; see [Creating a Standard for Configuration Settings](#).
- 2 Click and drag a standard from **Data Source Explorer** to the **Comparison Source**, or click **Select Standard** and choose a standard from the dialog.

To add a configuration archive to the job

- 1 If you have not already, create a configuration archive; see [Creating a Configuration Archive](#).
- 2 Click and drag an archive from **Data Source Explorer** to the **Comparison Source** box or **Target**, or click **Use an Archive** and choose an archive from the dialog.

To add multiple targets to the comparison job

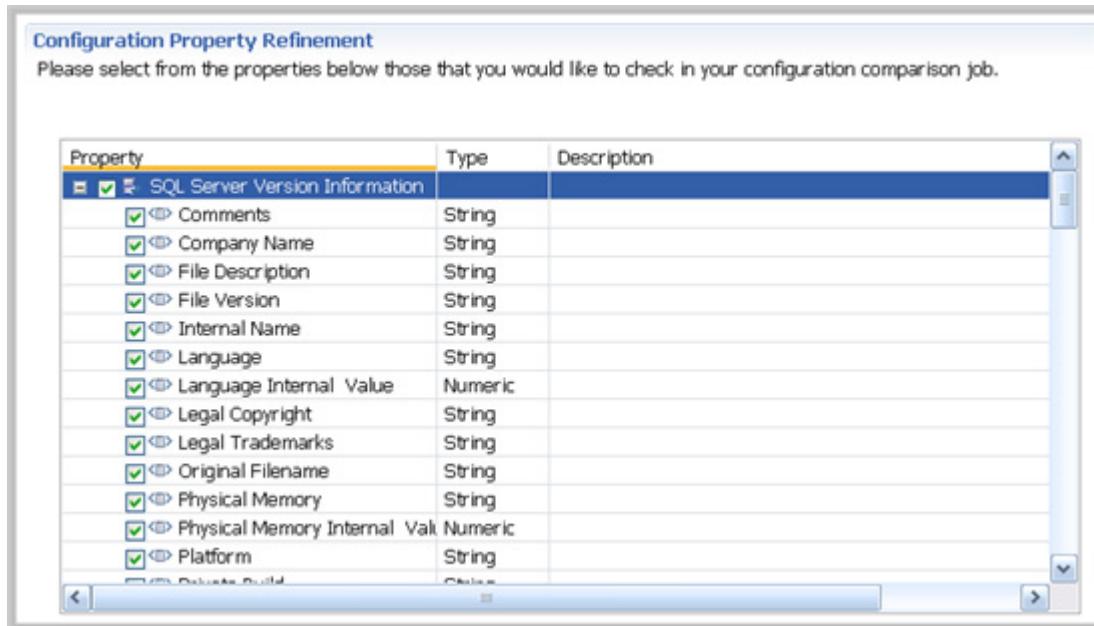
- 1 At the bottom of the pane, click **Add More Targets**.
- 2 Drag and drop additional data sources or archives from **Data Source Explorer** to the **Configuration Comparison Target** box, or click the **Add Archive** or **Add Data Source** commands beneath the table and select the appropriate target from the dialog.



When you are finished setting up the source and target or targets for the configuration comparison job, click **Refinements** to review the properties that will be compared.

REFINE PROPERTY COMPARISONS

After you specify a comparison job source and its target or targets, you can customize the comparison properties using the **Refinements** tab of the **Configuration Comparison Job** editor.



To specify the properties used in the comparison job

- 1 In the **Refinements** tab, click the category to see all of its properties.
- 2 Select the configuration properties you wish to compare, and deselect the rest.

After you have set the comparison properties, you can set the job options that pertain to how the job is executed.

SPECIFY CONFIGURATION COMPARISON JOB OPTIONS

The **Options** tab in the **Configuration Comparison Job** editor contains the **Extract DDL** option. If you select it, the synchronization code produced by the job comparison will appear in a separate editor window when the comparison is done.

SET CONFIGURATION COMPARISON JOB NOTIFICATIONS

When a job completes, DB Change Manager provides various ways of notifying you. See [Setting Job Notifications](#) for details.

RUN THE CONFIGURATION COMPARISON JOB

The configuration comparison job runs when you click the **Run Job** icon in the **Configuration Comparison Job** editor.



While it runs, the **Comparison Results** tab appears and displays the progress of the job.

VIEW THE RESULTS OF THE CONFIGURATION COMPARISON

When a job is complete, the **Comparison Results** tab displays the results.

Property	Source	Op	TORLABSQL05_1
Summary			
Not Found			27.8%
Matched			53.3%
Did Not Match			18.9%
Core SQL Properties			

The **Comparison Summary** shows:

- The number of matching properties
- The number of mismatched properties
- The number of properties that are in the source but not in the target

The **Comparison Results** table lists the details. Scroll down to see more.

Property ^	Source	Op	TORLABSQL05_1
show advanced options	1	EQU...	✔ 1
set working set size	1	EQU...	✘ 0
server trigger recursion		EQU...	⚠ 1
scan for startup procs	0	EQU...	✔ 0
Replication XPs		EQU...	⚠ 0

Comparison Results includes:

- The **Property** column lists settings found on the source.
- The **Source** column contains the value of the property on the source.
- The **Op** column provides the operator used in the job comparison that defines what constitutes a match. For most jobs, the operator is always EQUALS. For jobs that use a standard, it can be another operator.
- The last column, or columns if multiple targets were used, includes the target name and indicates the matches:
 - A check mark icon, along with the value of the property, indicates a match.
 - An X icon, along with the value of the property, indicates a mismatch.
 - A warning icon, along with the value of the property, indicates a value in the target that was not set in the source.

CREATE A JOB REPORT

After running a comparison job, you can create a job report using the **View Report** button on the **Results** tab. See [Creating a Job Report](#) for more information.

SYNCHRONIZE CONFIGURATIONS

If you want the configuration information to be the same between the source and target or targets, DB Change Manager can generate a script that will resolve the differences.

To create a SQL synchronization script

- 1 In the **Resolution** pane of the **Comparison Results** tab, click **Generate a SQL Script**.
- 2 In the **SQL Editor**, review the script and edit it as needed.

EXECUTE AN SQL SYNCHRONIZATION SCRIPT

You can run synchronization scripts in the **SQL Editor**. For more information, see [SQL Editor](#).

To run the SQL synchronization script

- In the **SQL Editor**, click the **Execute SQL** icon.

DB Change Manager executes the code on the target data source and displays the results.

CREATING A STANDARD FOR CONFIGURATION SETTINGS

A configuration standard is an object used in configuration comparison jobs that stores data source configuration properties. You compare a configuration standard against registered data sources and archives to determine if and how the latter differs from your company-standard values and ranges.

A configuration standard has a list of properties to which you assign values. Properties are derived from data sources, archives, or other standards. You assign each property value a range or set an operator to be used in comparisons.

The following tasks step you through creating a configuration standard:

- 1 [Create or Open a Configuration Standard](#)
- 2 [Describe and Save the Standard](#)
- 3 [Add Standard Sources](#)
- 4 [Define Configuration Standard Properties](#)
- 5 [Use a Configuration Standard in Configuration Comparison Jobs](#)

See [Comparing Database Configuration Settings](#) for more information.

CREATE OR OPEN A CONFIGURATION STANDARD

You create or modify a Configuration Standard in the Configuration Standard Editor. For more details about the editor, see [Configuration Standard Editor](#) in the [User Interface Reference](#).

To create a new configuration standard

- Select **File > New > Configuration Standard**.

The **Configuration Standard** editor opens.

To open an existing configuration standard

- 1 In **Project Explorer**, click **Standards > Configuration Standards**.
- 2 Double-click a standard from the list.

The **Configuration Standard** editor opens.

DESCRIBE AND SAVE THE STANDARD

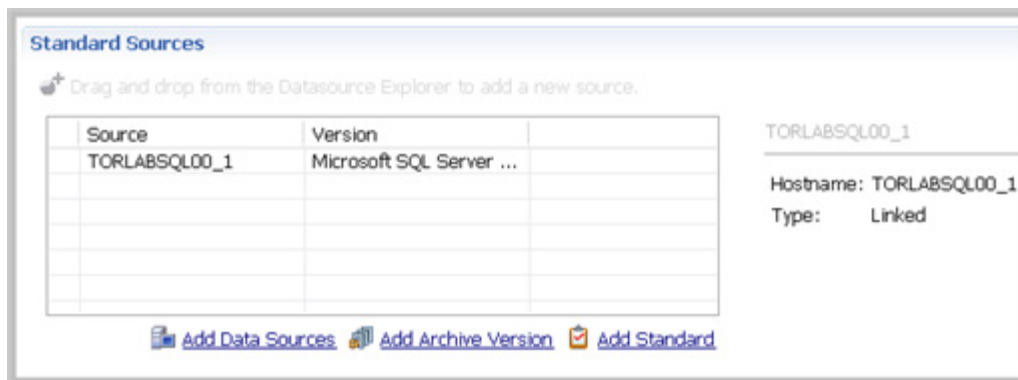
The **Job Name and Description** pane shows where the job is stored in **Project Explorer**. When you click the **Details** button, you can see the project's file directory path.

To describe and save a configuration standard

- 1 Type a new name for the job in the **Name** field. Enter a meaningful name that will quickly and uniquely identify the job in other views and dialogs.
- 2 Add relevant information in the **Notes** field.
- 3 Click **File > Save** or **File > Save All**, or highlight the editor window and press **Ctrl+S**.

ADD STANDARD SOURCES

The **Standard Sources** table lists the registered data sources, archives, and other standards from which properties for the new standard will be derived.



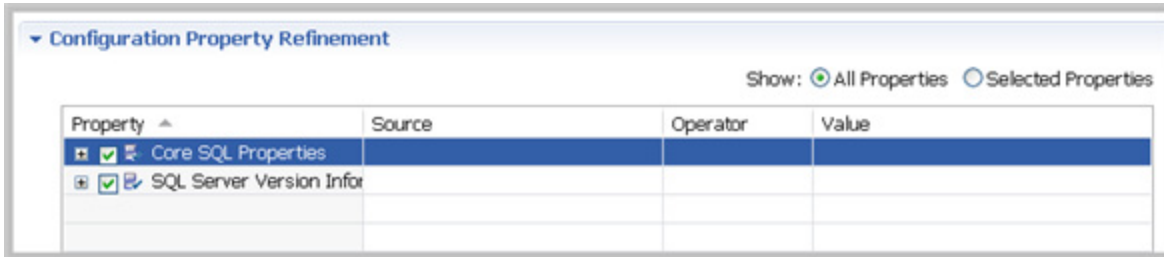
To add properties to a standard

- Drag and drop a data source, archive, or standard from **Data Source Explorer** to the **Standard Sources** table. Alternatively, use the **Add Data Source**, **Add Archive Version**, or **Add Standard** buttons beneath the table.

All of the properties of the selected data source, archive, or standard populate the new standard.

DEFINE CONFIGURATION STANDARD PROPERTIES

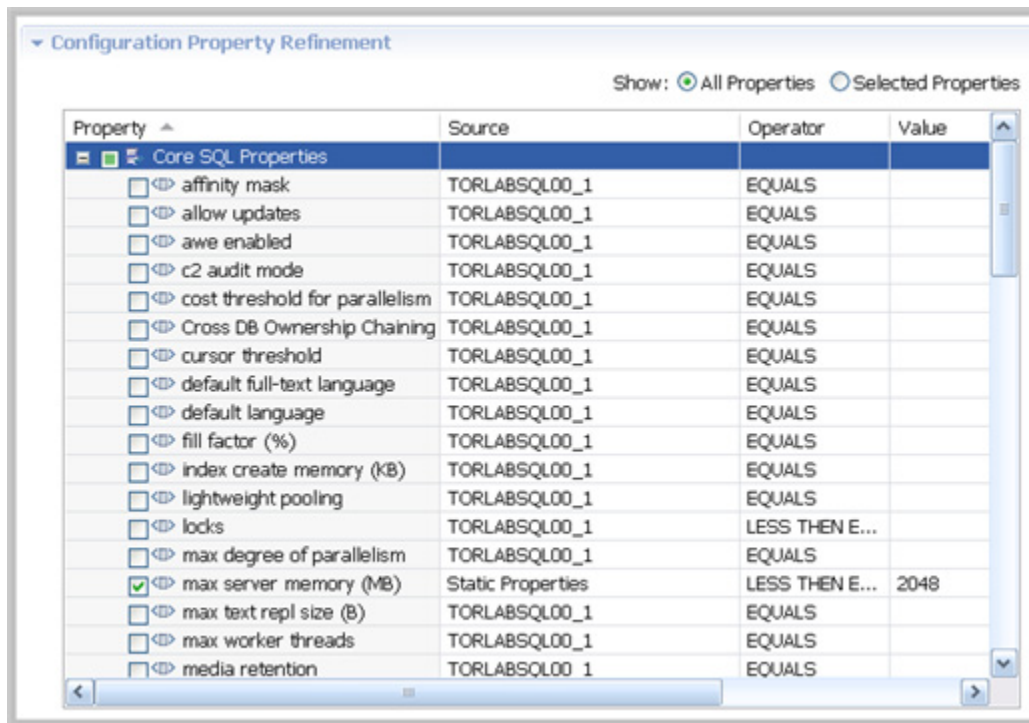
The Configuration Property **Refinement** tab displays properties and corresponding sources applied to the standard.



You can include or exclude properties in the standard, and set thresholds and individual property values in the **Operator** column. The following operators apply:

- Less Than
- Greater Than
- Greater Than Equal To
- Equals
- Less Than or Equal To
- Not Equals

The **Value** column indicates the value that the **Operator** definition must match.



To define the properties of a configuration standard

- 1 Deselect the properties you do not want to consider using the check boxes.
- 2 Click **Show: Selected Properties** to narrow down the list in the window.
- 3 For a property you want to use in the standard, click its appropriate row value in the **Operator** column and choose a selection from the drop down menu.
- 4 Type a string or a regular expression in the corresponding **Value** column.
- 5 Repeat the previous steps until you have defined the properties you require.
- 6 Click **File > Save** or **File > Save All**, or highlight the editor window and press **Ctrl+S**.

USE A CONFIGURATION STANDARD IN CONFIGURATION COMPARISON JOBS

You can use a configuration standard as a source in configuration comparison jobs, the same way as registered data sources or archives. You cannot use a standard as a comparison target.

To add a standard to a configuration comparison job

- 1 Drag and drop the standard from the **Project Explorer** to the comparison source box in the **Configuration Comparison Job** editor.
- 2 Continue to define and run the job. See [Comparing Database Configuration Settings](#) for more information.

CREATING A CONFIGURATION ARCHIVE

A configuration archive is a snapshot of the configuration parameters of a registered data source. You use a configuration archive in configuration comparisons against other data sources, standards, and archives to monitor property values and provide backups.

You can periodically update archives with the current values of an associated data source to create new versions. Older versions are retained with an archive and can be recalled as needed.

Archives are listed in **Data Source Explorer** under their corresponding data source.

The following tasks step you through creating a configuration archive:

- 1 [Create or Open a Configuration Archive Job](#)
- 2 [Describe the Configuration Archive Job](#)
- 3 [Specify a Configuration Archive Source](#)
- 4 [Refine Configuration Archive Properties](#)
- 5 Optionally request email, logs, or system tray notifications; see [Setting Job Notifications](#)
- 6 [Create a Version and Save the Archive Job](#)
- 7 Optionally generate a job report; see [Create an Archive Job Report](#)
- 8 [Use an Archive in a Configuration Comparison Job](#)

CREATE OR OPEN A CONFIGURATION ARCHIVE JOB

You create or modify a configuration archive in the **Configuration Archive** editor. For more details about the editor, see "[Configuration Archive Job Editor](#)" on page 171 in the [User Interface Reference](#).

To create a new configuration archive job

- Select **File > New > Configuration Archive Job**.

The **Configuration Archive** editor opens.

To open an existing configuration archive job

- 1 **File > Open Job**.
- 2 Select a configuration job from the list.

- 3 Click **OK**.

The **Configuration Archive** editor opens.

DESCRIBE THE CONFIGURATION ARCHIVE JOB

The **Archive Name and Description** pane contains more than just the **Name** field. It shows where the job is stored in **Project Explorer**. When you click the **Details** button, you can see the project's file directory path.

To describe a configuration archive job

- 1 Type a new name for the job in the **Name** field. Enter a meaningful name that will quickly and uniquely identify the job in other views and dialogs.
- 2 Add relevant information in the **Notes** field.

SPECIFY A CONFIGURATION ARCHIVE SOURCE

The configuration **Archive Source** table specifies the data source from which the archive is derived.

To specify and save a configuration archive source

- 1 Drag and drop the data source to be archived from **Data Source Explorer** to the configuration **Archive Source** box.

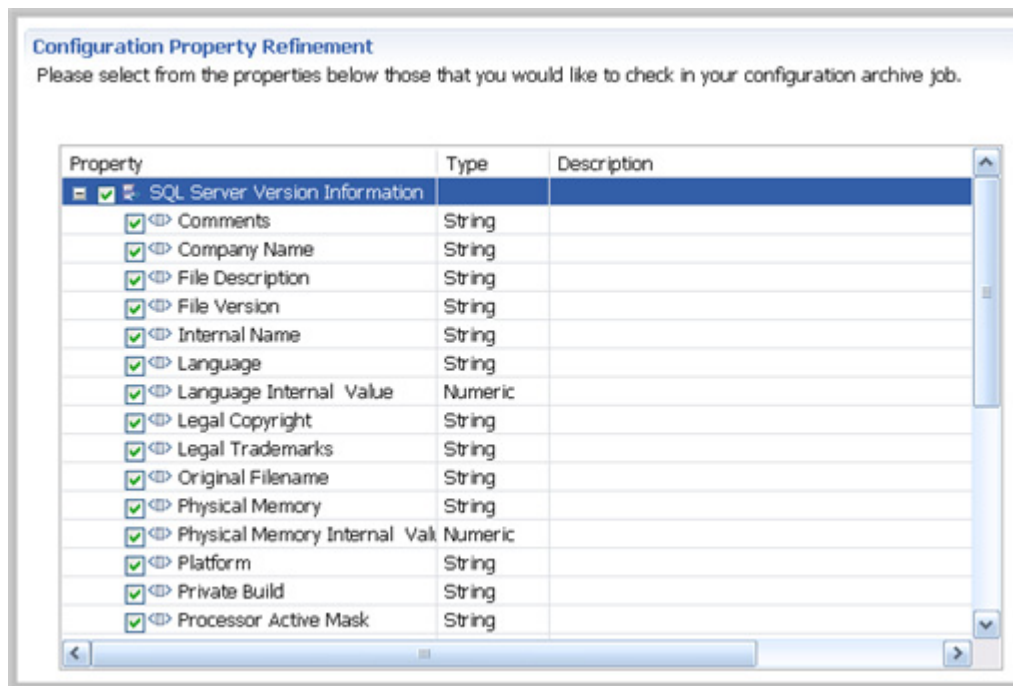
The data source is listed by name, database type, and host.
- 2 Click **File > Save** or **File > Save All**, or highlight the editor window and press **Ctrl+S**.

REFINE CONFIGURATION ARCHIVE PROPERTIES

When you create a new archive, by default all data source properties are included. However, you may choose to archive only a specific group of properties. For example, you may want to define an archive that retains only the security properties of the associated data source.

You can create multiple archives for the same data source. For example, another archive may retain only the performance properties.

Archive configuration properties are listed on the **Refinements** tab of the **Archive Job** editor.



To specify configuration archive properties

- Use the check boxes to select or deselect properties.

Within the same archive, properties are retained across versions unless you specify new values.

NOTE: Each time you create a **new** archive, DB Change Manager includes all properties of the specified data source and you must again refine the properties.

SET CONFIGURATION ARCHIVE JOB NOTIFICATIONS

When an archive job completes, DB Change Manager provides various ways of notifying you. See [Setting Job Notifications](#) for details.

SPECIFY CONFIGURATION ARCHIVE JOB OPTIONS

The **Options** tab in the **Configuration Comparison Job** editor contains the **Extract DDL** option. If you select it, the synchronization code produced by the job comparison will appear in a separate editor window when the comparison is done.

CREATE A VERSION AND SAVE THE ARCHIVE JOB

When you first run an archive job, DB Change Manager creates the first version of the archive. A version stores snapshots of the same archive job for backup or tracking comparison purposes. Versions are listed in the **Archive Job History** table on the **History** tab and in the **Data Source Explorer**.

To create a version and save a configuration archive job

- 1 In the **Configuration Archive Job** editor, click **Run Job**.

The **New Unsaved Version** tab lists the properties of the archive.

Archive Summary

Source: TORLABSQL00_1

New Unsaved Version
 ✓ 65 properties archived.

[View Report](#) [Export Version...](#)

Archive Results

Name	Operation	Value	Description
SQL Server Version Information			
Windows Version Internal	EQUALS	170393861	
Windows Version	EQUALS	5.1 (2600)	
Special Build Internal Value	EQUALS	133627904	
Special Build	EQUALS	[NULL]	
Product Version Internal Value	EQUALS	524288	
Product Version	EQUALS	8.00.2039	
Product Name	EQUALS	Microsoft SQL Server	
Product ID	EQUALS	[NULL]	
Processor Type Internal Value	EQUALS	586	
Processor Type	EQUALS	PROCESSOR_INTEL_PENTIUM	
Processor Count Internal Value	EQUALS	1	
Processor Count	EQUALS	1	
Processor Active Mask Internal Value	EQUALS	1	
Processor Active Mask	EQUALS	00000001	

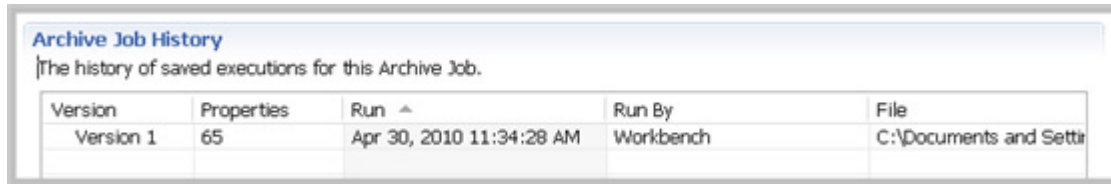
Overview Refinements Options Notification History **New Unsaved Version**

- 2 Highlight the tab and select **File > Save or File > Save All**.

The tab shows that the version is saved.

- 3 Click the **History** tab.

The **Archive Job History** table lists the archive version, the number of properties retained by the version, a time stamp, and whether the archive job was run from the workbench or the command line.



The screenshot shows a window titled "Archive Job History" with a subtitle "The history of saved executions for this Archive Job." Below the subtitle is a table with the following data:

Version	Properties	Run	Run By	File
Version 1	65	Apr 30, 2010 11:34:28 AM	Workbench	C:\Documents and Settir

You create additional versions within the same archive job each time you click **Run Job**. The new version contains the latest values for the archive job properties as specified on the **Refinements** tab.

To open an archive version

- In the **History** tab or in the **Job Explorer**, double-click the version.

The selected version opens in a new tab of the **Archive Job** editor.

CREATE AN ARCHIVE JOB REPORT

After running a comparison job, you can create a job report using the **View Report** button on the **Results** tab. See [Creating a Job Report](#) for more information.

USE AN ARCHIVE IN A CONFIGURATION COMPARISON JOB

You can use a configuration archive in configuration comparison job the same way as registered data sources or standards.

To add a configuration archive to a configuration comparison job

- Drag and drop an archive version from **Data Source Explorer** to the source or target boxes in the **Configuration Comparison Job** editor. Alternatively, if a data source is selected as the source or target, select **Use an Archive** in the respective box and use the drop down menu to choose an archive.

See [Comparing Database Configuration Settings](#) for more information.

COMPARING SCHEMAS

You can create a schema comparison job to capture, compare, and synchronize database schemas. In addition, you can compare a schema against an archived version of a schema. See:

- [Creating a Schema Archive](#)

The following tasks step you through creating and running a schema comparison job:

- 1 [Create or Open a Schema Comparison Job](#)
- 2 [Describe and Save the Schema Comparison Job](#)
- 3 [Specify a Schema Comparison Source and Targets](#)
- 4 [Refine Schema Comparisons](#)
- 5 [Customize Schema Mapping](#)
- 6 [Specify Schema Comparison Job Options](#)
- 7 Optionally request email, logs, or system tray notifications; see [Setting Job Notifications](#)
- 8 [Run the Schema Comparison Job](#)
- 9 Optionally generate a job report; see [Creating a Job Report](#)

After comparing the data, you can synchronize it to match; see [Synchronizing Data Repositories](#).

CREATE OR OPEN A SCHEMA COMPARISON JOB

You create, modify, and run a schema comparison job in the **Schema Comparison Job** editor. For more details about the editor, see [Schema Comparison Job Editor](#) in the [User Interface Reference](#).

To create a new schema comparison job

- Select **File > New > Schema Comparison Job**.

The **Schema Comparison Job** editor opens.

To open an existing schema comparison job

- Select **File > Open Job**. Select a job from the list and click **OK**.

The **Schema Comparison Job** editor opens.

DESCRIBE AND SAVE THE SCHEMA COMPARISON JOB

The **Job Name** and **Description** pane shows where the job is stored in **Project Explorer**. When you click the **Details** button, you can see the project's file directory path.

If the job will be used to monitor compliance against company standards or policies, you can add it to the [Compliance Explorer](#) list. This list lets you quickly see all the jobs that have recently passed or failed the compliance criteria you set.

To describe and save a job

- 1 Type a new name for the job in the **Name** field. Enter a meaningful name that will quickly and uniquely identify the job in other views and dialogs.
- 2 Add relevant information in the **Notes** field.
- 3 Click **Track Results in Compliance Explorer** if you want the job to be listed in the [Compliance Explorer](#).
- 4 Click **File > Save** or **File > Save All**, or highlight the editor window and press **Ctrl+S**.

SPECIFY A SCHEMA COMPARISON SOURCE AND TARGETS

The **Schema Comparison Source** and **Schema Comparison Target** panes identify what data sources or schema archives are compared when the job runs. The **Schema Comparison Job** editor displays the **Name**, **Type**, and **Host** of the source and target.

You can also use a schema archive in a schema comparison job, to compare an older schema against a new one. See [Creating a Schema Archive](#) for more information.

In addition, you can use the radio buttons to specify whether to compare the current properties of the live data source, or an archived set of properties previously stored in DB Change Manager.

- Select **Use Live Sources** to compare the current properties of the data source when the job is executed.
- Select **Use an Archive** and select the schema archive and version from the pulldown menu.

To add a data source to a schema comparison job

- Click and drag a data source from **Data Source Explorer** to the **Schema Comparison Source** or **Schema Comparison Target** box, or click **Change Data Source** and choose a data source from the dialog box.

To add a schema archive to a schema comparison job

- Click and drag a schema archive from **Data Source Explorer** to the **Schema Comparison Source** or **Schema Comparison Target** box.

To change a schema archive or data source in a schema comparison job

- Click **Change Data Source** in the appropriate box and select a new data source from the dialog, or drag a different data source from **Data Source Explorer** to the box.

CROSS-PLATFORM SCHEMA COMPARISONS SUPPORTED BY DB CHANGE MANAGER

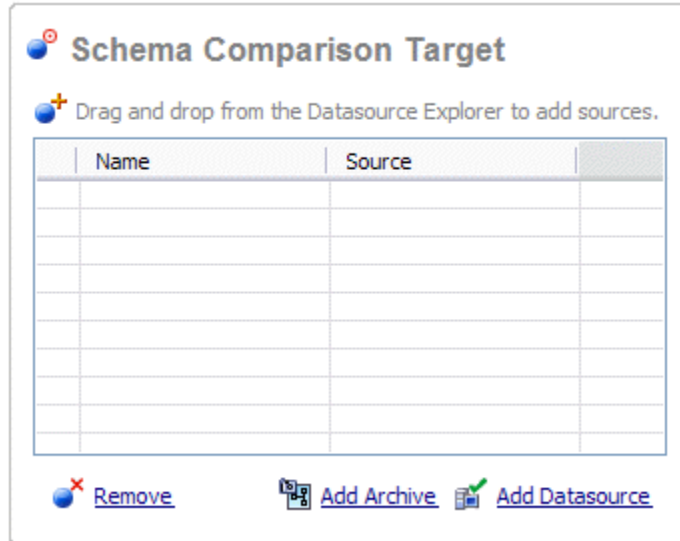
DB Change Manager supports cross-platform schema comparisons. However, such jobs are limited to the following DBMS combinations and data source objects:

DBMS Source/Target	Objects Compared
Sybase/SQL Server	Tables, Unique Keys, Check Constraints, Foreign Keys, Indexes, Primary Keys, Views
Oracle/SQL Server	Tables, Unique Keys, Check Constraints, Foreign Keys, Indexes, Primary Keys, Views
Firebird/InterBase	Tables, Unique Keys, Check Constraints, Foreign Keys, Indexes, Primary Keys, Views

If you attempt to compare two data sources that are not compatible, the **Schema Comparison Job** editor will not let you place objects in the source or target panes.

ADD MULTIPLE SCHEMA COMPARISON JOB TARGETS

You can add more targets to compare a source schema against multiple target schemas.



To add multiple targets to a schema comparison job

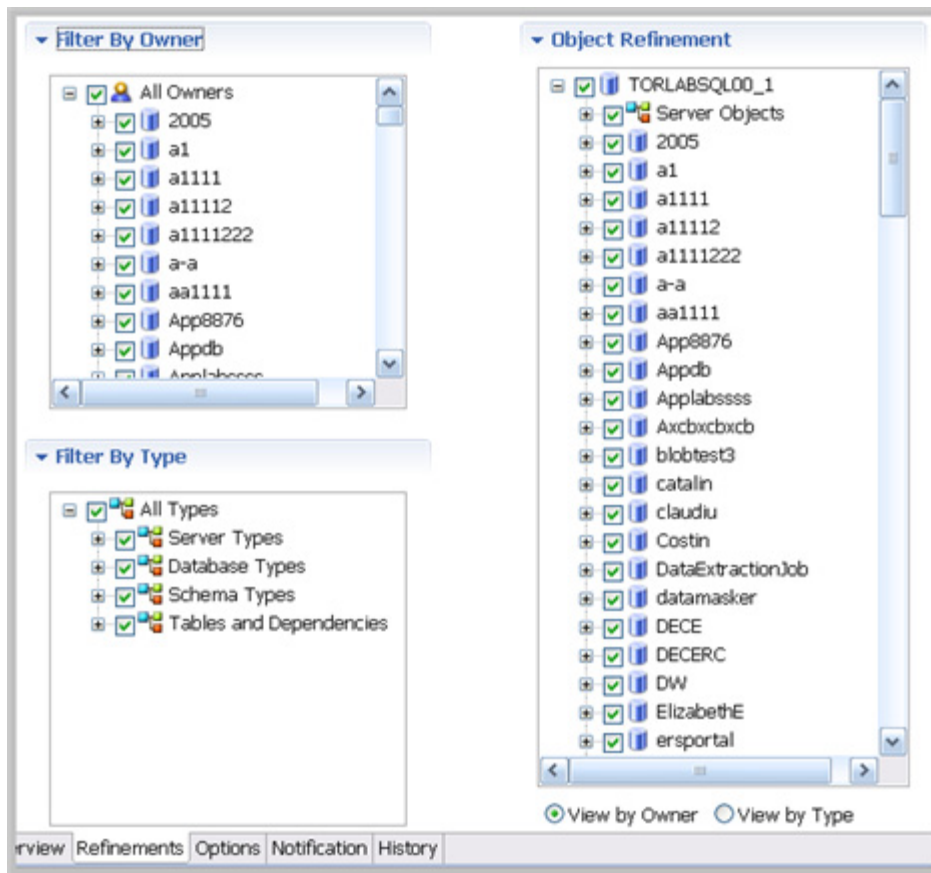
- Drag and drop additional data sources or schema archives from **Data Source Explorer** to the **Schema Comparison Target** box, or click the **Add Archive** or **Add Data Source** buttons beneath the table and select the appropriate target from the dialog box.

REFINE SCHEMA COMPARISONS

You can refine the schema comparison to use only the schema objects you want to include using the **Refinements** tab in the **Schema Comparison Job** editor.

The **Object Refinement** pane lists the database objects of the comparison source. Objects are listed by owner (if applicable to the database platform) or by object type.

The **Filter by Owner** and **Filter by Type** panes let you add or remove objects from the **Object Refinement** pane by owner (if applicable to the database platform) or by object type.



To specify the objects used in a schema comparison job

- 1 In the **Refinements** tab, select or deselect objects in the **Filter by Owner** and/or **Filter by Type** panes to add or remove objects from the **Object Refinement** pane.
- 2 Select or deselect the objects in the **Object Refinement** pane.

CUSTOMIZE SCHEMA MAPPING

Mapping pairs objects between the job source and target so that the objects can be compared. You can see and modify them using the **Mapping** tab of the **Schema Comparison Job** editor.

NOTE: Schema mapping is not available for Firebird and InterBase, as they do not have owners.



DB Change Manager automatically maps schema objects based on its interpretation of both schemas. Elements that do not match are initially excluded; you can manually pair them prior to running the job.

The **Refinements Mapping** table lists schema elements for the job source and target.

Refinements mapping

Select target: Ignore Case Ignore Spaces Ignore Underscores

Datoedwina	TORLABSQL00_2	
+	aaa	[Unmapped]
+	anmari	[Unmapped]
+	AppDb	AppDb
+	dt_issues	[Unmapped]
+	Extreme	[Unmapped]
+	licenseDB	licenseDB
+	master	master
+	model	model
+	msdb	msdb
+	Northwind	Northwind
+	PCenter	[Unmapped]
+	PERFCNTR	[Unmapped]
+	PERFCNTR_REPO	[Unmapped]
+	PERFCNTR_REPO2	[Unmapped]
+	PERFCNTR_REPO_suri	[Unmapped]
+	pman	[Unmapped]

 [Refresh Databases and Owners](#)  [Restore Default Mappings](#)

To redefine schema object mappings

- In the **Refinements Mapping** tab, click the target of the pair you want to change. Use the drop down menu to define the object you prefer.

To restore original schema object mappings

- Click **Restore Default Mappings** to revert to the original object mappings as determined by DB Change Manager.

SPECIFY SCHEMA COMPARISON JOB OPTIONS

Prior to running a job, you can set various options to control aspects of the comparison process. Set them using the **Options** tab.

The list of options differ depending on the DMBS of the source and target data sources.

Options differ according to the objects included in the job. For example, **Ignore Constraint Name Differences** will not appear if you exclude constants from the job.

The options you see in the job have defaults. You can change the defaults using **Preferences**.

Schema comparison job options include:

- **Synchronization Options**

These options specify how you want DB Change Manager to handle the synchronization script after it runs the comparison job, either:

- **Manual** causes DB Change Manager to display the results of the comparison, but it does not automatically attempt to repair any synchronization issues. If you choose this option, you will still be able to manually generate and run a synchronization script after the job runs.
- **Automatic** causes DB Change Manager to automatically generate *and run* a synchronization script immediately after the comparison job runs.

- **Include Object Options**

These settings determine what elements of the job source and target will be omitted when running a schema comparison job and synchronization. For more information, see [CM/Schema > Comparison Options > Include Objects Options](#).

- **Compare Options**

These settings affect how objects are treated in the comparison process. You can include or exclude specific object types in the job, and specify the behavior of the comparison job and synchronization scripts. For more information, see [CM/Schema > Comparison Options > Compare Options](#).

- **Decorator Options**

These settings affect particular database objects. For more information, see [CM/Schema > Comparison Options > Decorator Options](#).

- **Identifier Options**

These settings determine what elements of the source and target data source will be ignored during the schema comparison and synchronization process. For more information, see [CM/Schema > Comparison Options > Identifier Options](#).

- **DDL Extract Option**

This setting causes the DDL code to appear in a separate editor window after the comparison job runs. For more information, see [CM/Schema > Comparison Options > DDL Extract Options](#).

- **Custom Options**

This section lists options that you defined in the **Create New Schema Custom Compare Options** dialog. For more information, see [Select Schema Properties to Ignore During a Comparison](#)

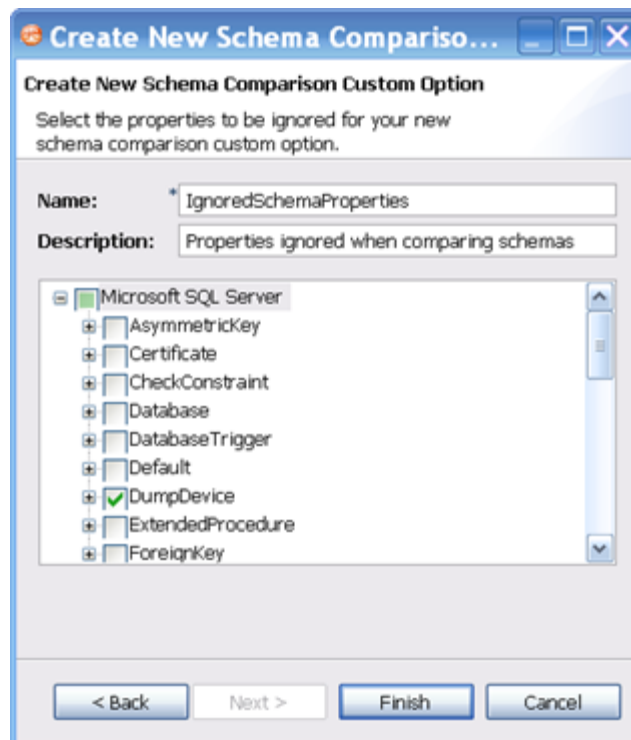
- As needed, DBMS-specific options settings are also listed; for more information, see [CM/ Schema > Comparison Options > Additional Options](#).

SELECT SCHEMA PROPERTIES TO IGNORE DURING A COMPARISON

You can create a custom list of schema properties that will be ignored by a schema comparison job.

To define schema properties to ignore

- 1 Click **File > New > Custom Schema Comparison Option**.
- 2 Name the custom properties.
- 3 Optionally, add a description.
- 4 Select the properties to ignore.



- 5 Click **Finish**.
- 6 Open a schema comparison job **Options** tab.
- 7 Under **Custom Options**, select the custom option.

SET SCHEMA COMPARISON JOB NOTIFICATIONS

When a job completes, DB Change Manager provides various ways of notifying you. See [Setting Job Notifications](#) for details.

RUN THE SCHEMA COMPARISON JOB

To run the schema comparison job

- Click the **Run** icon located in the upper right-hand corner of the **Schema Comparison Job** editor to run the job.



While the process runs, the **Comparison Results** tab appears and indicates the progress of the job.

Close results

Comparison Summary

Source: TORLABSQL00_1
 Target: TORLABSQL05_1
 Start Time: May 4, 2010 11:26:46 AM
 Estimated Remaining Time: Less Than 1 Minute
 Elapsed Time: 00:55 (mm:ss)
 Comparison progress: Current Object Type: Indexes Object Name: FranklynA.dbo.Supplier.dbc

32%

Comparison Results

Target	Task	Progress	Compare Index	Results
TORLABSQL05_1	Compare	In Progress		

When the job is complete, the **Comparison Results** pane shows information about the comparison.

Target	Task	Progress	Compare Index	Results
TORLABSQL05_1	Compare	Done with 0 errors	39% Match	Show Ir...

The **Comparison Results** pane includes these columns:

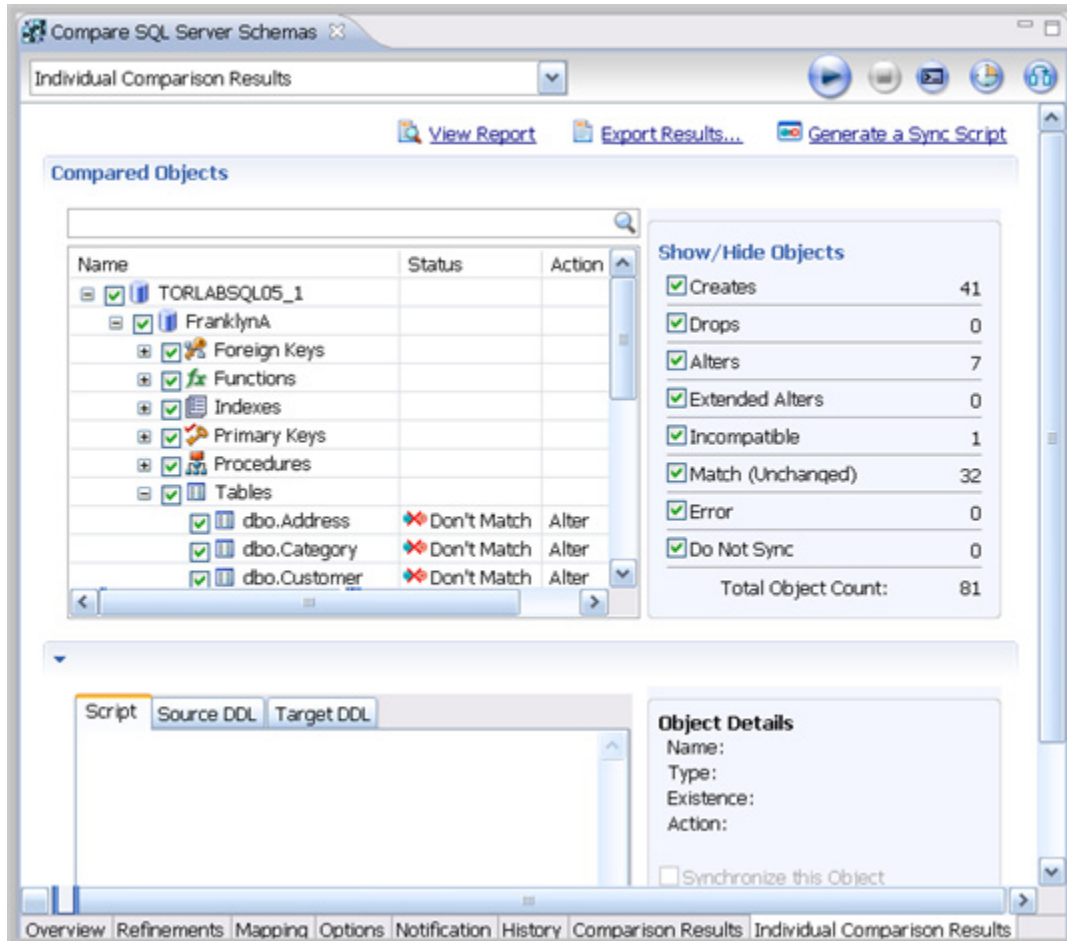
- The **Target** column shows the name of the target, and provides a link to open the **Individual Comparison Results** tab, which provides more details. For more information on viewing individual results, see [View Individual Schema Comparison Results](#).
- The **Task** column shows what kind of job was run.
- The **Progress** column shows whether the job is in progress or has completed, and how many errors were encountered.
- The **Compare Index** column indicates the percentage of schema that matched between the source and targets of the job.
- The **Results** column provides a **Show Individual Results** link to open the **Individual Comparison Results** tab, which provides more details. For more information on viewing individual results, see [View Individual Schema Comparison Results](#).
- The **Resolution** column provides a command that produces a synchronization script that, when run, automatically synchronizes the compared schema. For more information on generating a synchronization script, see [Synchronizing Schemas](#).
- The **Errors** column indicates how many errors were encountered during the comparison process, and provides a link to view them in a separate window.

VIEW INDIVIDUAL SCHEMA COMPARISON RESULTS

To see specific schema comparison results

- After a job runs, in the **Comparison Results** table, click **Show Individual Results**.

The **Individual Comparison Results** window appears.



To view the DDL between two schema comparison results

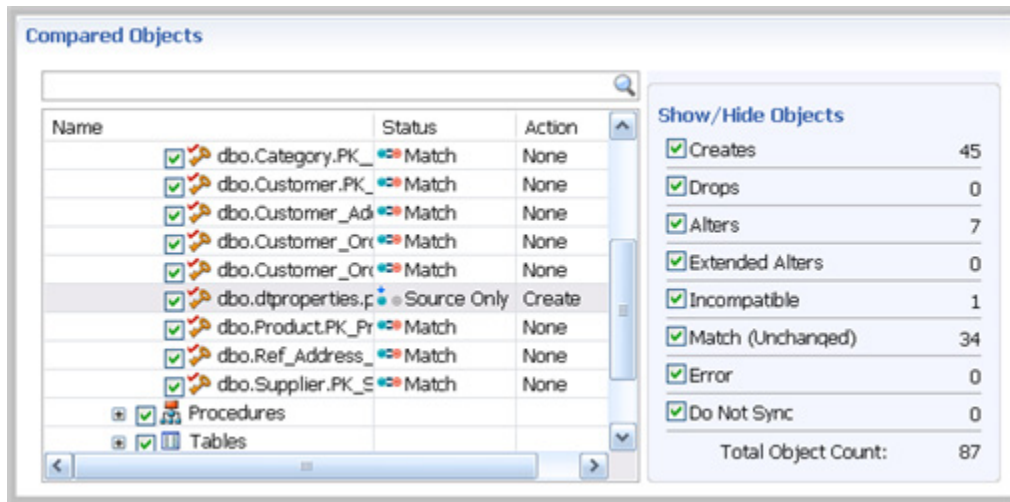
- Click **View Details**.

To save individual schema comparison results data to a text file

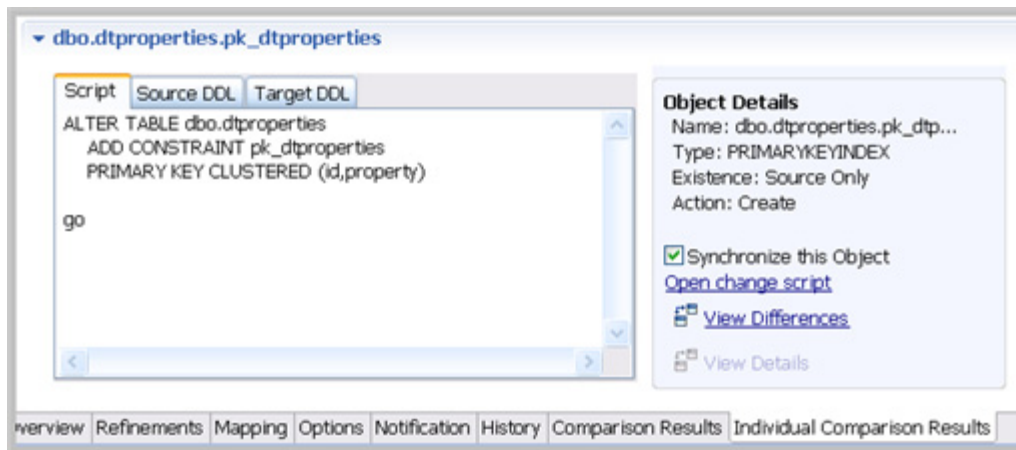
- Click **Export to File**.

To view the synchronization script and DDL between two individual objects

- 1 In the **Individual Comparison Results** window **Compared Objects** pane, select an object that does not match.



Script and DDL details appear in the pane below.



- 2 In the **Script** tab, view the SQL that will synchronize the objects.

CREATE A SCHEMA COMPARISON JOB REPORT

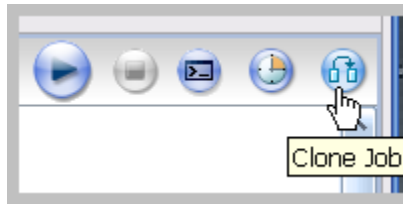
After running a comparison job, you can create a job report using the **View Report** button on the **Results** tab. See [Creating a Job Report](#) for more information.

CLONE A SCHEMA COMPARISON JOB

You can create a copy of a schema comparison job, using a different data source.

To clone a schema comparison job

- 1 After you have saved and run a job, in the **Schema Comparison Job** editor enables the **Clone Job** icon.



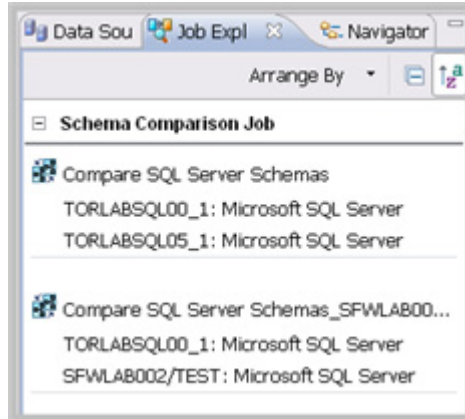
- 2 Click **Clone Job**.



- 3 In the **Data Source Selection** list, select the new target for the cloned job. If you need to add a new data source to the list, click **New Data Source**.

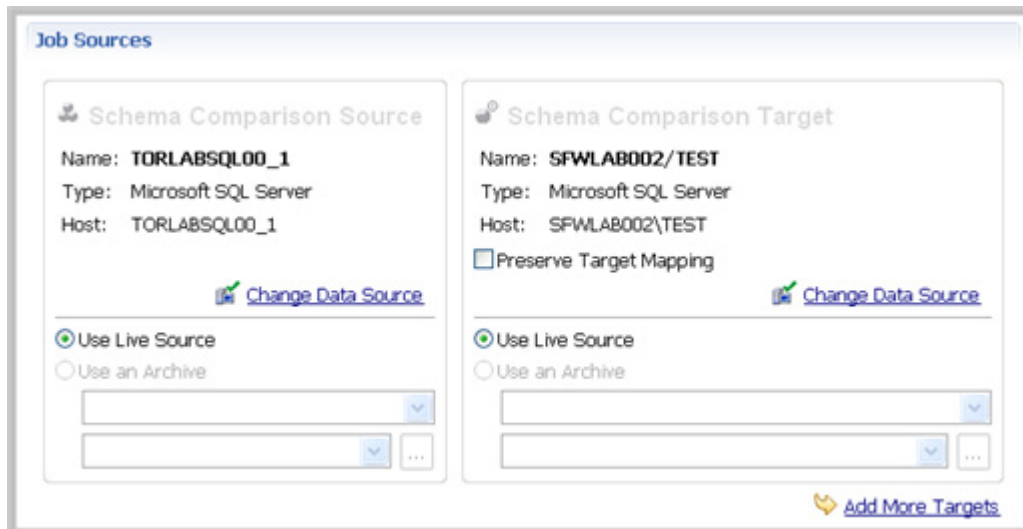
4 Click **OK**.

DB Change Manager lists the new, cloned job in the **Job Explorer**. You can modify and run the cloned job as just as any other schema comparison job. In addition, you can preserve the target mapping from the original job.



To preserve target mapping in a cloned schema comparison job

- 1 Open the cloned job in the **Schema Comparison Job** editor.
- 2 In the **Job Sources** target pane, click **Preserve Target Mapping**.



CREATING A SCHEMA ARCHIVE

A schema archive is a snapshot of the schema of a registered data source. You use a schema archive in schema comparisons against other data sources and archives. It is useful as a baseline to monitor data loss and inconsistencies when you move databases between testing and production environments.

Archive jobs are versioned. Older versions are retained within an archive.

Archive jobs are located in **Data Source Explorer** on the **Schema Archive Jobs** node of the corresponding data source.

The following tasks step you through creating a schema archive:

- 1 [Create or Open a Schema Archive Job](#)
- 2 [Describe and Save the Schema Archive Job](#)
- 3 [Specify a Schema Archive Job Source](#)
- 4 [Refine Schema Archive Properties](#)
- 5 [Specify Schema Archive DDL Options](#)
- 6 Optionally request email, logs, or system tray notifications; see [Setting Job Notifications](#)
- 7 [Create a Version and Save the Schema Archive Job](#)
- 8 [Use an Archive in Schema Comparison Jobs](#)
- 9 Optionally generate a job report; see [Creating a Job Report](#)
- 10 [Use an Archive in Schema Comparison Jobs](#)

CREATE OR OPEN A SCHEMA ARCHIVE JOB

You create or modify a schema archive in the **Configuration Archive Job** editor. For more details about the editor, see [Schema Archive Job Editor](#) in the [User Interface Reference](#).

To create a new schema archive

- Select **File > New > Schema Archive Job**.

The **Schema Archive Job** editor opens.

To open an existing schema archive

- 1 Select **File > Open Archive**.
- 2 Select a job from the list and click **OK**.

The **Schema Archive Job** editor opens.

DESCRIBE AND SAVE THE SCHEMA ARCHIVE JOB

The **Job Name and Description** pane shows where the job is stored in **Project Explorer**. When you click the **Details** button, you can see the project's file directory path.

To describe and save a configuration archive job

- 1 Type a new name for the job in the **Name** field. Enter a meaningful name that will quickly and uniquely identify the job in other views and dialogs.
- 2 Add relevant information in the **Notes** field.
- 3 Click **File > Save** or **File > Save All**, or highlight the editor window and press **Ctrl+S**.

SPECIFY A SCHEMA ARCHIVE JOB SOURCE

The **Schema Archive Source** table specifies the data source from which the archive job will be created.

To specify a schema archive job source

- Drag and drop the data source to be archived from **Data Source Explorer** to the **Schema Archive Source** box. The data source is listed by name, database type, and host.

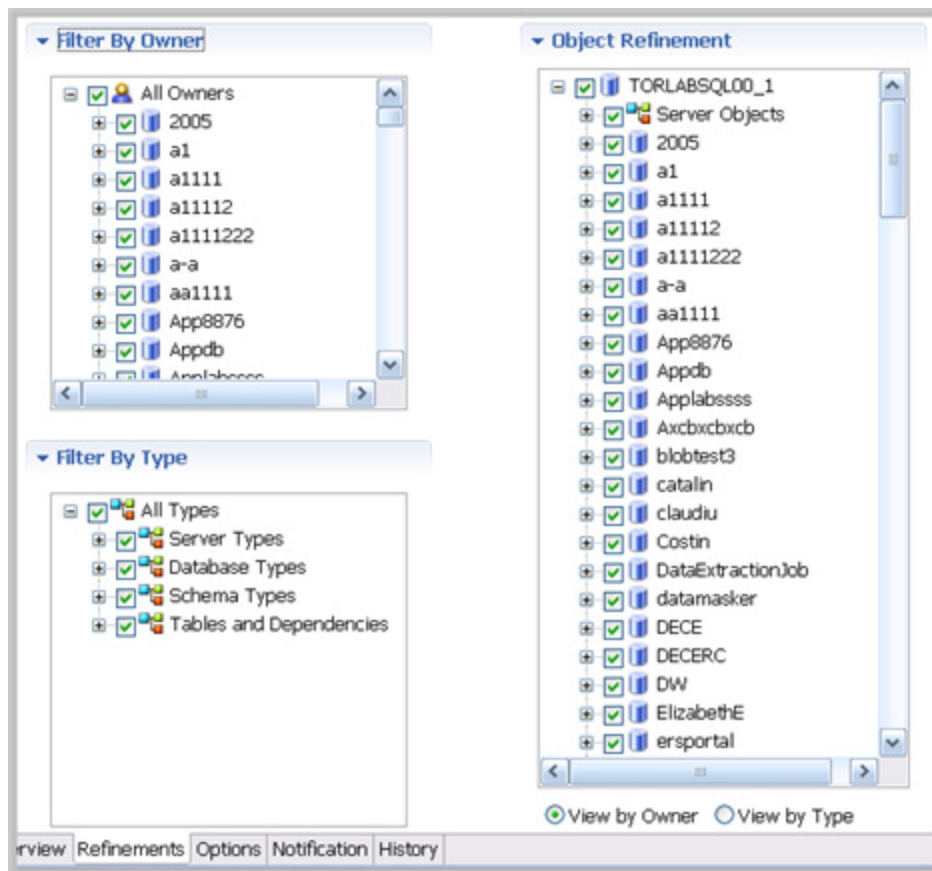
REFINE SCHEMA ARCHIVE PROPERTIES

When you create a new schema archive, by default all schema properties are included. However, you may choose to archive only a specific group of properties. For example, you may want to define an archive that retains a subset of the properties. You can create multiple schema archives for the same data source.

Archive job properties are located on the **Refinements** tab of the **Schema Archive Job Editor**, where they are listed by object name, owner (if applicable to the database platform), and type.

To refine schema archive properties

- 1 In the **Schema Archive** editor, click the **Refinements** tab.
- 2 DB Change Manager analyzes the data source and displays schema information in the **Refinements** tab.



- 3 Use the check boxes to select only the data source objects from which you want to create an schema archive.

Within the same archive, properties are retained across versions unless you specify new values.

NOTE: Each time you create a **new** schema archive, DB Change Manager includes all properties of the specified data source and you must again refine the properties.

SET SCHEMA ARCHIVE JOB NOTIFICATIONS

When a job completes, DB Change Manager provides various ways of notifying you. See [Setting Job Notifications](#) for details.

SPECIFY SCHEMA ARCHIVE DDL OPTIONS

The **Options** tab contains the **DDL Extraction Options** setting, where you define whether DDL is extracted and stored in files during the archive process. You can also obtain the DDL from the archive after the job runs. These options include:

- The default, **None**, means that the process does not separately save the DDL when building the schema archive.
- **One File** means that the DDL will also be saved to a single .SQL file.
- **Multiple Files** means that the DDL will also be saved to multiple .SQL files. Each object will have a corresponding file.

If you choose to store the DDL, you can find the directory location of the saved files on the **History** tab of the editor after archive versions are created.

To specify the DDL extraction process

- In the **Options** tab, select **None**, **One File**, or **Multiple Files** to indicate how the DDL will be stored.

CREATE A VERSION AND SAVE THE SCHEMA ARCHIVE JOB

When you first run an archive job, DB Change Manager creates the first version of the archive. A version stores snapshots of the same archive job for backup or tracking comparison purposes. Versions are listed in the **Archive Job History** table on the **History** tab and in the **Data Source Explorer**.

To create a version and save a schema archive job

- 1 In the **Archive Job Editor**, click **Run Job**.

The **New Unsaved Version** tab lists the properties of the archive.

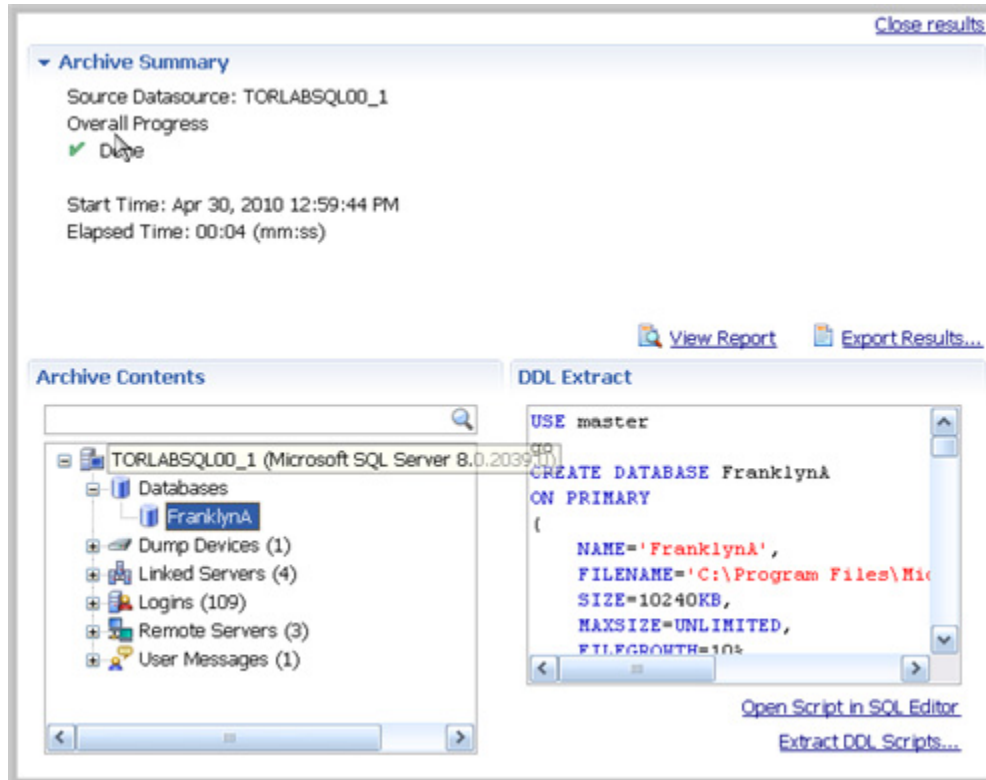
- 2 Highlight the tab and select **File > Save** or **File > Save All**.

A version appears in the **Archive Job History** table on the **History** tab containing the name of the archive, the number of properties retained by the version, a time stamp, and whether the archive job was run from the workbench or the command line.

To see the DDL

In the **Archive Contents** list, right-click on the object and select **Extract**.

The DDL appears in the **DDL Extract** pane.



You create additional versions within the same archive job each time you click **Run Job**. Each new version contains the latest values for the archive job properties you specified on the **Refinements** tab.

To edit the DDL

- 1 Open the **Version** tab of the schema archive.
- 2 At the bottom of the tab, click the **Open Script in SQL Editor** button.

The **SQL Editor** appears. For more information, see [SQL Editor](#).

To extract the DDL of the archive contents to a file

- 1 Open the **Version** tab of the schema archive.
- 2 In the **Archive Contents** list, right-click on the object and select **Extract DDL Scripts**.

The **Extract DDL Scripts** window opens. For more information, see [Extract DDL Scripts](#).

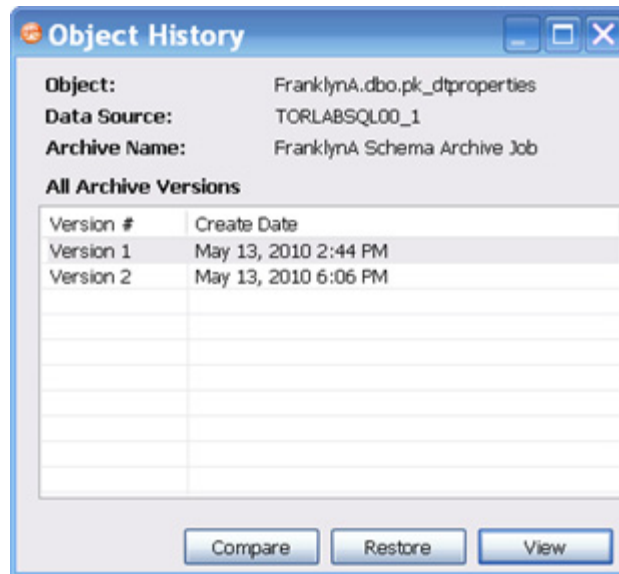
REVIEW THE HISTORY OF ARCHIVED OBJECTS

You can compare two versions of an object and restore an object to a previous version.

To see the history of an archived object

- 1 In the **Archive Contents** list, right-click on the object and select **History**.

The **Object History** window appears.



- 2 Select two versions.
- 3 Click **Compare**.

DB Change Manager compares the two and reports on the differences, if any.

To restore an archived object to its previous version

- 1 In the **Object History** window, select two versions.
- 2 Click **Restore**.

The newer version will be replaced by a copy of the older version.

CREATE A JOB REPORT

After running a schema archive job, you can create a job report using the **View Report** button on the **Version** tab. See [Creating a Job Report](#) for more information.

USE AN ARCHIVE IN SCHEMA COMPARISON JOBS

You can use a schema archive in schema comparison job.

To add a schema archive to a schema comparison job

- In the **Schema Comparison Job** editor, drag and drop an archive version from **Data Source Explorer** to the **Comparison Source** or **Comparison Target** boxes.

Alternatively, if a data source is selected as the source or target, select **Use an Archive** in the respective box and use the drop down menu to choose an archive.

See [Comparing Schemas](#) for more information.

SYNCHRONIZING SCHEMAS

This section contains these topics:

- [Resolve Schema via a Generated SQL Script](#)
- [Execute a SQL Synchronization Script](#)
- [Configure an SQL Session](#)

RESOLVE SCHEMA VIA A GENERATED SQL SCRIPT

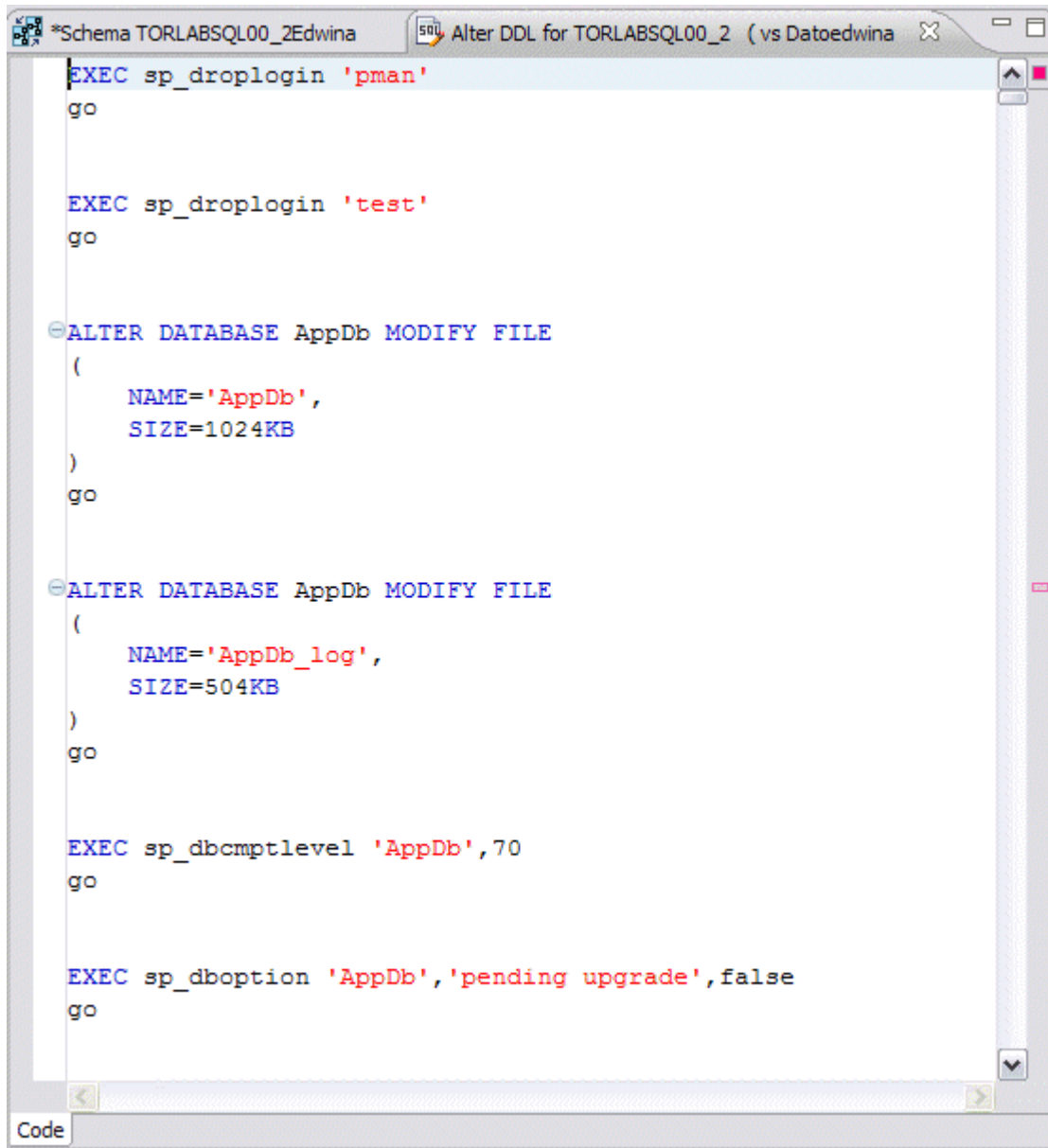
Once a schema comparison job has completed, you can choose to run synchronization commands that match the schemas to each other. The synchronization process generates CREATE, DROP, and ALTER statements, and stored procedures on the source and target data sources.

To generate a SQL synchronization script

- On the **Results** tab, click **Sync All in the Resolution** column of the **Comparison Results** table.

A synchronizing script is produced in another window.

The script can be modified manually, executed as needed, and saved at any time.



```
*Schema TORLABSQL00_2Edwina Alter DDL for TORLABSQL00_2 (vs Datoedwina)
EXEC sp_droplogin 'pman'
go

EXEC sp_droplogin 'test'
go

ALTER DATABASE AppDb MODIFY FILE
(
    NAME='AppDb',
    SIZE=1024KB
)
go

ALTER DATABASE AppDb MODIFY FILE
(
    NAME='AppDb_log',
    SIZE=504KB
)
go

EXEC sp_dbcmptlevel 'AppDb',70
go

EXEC sp_dboption 'AppDb','pending upgrade',false
go
```

Code

EXECUTE A SQL SYNCHRONIZATION SCRIPT

You execute SQL synchronization scripts using commands in the **Run** menu or the command icons located along the top right of **SQL Editor**.



When an SQL file is open in the workspace, select it and choose a database and an associated catalog on which you want to execute file using the drop down menus in **SQL Editor**. You can click the execute icon to execute the code on the specified database and catalog, start a transaction or commit a transaction, or modify the SQL session options prior to execution.

To execute SQL code

- 1 Open the SQL file you want to run.
- 2 Make sure that it is associated with the correct database.
- 3 Click the **Execute SQL** icon.

DB Change Manager executes the code on the data source you specified and displays the results in the same tab or in a new tab.

To execute a transaction

- 1 Open the transaction file you want to run.
- 2 Make sure that it is associated with the correct database.
- 3 Click the **Start Transaction** icon.

DB Change Manager executes the transaction on the data source you specified.

To commit a transaction

- 1 Open the transaction file you want to commit.
- 2 Make sure that it is associated with the correct database.
- 3 Click the **Commit Transaction** icon.

DB Change Manager commits the transaction on the data source you specified.

You can set transactions to auto-commit prior to execution on the **SQL Execution** node of the **Preferences** panel.

NOTE: The synchronization script generation process does not support all object parameters. You should add the following object parameters, as needed, after a synchronization script is generated: fillfactor, consumers, allow_dup_row, and statistics.

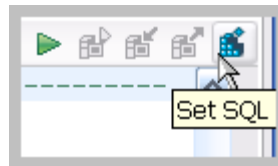
You can set transactions to auto-commit prior to execution on the **SQL Execution** node of the **Preferences** panel.

CONFIGURE AN SQL SESSION

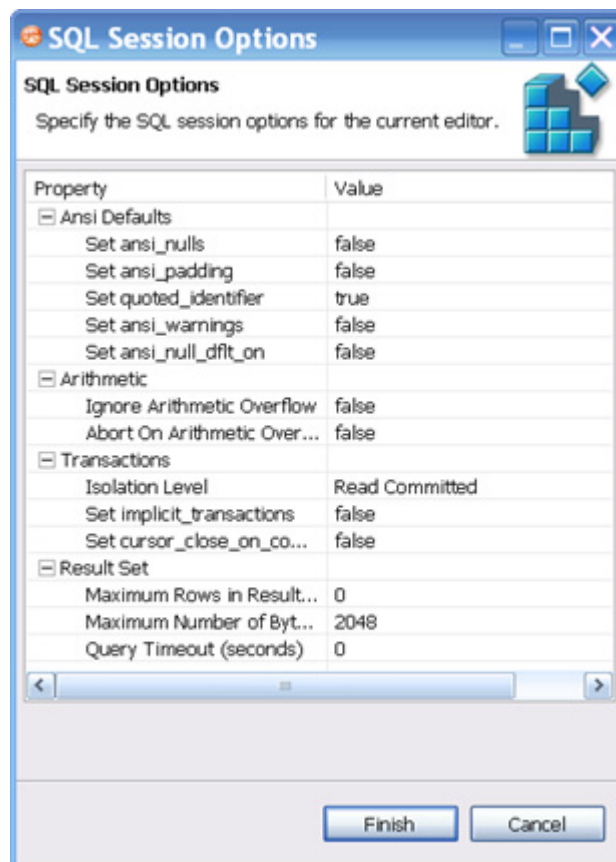
The **SQL Session Options** dialog provides configuration parameters that indicate to DB Change Manager how to execute code in the development environment.

To modify SQL session options

- 1 In the **SQL Editor**, click the **Set SQL Session Options** icon.



The **SQL Session Options** dialog appears.



- 2 Click on individual parameters in the **Value** column to change the configuration of each property.

3 Click **Finish**.

The session options will be changed and DB Change Manager will execute the code as specified when you launch the script.

NOTE: Session options only apply to the corresponding editor and are not retained when executing multiple SQL files.

After running the job, you can generate a job report that summarizes the archived configuration information. You can also generate job reports for any archive versions that exist in the system. Click **View Report** on the **Results** tab. See [Creating a Job Report](#) for more information.

MASKING DATA

Data masking is a way of securing sensitive data during the development or testing phases of a database development project. It is often performed as a security or compliance measure that protects important information. By masking valid production data, you can provide a copy of the data that is “scrambled” but still represents your production environment.

DB Change Manager lets you specify masking rules for moving data between a source and a target in a data comparison job. You can set rules for individual columns, tables, and entire databases. When you run a data comparison with the **Automatically Synchronize** option on, the data on the target is replaced with data from the source and any items configured with a masking rule will be masked. You can then use the masked data in your development and testing environments.

The following tasks describe how to create masked data:

- 1 [Create a New Masking Job](#)
- 2 [Specify a Masking Job Name](#)
- 3 [Specify Source and Target Data Sources](#)
- 4 [Set to Automatically Synchronize](#)
- 5 [Specify Mapping Options](#)
- 6 [Specify Masking Rules for Individual Columns](#)
- 7 [Run the Database Masking Job](#)
- 8 [See the Masked Data](#)

CREATE A NEW MASKING JOB

Use the **Data Comparison Job Editor** to define a masking job.

To start creating a new Data Masking Job:

- Select **File > New > Data Comparison Job**.

SPECIFY A MASKING JOB NAME

The name of the job distinguishes it between other jobs in the application environment.

To name a masked data job

Type the name of the job in the **Name** field of the **Data Masking Job Editor**.

Job Name and Description

Name:

Notes:

Project: **Change Management**

Track results in Compliance Explorer

SPECIFY SOURCE AND TARGET DATA SOURCES

The **Job Sources** section of the **Data Comparison Job Editor** contains the **Data Comparison Source** box, which identifies the original data source, and **Target Data Source** box, which identifies the data source where the data will be altered.

To specify source and target data sources

- 1 Click and drag a data source to be masked from the **Data Source Explorer** to the **Data Comparison Source** box. Drag and drop a target data source, where the mask will be applied, from the **Data Source Explorer** to the **Data Comparison Target** box.
- 2 Use the drop down **Database** menu to specify the database to be masked for the **Data Comparison Source**, and the database to which you want to apply masking definitions on the **Data Comparison Target**.

Job Sources

Data Comparison Source

Name:

Type:

Host:

[Select Datasource](#)

▼

Data Comparison Target

Name:

Type:

Host:

[Select Datasource](#)

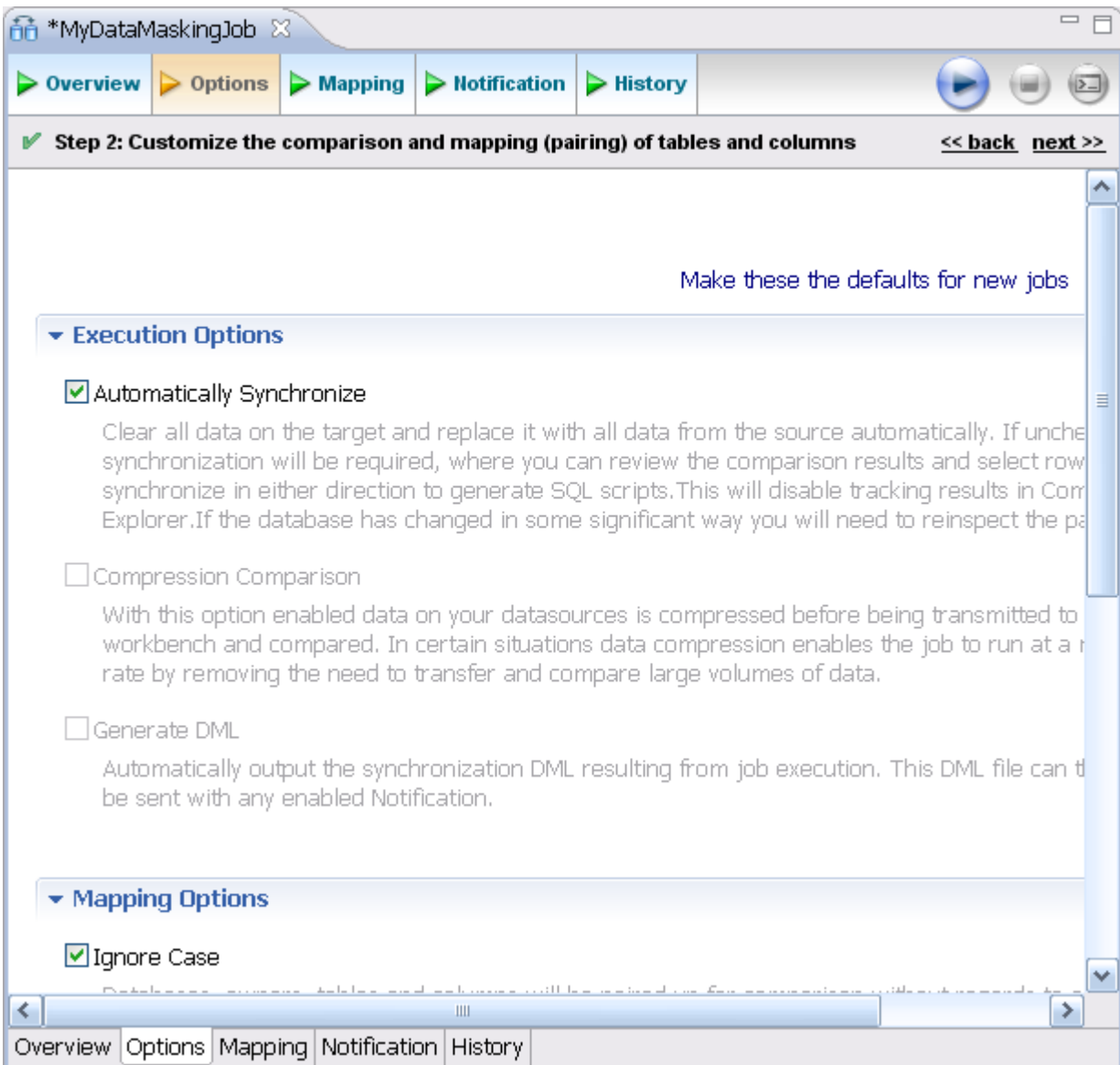
▼

SET TO AUTOMATICALLY SYNCHRONIZE

Prior to running a masking job, set it to automatically clear all the data on the target and replace it with new, masked data.

To set synchronization

- 1 Click on the **Options** tab.
- 2 Select the **Automatically Synchronize** check box.



SPECIFY MAPPING OPTIONS

Use the **Mappings** tab to specify the tables to mask.

You can apply a **Default Masking Rule** to selected columns using the pulldown menus. The options are:

- **Use Default:** The default masking rule as set in the **Database Mapping** box.
- **None:** No masking will be applied.
- **Randomize:** Replaces characters with random other characters.
- **Shuffle:** Reorders the values in different rows; for example, in a First Name column Bob, Ted, and Alice can change to Alice, Bob, and Ted.

Use the check boxes in the left-hand column to de-select any columns that you do not want to include in the data masking job process.

To specify mapping options

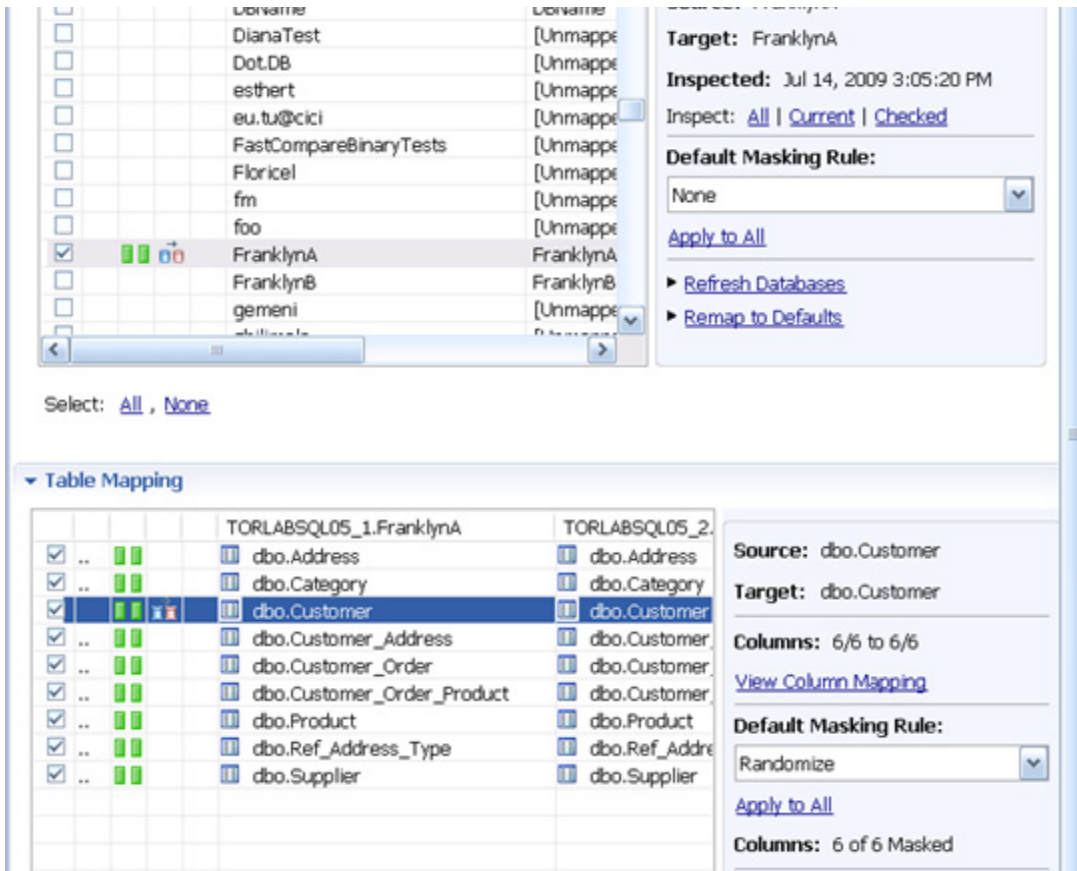
- 1 Click on the **Mappings** tab.
- 2 Select check box of the database you want to mask.
- 3 Click on the **Yes, retrieve the objects in the pair** button to inspect the tables and columns. The **Table Mapping** section opens.
- 4 Select a **Default Masking Rule** for the in the **Database Mapping** box.
- 5 Select a **Default Masking Rule** in the **Table Mapping** box.

SPECIFY MASKING RULES FOR INDIVIDUAL COLUMNS

You can specify different masking rules for different columns.

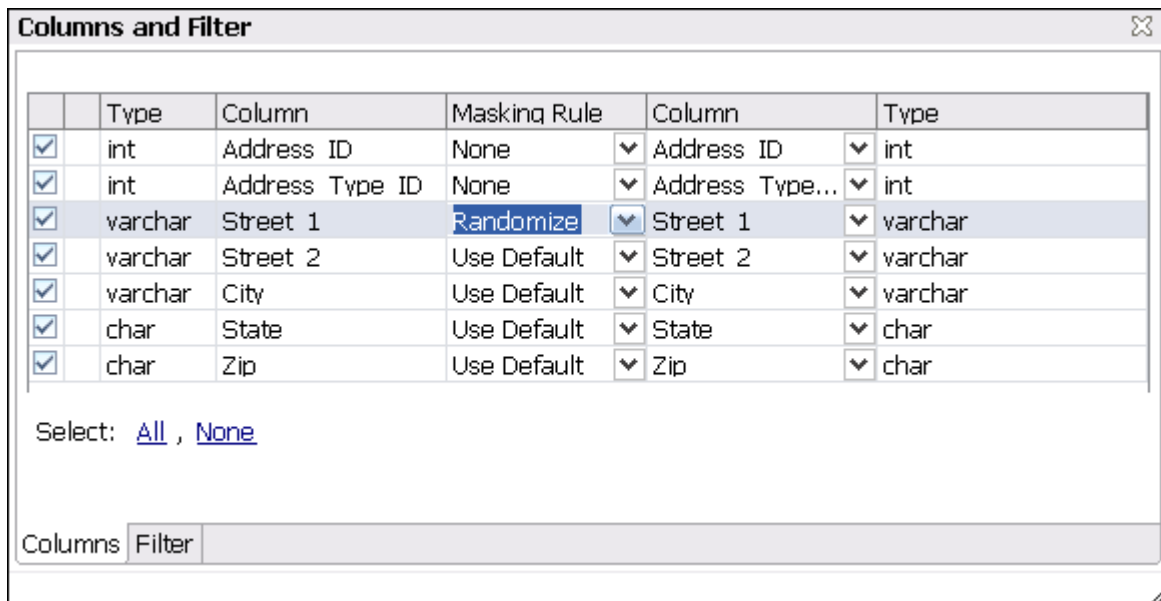
To specify a masking rule for a column

- 1 In the **Table Mapping** box, click **View Column Mapping**.
- 2 Click on the data **Masking Rule** option for the column and select from the pulldown menu one of the following:
 - **Use Default:** The default masking rule as set in the **Database Mapping** box.
 - **None:** No masking will be applied.
 - **Randomize:** Replaces characters with random other characters.



- **Shuffle:** Reorders the values in different rows; for example, in a **First Name** column Bob, Ted, and Alice can change to Alice, Bob, and Ted.

3 Close the window.



RUN THE DATABASE MASKING JOB

To execute the data masking job

- 1 Click **File > Save** to save the job.
- 2 Click the **Run Job** icon in the upper right-hand corner of the screen.
- 3 Click **OK** in the **Confirmation** dialog box to agree that you want to overwrite the target data.

The **Results** tab displays the progression of the masking job as it executes, and the **Rows Written** column indicates how many rows of data have been masked by the job.

Close results

Auto Synchronization Summary

58%

FranklynA.Ref_Address_Type - FranklynA.Ref_Address_Type
 Data Movement Progress: 5002 Rows
 Elapsed Time: 00:40 (mm:ss)

Auto Synchronization Results

EXTOVMGL1501	EXTOVMGL1502	Progress	Results	Migration Status
FranklynA	FranklynA	77%		✓ 2 rows moved
Contacts	Contacts	0%		✓ 5000 rows moved

Overview Options Mapping Notification History Results

SEE THE MASKED DATA

You can see the masked data by running a new **Data Comparison** job and looking at the changes.

- See [Comparing Data](#) for instructions on running a **Data Comparison** job.

For rows that do not have masking applied, you should see "0" in the **Different** column of the **Results Overview** box. For rows that are masked, you should see the number of rows that changed.

In the **Selected Table** box, you can see the masked data in the target table. For example, the First and Last Name data shown below was masked using the **Randomize** setting.

The screenshot shows the 'Results Overview' and 'Selected Table' sections of the Data Comparison tool. The 'Results Overview' table shows a comparison between 'Source: Fran...' and 'Target: Fra...'. The 'Different' column shows 0 for 'dbo.Category' and 348 for 'dbo.Customer'. The 'Selected Table' section shows two side-by-side tables for 'TORLABSQL05_1.FranklynA.dbo.Customer' and 'TORLABSQL05_2.FranklynA.dbo.Customer'. The first table shows original names like 'James Battiste', while the second table shows randomized names like 'Gxibk Yxssfmsl'.

Source: Fran...	Target: Fra...	Matched	Only in TORL...	Only in TORL...	Different
dbo.Category	dbo.Category	16	0	0	0
dbo.Customer	dbo.Customer	0	0	0	348

...	Customer ID	First Name	Last Name
	734	James	Battiste
	735	Samual	Philpott
	736	Noel	Arra
	737	Gaston	Amesqui

...	Customer ID	First Name	Last Name
	734	Gxibk	Yxssfmsl
	735	Jaciab	Fwzfiac
	736	Nael	Eaee
	737	Okeybx	Oaseqfw

MASKING DATA > SEE THE MASKED DATA

AUTOMATING BATCH SCRIPTS

This section contains these topics:

- [Generating Command Line Scripts](#)
- [Using the changec Command Line Program](#)

GENERATING COMMAND LINE SCRIPTS

Each job editor has a **Generate Command Line** button to launch the Script Generation Wizard, which helps you create batch files that automatically run one or more jobs.



By default, each command line script picks up the name of its job.

You can specify output files in CSV, XML, RTF, HTML, or PDF format.

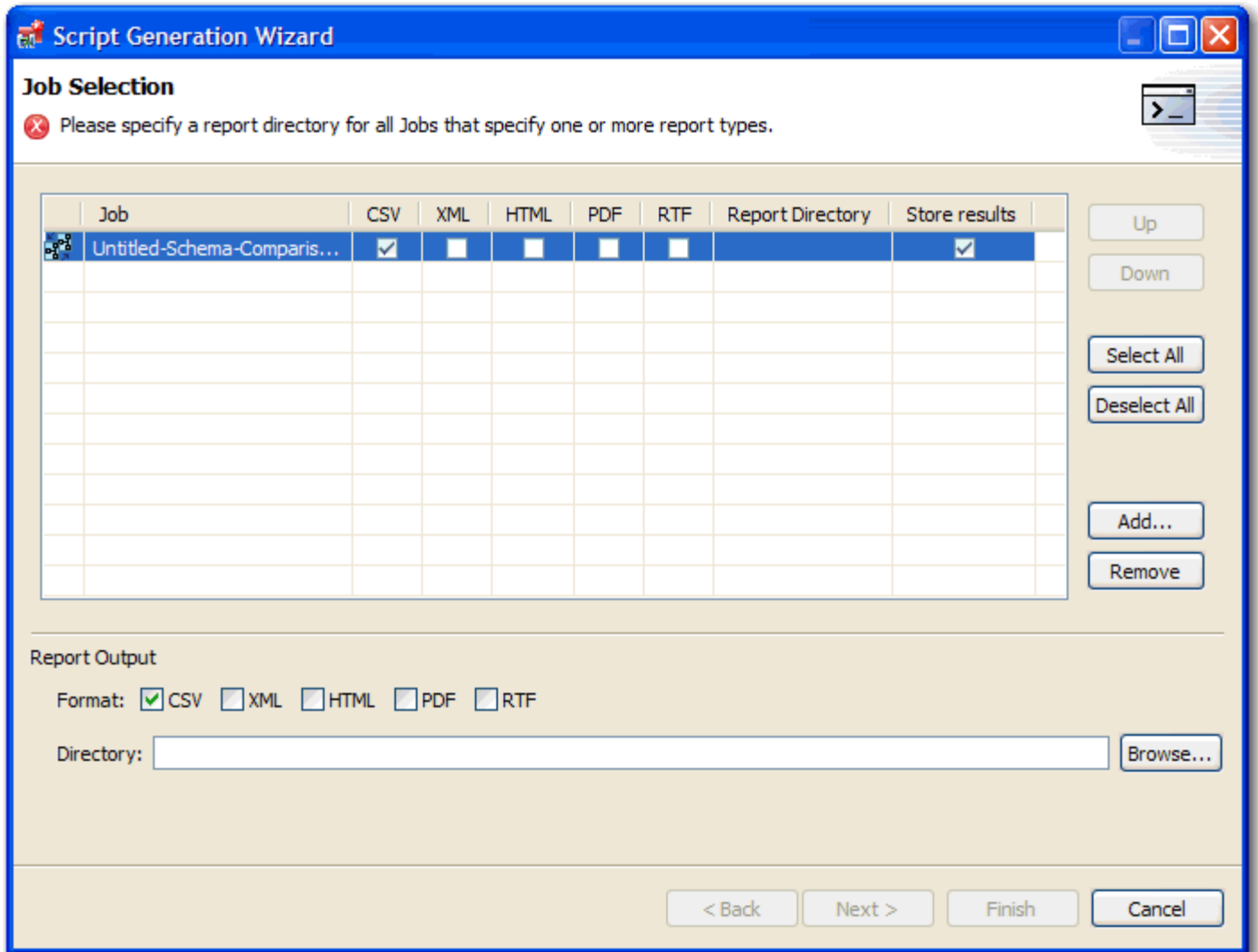
The Report Directory column indicates the location where reports will be created.

The Store Results column indicates if the job results will be saved once the job has executed.

NOTE: Make sure that the job has been saved.

To create a script file:

- 1 In the job editor, select the **Generate Command Line** icon located in the upper-right hand corner.
- 2 In the Script Generation Wizard, locate the job in the table.
- 3 Check the job report output format you prefer.
- 4 Click and specify the **Report Directory** where the report file will be stored.
- 5 If you want to create a batch file to run multiple jobs, click Add, and then add the other jobs.
If needed, reorder the jobs using the Remove, Up, and Down buttons. For example, an archive needs to be created before its version can be used in a comparison job.
- 6 Click **Next**.
- 7 Click and specify the location where you want to store the batch script file.
- 8 Select the script format, **.bat** or **.xml**.
- 9 Click **Finish**. The job immediately runs.



USING THE CHANGECC COMMAND LINE PROGRAM

The changecc command line program runs DB Change Manager job scripts. It is located in the DB Change Manager installation directory.

changecc options include:

- The **-job** option specifies the jobs to run. Its parameter is the fully-qualified job file name and location.
- The **-output** option specifies reporting format for the job. Its parameters are: **CSV**, **XML**, **RTF**, **HTML**, or **PDF**.
- The optional **-dontstore** flag specifies that the results for the job will not be stored.

For example, the syntax below runs the MyConfigCompare database comparison job and generates a report in PDF format:

```
-job
"C:\Users\test1\change_workbench_5_0\workspace\DatabaseChange\MyConfigCompare.dcj"
-output PDF "c:/report"
```

To execute multiple jobs from the same command line, use the following syntax:

```
changec -bulk bulk_file_name
```

Where *bulk_file_name* is a file where each row has a **-job** option.

The **-job** option accepts these regular expressions:

- ** indicates all sub folders in a directory location.
- * indicates any string.

For example, the following command runs all jobs in all projects and folders in the workspace:

```
changec -job workspace:\**\*.caj
```

To run a job from the command line:

- From the command line, enter a command with the following format:

```
changec -job "path_and_name.dcj" -output report_format "report_directory_path"
```

The job immediately runs and saves the report in the specified directory.

DATABASE MANAGEMENT SYSTEM REFERENCE

These topics describe the operations permissions and version information regarding the database management system (DBMS) or systems you use with DB Change Manager:

- [DBMS Operations Permissions](#)
- [Version Information Reference](#)

DBMS OPERATIONS PERMISSIONS

When DB Change Manager performs operations based on data source comparisons, there are minimal permissions required in some cases, dependent on data source platform and the action itself.

- [Permissions Required for DB2 for LUW](#)
- [Permissions Required for Firebird and InterBase](#)
- [Permissions Required for SQL Server](#)
- [Permissions Required for Oracle](#)
- [Permissions Required for Sybase](#)

PERMISSIONS REQUIRED FOR DB2 FOR LUW

Task	Permissions for Own Objects	Permissions for All Objects
Archive and Compare Schemas		
Alter Schema Object		
Configuration Settings		
Compare Data	Select privilege on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job.	Select privilege on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job.
Synchronize/Mask/Move Data	Insert/Update/Delete privileges on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job.	Insert/Update/Delete privileges on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job.

PERMISSIONS REQUIRED FOR FIREBIRD AND INTERBASE

Task	Permissions for Own Objects	Permissions for All Objects
Archive and Compare Schemas	SYSDBA or Database Owner	SYSDBA or Database Owner
Alter Schema Object	SYSDBA or Database Owner	SYSDBA or Database Owner
Configuration Settings	No special privileges are required. DB Change Manager queries the JDBC metadata for basic version information only for configuration content.	No special privileges are required. DB Change Manager queries the JDBC metadata for basic version information only for configuration content.
Compare Data	Select privilege on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job.	Select privilege on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job.
Synchronize/Mask/Move Data	Insert/Update/Delete privileges on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job.	Insert/Update/Delete privileges on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job.

PERMISSIONS REQUIRED FOR SQL SERVER

Task	Permissions for Own Objects	Permissions for All Objects
Archive and Compare Schemas	No special privileges are required. Encrypted stored procedures cannot be viewed.	No special privileges are required. Encrypted stored procedures cannot be viewed.
Alter Schema Object	Permission to connect to the MSDB database. No special privileges for objects owned by the user.	sysadmin On SQL Server 2000, you must enable sp_configure allow updates, and have permission to connect to the MSDB database.
Configuration Settings	Not applicable	sysadmin
Compare Data	Select privilege on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job. Encrypted stored procedures cannot be viewed.	Select privilege on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job. Encrypted stored procedures cannot be viewed.
Synchronize/Mask/Move Data	Insert/Update/Delete privileges on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job.	Insert/Update/Delete privileges on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job.

PERMISSIONS REQUIRED FOR ORACLE

Task	Permissions for Own Objects	Permissions for All Objects
Archive and Compare Schemas	No special privileges are required	SELECT_CATALOG_ROLE
Alter Schema Object	No special privileges are required	DBA role to alter system objects
Configuration Settings	Not applicable	DBA role
Compare Data	Select privilege on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job.	Select privilege on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job.
Synchronize/Mask/Move Data	Insert/Update/Delete privileges on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job.	Insert/Update/Delete privileges on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job.

PERMISSIONS REQUIRED FOR SYBASE

Task	Permissions for Own Objects	Permissions for All Objects
Archive and Compare Schemas	No special privileges are required. Stored procedures cannot be hidden, and syscomment.text read must be allowed in order to compare or archive stored procedures.	No special privileges are required. Stored procedures cannot be hidden, and syscomment.text read must be allowed in order to compare or archive stored procedures.
Alter Schema Object	No special privileges are required	sa_role Enable sp_configure allow updates
Configuration Settings	Not applicable	sa_role
Compare Data	Select privilege on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job. The compression comparison function is unavailable for Sybase at this time.	Select privilege on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job. The compression comparison function is unavailable for Sybase at this time.
Synchronize/Mask/Move Data	Insert/Update/Delete privileges on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job.	Insert/Update/Delete privileges on the relevant tables and views. Schema archive permissions on the database or schema objects used in the job.

VERSION INFORMATION REFERENCE

These topics describe the version information used when comparing, archiving, or creating a configuration standard:

- [Firebird and InterBase Version Information](#)
- [IBM DB2 for LUW Version Information](#)
- [Oracle Version Information](#)
- [SQL Server Version Information](#)
- [Sybase Version Information](#)

FIREBIRD AND INTERBASE VERSION INFORMATION

For Firebird and InterBase:

Version: Revision, Minor, Major, Full, and Build Number

Product: Name and Information

IBM DB2 FOR LUW VERSION INFORMATION

Parameter	Description
INST_NAME	Name of the current instance.
IS_INST_PARTITIONABLE	Indicates if the current instance is a partitionable database server instance. Possible return values are 0 if it is not a partitionable database server instance, and 1 if it is a partitionable database server instance.
NUM_DBPARTITIONS	The number of database partitions. If it is not a partitioned database environment, this value is 1.
INST_PTR_SIZE	The bit size of the current instance (32 or 64).
RELEASE_NUM	The internal release number, as returned by the db2level command. For example, 03030106.
SERVICE_LEVEL	The service level, as returned by the db2level command. For example, DB2 v8.1.1.80.

Parameter	Description
BLD_LEVEL	The build level, as returned by the db2level command. For example, n041021.
PTF	The program temporary fix (PTF) identifier, as returned by the db2level command. For example U498350.
FIXPACK_NUM	The FixPak number, as returned by the db2level command. For example, 9.

ORACLE VERSION INFORMATION

On Oracle platforms, DB Change Manager provides a list of all products installed on the Oracle instance and their versions.

SQL SERVER VERSION INFORMATION

On SQL Server, the following parameters are included:

- Product Version (e.g. 9.00.3042.00)
- Platform (e.g. NT INTEL X86)
- Language
- Physical Memory
- Processor Count
- Processor Type
- Windows Version

NOTE: For entries that contain internal values, the numeric attribute is also available for standard comparisons. For example, the Product Version attribute has a character and an internal value of "8.00.2039" and 524288, respectively. DB Change Manager supplies the Product Version as "8.00.2039", and the Product Version Internal Value as 524288.

SYBASE VERSION INFORMATION

On Sybase platforms, two groups of configuration parameters are supplied:

- The Server Information group displays the results of executing sp_server_info, where the attribute name is the configuration parameter and the value is its value.

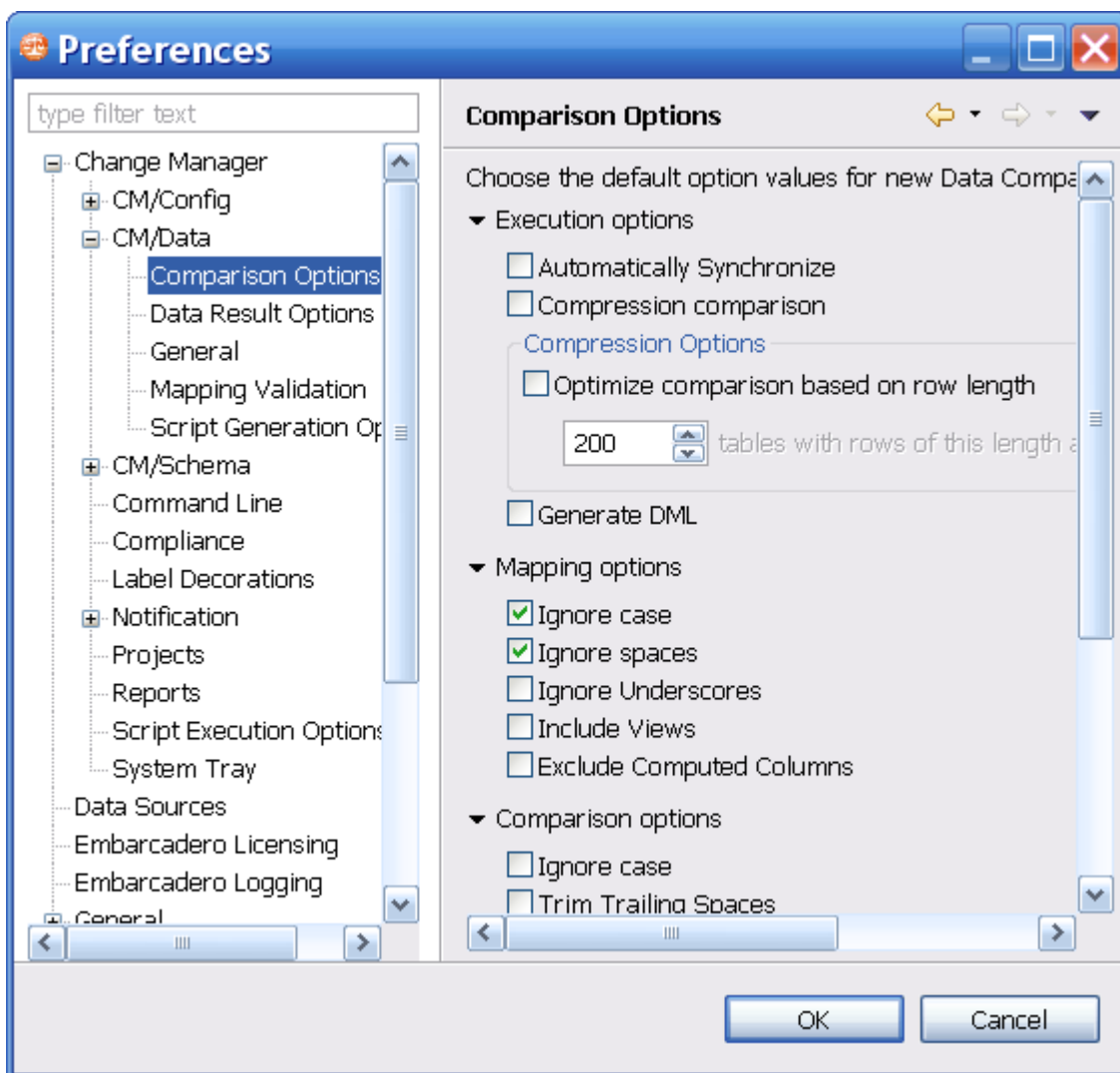
- The Version Information group, which contains the following parameters:
 - Build Number
 - Build Type
 - Compilation Date & Time
 - Compilation Kernel
 - Compilation OS Bit Size
 - Compilation OS Platform
 - EBF Version
 - Post Build Type
 - Product
 - Product Code
 - Version

PREFERENCES REFERENCE

To customize the interface, select **Window > Preferences**. The **Preferences** dialog lets you set general Eclipse as well as DB Change Manager options. Depending on your Eclipse environment, you may see additional nodes in the **Preferences**.

- [DB Change Manager Preferences](#)
- [Eclipse SQL Development Preferences](#)

Once you are familiar with the options, you can use the **Preferences** menu to select a specific category of DB Change Manager options.



DB CHANGE MANAGER PREFERENCES

The following sections describe preferences specific to DB Change Manager:

[CM/Config > Archive Options](#)

[CM/Config > Comparison Options](#)

[CM/Data > Comparison Options](#)

[CM/Data > Data Result Options](#)

[CM/Data > General](#)

[CM/Data > Mapping Validation](#)

[CM/Data > Script Generation Options](#)

[CM/Schema > Archive Options](#)

[CM/Schema > Comparison Options](#)

[CM/Schema > General](#)

[CM/Schema > Mapping Options](#)

[Command Line](#)

[Compliance](#)

[Label Decorations](#)

[Notification > Email Notification](#)

[Notification > Email Notification > Advance SettingsScript Execution Options](#)

[Notification > Email Notification > Contacts](#)

[Notification > Email Notification > Templates](#)

[Notification > File System Notifier](#)

[Notification > Popup Notification](#)

[Notification > Popup Notification > Advanced Settings](#)

[Projects](#)

[Reports](#)

[Script Execution Options](#)

[System Tray](#)

[Task Scheduler](#)

[Data Sources](#)

CM/CONFIG > ARCHIVE OPTIONS

You can set this option to take effect whenever you create a new configuration archive job:

- **Extract DDL**

By default, a new configuration archive will not include DDL. If you want each new archive to include the DDL, check this box and click **Apply**.

CM/CONFIG > COMPARISON OPTIONS

You can set this option to take effect whenever you create a new configuration comparison job:

- **Extract DDL**

By default, a new configuration comparison will not include DDL. If you want each new archive to include the DDL, check this box and click **Apply**.

CM/DATA > COMPARISON OPTIONS

The **Comparison Options** defaults correspond to the job options listed on the **Options** tab of the **Data Comparison Job** editor. See [Specify Execution, Mapping, Comparison, and Resolution Options](#) for more information about these options.

EXECUTION OPTIONS

Execution Options control aspects of processing the job:

- **Automatically Synchronize** indicates that the job will automatically attempt to synchronize any issues it finds between the source and target data sources upon completing comparison execution. If you leave this option unchecked, you will be able to generate a synchronization script at a later point, after job completion.
- **Compression Comparison** speeds up a job by compressing row data for faster transmission and comparison. The compression does not affect the accuracy of a job. This option is useful with slow networks, with large volumes of data in each row, and with faster database servers.

NOTE: This option is not supported for Firebird, InterBase, and Sybase.

- **Optimize comparison based on row length**

The value determines when compression is applied on a table during the data comparison process, based total size of all of the rows. If the size of the rows is greater than the this value, then compression is automatically enabled.

- The **Generate DML** option indicates that the synchronization DML used in the job will automatically be placed in a separate file. If this option is selected, you can specify on the **Notifications** tab to send the resulting output with any notification.

MAPPING OPTIONS

Mapping Options control how the data sources in the job are mapped to each other:

- **Ignore Case** indicates that objects are analyzed without regards to case. For example, the table "Product_Prices" will be compared to the table "PRODUCT_PRICES".
- **Ignore Spaces** indicates that spaces will be ignored when analyzing objects. For example, the table "Product Prices" will be compared to the table "ProductPrices".
- **Ignore Underscores** disregards underscores in object names for comparison purposes. For example, the table "Product_Prices" will be compared to the table "ProductPrices".
- **Include Views** indicates that the comparison process will include views during the mapping process. If this option is not selected, views are automatically excluded from the comparison.
- **Exclude Computed Columns** indicates that the comparison does not include columns whose values are computed in the mapping and comparison processes.

See [Customize Database Mapping](#) for more information about these options.

COMPARISON OPTIONS

The **Comparison Options** specify how the job handles naming conventions:

- **Ignore Case** indicates the job ignores case when comparing data. For example, "Apple" matches a corresponding value of "apple".
- **Trim Trailing Spaces** indicates the job comparison ignores extra spaces at the end of values when comparing data. For example, "Apple " matches a corresponding value of "Apple".
- **Match Empty Strings with Nulls** indicates that the job comparison considers empty values to be the same as null values for matching purposes.
- **Cache Long Datatype Values** indicates that long data types (blob, clob, varchar(max), etc.) values that do not match are cached during comparison and available for resolution during that stage of the process. When this value is disabled, the results of the job indicate that the values did not match but resolution is unavailable. Individual long datatype values can be retrieved when exploring results regardless of the selection of this property.
- **Check Rows from Source** checks the rows that exist in the source tables. If the option is not selected, the rows that exist on the source tables will not be included in the comparison.
- **Check Rows from Target** checks the rows that exist in the target tables. If the option is not selected, the rows that exist on the target tables will not be included in the comparison.

RESOLUTION OPTIONS

The **Resolution Options** section determines how the job resolves any discrepancies discovered during the comparison process.

NOTE: These options are not supported for Firebird and InterBase.

- **Drop and Recreate Foreign Keys** indicates that all foreign keys are dropped prior to applying changes during the resolution process. This option should be used with **Wrap Statements in Transaction** to ensure that foreign keys do not remain dropped in the event that the synchronization script used to resolve it fails.

NOTE: This option is only available for SQL Server scripts.

- **Wrap Statements in a Transaction** utilizes a transaction and rolls back any repository changes if it encounters an error.

NOTE: This option is not supported for DB2 for LUW scripts.

- **Enable Identity Column Inserts and Updates** enables changes to identity column values on SQL Server and Sybase platforms. The update and insert statements are automatically included in the synchronization script when this option is selected.

CM/DATA > DATA RESULT OPTIONS

The **Data Result Options** affect how data comparison results are processed:

- **Cache options**

The **Comparison result cache location** specifies the directory path of the result cache, where job data is captured. Use the **Default Location**, or de-select it and browse to a new directory location.

NOTE: This is a temporary cache. Large data comparison jobs may require many system resources.

- **Refresh Interval**

The **Data Movement Progress Refresh Interval** determines how quickly the progress of the data comparison is updated, in seconds.

CM/DATA > GENERAL

The **General** options relate to data mapping:

- **Remembered dialog decisions**

When you select a default masking rule on a database or table, this option affects the masking rule used on the corresponding tables or columns. **Yes** means the rule will be used on all. **No** means the tables or columns will use the rule set explicitly for them. The default **Prompt** will allow you to make the decision at the time you select the masking rule on the database or table.

- **Don't show again decisions**

When the "automatically synchronize" option was toggled, some mappings became invalid and were removed. If you want to see this information, select this box.

CM/DATA > MAPPING VALIDATION

DB Change Manager performs a variety of validation checks when it compares or synchronizes data. You can specify severity responses for mapping validation checks on comparison and synchronization jobs. By default, some checks always indicate an error. For other checks, you can configure the response.

Mapping Validation ← ▾ → ▾ ▾

Choose the severity level for each mapping validation check.
Checks which can't have their severity changed will be disabled.

▼ **Comparison Mapping Validation**

Comparison Key required for comparison: Error ▾

Column data type mapping compatibility: Error ▾

▼ **Auto Synchronization Mapping Validation**

Relational table cycles: Error ▾

Parent tables not mapped: Error ▾

Child tables not mapped: Error ▾

Source and Target Table Mapping Validation: Error ▾

Check Constraints on target table: Warning ▾

Data truncation with length differences: Error ▾

Data truncation with scale/precision differences: Error ▾

Nullable column to Non-Nullable column: Error ▾

Unmapped Non-Nullable target columns without a default value: Error ▾

Non-Unique column to Unique column: Error ▾

The target table contains computed column(s) which generate their own data.: Error ▾

The job requires at least one pair of columns mapped: Error ▾

Unmapped source columns: Warning ▾

Unmapped target columns: Warning ▾

JDBC Driver support for data types: Error ▾

Data Type conversion support on Data Movement: Error ▾

Restore Defaults Apply

- **Ignore** means there will be no indication regarding that specific validation check.
- **Warning** means that a warning message indicating that there is a problem with the mapping validation will be issued.
- **Error** means that an error message indicating that there is a problem with the mapping validation will be issued.

Once you configure the responses, click **Apply**.

CM/DATA > SCRIPT GENERATION OPTIONS

Script Generation Options affect how DB Change Manager builds a synchronization script for a data comparison job. These options are useful if the match key for a table in a data comparison job does not reflect uniqueness. For example, if a match is performed on last_name, and there are several values of "Smith" in a table, the data synchronization may have unexpected results.

- **Warn about duplicate rows**

- By default, DB Change Manager will issue a warning whenever it encounters duplicate rows while generating a synchronization script.

- **Exclude duplicate rows**

If you want to automatically ignore duplicate rows while generating a synchronization script, select this option.

- **Set quoted identifier ON/OFF**

For SQL Server and Sybase only, if the quoted identifier database option is off, this option will prefix the data synchronization script with `SET QUOTED IDENTIFIER ON` and suffix it with `SET QUOTED IDENTIFIER OFF`.

Once you configure the responses, click **Apply**.

CM/SCHEMA > ARCHIVE OPTIONS

You can specify defaults for how to extract DDL when the option is enabled in the **Schema Archive Job** editor.

- **None** means that DB Change Manager will create the archive in the development environment workspace. You can then extract the DDL directly from the archive.
- **One File** means that DB Change Manager will export the objects of the archive job to a single .SQL file, in addition to building the archive in the development environment. You can locate the SQL file in the **Project Explorer**.
- **Multiple Files** means that DB Change Manager will export each object of the archive job into individual .SQL files, in addition to building the archive in the development environment. You can locate the SQL files in the **Project Explorer**.

Once you configure the responses, click **Apply**.

CM/SCHEMA > COMPARISON OPTIONS

You can set the default job options for schema comparison jobs. Choose a database tab and then select the check box of the appropriate job option to indicate that it is on by default.

The list of options differ depending on the DMBS of the source and target data sources.

The parameters in the panel correspond to the job options listed on the **Options** tab of the **Schema Comparison Job** editor. The option categories include:

- [CM/Schema > Comparison Options > Include Objects Options](#)
- [CM/Schema > Comparison Options > Compare Options](#)
- [CM/Schema > Comparison Options > Decorator Options](#)
- [CM/Schema > Comparison Options > Identifier Options](#)
- [CM/Schema > Comparison Options > DDL Extract Options](#)
- [CM/Schema > Comparison Options > Additional Options](#)

After you reconfigure any options, click **Apply**.

CM/SCHEMA > COMPARISON OPTIONS > INCLUDE OBJECTS OPTIONS

The **Include Object Options** settings determine what elements of the job source and target will be omitted when running a schema comparison job and synchronization. The list of options differ depending on the DMBS of the source and target data sources.

The **Include Object Options** include:

- The **Exist in Source Only (Create)** option specifies that objects only existing on the source will appear in the job results.
- The **Exist in Source and Target and Don't Match (Alter)** option specifies that objects that differ between the source and the target will appear in the job results.
- The **Force Extended Alter** option specifies that the synchronization process will always drop and recreate objects that don't match instead of using an ALTER statement on existing objects.
- The **Include Dependencies** option indicates that dependent objects will be included in the generated DDL.
- The **Exist in Target Only (Drop)** option specifies that objects only existing on the target will appear in the job results.
- The **Generate Data Copy DDL** indicates that data copy DDL will be generated on extended ALTER statements in the automated synchronization script.

CM/SCHEMA > COMPARISON OPTIONS > COMPARE OPTIONS

Compare Options settings affect how objects are treated in the comparison process. You can include or exclude specific object types in the job, and specify the behavior of the comparison job and synchronization scripts.

TIP: The list of options differ depending on the DMBS of the source and target data sources. If an option is not supported for the database, it is not available in the **Preferences**.

The **Compare Options** include:

- **Ignore Storage** ignores storage comparison when running the job.
- **Ignore Partition Differences** ignores partition differences when running the job. (Oracle only)
- **Ignore Column Order** ignores table column order differences when running the job.
- **Ignore Check Constraints Quotes** ignores quotes in check constraints when running the job.
- **Ignore Table Comment** ignores table comments when running the job.
- **Ignore Text Case** ignores syntax case in text object comparisons when running the job. (views, procedures, trigger, etc.)
- **Ignore Text Comments** ignores comments in text objects when running the job.
- **Ignore Text Whitespace** ignores white space when running the job.
- **Ignore Tablespace File Name** ignores tablespace file names when running the job.
- **Ignore Password Differences** ignores differences in object passwords when running the job. (group, login, role, user, etc.)
- **Ignore Tablespace** ignores tablespaces for table, index, cluster, etc. (IBM only)
- **Ignore Object Permissions** ignores the permissions on objects when running a job.
- **Ignore Logging/No Logging Differences** ignores object differences where one is set to **Logging** and the other is set to **No Logging**. (Oracle only)
- **Ignore Sequence Start Value** ignores the **Start Value** property of an Oracle Sequence. (Oracle only)
- **Ignore Data Capture** ignores the **Data Capture** property for tables. (IBM only)
- **Ignore File Group** ignores the **File Group** property for tables and indexes. (SQL Server only)
- **Ignore Segment** ignores the **Segment** property for tables, indexes, and partition definitions. (Sybase only)

CM/SCHEMA > COMPARISON OPTIONS > DECORATOR OPTIONS

Decorator Options affect particular database objects.

Decorator Options are not supported for Firebird and InterBase.

The **Decorator Options** include:

- **Preserve Target File Group** (on extended alter) (SQL Server)

When an extended alter is generated for an object, the target objects file group will remain the same as the original object if this option is enabled.

If this option is disabled, the targets objects file group will be set to the one used by the source object.

- **Preserve Target Storage** (Oracle)

The functionality is similar to **Preserve Target File Group** but this option applies to the storage properties of Oracle objects.

- **Preserve Target Tablespace** (Oracle)

The functionality is similar to **Preserve Target File Group** but this option applies to the tablespace properties of Oracle objects.

- **Preserve Target Segment** (Sybase)

- **Recompile**

This option triggers a recompilation for dependent procedures, when the target object needs to be dropped and recreated (extended altered).

- **Comment Out Extended Alters for Tablespaces**

An extended alter is essentially a drop and then a recreate. For tablespces, an extended alter is a dangerous operation that may cause data loss.

This option is by default enabled and generates commented out DDL for tablespace extended alter.

CM/SCHEMA > COMPARISON OPTIONS > IDENTIFIER OPTIONS

The **Identifier Options** panel contains parameters that determine what elements of the source and target data source will be ignored during the schema comparison and synchronization process.

- The **Ignore Name Case** option specifies that the syntax case will be ignored when doing comparisons. By default, it is deselected.
- The **Ignore Constraint Names** option specifies that differences in table constraint names will be ignored when doing comparisons. By default, it is deselected.

- The **Ignore Specific Name** option ignores SQL specific names in procedures, functions, and methods. By default, it is deselected.

The **Ignore Specific Name** option is available only for IBM DB2 LUW.

CM/SCHEMA > COMPARISON OPTIONS > DDL EXTRACT OPTIONS

The **DDL Extract Options** section contains the **Extract DDL** parameter. When it is selected, the DDL code appears separate editor window after the comparison job runs.

By default, **Extract DDL** is deselected.

CM/SCHEMA > COMPARISON OPTIONS > ADDITIONAL OPTIONS

Additional comparison options are listed at the bottom of the list of preferences. These include:

- **Enable Dependency Linking**

When this option is activated, the comparison will contain information about dependencies between objects. This information is critical for the ordering of the generated DDL. If the option is not enabled, the schema comparison is faster but when DDL is generated for more than one object, the ordering will not be correct.

By default, **Enable Dependency Linking** is deselected for Oracle, so all GRANT statements for Oracle objects are not extracted and compared.

CM/SCHEMA > GENERAL

The **General** options relate to schema comparison jobs:

- **Show uncomment alter tablespace script confirm dialog**

There is an alter tablespace script confirm dialog that will display by default. Deselect this option if you do not want to see this dialog.

- **Do not ask about running the created jobs**

Select this option if you want to ignore the system prompt when running schema comparison jobs. If you select this option, you can also select whether to automatically run a schema comparison job when an object is created.

- **Automatically launch jobs created from selected objects**

Select this option if you want to automatically run a schema comparison job when an object is created.

Once you configure new defaults, click **Apply**.

CM/SCHEMA > MAPPING OPTIONS

You can specify the default mapping job options available on the **Mapping** tab of the **Schema Comparison Job** editor:

- **Ignore Case**

By default, schema comparison jobs will ignore the case of the texts. Deselect this option if case is important to you.

- **Ignore Spaces**

By default, schema comparison jobs will ignore spaces in the texts. Deselect this option if spaces are important to you.

- **Ignore Underscores**

By default, schema comparison jobs will compare underscores in the texts. Select this option if you do want to ignore underscores.

- **Show Users (Oracle)**

By default, schema comparison jobs will show users from the source and target that have at least one object. For Oracle only, select this option if you want to show all users from the target, regardless of whether they have any objects in their schema, and map them with users from the source.

Once you configure new defaults, click **Apply**.

COMMAND LINE

You can specify the **Default Script Output** for command line jobs:

- **Windows (*.bat)**

By default the command line interface produces batch files.

- **Ant (*.xml)**

Click **Ant (*.xml)** if you want the command line interface to produce XML files by default.

Once you configure new defaults, click **Apply**.

See [Generating Command Line Scripts](#) for more information.

COMPLIANCE

You can specify default compliance ranges for comparison results. These ranges are percentage amounts that specify when to pass, warn, or fail a configuration comparison job in the [Compliance Explorer](#). The compliance options include:

- **Compliance Met**

By default this value is 100%, which allows a job to pass.

- **Compliance Critical**

By default this value is 75, meaning that 75-99% compliance will result in a warning.

- **Compliance Not Met**

By default this value is 0, meaning that 0-74% compliance will result in a failure.

Once you configure new defaults, click **Apply**.

LABEL DECORATIONS

Label Decorations provides a link to the standard Eclipse **Preferences** section where you can specify decorations for an object in the Eclipse interface.

In the **Job Decoration Text** panel, select **Project** to add a decoration in the **Project Explorer**.

Once you configure new defaults, click **Apply**.

NOTIFICATION > EMAIL NOTIFICATION

You can specify email server parameters that will enable email notifications for jobs:

- **Host**

The name of the email host server is required.

- **Port**

The default port number is 25.

- **Sender Address**

The email address to use for sending email from DB Change Manager to others.

- **User Name**

The user name of the sender on the mail server.

- **Password**

The password of the sender on the mail server.

Use the **Test Email** button to validate the settings. DB Change Manager will prompt you for an email address, and then send a test notification email to that address.

Once you have validated the email settings, click **Apply**.

NOTIFICATION > EMAIL NOTIFICATION > ADVANCE SETTINGS

Advanced Settings lets you map SMTP properties to values. Refer to <https://java.sun.com/products/javamail/downloads/index.html> and <https://java.sun.com/products/javamail/FAQ.html> for more information.

SMTP properties are listed at: <http://java.sun.com/products/javamail/javadocs/com/sun/mail/smtp/package-summary.html>.

NOTIFICATION > EMAIL NOTIFICATION > CONTACTS

Contacts are email addresses used in email notifications. When a new notification is sent, it will be sent to all of the addresses added to the contact list.

To add a contact

- 1 In **Preferences > Notification > Email Notification > Contacts**, click **Add**.
- 2 Enter an email address.
- 3 Click **OK**.

You can also import existing contact lists from XML files, or export the current contact list as an XML file.

NOTIFICATION > EMAIL NOTIFICATION > TEMPLATES

You can create and edit email templates that format the texts used in email notifications. The default template shows how the templates work. You can add a template and customize the values of the template variables. Actions include:

- **Edit**

The **Edit** button opens the **Edit Template** dialog for the selected template.

TIP: Press Ctrl-Space on the **Notification** screen to open a list of variables.

- **Add**

The **Add** button opens the **Add Template** dialog.

- **Remove**

The **Remove** button removes the selected template.

To create a new email notification template

- 1 Select the default template.
- 2 Click **Edit**.

- 3 Click the **Template Key** icon on the left to see the template variables that are available.
- 4 In the **Body** section, select and copy the template text.
- 5 Click **Cancel**.
- 6 In **Preferences > Templates**, click **Add**.
- 7 In the **Add a Template** dialog **Body** section, paste the copied text.
- 8 Edit the text. Click the **Insert template key** dropdown to select from a list of variables.
- 9 Name the template.
- 10 Enter the **Subject** content that will be the subject of your email. You can use variables in a subject.

NOTIFICATION > FILE SYSTEM NOTIFIER

You can specify where file system notifications will be stored using **Preferences > Notification > File System Notifier** options, including:

- **Enable file system notification service**

By default, file system notifications are not enabled. Select this option to enable them.

- **Report, Log, and Script Directories**

By default, file system notifications are created in the Documents and Settings/username/Change Manager Results directory. Click Browse to change the default directories.

Once you have validated the email settings, click **Apply**.

NOTIFICATION > POPUP NOTIFICATION

You can set the default options for System Tray notifications, including:

- **Enable system tray notification service**

By default, system tray notifications are not enabled. Select this option to enable them.

- **Display notification time**

By default, the notification lasts for 3000 milliseconds.

- **Show only when minimized to tray**

By default, the popup will display when the System Tray is minimized.

- **Play sound**

By default, if notifications are enabled, a sound will accompany a notification.

- **Custom sound**

You can change the default sound and select a .wav sound file.

Once you have validated the email settings, click **Apply**.

NOTIFICATION > POPUP NOTIFICATION > ADVANCED SETTINGS

You can change the default options for System Tray notifications. These options include:

- **Generate notification when job is run from**

By default, if notifications are enabled, a notification will be generated for each job run from the Workbench or from the command line. You can limit notifications to one or the other.

- **Generate notification when job finishes**

By default, a notification will be generated for each job that succeeds or fails. You can limit notifications to one or the other.

- **Generate notification for threshold**

This option is currently unavailable.

After you change the notification settings, click **Apply**.

PROJECTS

The **Projects** option applies to the **Project Explorer**:

- **Suppress Project Required Warning Dialog**

This option disables the warning dialog when using the **Project Explorer**. By default it is disabled.

After you change the projects settings, click **Apply**.

REPORTS

The Reports settings let you customize the look, feel, and format of reports issued when you generate a job report for comparison jobs.

- The **Report Style** tab

- **Logo**

You can browse to upload a logo. The logo should be 80 pixels wide by up to 420 pixels tall.

- **Accent color**

You can select an accent color from the color palette.

After you change the style settings, click **Apply**.

- The **Paper Setup** tab

- **Paper size**

You can use the dropdown list to select from a list of standard paper sizes, and choose between **Portrait** or **Landscape** orientations.

- **Margins**

You can set the **Top**, **Right**, **Left**, and **Bottom** margins for the paper. If you prefer not to use inches, set the **Unit** option to either centimeters or millimeters.

After you change the paper setup options, click **Apply**.

- The **Contents** tab

- **CM/Config: Include matching configuration parameters**

By default, matching configuration parameters are included in a report.

- **CM/Data: Include matching data fields (database level)**

By default, matching data fields in databases are included in a report.

- **CM/Data: Include matching data fields (table level)**

By default, matching data fields in tables are included in a report. You cannot change this option.

- **CM/Data: Include detail reports in CM/Data notifiers**

By default, details are included in a data job notification.

- **CM/Schema: Include matching schema objects**

By default, matching schema objects are included in a report.

- **CM/Schema: Include DDL in Schema Reports**

By default, DDL is included in a report. If you do not need the DDL in the report, and to improve performance, deselect this option.

After you change the contents options, click **Apply**.

- The **Font** tab

- **Path**

By default, DB Change Manager includes the FreeSerif.ttf font for use in reports. You can change this to another font on your system.

- **PDF Encoding**

If you change the font, make sure that the appropriate PDF encoding is set. Leave the field blank to use the default encoding.

After you change the font options, click **Apply**.

SCRIPT EXECUTION OPTIONS

The **Script Execution Options** apply to automation scripts:

- **Connection Auto Commit**

By default, auto commit applies to synchronization scripts for data comparison jobs.

If you change the script execution settings, click **Apply**.

SYSTEM TRAY

The **System Tray** options affect the Windows System Tray:

- **Show System Tray Icon**

By default, the System Tray icon is shown.

- **Minimize to System Tray**

You can minimize the system tray by selecting this option.

If you change the System Tray settings, click **Apply**.

TASK SCHEDULER

The **Task Scheduler** options affects batch files:

- **Always schedule auto generated batch files**

By default, automatically generated batch files do not run on schedule. Select this option to require scheduling.

- **Batch files directory**

By default, batch files are saved in Documents and Settings\username\change_manager_x\workspace\Change Management\AutoGeneratedBatchFiles. You can change this to another directory.

If you change the task scheduler settings, click **Apply**.

DATA SOURCES

The **Data Sources** preferences specify where registered data source definitions and metadata are stored:

- **Data Sources**

By default, batch files are saved in Documents and Settings\username\Application Data\Embarcadero\Data Sources. You can change this to another directory.

- **Metadata**

By default, metadata files are saved in Documents and Settings\username\Application Data\Embarcadero\Data Sources\metadata. You can change this to another directory.

If you change the data source file location settings, click **Apply**.

ECLIPSE SQL DEVELOPMENT PREFERENCES

Some **SQL Development** preferences are important for DB Change Manager jobs, including:

- [SQL Development > SQL Editor](#)
- [SQL Development > SQL Editor > Code Assist](#)[SQL Development > SQL Editor > Code Formatter](#)
- [SQL Development > SQL Editor > Code Formatter](#)
- [SQL Development > SQL Editor > Results Viewer](#)
- [SQL Development > SQL Editor > Syntax Coloring](#)
- [SQL Development > SQL Filters](#)

SQL DEVELOPMENT > SQL EDITOR

The **SQL Editor** options include:

- **Enable SQL Parser**

Deselecting **Enable SQL Parser** will disable many of the “smart” SQL editor features, including code formatting, auto completion, semantic validation, and hyperlinks. For better performance, you may disable the parser for files above a specified size.

- **Parsing Delay**

This option specifies the amount of time that the parser delays prior to activating features after a keystroke.

- **Severity Level for Semantic Validation Problems**

This option determines how semantic code errors are flagged in the **SQL Editor** and the **Problems** view.

SQL DEVELOPMENT > SQL EDITOR > CODE ASSIST

The **Code Assist** options affect code completion features in the SQL Editor:

- **Enable Auto Activation**

By default, **Code Assist** is enabled. The **Auto-activation delay** determines how long before the **Code Assist** feature activates.

- **Insert Single Proposals Automatically**

By default, if only a single code completion suggestion is returned, it is automatically inserted. To disable automatic inserts, deselect this option.

- **Fully Qualify Completions Automatically**

By default, code completion results are the minimum required to identify the object. Select this option to return specific, fully-qualified code completions.

- **Code Assist Color Options**

By default, the background color of code completion proposals is white and the foreground is black. You can modify these options using the color picker.

SQL DEVELOPMENT > SQL EDITOR > CODE FORMATTER

The **Code Formatter** options configure code formatting in the **SQL Editor**. DB Change Manager provides a built-in profile.

You can choose from a drop down list of formatting profiles, and the **Preview** box displays how each profile formats code. You can import your own custom **Code Formatter** profile.

SQL DEVELOPMENT > SQL EDITOR > RESULTS VIEWER

The Results Viewer options specify how the **Results** view displays results.

- **Grid refresh interval**

This option defines the speed in milliseconds that the **Results** view refreshes.

- **Maximum result rows to sort**

By default, the results will sort up to 25,000 rows.

- **Maximum number of result sets**

By default, up to 100 results sets will be included.

- **Stripe the rows of the results table**

This option adds intermittent highlighted bars in the **Results** view.

- **Display results in separate tab in the SQL Editor**

This option will open the **Results** view in a separate tab on the Workbench.

- **Suppress execution messages**

Select this option to disable system messages in the **Results** view.

- **Results Format**

By default, **Results** are in a grid format. You can change that to HTML, XML, or text format.

SQL DEVELOPMENT > SQL EDITOR > SYNTAX COLORING

The **Syntax Coloring** options change the look and feel of code syntax in the **SQL Editor**. Use the **Element** box to select the comment type or SQL element you want to modify, and assign color options on the right. The **Preview** box displays sample code that updates according to the changes you make.

SQL DEVELOPMENT > SQL FILTERS

The **SQL Filters** options specify whether or not to ignore system objects when performing jobs. By default, system objects are ignored.

For more information, see [Filtering Data Source Objects](#).

USER INTERFACE REFERENCE

This section describes the various dialogs, editors, and menus available in DB Change Manager. Understanding how the DB Change Manager user interface functions will help you work faster and more efficiently.

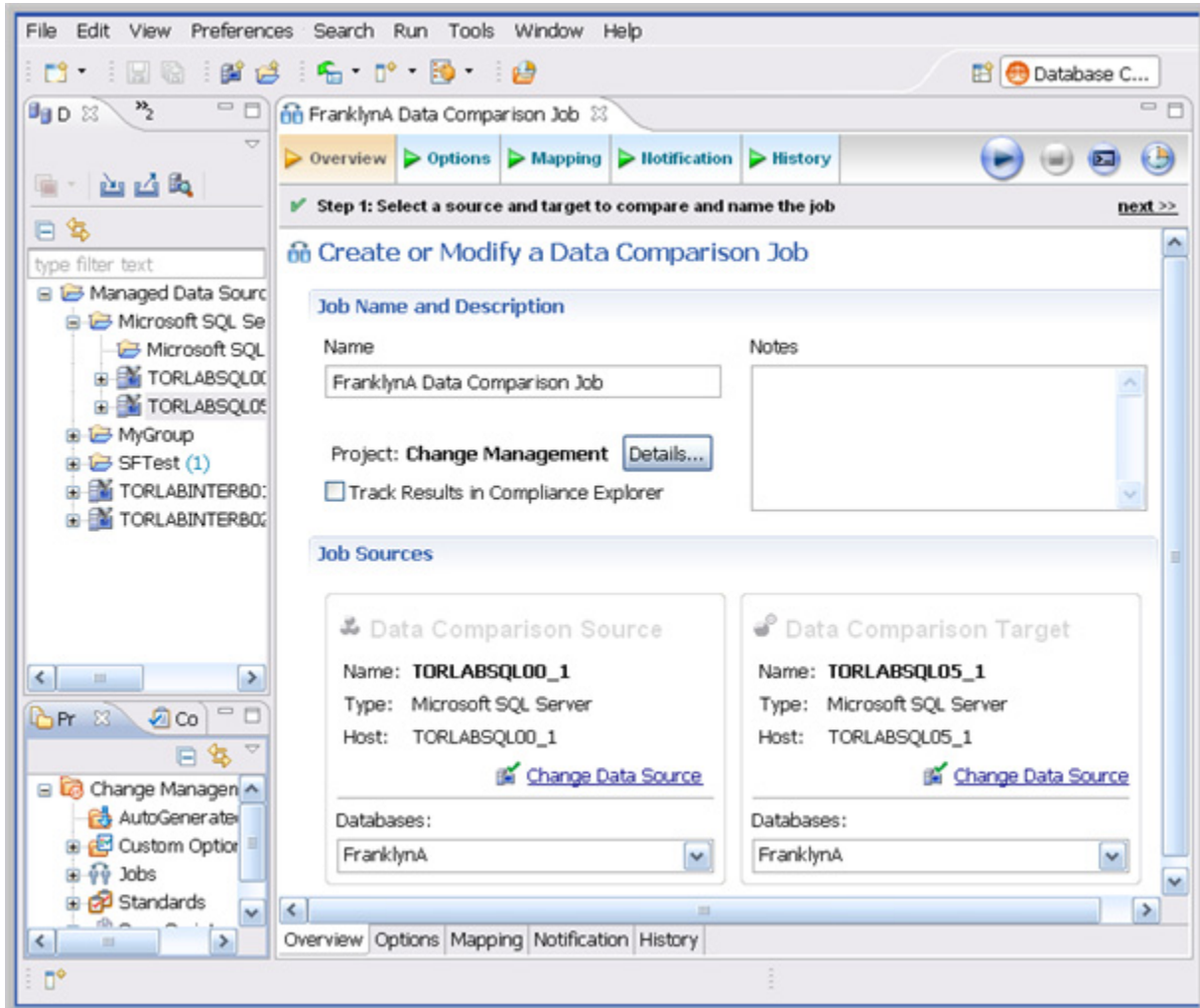
The following interface components compose the DB Change Manager work environment:

- [The DB Change Manager Workbench](#)
- [Welcome Page](#)
- [Workspace](#)
- [Views](#)
 - [Job Explorer](#)
 - [Data Source Explorer](#)
 - [Project Explorer](#)
 - [Navigator View](#)
 - [Compliance Explorer](#)
 - [SQL Errors View](#)
 - [Data Diff View](#)
 - [SQL Log](#)
 - [SQL Errors Log](#)
 - [Search View](#)
- [Editors](#)
 - [Data Comparison Job Editor](#)
 - [Configuration Comparison Job Editor](#)
 - [Configuration Standard Editor](#)
 - [Configuration Archive Job Editor](#)
 - [Schema Comparison Job Editor](#)
 - [Schema Archive Job Editor](#)
 - [SQL Editor](#)
 - [SQL Session Options](#)

THE DB CHANGE MANAGER WORKBENCH

The Workbench is the DB Change Manager development environment. It provides the interface to create, manage, and navigate DB Change Manager processes through interaction with views, editors, and menus.

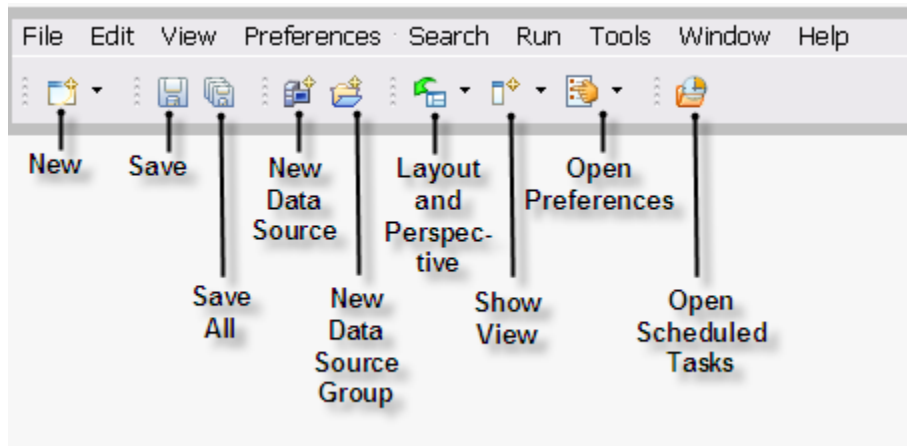
Based on Eclipse standards, the individual parts of the Workbench share certain elements in common and provide you with a uniform, integrated system for working in DB Change Manager. In addition, you can customize aspects of the environment to suite your development requirements.



The Workbench space is composed of [Views](#) and [Editors](#), as well as the **Menu Bar** and **Command Toolbar**.

- The **Menu Bar** options control functionality and processes. They apply either to the whole application, or to individual views and editors, depending on the what has the focus of the mouse.

- The **Command Toolbar** icons run commonly-used **Menu Bar** commands.

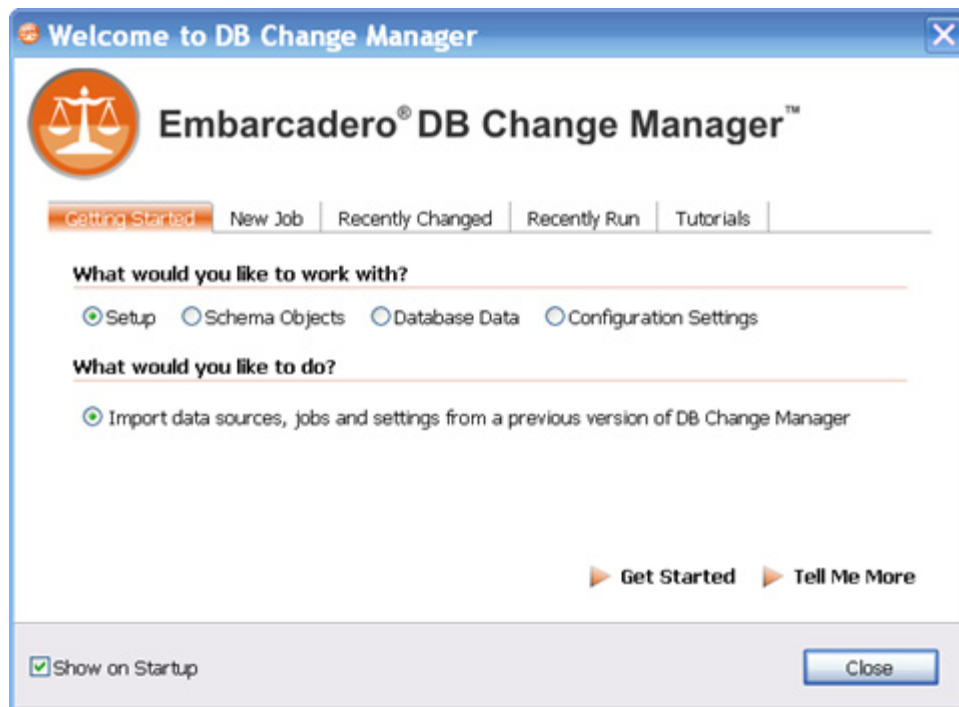


This reference includes:

- [Welcome Page](#)
- [Workspace](#)
- [Views](#)
- [Editors](#)

WELCOME PAGE

The **Welcome page** opens when you launch DB Change Manager. It provides easy access to information for new users, and helps you get started with the application.

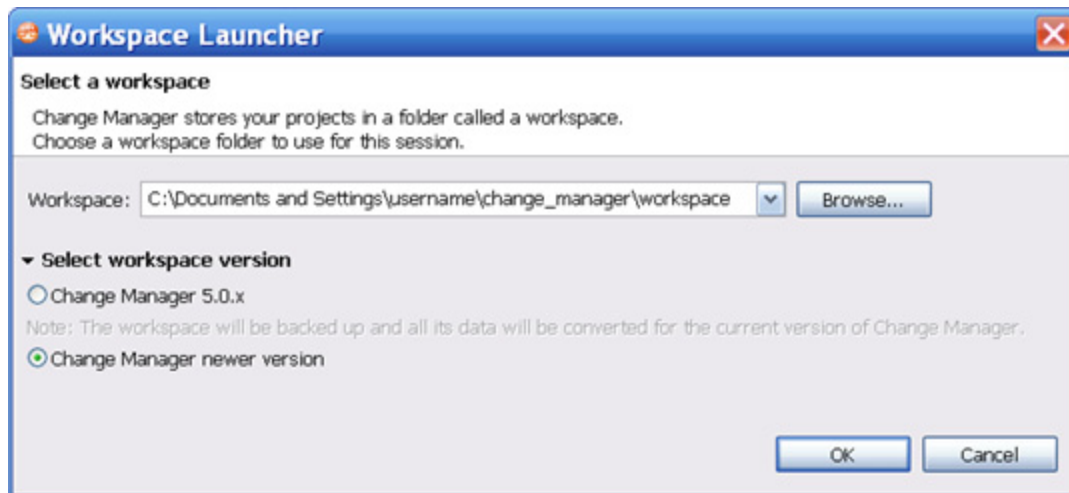


- The **Getting Started** tab presents commonly-performed tasks. Choose the options you want and click **Get Started**. A list of related instructions appears.
- The **New Job** tab provides a list of jobs that you can run. Click the job you want to start, and the appropriate editor opens.
- The **Recently Changed** tab displays jobs that have been recently modified. Select a job from the list and click **Open Job** to open the job in the Workbench.
- The **Recently Run** tab displays jobs that have been recently executed. Select a job from the list and click **Open Job** to open the job in the Workbench.
- The **Tutorials** tab lists DB Change Manager tutorials which describe how to perform various tasks. Choose a tutorial and click **Start the Tutorial** to begin.
- The **Show on Startup** check box controls whether the **Welcome page** opens when you start DB Change Manager.

WORKSPACE

A workspace is the location where your project files are stored. When you have many Eclipse projects, workspaces make it easier to organize and manage your work.

You can switch between multiple DB Change Manager workspaces, or convert workspaces to new versions of DB Change Manager.



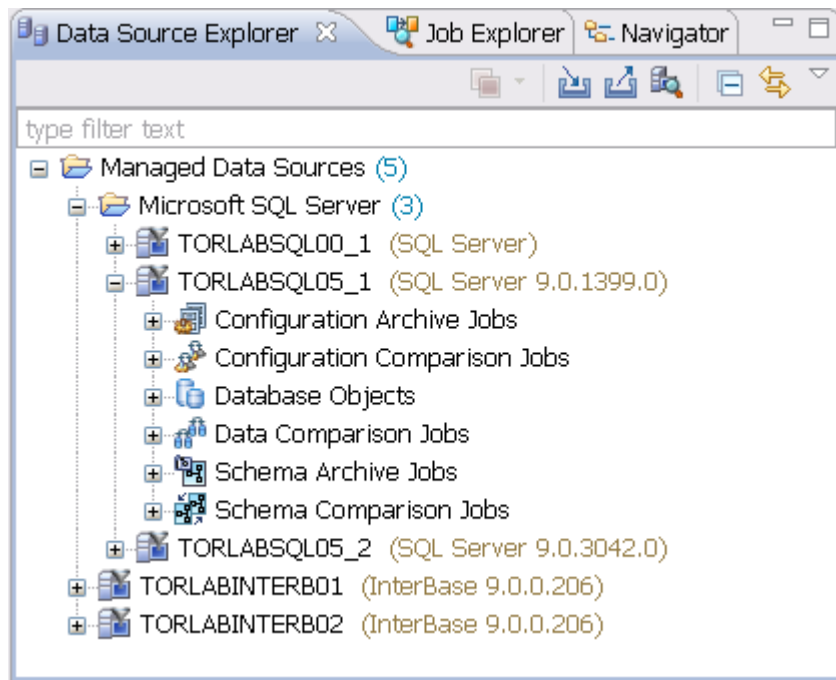
To switch DB Change Manager workspaces

- 1 From the **Main** menu, click **File > Switch Workspace**.
- 2 In the **Workspace Launcher** dialog, browse to the directory and select it.
- 3 Select the version of the older workspace.
- 4 Click **OK**.

VIEWS

A view is a Workbench interface component typically used to navigate a hierarchy of information or list, open editors for jobs, or display properties or settings.

Views complement editors and provide alternative presentations or navigations of the information in the Workbench. For example, **Data Source Explorer** provides a list of all registered data sources in DB Change Manager, organized by group folder.



Views can be active or inactive, but only one view can be active at any one time. The title of an active view is highlighted and the view is the target for common commands such as cut, copy, and paste. Views can be stacked with other views to create tabbed notebooks.

All views have two ways to access command menus:

- Right-click the view tab
- Click the icon in the view toolbar

Here are some of the ways you can interact with views:

To move a view

- Click the tab of the view and drag your mouse. A frame indicates where the view will fit in the Workbench. Release the button to move the view to its new location.

To toggle a floating view

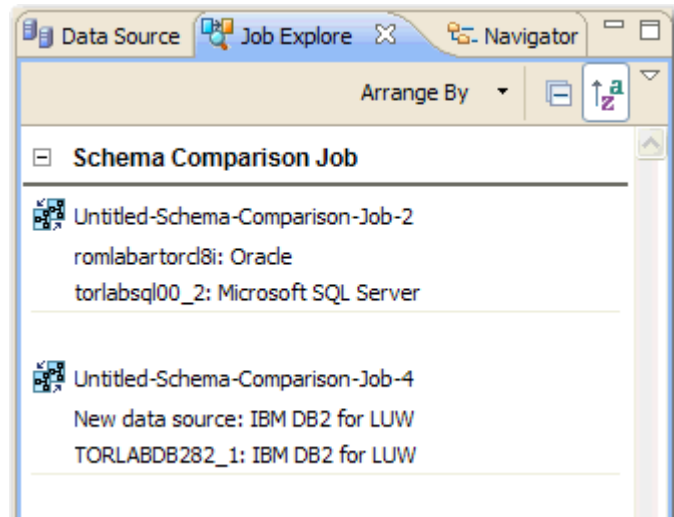
- Right-click the tab of the view and select **Detached**. The view switches to a floating window format that can be used to position it overtop of 'embedded' views in the Workbench.

To build a tabbed view

- Drag a view over another view and release the mouse button when the frames meet. Both views overlap and create tabs within the same frame. You can combine multiple views in this way.

JOB EXPLORER

Job Explorer provides an interface for managing existing jobs in DB Change Manager.



Use **Job Explorer** when searching for jobs of a specific type or name. You can list jobs according to a wide variety of categories.

To view jobs in Job Explorer

- 1 Navigate to **Job Explorer** and click **Arrange By** to view a list of categories.
- 2 Select a category. **Job Explorer** displays all jobs based on your new selection.

To open a job in its editor

- 3 In **Job Explorer**, double-click a job.

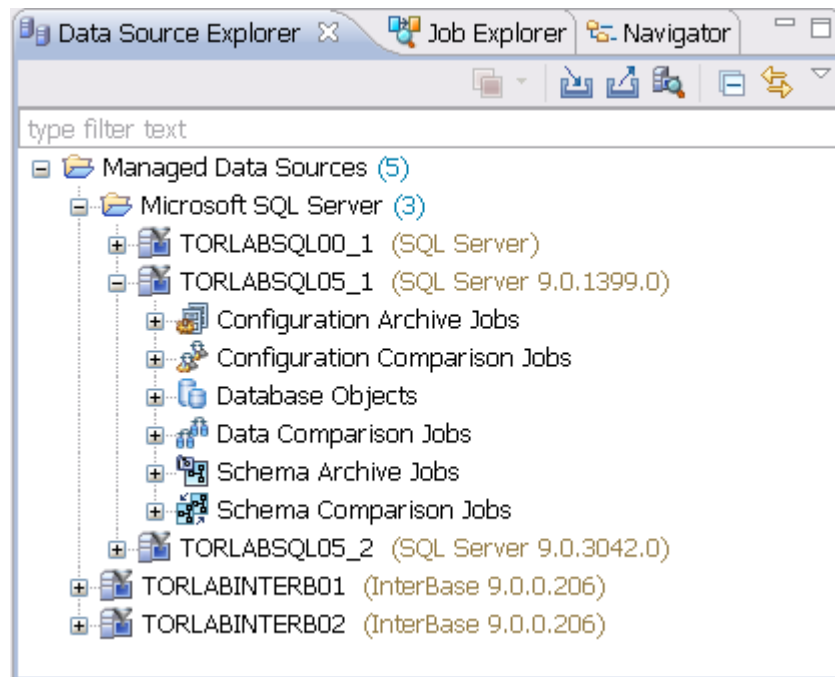
VIEWING CHANGE HISTORY

Each time a job runs, its the history is recorded and listed in the **Job Explorer**. For more information, see [Job Explorer](#).

You can see the history of a specific job in the its job editor **History** tab.

DATA SOURCE EXPLORER

Data Source Explorer lists registered data sources and the comparison jobs and archives associated with them.



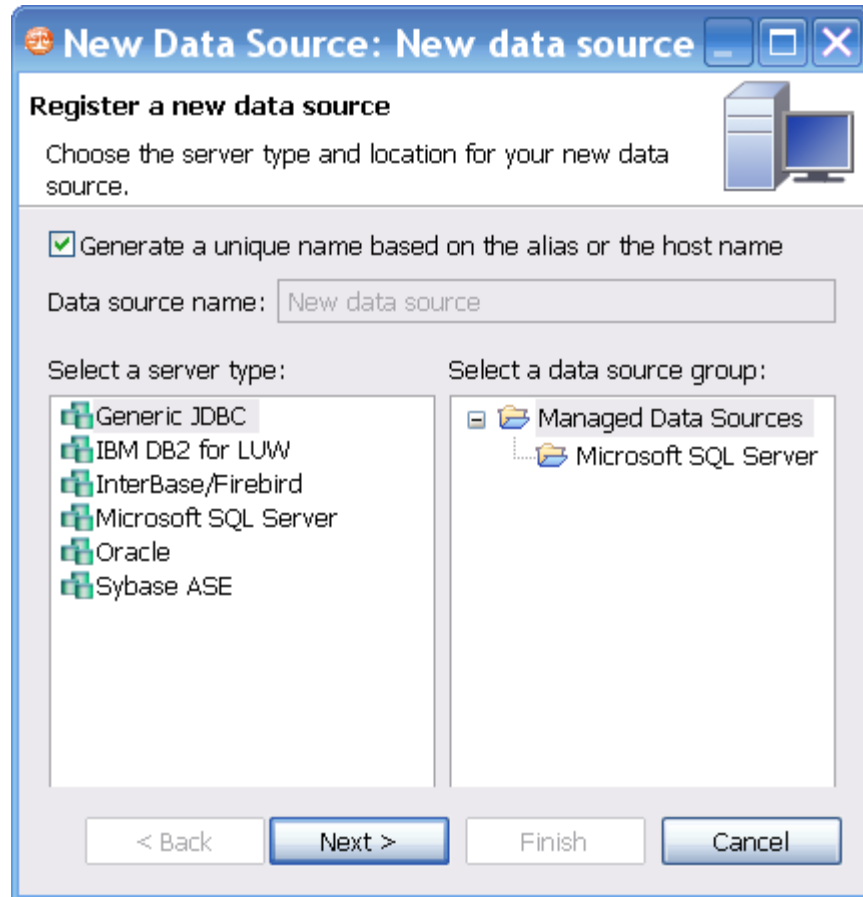
When you register a new data source, you specify its connection information. Once registered, it remains stored in a catalog and does not need to be re-registered. It can be archived and used in multiple jobs.

You can organize data sources based on working sets. A working set is a user-defined collection of data sources. Click the menu icon in the **Data Source Explorer** toolbar and choose **Select Working Set**. In the **Select Working Set** dialog, click **New** to define one.

When you select a working set, **Data Source Explorer** only shows those data sources you specified for the set. To change back to the default view, click the menu icon and choose **Deselect Working Set**.

To add a data source

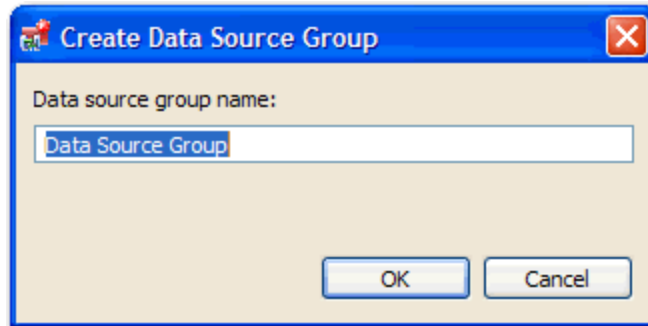
- 1 Right-click on the **Managed Data Sources** node in **Data Source Explorer** and select **New > Data Source**. The **Add Data Source** dialog appears.



- 2 Follow the steps in the **Data Source Wizard** by selecting the data source platform and entering the appropriate connectivity parameters in the fields provided. Click **Finish** when you are finished. The new data source appears in **Data Source Explorer**.

To create Data Source Group folders and organize Data Source Explorer objects

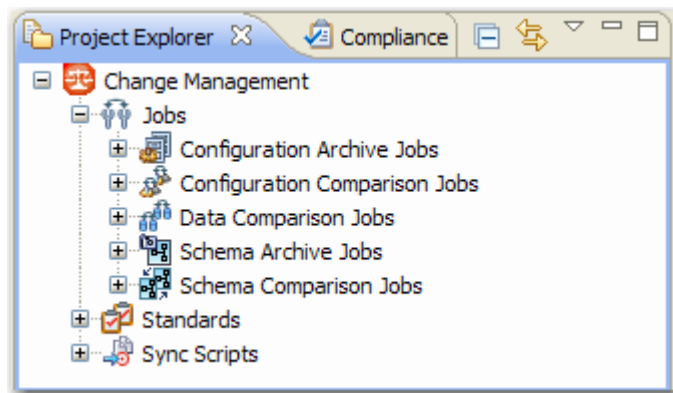
- 1 Right-click on the **Managed Data Sources** node in **Data Source Explorer** and select **New > Data Source Group**. The **Create Data Source Group** dialog appears.



- 2 Enter a meaningful name for the new group in the **Name** field and choose a parent folder, if any.
- 3 Click **Finish**. The **Data Source Group** folder is added to **Data Source Explorer**. You can drag and drop data sources in the folder or add new data sources to the folder.

PROJECT EXPLORER

Project Explorer provides a list that organizes jobs, standards, and synchronization scripts into project folders.



The list is organized by **Jobs**, **Standards**, and **Sync Scripts**. Double-click on the objects to open them in their editors.

A group of administrators can share a set of project files by connecting it to a source code control system and using the **Navigator**.

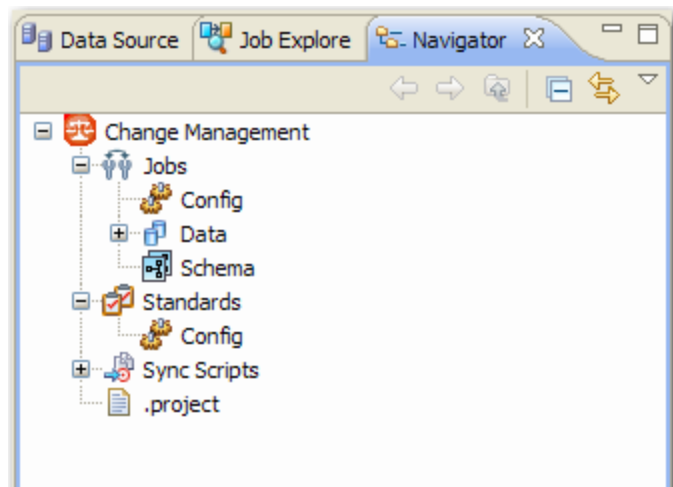
You can use the **File > New > Project** command in the Main menu to create project folders to organize and manage your jobs. When you create a new project, a folder is added to **Project Explorer** with the name that you specified. You can add or delete files under this folder by dragging and dropping them from the **Change Management** branch.

To create a new project

- 1 Select **File > New > Project**. The **New Project** wizard appears.
- 2 Expand the **General Node** and select **Project**.
- 3 Click **Next**.
- 4 Type a project name into the field provided. The Location field indicates where the project files are stored on your machine. By default, all project data is stored in the location where you installed DB Change Manager. If you want to change this location, deselect the **Use Default Location** check box and define a new directory location.
- 5 Click **Finish**. The new project folder appears in **Project Explorer**.

NAVIGATOR VIEW

The **Navigator** view lists all files in each of your projects. You use the **Navigator** to share files with a source control system for jobs, results, standards, and scripts.



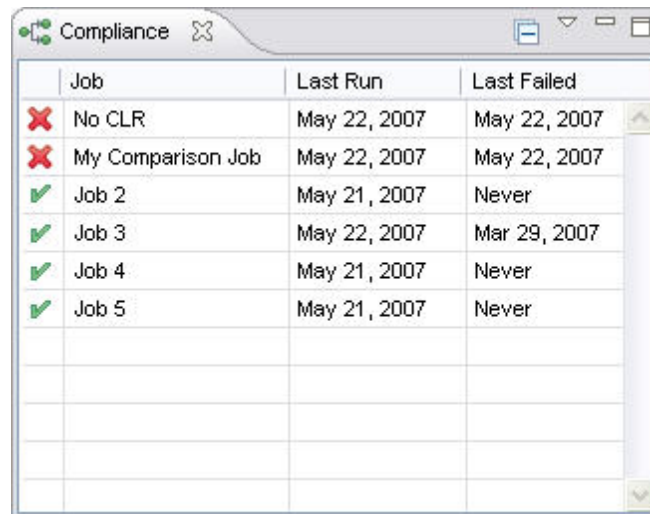
Whenever a job, standard, or synchronization script is generated by the system, its output is saved to a file that is stored in a directory and displayed in the **Navigator**. These files can then be shared in a team environment using a source control system.

All jobs, synchronization scripts, and archive versions are stored in XML format, so that objects can be merged and differentiated between multiple versions of the same object.

NOTE: Before you can use source code control, you need to install an Eclipse third-party plugin and the appropriate client tools.

COMPLIANCE EXPLORER

The **Compliance Explorer** provides you a quick way to see comparison jobs that may have failed compliance checks.



	Job	Last Run	Last Failed
✘	No CLR	May 22, 2007	May 22, 2007
✘	My Comparison Job	May 22, 2007	May 22, 2007
✔	Job 2	May 21, 2007	Never
✔	Job 3	May 22, 2007	Mar 29, 2007
✔	Job 4	May 21, 2007	Never
✔	Job 5	May 21, 2007	Never

The **Compliance Explorer** lists comparison jobs that have been run and specifies:

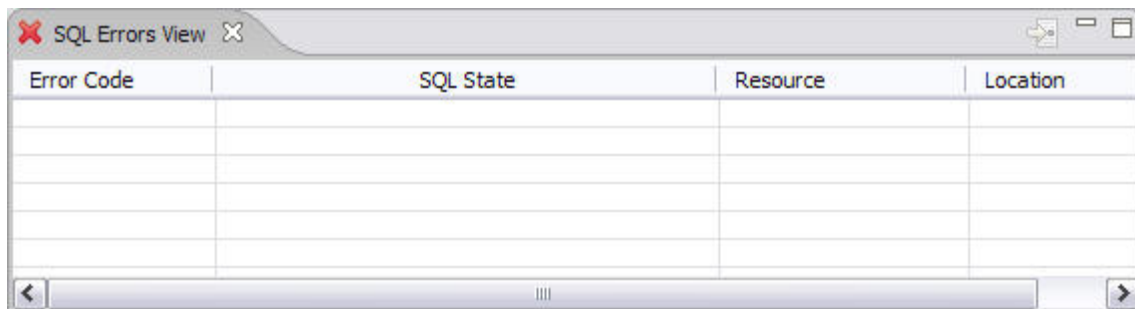
- The day the job was last run
- The day the job last failed, if applicable

By default, jobs are not listed in the **Compliance Explorer**. To list a job here, select the **Track Results in Compliance Editor** check box on the **Overview** tab of the specific job.

SQL ERRORS VIEW

When comparison jobs run, you can optionally generate a SQL script that can synchronization the data repositories of the data sources. A synchronization script can be saved and retrieved at any time, and executed within the application.

The **SQL Errors** view logs any errors that may occur when executing SQL scripts.

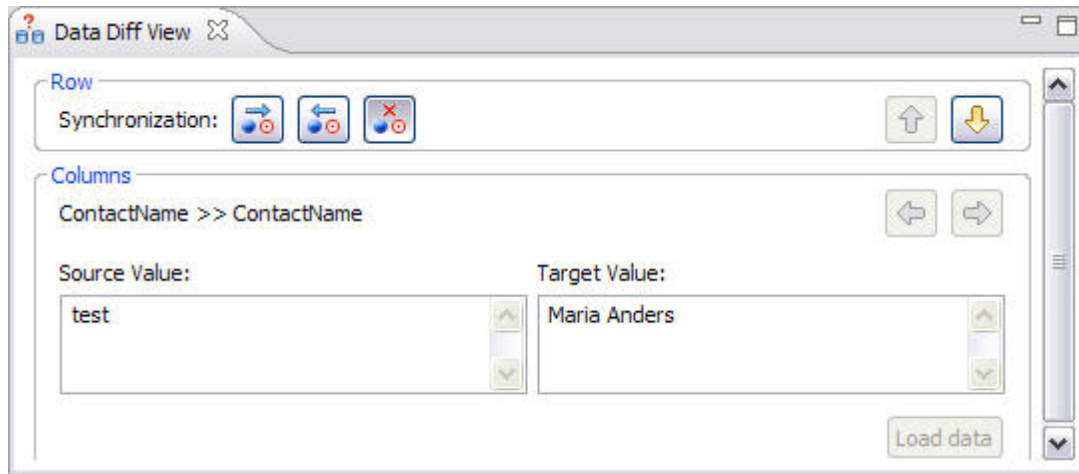


Error Code	SQL State	Resource	Location

SQL Errors are listed by **SQL Error Code**, **SQL State**, **Resource**, and **Location**.

DATA DIFF VIEW

The **Data Diff** view displays the values of a mismatched row pairs when viewing the results of a data comparison job. The view contains row synchronization icons and the specific data values of a selected pair.



You can specify how data is resolved in the **Data Diff** view by selecting one of the appropriate icons beside the **Synchronization** prompt. By row, you can choose to **Make Target Match Source for Selected Row**, **Make Source Match Target for Selected Row**, or **Do Not Resolve Selected Row**, respectively.

You can add the **Data Diff View** to the Workbench by selecting **Window > Show View > Data Diff View**.

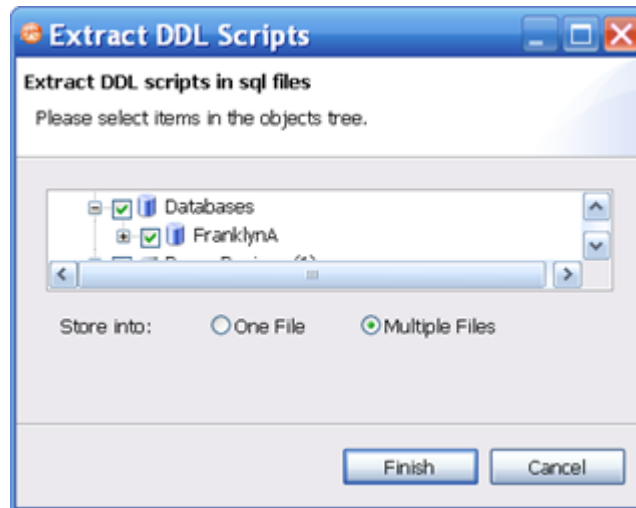
This view lets you specify changes without resolving the repositories of both data sources with synchronization scripts.

EXTRACT DDL SCRIPTS

To extract the DDL of the archive contents to a file

- 1 Open the **Version** tab of the archive.
- 2 In the **Archive Contents** list, right-click on the object and select **Extract DLL Scripts**.

The **Extract DLL Scripts** window opens.



- 3 Select the object for which you want to save scripts.
- 4 Select **One File** to save the DDL for all selected objects to one file.
Select **Multiple Files** to save the DDL for all selected objects into single files for each object.

SQL LOG

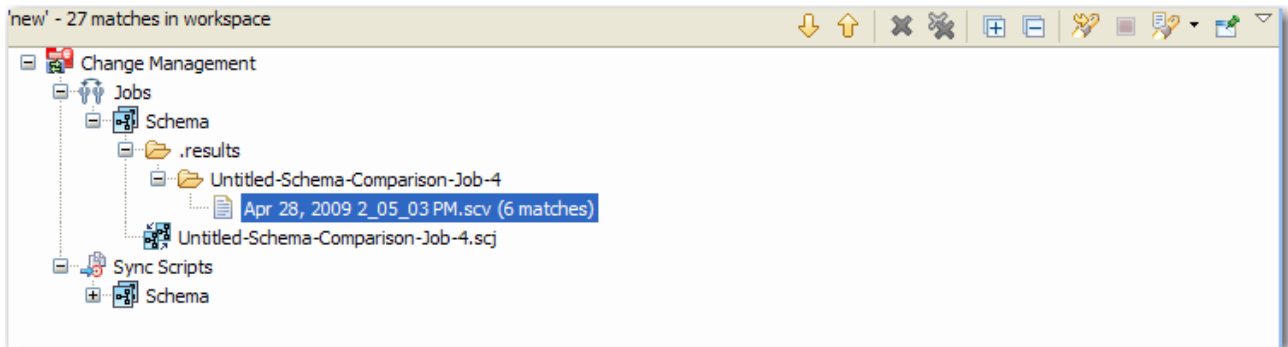
The **SQL Log** captures all SQL commands executed by **SQL Editor** and the system. **SQL Log** entries are listed by **SQL Statement** name, **Date** issued, **Host/Server**, **Service**, **User**, **Source**, and the **Time** (in milliseconds) it took to execute the command.

SQL ERRORS LOG

The **SQL Errors** log automatically logs all SQL errors encountered when SQL commands are executed through DB Change Manager. Errors are listed by **Error Code**, **SQL State**, error **Details**, **Resource**, and the **Location** of the error in the SQL file.

SEARCH VIEW

The **Search** view displays result values when you issue the **Search** command from the **Main Menu**.



The returned results are organized by the job type where the file resides, or by synchronization script, depending on the scope you specified. Navigate to the file that you want to open and double-click it. The file will open in a separate editor window.

In addition, the **Search** view toolbar provides shortcuts to preferences, a filter option when viewing results, and history commands to return or review old search results.

For more information on searching for files, see [Searching for Database Objects and Synchronization Scripts](#).

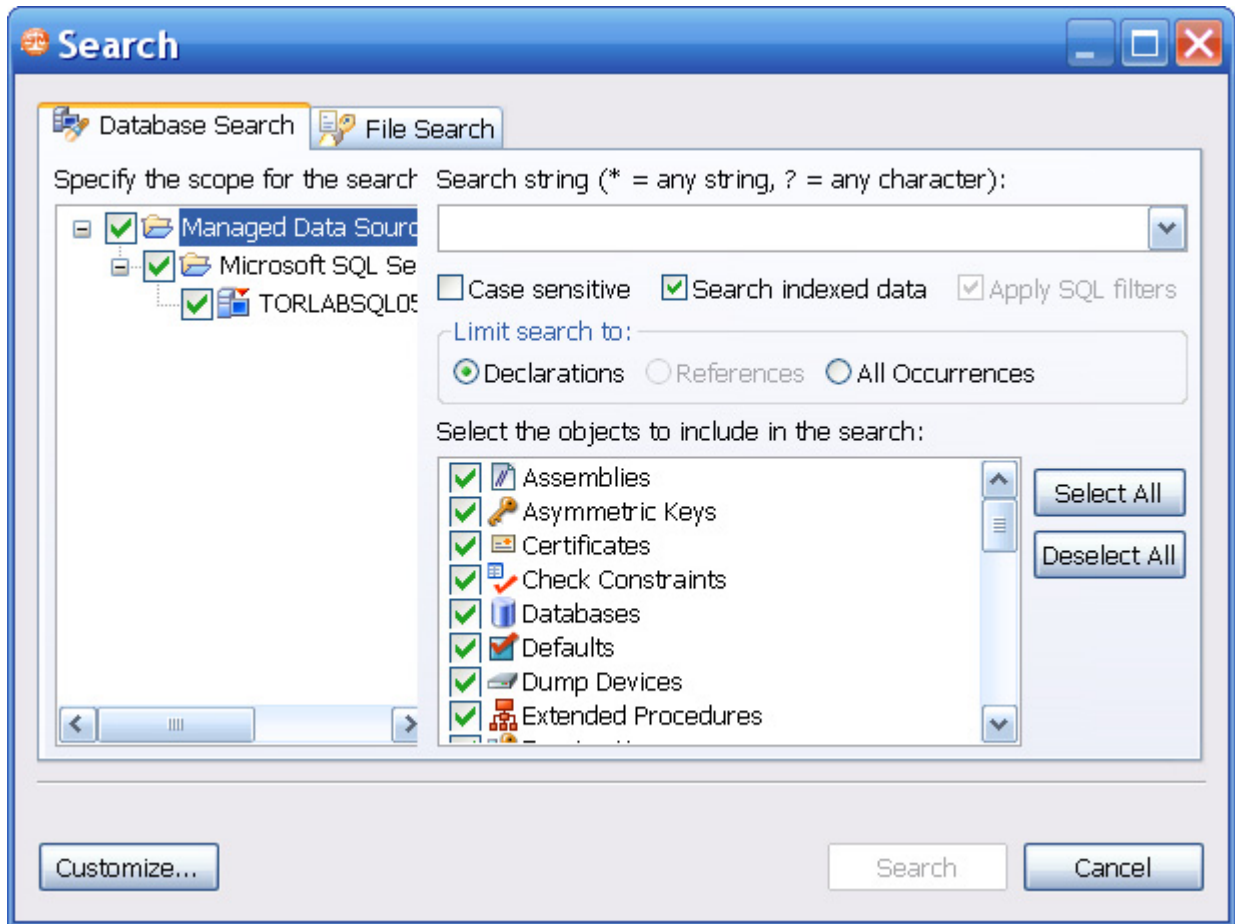
SEARCHING FOR DATABASE OBJECTS AND SYNCHRONIZATION SCRIPTS

The **Search** function provides a means of searching for database objects you have registered, and synchronization scripts that have been generated by DB Change Manager.

To open the **Search** dialog, click the Search icon in the Toolbar, or select **Search > Search** from the **Main Menu**.

The **Search** dialog contains two tabs:

- **Database Search**
- **File Search**



To search for databases

- 1 Navigate to the **Database Search** tab.
- 2 Use the parameter fields provided to search and locate a database:
 - In the box on the left-hand side of the window, use the check boxes to select the **Data Source Groups** you want to search in.
 - In the **Search String** field, type a value to search. Use the * character to indicate wildcard string values, and the ? character to indicate wildcard character values. The \ character indicates an escape character for literals. (* ? and /).
 - Select **Case Sensitive** to take case into account when searching for string matches.
 - Select **Search Indexed Data** to include any information that has been retained by DB Change Manager's indexing function.
 - Select **Apply SQL Filters** to apply any filters you may have associated with the registered data sources in the system.
 - Specify if the search is limited to **Declarations, References, or All Occurrences** in the **Limit Search to** section.
 - In the box in the lower right-hand corner of the screen, select and de-select the object types you want to include in the search. Only selected objects that match your search criteria will be returned.
- 3 Click **Search**. DB Change Manager returns a list of all possible script selections based on the variables you entered.

To search for synchronization scripts

- 1 Navigate to the **File Search** tab.
- 2 Use the parameter fields provided to search and locate a synchronization script:
 - Type a value to search in the **Containing Text** field. Use the * character to indicate wildcard string values, the ? character to indicate wildcard character values, and the \ character to indicate an escape character for literals. (* ? and /).
 - Select **Case Sensitive** to take case into account when searching for string matches.
 - Select **Regular Expression** to identify the string as a regular function in terms of search criteria.
 - In the **File Name Pattern** field, specify the extension name of the files to search for explicitly. If the value in this field is a * character, the search function will search all files regardless of extension. Manually type the extensions to indicate the file type (separate multiple file types with commas), or click **Choose** and use the **Select Types** dialog to select the file extensions the process will use to search for the string. For example, selecting *.caj will search for configuration archive job files, exclusively.
 - Select **Consider Derived Resources** to include derived resources in the search.
- 3 Click **Search**. DB Change Manager returns a list of all possible script selections based on the variables you entered.

EDITORS

Editors are Workbench interface components that enable DB Change Manager jobs.

- [Data Comparison Job Editor](#)
- [Configuration Comparison Job Editor](#)
- [Configuration Standard Editor](#)
- [Configuration Archive Job Editor](#)
- [Schema Comparison Job Editor](#)
- [Schema Archive Job Editor](#)
- [SQL Editor](#)

The following actions can be performed on all editors:

To move an editor

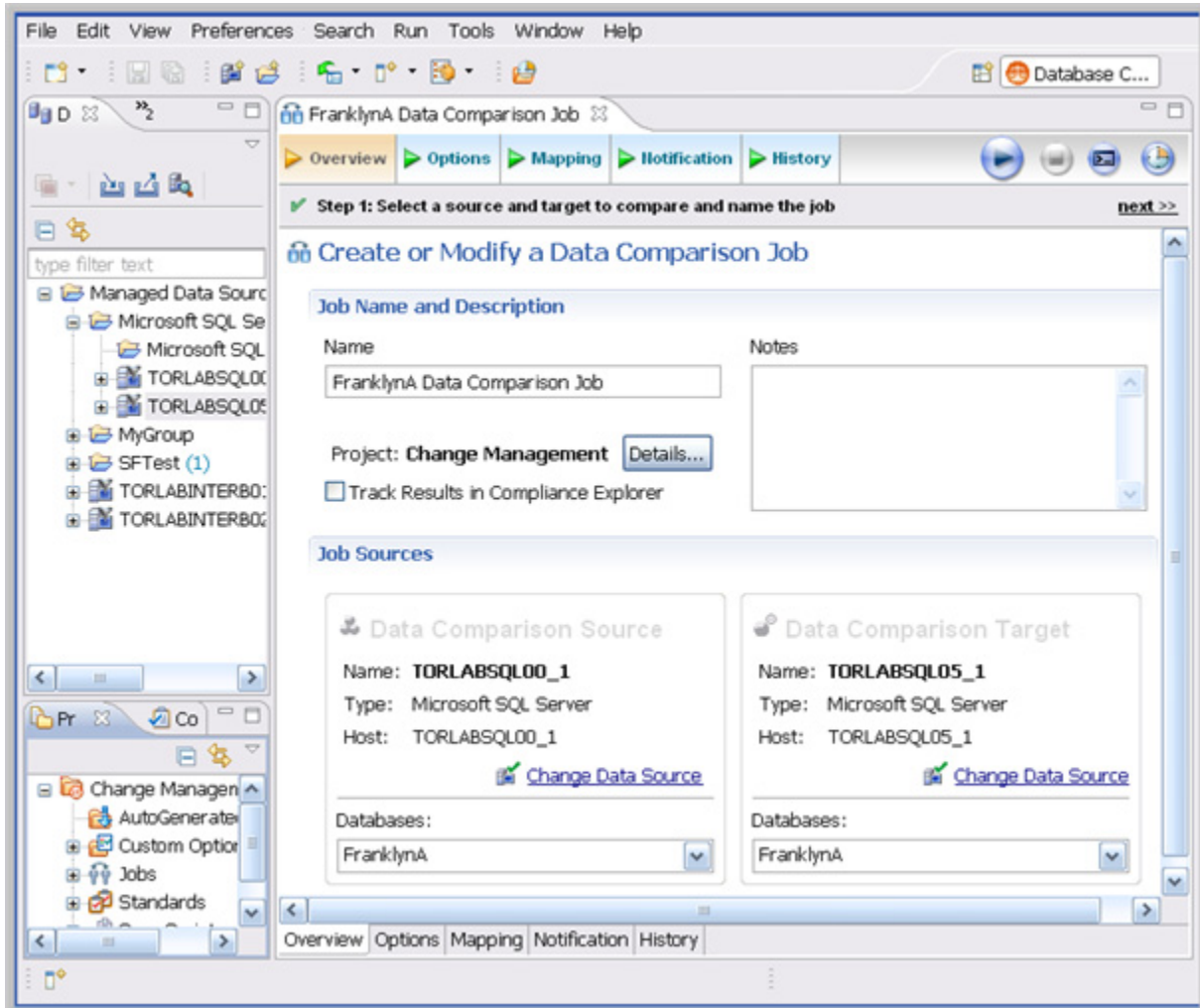
- Click the tab of the editor and drag your mouse. A frame indicates where the editor will fit in the Workbench. Release the button to move the editor to its new location.

To build a tabbed frame

- Drag the editor over another editor and release the mouse button when the frames meet. Both editors overlap and create a tabbed frame. Multiple editors can be combined in this format.

DATA COMPARISON JOB EDITOR

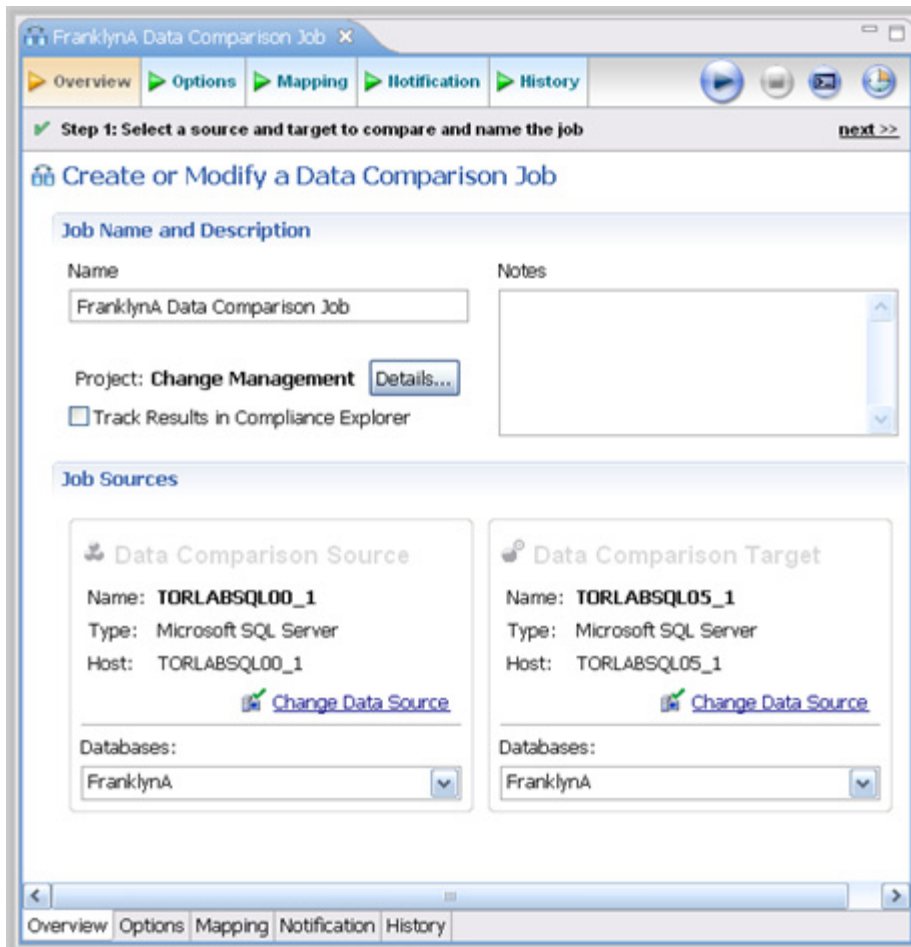
The **Data Comparison Job** editor compares the data repositories of two specified data sources at the table and row levels. In addition, the editor lets you specify table mappings and data matching functions that synchronize it at the global, table, and row level. It also provides high-level mapping refinement options for a job, specific to the data source.



For more information on comparing the data repositories of two specified data sources, see [Comparing Data](#).

CONFIGURATION COMPARISON JOB EDITOR

The **Configuration Comparison Job Editor** compares the configuration settings of two specified data sources. Additionally, the editor provides the ability to define standard settings templates and archive individual data source properties to be used in comparison jobs.

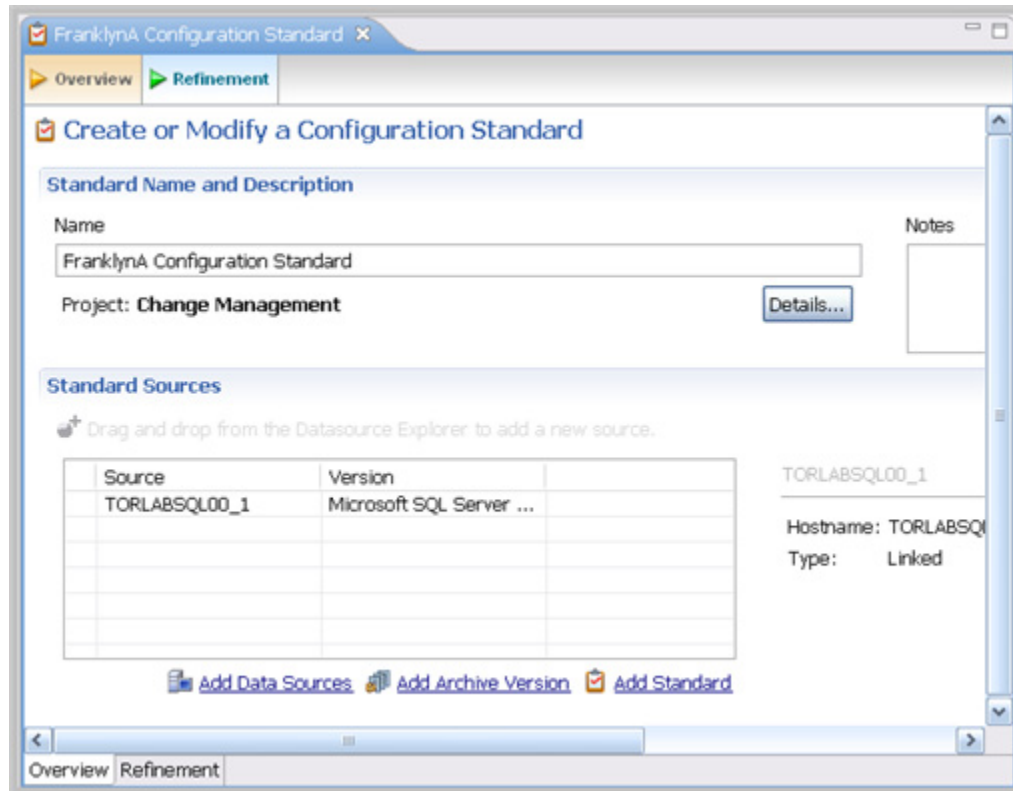


For more information on comparing the configuration settings of two specified data sources, see [Comparing Database Configuration Settings](#).

CONFIGURATION STANDARD EDITOR

The **Configuration Standard** editor defines a configuration standard. A configuration standard is a set of customized data source configuration properties. Once defined, standards are used in configuration comparison jobs against data sources and archives to compare against a specific set of values and threshold operators. You use standards in configuration comparison jobs to ensure that configuration values fall within acceptable ranges.

If you want to add the properties of an existing data source or archive to a new standard, click and drag the appropriate object from **Data Source Explorer** to the **Standard Sources** table. All of the properties of the selected data source or archive populate the new standard.

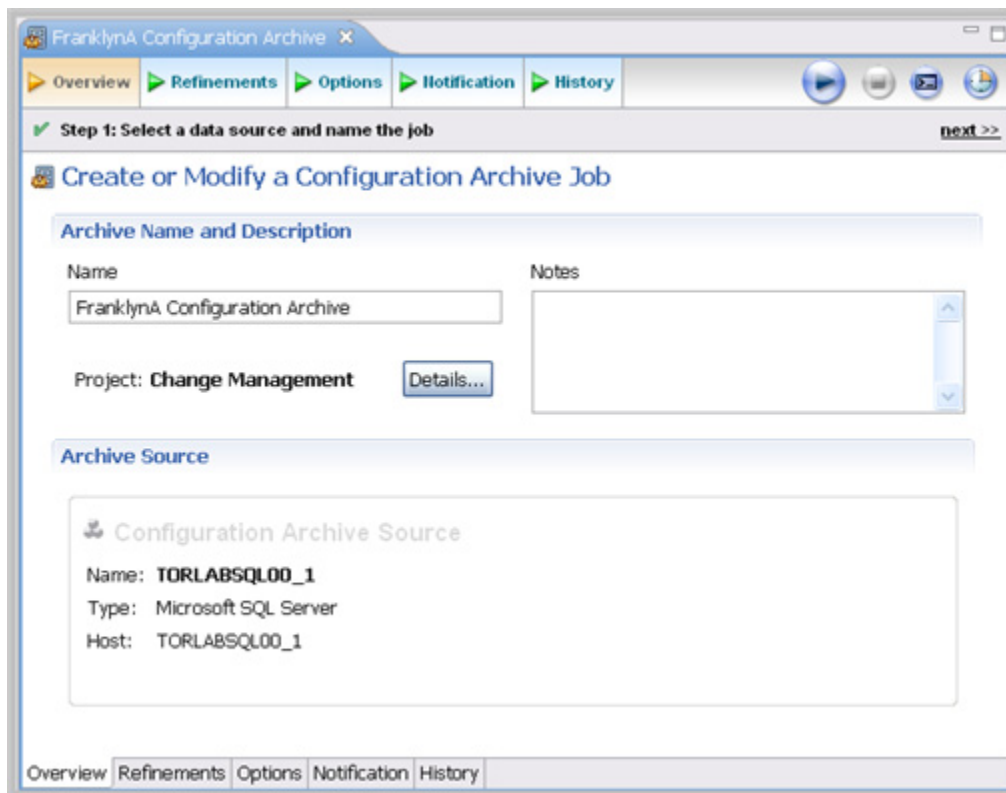


For more information on creating configuration standards, see [Creating a Standard for Configuration Settings](#).

CONFIGURATION ARCHIVE JOB EDITOR

The **Configuration Archive Job** editor provides the ability to capture a snapshot of existing data source configuration values. You can version the same archive multiple times, so you can store and use archive configuration values from different points in time.

Configuration archives are snapshots of existing data source configuration values. Each data source in your environment can contain multiple archive versions. These versions can be used in configuration comparison jobs against other data sources, archives, or standards.



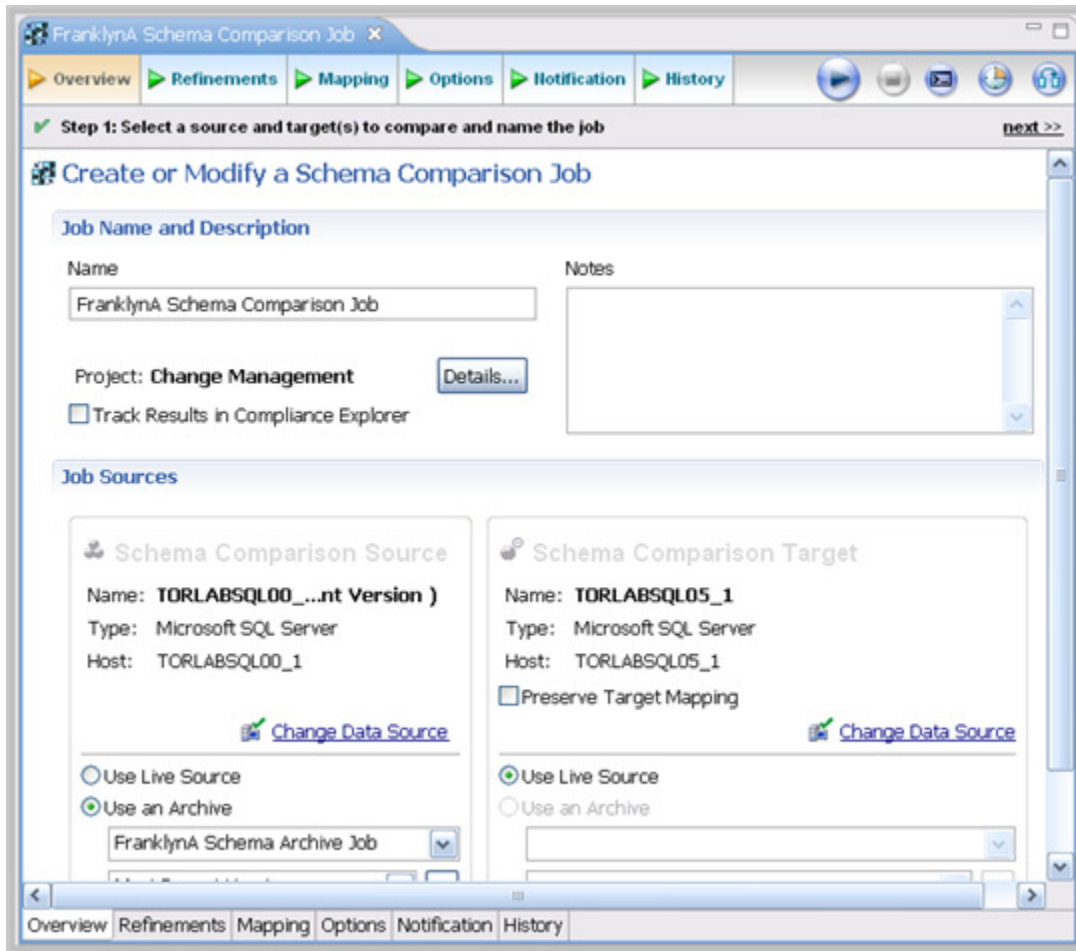
For more information on creating configuration archives for data sources, see [Creating a Configuration Archive](#).

SCHEMA COMPARISON JOB EDITOR

The **Schema Comparison Job** editor compares two schemas on registered data sources, to identify differences between them. You can compare the schemas of two live data sources, a live data source and an archive, or two archives. For example, in order to ensure that approved changes are rolled forward between testing and production environments, you could first archive a baseline schema and use it to compare with the live environment once the move is completed.

DB Change Manager can support cross-platform schema comparisons, but such jobs are limited to the following DBMS combinations and data source objects:

DBMS Source/Target	Objects Compared
Sybase/SQL Server	Tables, Unique Keys, Check Constraints, Foreign Keys, Indexes, Primary Keys, Views
Oracle/SQL Server	Tables, Unique Keys, Check Constraints, Foreign Keys, Indexes, Primary Keys, Views

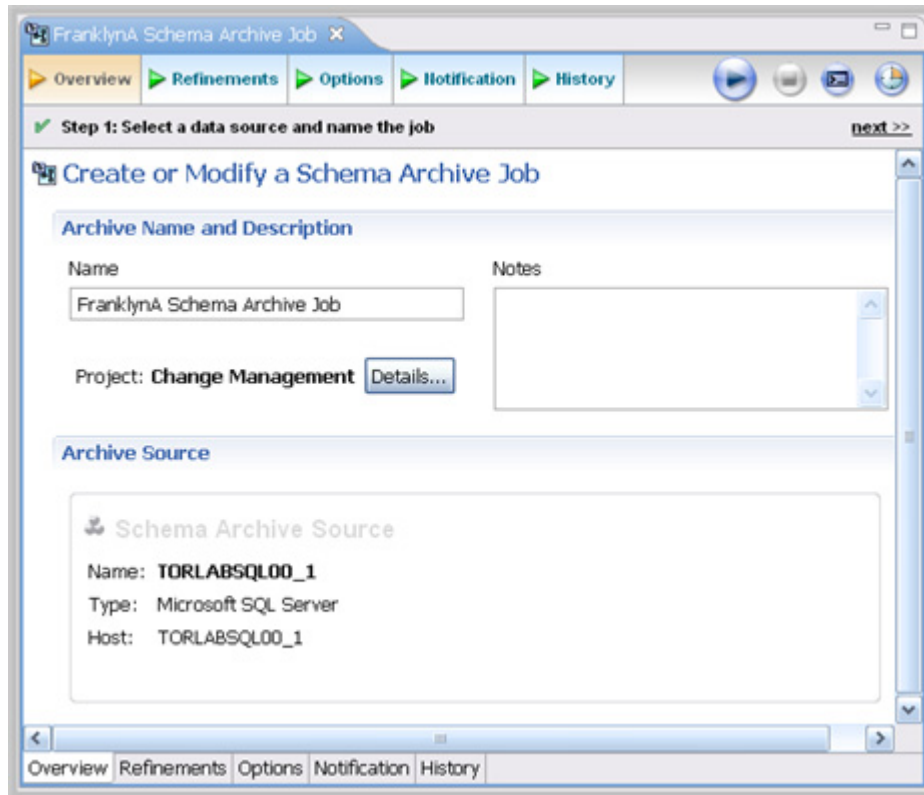


For more information on comparing the schemas of two data sources, see [Comparing Schemas](#).

SCHEMA ARCHIVE JOB EDITOR

The Schema Archive Editor provides the ability to capture a 'snapshot' of an existing schema archive. The interface provides the ability to version the same archive multiple times, so users can store and utilize archived schemas from different points in time.

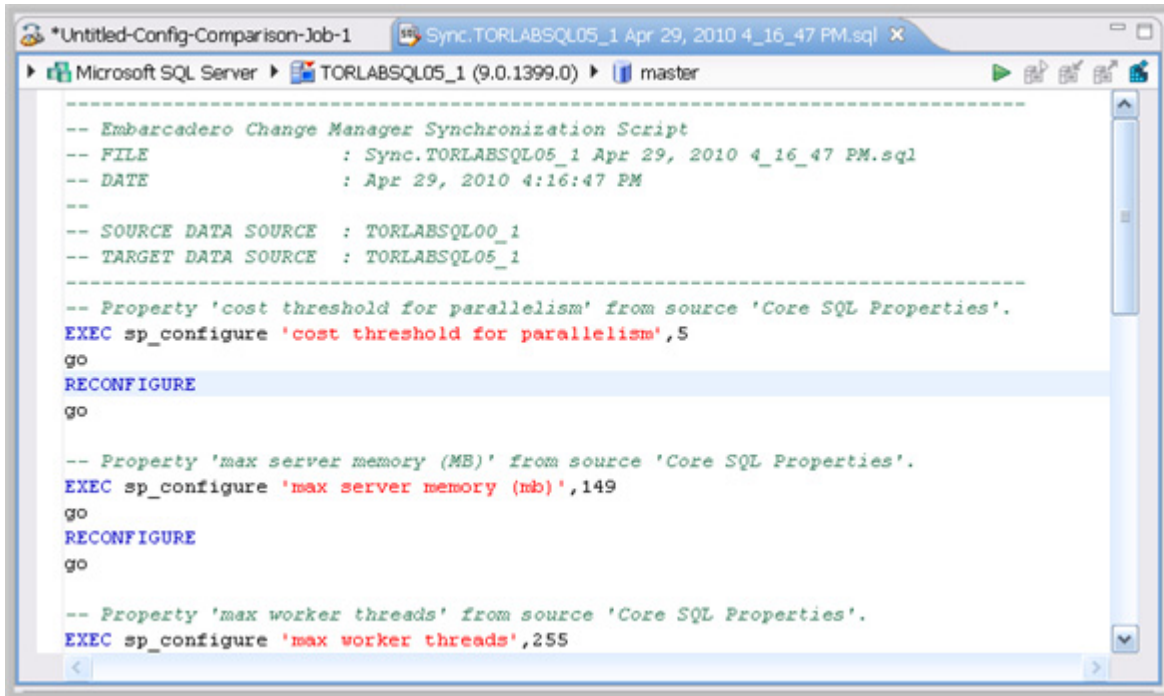
Schema archives are snapshots of existing schema values. Each data source in your environment can contain multiple archive versions. These versions can be used as baselines in schema comparison jobs against other data sources, archives, or standards.



For more information on creating [Creating a Schema Archive](#).

SQL EDITOR

In the **SQL Editor** you can view and modify job scripts.



When DB Change Manager initially generates a script, it appears in **SQL Editor**. By default, new scripts are automatically saved and appear in the **Navigator** and **Project Explorer** in a default directory specified by DB Change Manager. You can edit the SQL scripts and save your changes.

When an SQL file is open in the Workspace, select it and choose a database and an associated catalog on which you want to execute file using the drop down menus in **SQL Editor**. You can execute the code on the specified database and catalog, start a transaction or commit a transaction, or modify the SQL session options prior to execution.

To run SQL code

- Open the SQL file you want to run, ensure it is associated with the correct database
- Click the **Execute** icon. DB Change Manager executes the code on the data source you specified. Results are displayed in the same tab or in a new tab.

To execute a transaction

- Open the transaction file you want to run and ensure it is associated with the correct database, and then click the **Start Transaction** icon. DB Change Manager executes the transaction on the data source you specified.

To commit a transaction

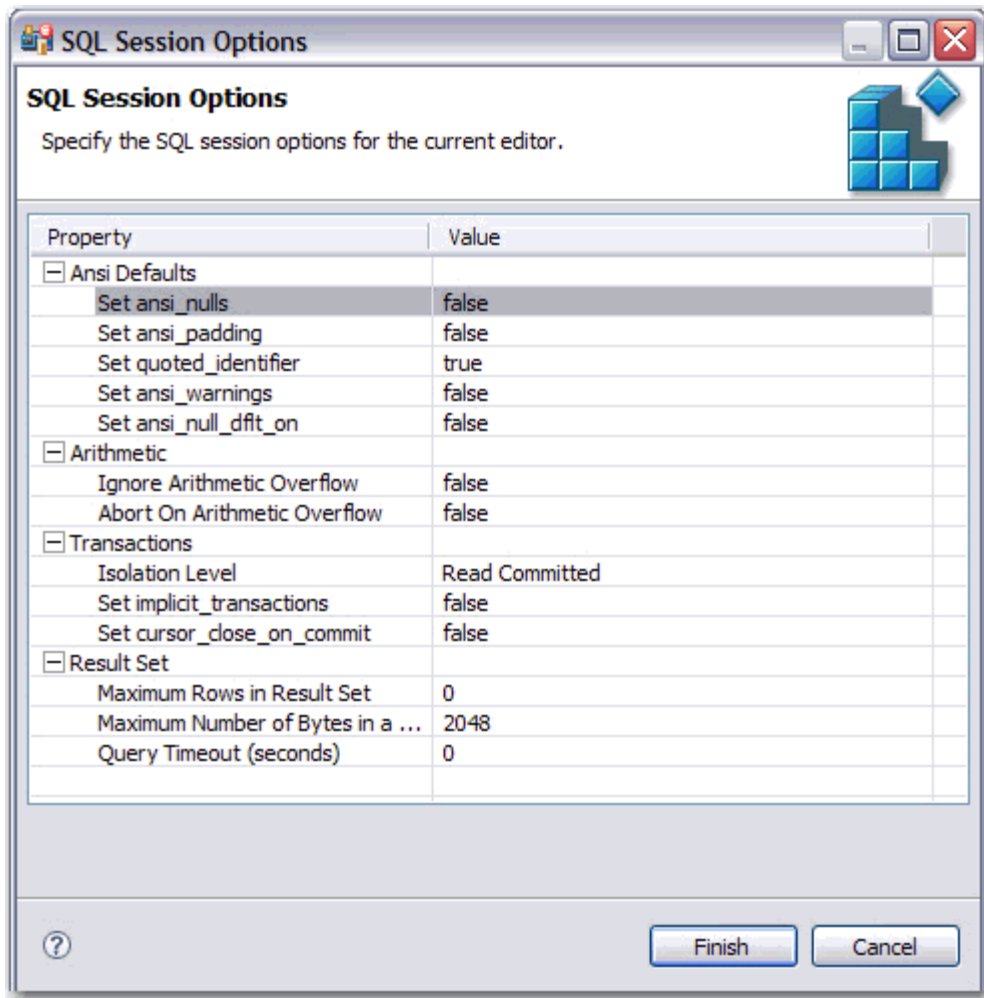
- Open the transaction file you want to commit, ensure it is associated with the correct database, and then click the Commit Transaction icon. DB Change Manager commits the transaction on the data source you specified.

You can set transactions to auto-commit prior to execution on the **SQL Execution** node of the **Preferences** panel.

NOTE: The synchronization script generation process does not support all object parameters. The following object parameters should be manually added after a synchronization script is generated, as required: fillfactor, consumers, allow_dup_row, and statistics.

SQL SESSION OPTIONS

The **SQL Session Options** dialog provides configuration parameters for running code in the development environment.

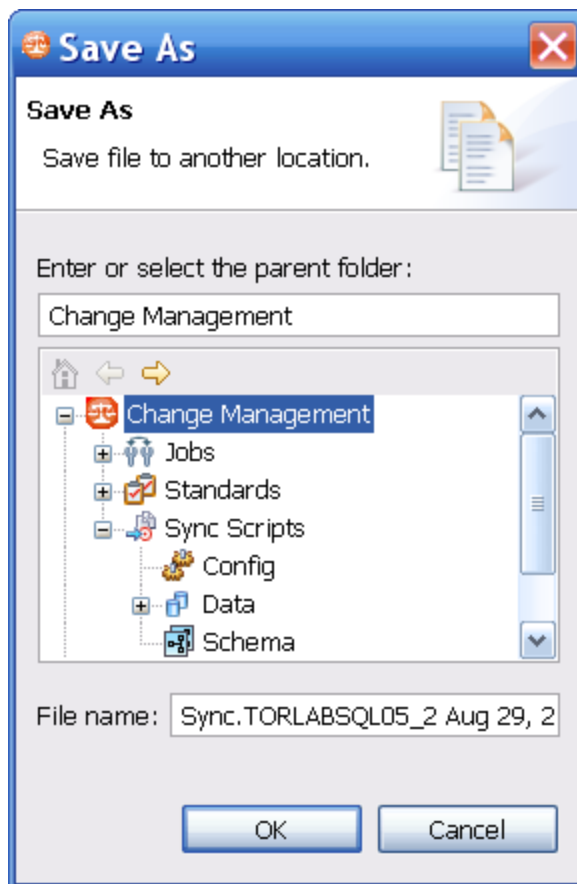


To modify SQL session options

- 1 On the Toolbar, click the **SQL Session Options** icon.
- 2 In the **SQL Session Options** dialog, click on individual parameters in the **Value** column to change the configuration of each property, as specified.
- 3 Click **Finish**.

The session options will be changed and DB Change Manager will execute the code as specified when you launch the script.

NOTE: Session options only apply to the corresponding editor and are not retained when executing multiple SQL files.



You can rerun scripts that are saved in DB Change Manager by double-clicking the script name in the **Navigator** view.

ECLIPSE PLUGIN EXTENSIONS REFERENCE

DB Change Manager publishes extensions that can be implemented as plugins to customize the application.

The **Notifications extension** helps you code a plugin for DB Change Manager that implements any type of job execution notification required for your organization. DB Change Manager includes email notification by default, and you can code additional Java plugins to add to DB Change Manager.

The **Configuration Parameters extension** helps you code a plugin for DB Change Manager that adds additional configuration parameters and values for configuration standards, archive jobs, and comparisons. DB Change Manager includes plugins for standard configuration properties and version information, and you can code additional plugins that query data sources using Java JDBC for additional or environment-specific information.

To write Eclipse plugins, you need to have some experience coding in Java.

For instructions, see:

- [Creating and Implementing a Notification Plugin](#)
- [Creating and Implementing a Custom Configuration Plugin](#)

CREATING AND IMPLEMENTING A NOTIFICATION PLUGIN

Notifications can be set to alert you if a job is successful, unsuccessful, or either result. Additionally, a notification can also indicate if a job was launched from the Workbench or from a command line, and can produce a report in a number of different file formats.

You can select the following options and option parameters using the job notification plugin:

Notification Option	Option Parameters
Notify When Run From	Workbench, Command Line, Both
Attached Report	None, CSV, XML, HTML, PDF, RTF
Send Notification	Always, Only on Success, Only on Failure

Notification parameters are set independently on each job configured in DB Change Manager. Additionally, each set of notification options apply individually per job selected.

The notification plugin must be defined in Eclipse and then activated in DB Change Manager.

The following tasks provide a high-level overview of defining and implementing the notification plugin:

- 1 [Creating a Plugin Project](#)
- 2 [Configuring the Target Platform](#)

- 3 [Defining plugin Dependencies](#)
- 4 [Implementing a Notifier Class](#)
- 5 [Implementing a Viewer Contribution](#)
- 6 [Specifying Setup Constants](#)
- 7 [Implementing the DirectoryNotifier Class](#)
- 8 [Implementing the DirectoryNotifierViewerContribution Class](#)
- 9 [Registering the Extensions](#)
- 10 [Deploying the Notifier Plugin](#)

CREATING A PLUGIN PROJECT

In order to start building a plugin, you need to create a new plugin platform in Eclipse.

To create a plugin project

- 1 Select **File > New > Other**.
The **Select a Wizard** dialog appears.
- 2 Select **Plugin Project** and click **Next**.
- 3 Type a name for the plugin in the appropriate field, and leave the remainder of the parameters as they appear. Click **Next**.
- 4 Retain the default parameters on the next dialog and click **Finish**.

The new plugin project is created in Eclipse and is ready for plugin development.

CONFIGURING THE TARGET PLATFORM

The target platform for the plugin development process needs to be identified in DB Change Manager. This will indicate to the new plugin for what product it extends, and will grant the plugin access to the system.

- 1 Select **Window > Preferences**.
The **Preferences** dialog appears.
- 2 Choose the **Plugin Development > Target Platform** node and click **Browse**.
- 3 Navigate to the install directory for DB Change Manager.
- 4 Click **Apply** and **OK**.

The **Preferences** dialog closes and the target platform of the plugin is now indicated.

DEFINING PLUGIN DEPENDENCIES

The new plugin requires a pair of dependency definitions on the DB Change Manager Notification plugin:

- `com.embarcadero.change.notifications`
- `org.eclipse.ui.forms`

- 1 Navigate to the `META-INF` folder in your project and double-click `MANIFEST.MF`.

The `MANIFEST.MF` file opens in Eclipse.

- 2 Select the **Dependencies** tab and click **Add**.

- 3 Choose `com.embarcadero.change.notifications` and click **OK**.

The dependency is added.

- 4 Choose `org.eclipse.ui.forms` and click **OK**.

The dependency is added.

- 5 Press `Ctrl+S` to save the changes, or choose **File>Save** from the menu to retain the new dependencies and close the editor.

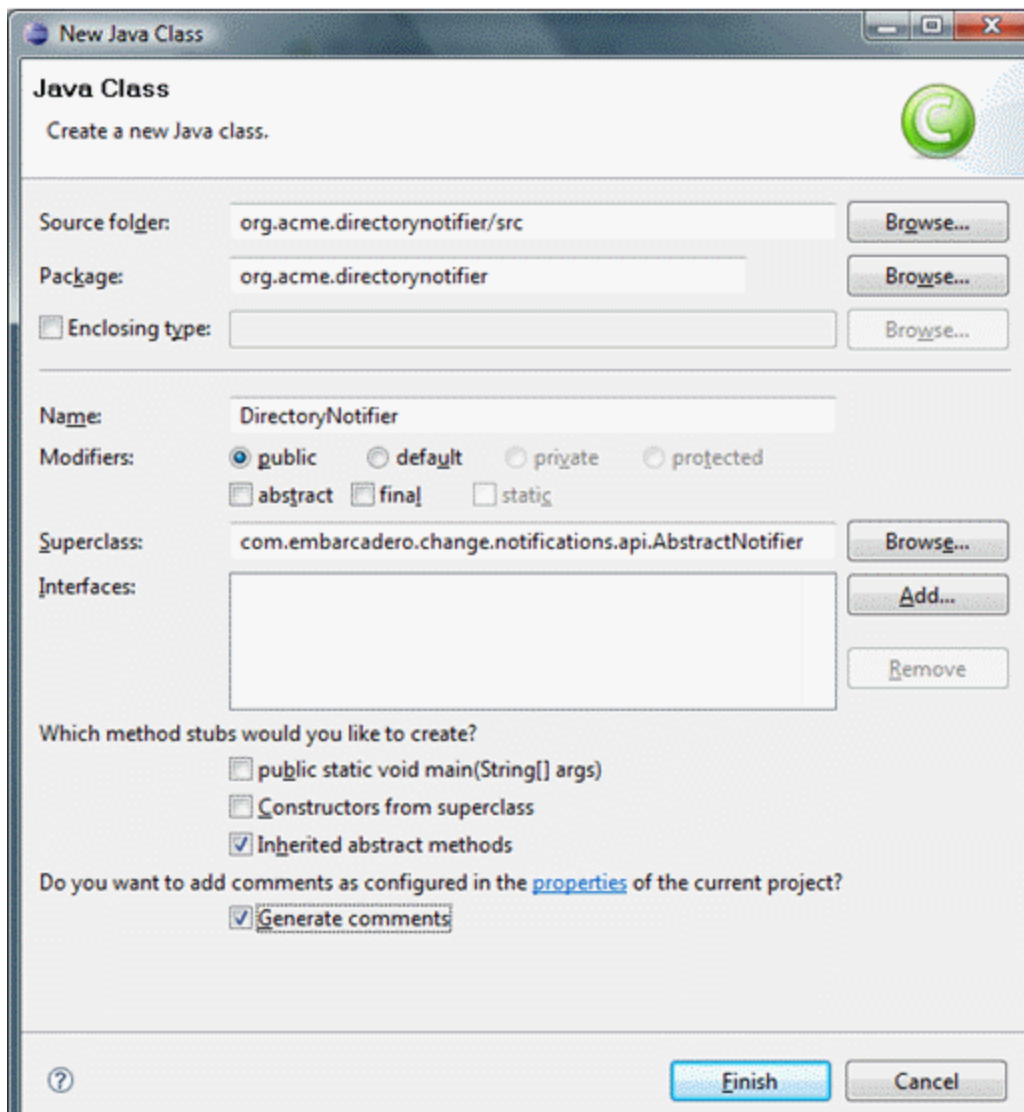
IMPLEMENTING A NOTIFIER CLASS

Each notification plugin requires a `Notifier` class that must implement the interface `com.embarcadero.change.notifications.api.INotifier`. To do this, you subclass `com.embarcadero.change.notifications.api.AbstractNotifier` and customize it.

NOTE: If `AbstractNotifier` or `INotifier` cannot be found in Eclipse, the target platform or plugin dependencies have not been configured properly. Ensure that you have configured these prerequisites and then search again for the notifier classes.

- 1 Right-click on your project and select **New > Class**.

The **New Java Class** dialog appears.



2 Enter the appropriate information in the fields provided to define the new Java class:

- In the **Name** field, type "DirectoryNotifier".
- In the **Package** field, type "org.acme.directorynotifier".
- In the **Superclass** field, type "com.embarcadero.change.notifications.api.AbstractNotifier".

3 Click **Finish** to create the new class.

Once you have defined the `DirectoryNotifier` class, two methods require implementation:

- `isReportSupported` returns a Boolean value indicating if the notification can include a report. The directory notifier will return true, since reports can be placed in a directory, but a notification may not support reports. This method is called when creating job editors to determine if the reporting section will be shown for this type of notification.
- `sendNotification` is called after a job runs and is responsible for the notification. The three parameters are as follows:

Parameter	Type	Description
<code>notifierData</code>	<code>INotifierData</code>	This instance contains the configuration for the notification of the job just run.
<code>jobMetaData</code>	<code>Map<String,String></code>	This map contains information about the job execution. The keys in the map correspond to entries in the <code>NotificationPropertyEnum</code> . For example, to get the data and time of the execution, you would code: " <code>jobMetaData.get(NotificationPropertyEnum.DATE_TIME.getTag())</code> " Note: The keys are the tag of the enum entries, so ensure that <code>getTag()</code> is called when accessing <code>jobMetaData</code> .
<code>notificationInfo</code>	<code>IReportGenerator</code>	Used to generate the report and access any DDL or sync script output.

IMPLEMENTING A VIEWER CONTRIBUTION

Each notifier can be configured for a job in DB Change Manager.

The notifier will enable a user to select a target directory in which to put results. The email notifier that is supplied with the application enables users to select the email template that will be used to generate the message as well as the addresses of those who will receive it.

In order to add a directory field to the job editor, you write a class that implements the interface named `com.embarcadero.change.notifications.api.INotifierViewerContribution`.

- Create a new class named "DirectoryNotifierViewerContribution" and follow the same steps you used to create the `DirectoryNotifier` class. The exception to this process is that you need to leave the `Superclass` parameter set to "java.lang.Object", as well as adding the interface `INotifierViewerContribution` to the list of implemented interfaces.

SPECIFYING SETUP CONSTANTS

The notifier only allows users to select a directory in which reports and outcomes are written whenever jobs are executed.

The `DirectoryNotifierConstants` class is a small constants class that stores the property name for storage and retrieval purposes.

- 1 Create a new class named `DirectoryNotifierConstants`.
- 2 Enter the following code for the class definition:

```
/**
 * Constants for this network share notifier
 */
public final class DirectoryNotifierConstants
{
    // Overview file name
    public static final String OVERVIEW_FILE_NAME = "outcome.txt";

    // Persisted Job Settings Key
    public static final String TARGET_DIRECTORY = "directorynotifier.targetdir";

    /**
     * Constant utility class cannot be instantiated
     */
    private DirectoryNotifierConstants()
    {
    }
}
```

IMPLEMENTING THE DIRECTORYNOTIFIER CLASS

The `DirectoryNotifier` class can be coded with two return methods:

- `isReportSupported`
- `SendNotification`

To implement the `isReportSupported` method

Enter the following code:

```
public boolean isReportSupported()
{
    return true;
}
```


To implement the SendNotification method

- 1 To add the `notifierProperties` and assign the `targetDirectory` to a variable to provide a directory that the user selected for the executed job, add the following code:

```
public void sendNotification ( INotifierData notifierData,
                             Map<String, String> jobMetaData,
                             notificationInfo

    Map<String, String> notifierProperties = notifierData.getNotificationProperties();

    String targetDirectory = notifierProperties.get (
        DirectoryNotifierConstants.TARGET_DIRECTORY );
```

- 2 Enter the following code to define that the report and output file need to be placed in a directory whose name will be composed of the job execution date and time.

NOTE: If a directory cannot be created, the code below also includes a `RuntimeException`.

```
//Create a new directory named by the execution date and time
String dateTime = jobMetaData.get ( NotificationPropertyEnum.DATE_TIME.getTag() );
File dir = new File (targetDirectory +
    File.seperatorChar +
    dateTime.replace ( ':', '-' ) );
boolean createdDirectory = dir.mkdir();
if ( !createdDirectory )
{
    throw new RuntimeException ( "Create Failed: " +
dir.getAbsolutePath () );
}
```

- 3 Enter the following code to define the outcome file, which includes each of the job execution parameters:

```
try
{
    //Write the results to an output file
    File f = new File ( dir.getAbsolutePath() +
        File.separatorChar +
        DirectoryNotifierConstants.OVERVIEW_FILE_NAME );
    f.createNewFile();

    BufferedWriter writer = new BufferedWriter ( new FileWriter ( f ) );

    String newLine = System.getProperty ( "line.separator" );

    writer.write ( "Outcome: " +
        jobMetaData.get (
            NotificationPropertyEnum.JOB_OUTCOME.getTag() ) +
        newLine );
    writer.write ( newLine );
    writer.write ( "Date & TimeL " +
        jobMetaData.get (
            NotificationPropertyEnum.DATE_TIME.getTag() ) +
        newLine );
    writer.write ( "Elapsed Time: " +
        jobMetaData.get (
            NotificationPropertyEnum.ELAPSED_TIME.getTag() ) +
        newLine );
    writer.write ( newLine );
    writer.write ( "Name: " +
        jobMetaData.get (
            NotificationPropertyEnum.JOB_NAME.getTag() ) +
        newLine );
    writer.write (newLine );
    writer.write ( "Host: " +
        jobMetaData.get (
            NotificationPropertyEnum.JOB_HOST.getTag() ) +
        newLine );
    writer.write ( "Job Type: " +
        jobMetaData.get (
            NotificationPropertyEnum.JOB_TYPE.getTag() ) +
        newLine );
    writer.write ( "Module: " +
        jobMetaData.get (
            NotificationPropertyEnum.MODULE.getTag() ) +
        newLine );
    writer.write ( newLine );
    writer.write ( "Sources: " +
        jobMetaData.get (
            NotificationPropertyEnum.JOB_SOURCES.getTag() ) +
        newLine );
    writer.write ( "Targets: " +
        jobMetaData.get (
            NotificationPropertyEnum.JOB_TARGETS.getTag() ) +
        newLine );
    writer.write ( newLine );
    writer.write ( "Job Notes: " +
        jobMetaData.get (
            NotificationPropertyEnum.JOB_NOTES.getTag() ) +
        newLine );

    writer.flush();
    writer.close();
}

catch (Throwable t )
{
    throw new RuntimeException ( t );
}
```

- 4 Enter the following code to generate the report. This code also provides a null report if the report type is set to none at execution time.

```
//Create the report if required
File reportFile = notificationInfo.getReport() ;

if ( reportFile !=null )
{
    File destinationReportFile = new File
        dir.getAbsolutePath() +
        File.separatorChar +
        reportFile.getName() );
    reportFile.renameTo ( destinationReportFile );
}
```

Finally, output any DDL or synchronization SQL:

```
//Write the script
File scriptFile=notificationInfo.getScript();

if ( scriptFile !=null)

    File DestinationScriptFile = new File (
        dir.getAbsolutePath() +
        File.separatorChar +
        scriptFile.getName ( ) );
    scriptFile.renameTo (destinationScriptFile);
```

IMPLEMENTING THE DIRECTORYNOTIFIERVIEWERCONTRIBUTION CLASS

Add the code below to the `DirectoryNotifierViewerContribution` class to provide functionality for the interface:

- 1 The `DirectoryNotifierViewerContribution` class requires a three class-level private attributes. These store notification data, which will be the composite on which any options will then be placed, as well as the `folderField`, which is used to enter a file directory address.

```
private INotifierData notificationData;
private Composite body;
private Text folderField;
```

- 2 The `createFormContent` attribute creates a new composite and provides a layout template for options and entry fields. The code below adds a label and a text field to the interface:

```
public Composite createFormContent ( Composite parent, FormToolkit toolkit )
    body = toolkit.createComposite ( parent );
    body.setLayout ( new GridLayout ( ) );

    toolkit.createLabel ( parent, "Target Directory:" );

    String folder = notificationData == null ? " " :
        notificationData.getNotificationProperties().get (
            DirectoryNotifierConstants.TARGET_DIRECTORY );

    folderField = toolkit.createText ( parent, folder, SWT.SINGLE | SWT.BORDER );
    folderField.setLayoutData ( new GridData ( SWT.FILL, SWT.CENTER, true, false ) );
    folderField.addModifyListener ( new ModifyListener ( ) {
        public void modifyText ( ModifyEvent e )
        {
            notificationData.setNotificationPropert (
                DirectoryNotifierConstants.TARGET_DIRECTORY,
                folderField.getText().trim() );
        }
    } );

return body;
```

The code does the following:

- Creates a new composite body, and sets it to have a grid layout.
- Creates a label for the target directory field.
- Retrieves the folder from the job, if the job is new or unsaved, or defaults to an empty string if the result is null.
- Initializes the `folderField` text element and sets it to `justify`.
- Adds a listener for changes on the `folderField` element that updates the `notificationData`, thus enabling save actions.
- Returns the body to be added to the DB Change Manager job editor to which this class applies.

- 3 Add the `setData` method using the following code, which is called to update the user interface with job information. This method accepts and stores the new instance of `INotifierData` containing job specific data for the notifier. If the `folderField` has already had something entered, the notification properties will be initialized and store the value that was previously entered.

```
public void setData ( INotifierData notificationData )
    this.notificationData = notificationData;

    if ( folderField !=null && !folderField.isDisposed() &&
        folderField.getText().trim().length() > 0 ) )

        if ( notificationData !=null )

            Map<String, String> propMap
                notificationData.getNotificationProperties();

            if ( propMap == null )
            {
                notificationData.setNotificationProperties(
                    new HashMap<String, String>() );

                propMap = notificationData.getNotificationProperties();
            }

            notificationData.setNotificationProperty (
                DirectoryNotifierConstants.TARGET_DIRECTORY,
                folderField.getText() );
        }
    }
```

Use the following code to implement the `setEnabled` method:

```
public void setEnabled ( boolean enabled )
{
    folderField.setEnabled( enabled );
}
```

Use the following code to implement the `dispose` method:

```
public void dispose()
{
    if ( body !=null && !body.isDisposed() )
    {
        body.dispose();
        body = null;
    }
}
```

REGISTERING THE EXTENSIONS

Once you have coded the plugin and implemented the notifier and user interface, you include definitions that explain how DB Change Manager should interact with the plugin.

You register the extensions that define the `Notifier` and `NotifierViewerContribution` to DB Change Manager. When the plugin is added to DB Change Manager, the extensions are registered and DB Change Manager understands the purpose of the plugin.

- 1 Open the `MANIFEST.MF` file.
- 2 Navigate to the **plugin.xml** tab and display the source.
- 3 Add the following code:

```
<extension
  point="com.embarcadero.change.notifications.notifier">

  notifier
    id="org.acme.directorynotifier"
    name="Directory Notifier"
    notifier="org.acme.directorynotifier.DirectoryNotifier"
    viewContribution=
      "org.acme.directorynotifier.DirectoryNotifierViewerContribution">
  </notifier>
</extension>
```

- 4 Save the file by pressing `Ctrl+S` or clicking **File>Save**.

When `org.acme.directorynotifier` is found, the notification extension point will be registered. The `notifier` and `viewContribution` attributes are the class names for two implementations and Eclipse will reflectively register these objects for use in DB Change Manager.

DEPLOYING THE NOTIFIER PLUGIN

Once the plugin has been defined, you need to deploy the plugin in DB Change Manager.

- 1 Stop DB Change Manager if it is currently running.
- 2 Select **File > Export**.
The **Export** dialog appears.
- 3 Choose **Deployable Plugins and Fragments** from the **Plugin Development** group.
- 4 Click **Next**.
- 5 Select `org.acme.directorynotifier` and ensure that the target directory is attribute to the DB Change Manager installation folder.
- 6 Click **Finish**.
- 7 Start DB Change Manager and create a new data comparison job. Define a source and target data source and then navigate to the **Notifications** tab.

The **Directory Notifier** has been added as a section to the tab.

- 8 Choose **Enable Directory Notifier** to enable the feature.
- 9 Specify the **Target Directory**. For example, `C:\ChangeManagerNotifications`, and choose **PDF** in the **Attached Report** section.

- 10 Execute the job and open the target folder. A new folder is created containing two output files from the job.

CREATING AND IMPLEMENTING A CUSTOM CONFIGURATION PLUGIN

Follow the instructions in these sections to code an Eclipse plugin for DB Change Manager that adds additional configuration parameters and values for configuration standards, archive jobs, and comparisons.

- 1 [Creating a Project Plugin](#)
- 2 [Configuring the Target Platform](#)
- 3 [Defining Plugin Dependencies](#)
- 4 [Implementing a Property Source](#)
- 5 [Determining Support](#)
- 6 [Defining Properties](#)
- 7 [Gathering Results](#)
- 8 [Synchronizing the Job](#)
- 9 [Registering Extensions](#)
- 10 [Deploying the Custom Configuration Plugin](#)

CREATING A PROJECT PLUGIN

In order to start building a plugin, you need to create a new plugin platform in Eclipse.

To create a plugin project

- 1 Select **File > New > Other**.
The **Select a Wizard** dialog appears.
- 2 Select **Plugin Project** and click **Next**.
- 3 Type a name for the plugin in the appropriate field, and leave the remainder of the parameters as they appear. Click **Next**.
- 4 Leave the parameters on the next screen as they appear and click **Finish**.

The new plugin project is created in Eclipse and is ready for plugin development.

CONFIGURING THE TARGET PLATFORM

The target platform for the plugin development process needs to be identified in DB Change Manager. This will indicate to the new plugin for what product it extends, and will grant the plugin access to the system.

- 1 Select **Window > Preferences**.

The **Preferences** dialog appears.

- 2 Choose the plugin **Development > Target Platform** node and click **Browse**.
- 3 Navigate to the install directory for DB Change Manager.
- 4 Click **Apply** and **OK**.

The **Preferences** dialog closes and the target platform of the plugin is now indicated.

DEFINING PLUGIN DEPENDENCIES

The plugin requires a dependency for configuration comparison jobs in DB Change Manager. This dependency can be set by opening the `MANIFEST.MF` file located in the `META-INF` directory of your project.

- 1 Select the **Dependencies** tab and click **Add**.
- 2 Choose `com.embarcadero.change.config.propertysource` and click **OK**.

The dependency is added to the plugin and appears in the **Dependencies** list.

- 3 Press CTRL-S, or Select **File>Save..**

Once the dependency definition has been saved, you can close the editor and the system will retain your selection.

IMPLEMENTING A PROPERTY SOURCE

Each configuration property source plugin must contain a class that extends `com.embarcadero.change.config.propertysource.api.ConfigurationPropertySource`.

NOTE: If `ConfigurationPropertySource` cannot be found in Eclipse, it means you have misconfigured the target platform or target dependencies.

- 1 Right-click on the project and select **New>Class**.

The **New Java Class** dialog appears.

- 2 Name the class `JDBCVersionInformation`, set the package to `org.acme.versioninformation`, and set the superclass to `com.embarcadero.change.config.propertysource.api.ConfigurationPropertySource`.

3 Click **Finish**.

The new class is generated.

DETERMINING SUPPORT

Each class that subclasses `ConfigurationPropertySource` contributes properties to configuration comparison job functionality.

The implementation uses the `isSupported` method, which determines if there are attributes available for a specific data source.

For example, if a user added the Sybase `@@Version` tag as a configuration property, it would be pointless to run the code for Oracle, SQL Server, or DB2 for LUW data sources.

The method has the following signature:

```
boolean isSupported ( IDatasourceIdentifier dataSourceIdentifier )
```

This method is called by DB Change Manager, for any registered content plugin when a data source is used in a configuration archive job, a standard, or a comparison job. When it encounters a data source identifier, it returns true or false and indicates if the property source provides parameters for the data source, or not.

Implement the following code to define this method:

```
public boolean isSupported ( IDatasourceIdentifier dataSourceIdentifier )
{
    return dataSourceIdentifier.getDatabaseVersion().getDBMSType() !=DBMSType.DB2;
}
```

DEFINING PROPERTIES

The `getAvailableProperties` function returns a set of instances of `IConfigurationProperty` instances.

There are several utility methods to help instantiate these instances, named of the form, `createConfigurationProperty`.

Implement the following code to define the `getAvailableProperties` method:

```
public Set(IConfigurationProperty> getAvailableProperties (
    IDatasourceIdentifier identifier )
{
    Set<IConfigurationProperty> results = new HashSet<IConfigurationProperty>();

    for (EVersionProperties prop : EVersionProperties.values() )
    {
        results.add ( createCOnfigurationProperty ( prop.getDisplayName(),
                                                    prop.name(),
                                                    "",
                                                    prop.getType(),
                                                    "" ) );
    }

    return results;
}
```

Because the `EVersionProperties` code cannot be resolved, you can add a Java enum to define the properties provided by the plugin. Create a new file named `EVersionProperties.java` and implement the following code:

```
package org.acme.versioninformation;

import java.sql.Connection;
import java.sql.SQLException;

import com.embarcadero.change.config.propertysource.api.ConfigurationPropertyType;

/**
 * An enumeration of version properties
 */
public enum EVersionProperties
{
    PRODUCT_NAME ( ConfigurationPropertyType.STRING, "Product Name" )
    {
        /**
         * @returns the JDBC metadata's information for product name
         * @throws SQLException if there eis a JDBC error
         */
        public String getValue ( Connection connection ) throws SQLException
        {
            return connection.getMetaData().getDatabaseProductName();
        }
    },
    PRODUCT_VERSION ( ConfigurationPropertyType.STRING, "Product Version" )
    {
        /**
         * @returns the JDBC metadata's information for product version
         * @throws SQLException if there is a JDBC error
         */
        public String getVAlue ( Connection connection ) throws SQLException
        {
            return connection.getMetaData().getDAtabaseProductVersion();
        }
    },
    MAJOR_VERSION ( ConfigurationPropertyType.NUMERIC, "Major Version" )
    {
        /**
         * @returns the JDBC metadata's information for major version
         * @throws SQLException if there is a JDBC error
         */
        public String getValue ( Connection connection ) throws SQLException
        {
            return " " + connection.getMetaData().getDAtaMajorVersion();
        }
    }
};

//Attributes
private ConfigurationPropertyType type;
private String name;

/**
 * Constructor to set entry attributes
 * @param type the type
 * @param name
 */
private EVersionProperties ( ConfigurationPropertyType type,
    String name)
{
    this.type = type;
    this.name = name;
}

/**
 * Given a JDBC connection return the parameter's value
 * @param connection the database connection
 * @throws SQLException if there is a JDBC error

```

```

*/
abstract String getValue ( Connection connection ) throws SQLException;

/**
 * @return the display name of the parameter
 */
public String getDisplayName()
{
    return name;
}

/**
 * @return the type of the parameter
 */
public ConfigurationPropertyType getType()
{
    return type;
}

/**
 * Finds a property of a name and returns it
 * @param name the name of the property
 * @return the property
 */
public static EVersionProperties getProperty ( String name )
{
    for ( EVersionProperties p : values() )
    {
        if ( p.name().equals ( name ) )
        {
            return p;
        }
    }

    throw new RuntimeException ( EVersionProperties.class.getName() + " not found for
    \" + name + "\" );
}

```

In the code above, the four entries that capture from JDBC the product name, version, minor version, and major version have been added to the enum.

GATHERING RESULTS

DB Change Manager gathers results when it collects all of the properties and values for display purposes or to store in an archive. The extraction method is called when a data source is indicated in a standard, or when an archive or comparison job is run.

```

public Set<IConfigurationPropertyResult> extract( Set<IConfigurationProperty>
properties, IDatasourceIdentifier dataSourceIdentifier )

```

A set of `IConfigurationPropertyResult` instances that correspond to the properties must be returned, and values need to be added for the properties returned from the data source, as follows:

```
public Set<IConfigurationPropertyResult> extract ( Set<IConfigurationProperty>
properties, IDatasourceIdentifier dataSourceIdentifier)
{
    try
    {
        Set<IConfigurationPropertyResult> results = new
HashSet<IConfigurationPropertyResult> ();

        Connection connection = getConnection (dataSourceIdentifier );

        for (IConfigurationProperty configProp : properties )
        {
            EVersionProperties prop = EVersionProperties.getProperty (
                configProp.getPropertyIdentifier() );
            String value = prop.getValue ( connection );

            results.add( createConfigurationPropertyResult( configProp, true, value )
);
        }
        return results;
    }
    catch ( SQLException sql )
    {
        throw new RuntimeException ( sql );
    }
}
```

In the above implementation, the code creates an empty set to hold the results and then retrieves the JDBC connection to the data source. For each retrieval, the enum constant `EVersionProperties` on which `getValue` is called to retrieve each property value from JDBC metadata. Finally, `createConfigurationPropertyResult` is called to create the result instance.

SYNCHRONIZING THE JOB

The synchronization method returns an empty array because users cannot synchronize version information.

```
public List<IPropertyResultsSynchronization> sync (
    IDatasourceIdentifier dataSourceIdentifier,
    List<IConfigurationPropertyResult> sourcePropertyValues,
    List<IConfigurationPropertyResult> targetPropertyValues )
{
    return new ArrayList<IPropertyResultsSynchronization>();
}
```

REGISTERING EXTENSIONS

Once you have completed coding the plugin, register it with DB Change Manager by implementing an Eclipse extension. When the plugin is added to DB Change Manager, the extensions are registered and DB Change Manager understands that the plugin adds a new type of notification to the module.

1 Open the `Manifest.MF` file and navigate to the **Extensions** tab.

2 Click **Add** to register an extension.

The **New Extension** dialog appears.

3 Choose `com.embarcadero.change.config.propertysource.contribution`.

4 Click **Finish** and close the dialog.

The extension is added to the **All Extensions** list.

5 Select the (source) entry on the right-hand side of the screen and enter the following details:

- `display-name: JDBC Metadata Version Information`
- `source-identifier: org.acme.jdbc.versioninformation.source`
- `class: org.acme.jdbcversioninformation.JDBCVersionInformation`

NOTE: The class name is used to enable DB Change Manager to instantiate the new class, while the display-name is the group under which the four JDBC metadata version properties will show.

6 Save the file by pressing CTRL-S or choose **File>Save** from the menu.

The class notification is complete, and DB Change Manager and Eclipse are informed about the property source. When the plugin is added to DB Change Manager, it will add your properties to configuration comparison and standard jobs.

DEPLOYING THE CUSTOM CONFIGURATION PLUGIN

To deploy the custom plugin

1 Shut down DB Change Manager if it is running.

2 Click **File>Export**.

The **Export** dialog appears.

3 Select **Deployable Plugins and Fragments** from the **Plugin Development** group and click **Next**.

4 Select **org.acme.jdbcversioninformation** and ensure the target directory is the same as the DB Change Manager installation folder.

5 Click **Finish**.

The new plugin is deployed.

Start DB Change Manager and create a new configuration archive job. When you select an Oracle, Sybase, or SQL Server data source, the **Refinements** and the **Results** tab will contain the four version properties you defined in the new plugin.