



Product Documentation

Embarcadero® Change Manager™

User Guide

Version 5.1.1

Published September 9, 2009

Copyright © 1994-2009 Embarcadero Technologies, Inc.

Embarcadero Technologies, Inc.
100 California Street, 12th Floor
San Francisco, CA 94111 U.S.A.
All rights reserved.

All brands and product names are trademarks or registered trademarks of their respective owners.

This software/documentation contains proprietary information of Embarcadero Technologies, Inc.; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited.

If this software/documentation is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

Restricted Rights Legend Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

If this software/documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with Restricted Rights, as defined in FAR 552.227-14, Rights in Data-General, including Alternate III (June 1987).

Information in this document is subject to change without notice. Revisions may be issued to advise of such changes and additions. Embarcadero Technologies, Inc. does not warrant that this documentation is error-free.

Contents

Welcome to Change Manager	9
Technical Requirements	9
Additional Product Information	10
Overview of Change Manager	11
What You Can Do With Change Manager	13
Commonly-performed Tasks	13
Comparing and Synchronizing Data	13
Comparing and Synchronizing Schemas	13
Managing Database Configurations	13
Creating Masked Data for Development Environments	14
Automating Jobs	14
Registering and Organizing Data Sources	15
Registering Data Sources on Startup	15
Scanning for Data Sources to Register	15
Registering a New Data Source	18
Creating a New Data Source Group	20
Importing and Exporting Data Sources	21
Creating a Job Report	24
Setting Job Notifications	27
Email Notifications	27
File System Notifications for Log Files	28
System Tray Notifications	29
Filtering Data Source Objects	32
Comparing Data	34
Create a Data Comparison Job	35
Describe and Save the Job	35
Specify the Source and Target Data Sources	36
Specify Execution, Mapping, Comparison, and Resolution Options	37
Execution Options	37
Mapping Options	38
Comparison Options	39
Resolution Options	40

Understand the Data Source Mapping Process	40
Customize the Data Source Mapping Process	42
Customize Database Mapping	42
Customize Table Mapping	43
Set Job Notification Options	44
Run the Data Comparison Job	45
Synchronizing Data Repositories	47
Selecting Data to Resolve	47
Running the SQL Synchronization Script	50
Configure an SQL Session	51
Comparing Database Configuration Settings	53
Create or Open a Configuration Comparison Job	53
Describe and Save the Job	54
Specify Source and Target Data Sources	55
Refine Property Comparisons	57
Specify Job Options	58
Set Job Notification Options	58
Run the Comparison Job	58
View the Results of the Comparison	59
Create a Job Report	60
Synchronize Configurations	60
Creating a Standard for Configuration Settings	63
Create or Open a Configuration Standard	63
Describe and Save the Standard	64
Add Standard Sources	64
Define Configuration Standard Properties	65
Use a Configuration Standard in Configuration Comparison Jobs	66
Creating a Configuration Archive	68
Create or Open a Configuration Archive Job	68
Describe and Save the Archive Job	69
Specify an Archive Source	69
Refine Properties to be Archived	70
Set Job Notification Options	71
Create a Version and Save the Archive Job	71
Create a Job Report	72
Use an Archive in a Configuration Comparison Job	72

Comparing Schemas	73
Create or Open a Schema Comparison Job	74
Describe and Save the Job	74
Specify a Comparison Source and Targets	75
Add Multiple Schema Comparison Job Targets	76
Refine Schema Comparisons	77
Customize Schema Mapping	78
Specify Job Options	79
Synchronization Options	79
Compare Options	80
Schema Comparison Options by Database Platform	80
Include Object Options	84
Identifier Options	85
DDL Extract Options	85
Set Job Notification Options	85
Run the Schema Comparison Job	86
Viewing Individual Schema Comparison Results	88
Creating a Schema Archive	90
Create or Open a Schema Archive Job	91
Describe and Save the Schema Archive Job	91
Specify an Archive Job Source	92
Refine Properties	93
Set Job Notification Options	94
Specify Extraction Options	94
Create a Version and Save the Archive Job	95
Compare Archive Contents and Extract DDL	95
Create a Job Report	95
Use an Archive in Schema Comparison Jobs	95
Synchronizing Schemas	97
Resolve Schema via a Generated SQL Script	97
Execute a SQL Synchronization Script	98
Configure an SQL Session	99
Masking Data	102
Create a New Masking Job	102
Specify a Masking Job Name	102

Specify Source and Target Data Sources	103
Set to Automatically Synchronize	104
Specify Mapping Options	105
Specify Masking Rules for Individual Columns	106
Run the Database Masking Job	107
See the Masked Data	108
Automating Batch Scripts	109
Generating Command Line Scripts	109
Using the change Command Line Program	110
Cheat Sheets	112
Cheat Sheet 1A: Creating a New Configuration Comparison Job	112
Cheat Sheet 1B: Creating a New Configuration Archive	116
Cheat Sheet 1C: Creating a New Configuration Standard	119
Cheat Sheet 2A: Creating a New Data Comparison Job	121
Cheat Sheet 3A: Creating a New Schema Comparison Job	127
Cheat Sheet 3B: Creating a New Schema Archive	130
DBMS Reference	133
DBMS Operations Permissions	133
Permissions Required for DB2 for LUW	133
Permissions Required for Firebird and InterBase	133
Permissions Required for SQL Server	134
Permissions Required for Oracle	134
Permissions Required for Sybase	135
Version Information Reference	135
Firebird and InterBase Version Information	136
IBM DB2 for LUW Version Information	136
Oracle Version Information	136
SQL Server Version Information	137
Sybase Version Information	137
Preferences Reference	138
Configuration Comparison Options	139
Data Comparison Options	139
Set Schema Comparison and Archive Options	140
Command Line	142
Compliance	142
Label Decorations	142
Notification	142
Projects	143

Comparison Options	143
Data Result Options	143
Script Execution Options	144
Script Generation Options	144
Jobs	144
Command Line	145
Mapping	145
Notification	145
Reports	145
Data Sources	146
SQL Editor	146
Code Assist	146
Code Formatter	146
Results Viewer	147
Syntax Coloring	147
SQL Filters	147
Extensions Reference	148
Extending Change Manager Functionality with Eclipse Plug-Ins	148
Creating and Implementing a Notification Plug-In	148
Creating and Implementing a Custom Configuration Plug-In	159
User Interface Reference	166
The Change Manager Workbench	167
Welcome Page	168
Views	169
Job Explorer	170
Viewing Change History	170
Data Source Explorer	171
Project Explorer	173
Navigator View	174
Compliance Explorer	175
SQL Errors View	175
Data Diff View	176
SQL Log	176
SQL Errors Log	176
Search View	177
Editors	180
Data Comparison Job Editor	181
Configuration Comparison Job Editor	182
Configuration Standard Editor	183

CONTENTS

Configuration Archive Editor	184
Schema Comparison Job Editor	185
Schema Archive Editor	186
SQL Editor.	187
Glossary	189

Welcome to Change Manager

Embarcadero® Change Manager™ helps database administrators and developers simplify and automate database change management to streamline development and ensure availability, performance and compliance.

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

Change Manager's compare, synchronization, data masking, and auditing capabilities 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 in less time. And by monitoring configuration settings, DBAs can ensure compliance with regulatory policies and performance standards, and maintain overall database performance and availability. Change Manager supports the most popular relational databases from a single, easy to use interface.

Change Manager offers:

- Schema "archive" capture, compare and synchronization
- Data compare and synchronization with custom mapping features
- Archive and compare database configuration settings
- Advanced alter script generation; handles object dependencies and preserves data
- Full Unicode Support
- Source Code Control (SCC) Integration
- Command-line API for scheduling jobs and event notification
- Data masking with shuffling and randomization

Technical Requirements

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

Hardware Requirements

The following minimum hardware requirements are required to run Change Manager:

- 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)
- Internet Explorer 6 Service Pack 1 or later (Windows) or Mozilla (Linux)

Operating System Requirements

Change Manager supports the following operating systems:

- Microsoft Windows XP (x86-32, Win32)
- Microsoft Windows Server 2003
- Microsoft Windows Vista

DBMS Support

Change Manager supports the following platforms:

- FireBird® 1.5, 2.0
- InterBase® 2007, 2009 (excluding the To Go Edition)
- IBM® DB2® LUW 8.0 - 9.5
- Oracle® 8i - 11g
- Microsoft SQL Server® 2000, 2005, and 2008
- Sybase® ASE 12.5 and 15, 15.0.1, and 15.0.2

Installation Notes

Change Manager can be installed as a standalone application (RCP installation) or as an Eclipse plug-in (Plug-in Installation). Eclipse is an open source development framework that supports application plug-ins to provide additional functionality.

The Eclipse plug-in version of Change Manager requires Eclipse version 3.3 or higher, and Sun Java Standard Edition 5.0 Update 11 or later, in addition to regular system requirements.

Before installing the plug-in version of Change Manager, ensure that Eclipse and Java are installed on your machine. You can download Eclipse from the following Web site: <http://www.eclipse.org/downloads>.

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

Overview of Change Manager

Change Manager helps you manage changes to your data, database configurations, and schemas. This topic gives you a quick overview of the user interface and describes basic concepts.

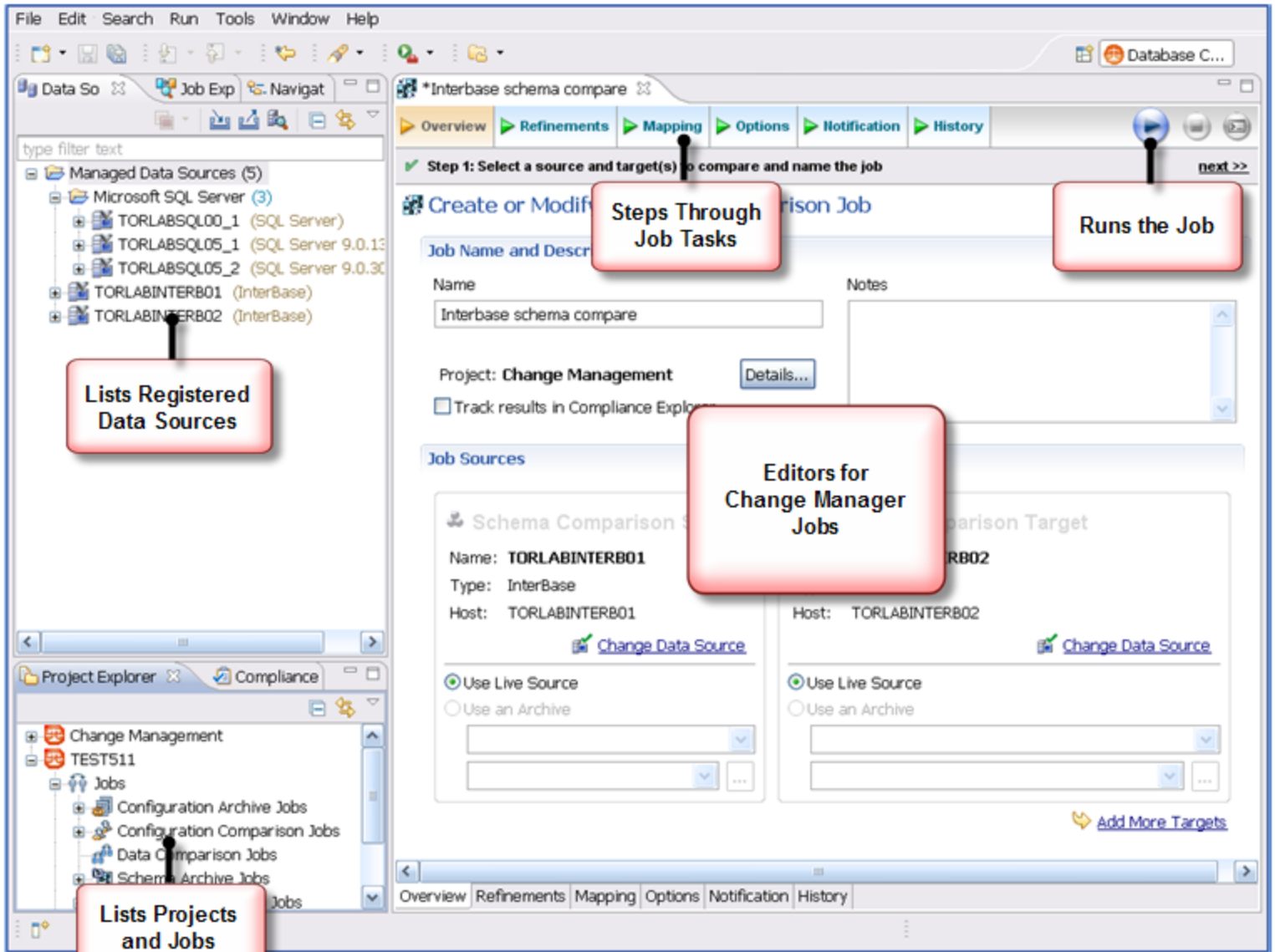
Change Manager uses the term “job” to mean a complete start-to-finish task, such as comparing two schemas. A job can generate a report of its activities. You can automate jobs using scripts.

Change Manager first must gather information about your databases. You register them with the system, and information is organized and displayed in the Change Manager interface. You can apply filters to narrow down or single out the particular objects you want to use in a job.

Change Manager supports the most popular databases, and each is slightly different than the others. These differences are noted in the interface and in the documentation. Make sure you have applied the relevant permissions for your database before using Change Manager. See the [DBMS Reference](#) for details.

Because Change Manager is an Eclipse-based product, you can extend it using plugins. See the [Extensions Reference](#) for more information.

Change Manager is based on the Eclipse interface. See the [User Interface Reference](#) for details about its views and editors.



What You Can Do With Change Manager

This section describes what you can do with Change Manager, and links to more detailed information about how to perform the tasks. In addition, Change Manager includes step-by-step Eclipse-style cheat sheets to help you learn the tasks. The cheat sheets are reproduced in the [Cheat Sheets](#) topic so that you can print them out for easy reference.

Commonly-performed Tasks

These tasks describe basic functions and help you use 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](#)
- [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 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.

- [Comparing Schemas](#)
- [Synchronizing Schemas](#)
- [Creating a Schema Archive](#)

Managing Database Configurations

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. 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 Jobs

You may need some jobs to run on a regular basis, and Change Manager can automate that task. The Script Wizard lets you define scripts that run jobs, and batch files to run multiple scripts.

- [Generating Command Line Scripts](#)

Registering and Organizing Data Sources

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. 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.

- [Registering Data Sources on Startup](#)
- [Scanning for Data Sources to Register](#)
- [Registering a New Data Source](#)
- [Creating a New Data Source Group](#)
- [Importing and Exporting Data Sources](#)

Registering Data Sources on Startup

When 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 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 to Register

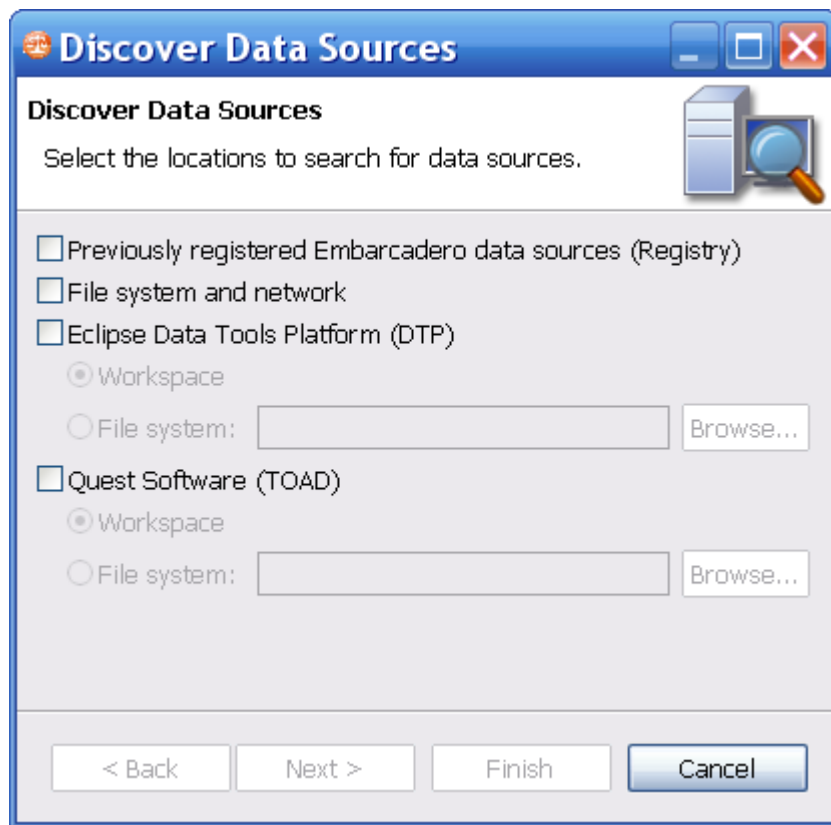
Change Manager can register data sources by scanning your systems.

To initiate a scan for data sources:

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



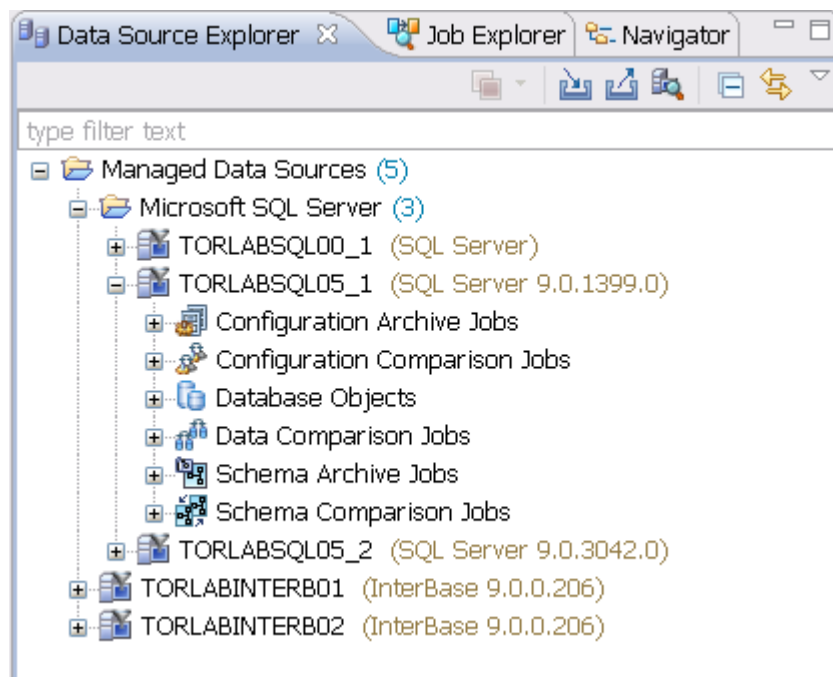
The **Discover Data Sources** dialog appears.



- 2 Choose the type of data sources you want to scan for and click **Next**.



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

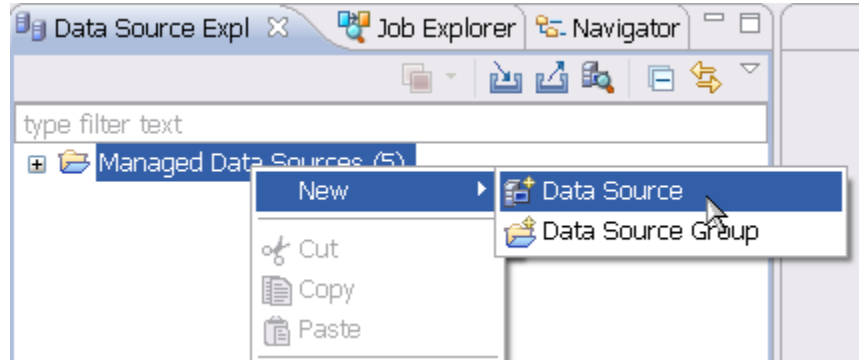


Registering a New Data Source

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

New Data Source: New data so...

Configure a new InterBase/Firebird data source

Enter InterBase/Firebird-specific connection and security credential information for the new data

InterBase/Firebird

Host/Instance: *

Port: 3050

Database: *

Security credentials

User name: *

Password:

Connect as: normal

Auto-connect (Saves and encrypts password)

Allow trusted connections

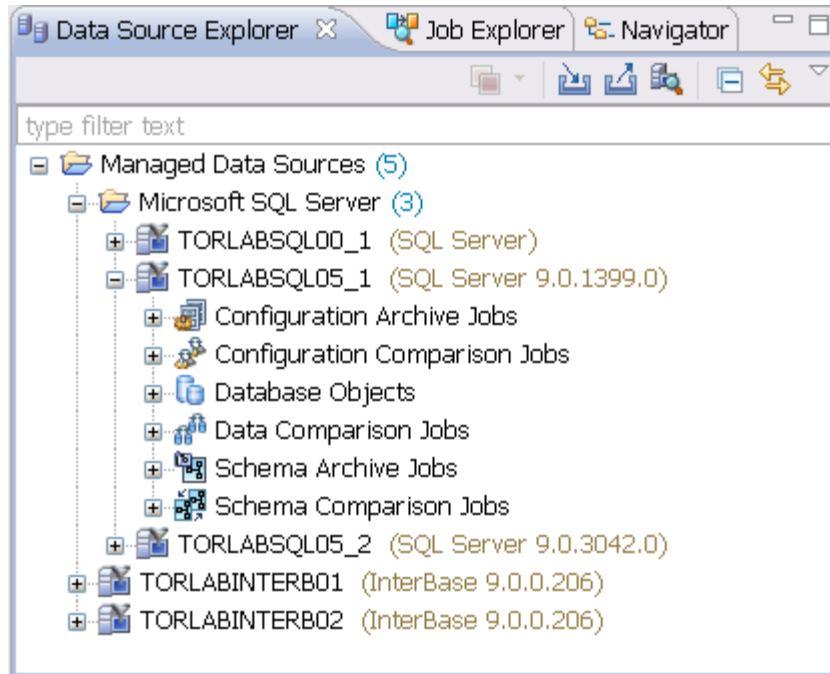
Test Connection

< Back Next > Finish Cancel

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

Creating a New Data Source Group

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



To create a data source group:

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

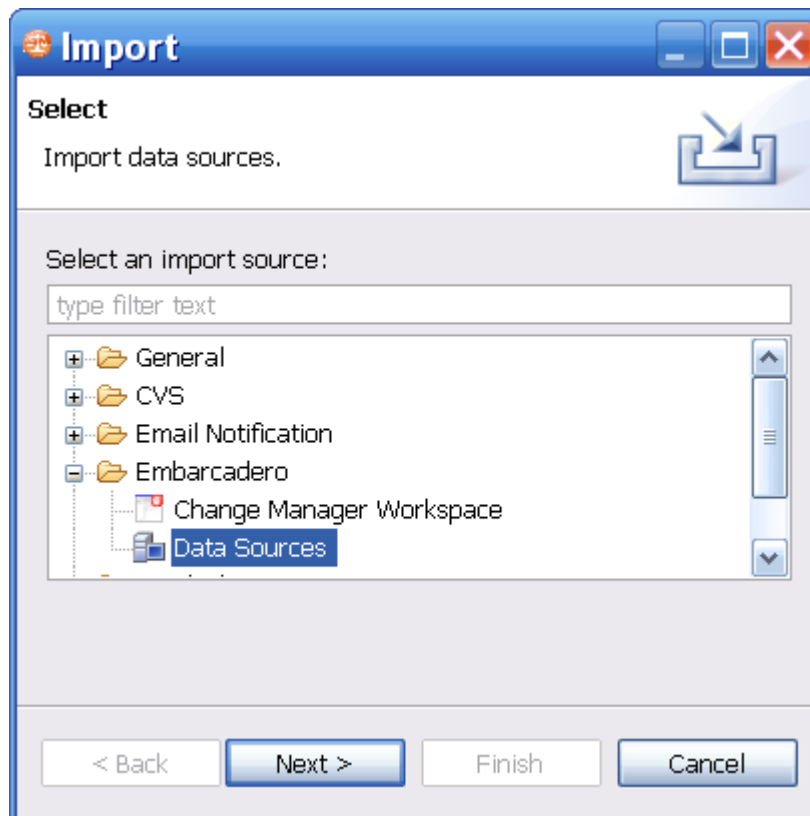
- Eclipse Data Tools Platform (DTP)
- Previously Registered Embarcadero Data Sources (File)
- Previously Registered Embarcadero Data Sources (Registry)
- Quest Software (TOAD)

Conversely, you can export your current data source catalog to a file, which can then be imported into other instances of 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)**.

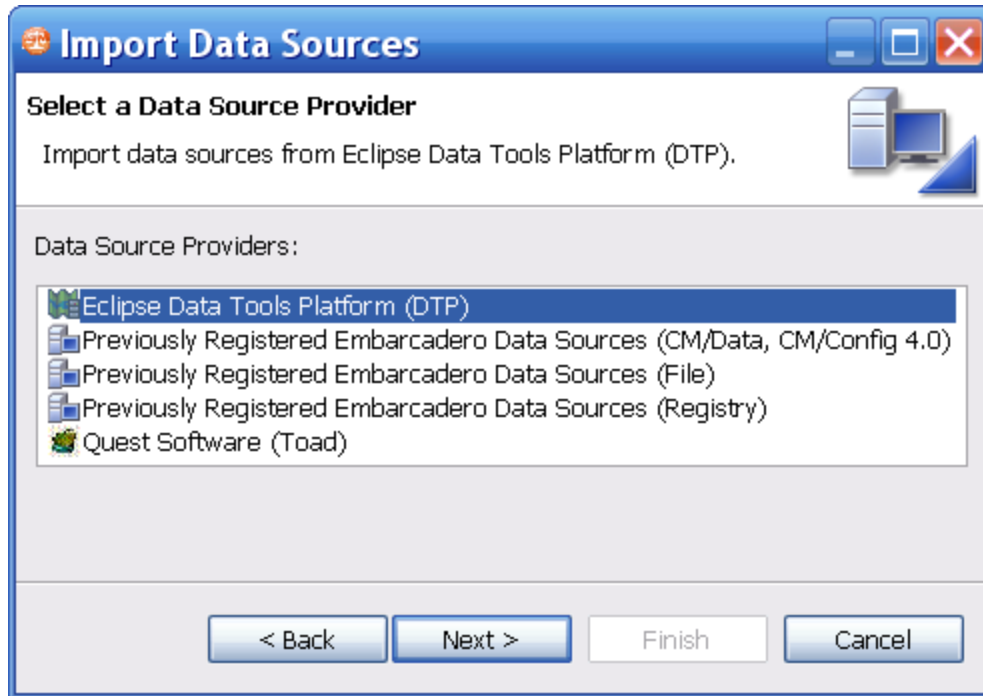
To import data sources:

Select **File > Import**.



- 4 Choose **Embarcadero > Data Sources**.

Click **Next**.



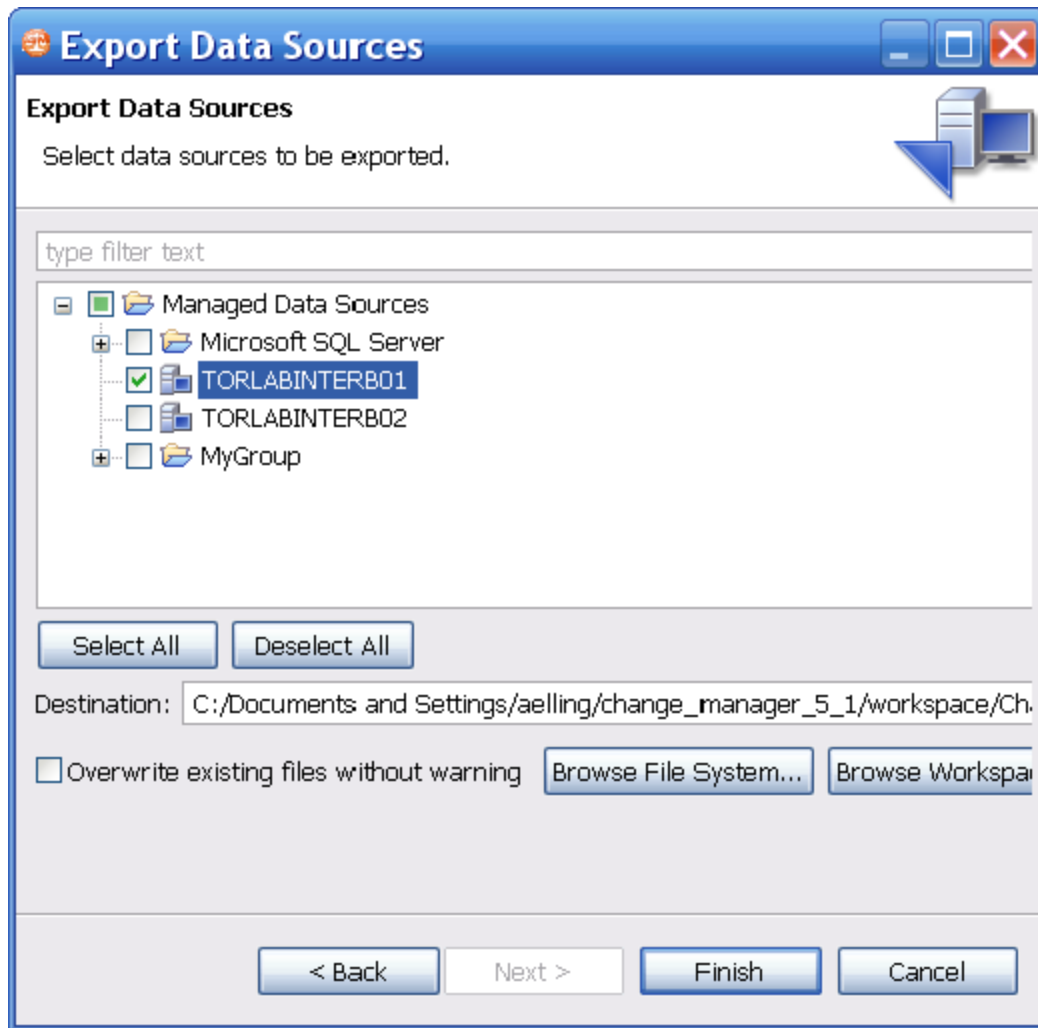
- 5 Choose a source from which you want to import the data sources.
- 6 Click **Next**.
- 7 Specify the details about the location of the import source.
- 8 Click **Finish**.

Data Source Explorer is automatically populated with the new data sources.

To export data sources:

- 1 Select **File > Export**. The **Export Data Sources** dialog appears.
- 2 Choose **Embarcadero > Data Sources**.

Click **Next**.




- 3 Browse the list and select a data source to export.
- 4 Type or use the **Browse File System** or **Browse Workspace** buttons to enter the directory path and file name.
- 5 Click **Finish**.

The data source is are exported as an XML file. You can import this file to other instances of Change Manager.

Creating a Job Report

After you have executed a job, you can see a job report using the **View Report** command on the **Results** tab. The report opens in a separate window and can be read on screen, sent to a printer, and saved to disk.



Data Comparison Summary Report
 Untitled-Data-Comparison-Job-1 (August 29, 2009 12:55 PM)

1 - Overview for Untitled-Data-Comparison-Job-1

Name: Untitled-Data-Comparison-Job-1
 Project: Change Management
 Job Host: USSVL-DELLING
 Start Time: Aug 29, 2009 12:55 PM
 Elapsed Time: 00:18 (mm:ss)
 Rows Matched: 39 - 0.58%
 Rows Not Matched: 6951 - 99.44%

2 - Source

Name: TORLABSQL05_1
 Type: Live Database
 Platform: Microsoft SQL Server 9.0.1399.0
 Alias: TORLABSQL05_1
 User: sa
 Protocol: TCP/IP

3 - Target

Name: TORLABSQL05_2
 Type: Live Database
 Platform: Microsoft SQL Server 9.0.3042.0
 Alias: TORLABSQL05_2
 User: sa
 Protocol: TCP/IP

4 - Comparisons

TORLABSQL05_1	TORLABSQL05_2	Progress	Compare Index
FranklynA	FranklynA	Done	0% Match

4.1 - FranklynA - FranklynA

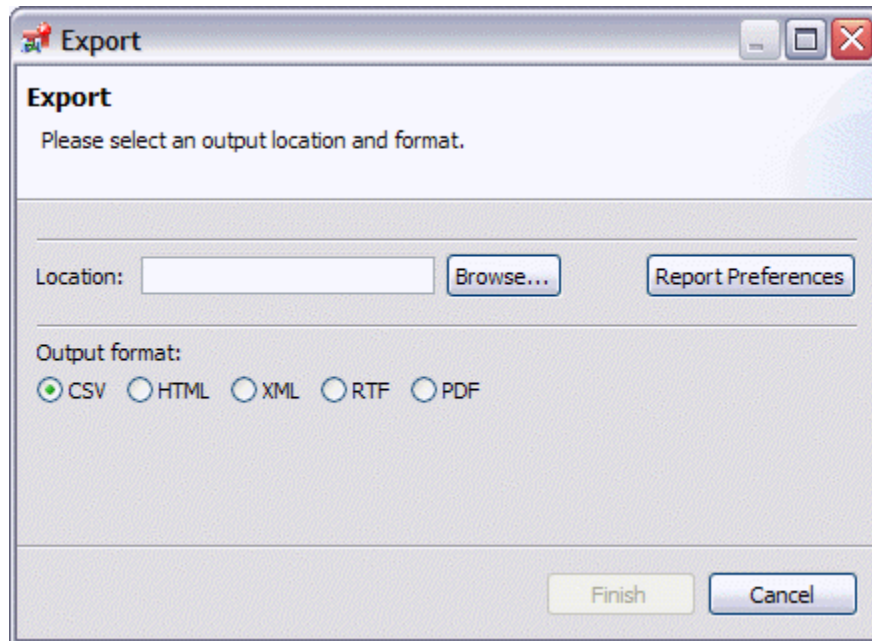
FranklynA	FranklynA	Matched	Only in source	Only in target	Difference
Address	Address	0	134	115	341
Category	Category	16	0	0	0
Customer	Customer	0	0	0	348
Customer_Address	Customer_Address	4	459	440	0

Embarcadero® Change Manager™ Number of Pages: 2 Page: 1

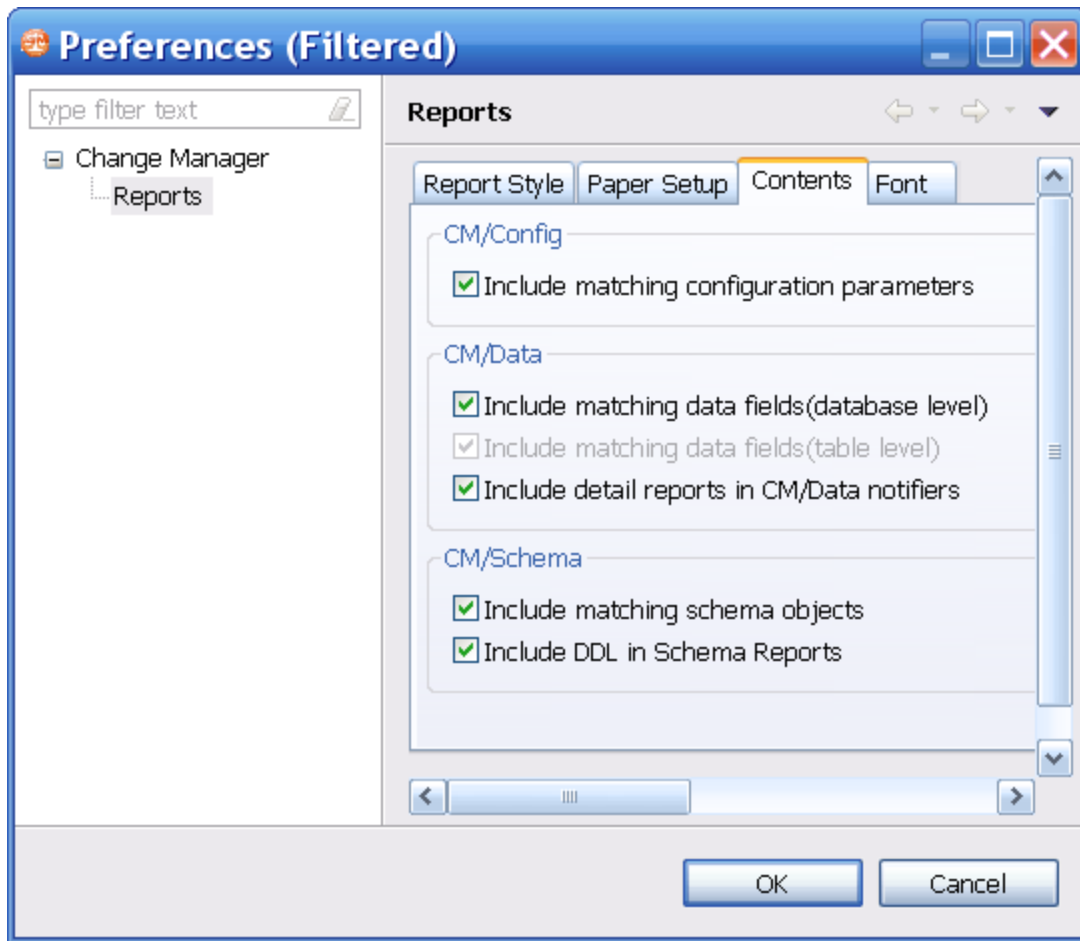
The **Export Results** command will produce a report file in a specified format at a directory location of your choosing.

To create a report file:

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



- 2 Select a directory using the **Location** field.
- 3 Click an output format.

4 Click **Report Preferences**.5 In the **Contents** tab, specify the criteria to include in the report.**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 Data Detail Reports in 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.

For Configuration Comparison Jobs:

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

Setting Job Notifications

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

- [Email Notifications](#)
- [File System Notifications for Log Files](#)
- [System Tray Notifications](#)

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 the 'Email Notifier' configuration window. At the top, there is a dropdown menu labeled 'Email Notifier'. Below it, a checkbox labeled 'Enable Email Notifier' is checked. There are two text input fields: 'To:' and 'Template:'. The 'Template:' field contains the text 'Default Email Notification Template'. Below these fields is a link that says 'Show Email Notification Preferences'. The window is divided into two main sections: 'Send Notification' and 'Attachments'. The 'Send Notification' section has three sub-sections: 'When Run From:' with radio buttons for 'Workbench', 'Command Line', and 'Both' (selected); 'Job Outcome:' with radio buttons for 'Only on Success', 'Only on Error', and 'Both' (selected); and 'For Threshold:' with checkboxes for 'Pass', 'Warn', and 'Fail' (all checked), and a link 'Open Compliance Page...'. The 'Attachments' section has three sub-sections: 'Report:' with radio buttons for 'None' (selected), 'CSV', 'XML', 'HTML', 'PDF', and 'RTF'; 'Log Files:' with radio buttons for 'Always' and 'Only on Error' (selected).

Click **Enable Email Notifier** to enable email notification. When you select this check box, a list of notification parameters appear.

Enter the email addresses to which the notification will be sent in the **To** field. You can specify an email format template in the **Template Field**.

The **Send Notification** options specify when a notification email should be sent based on the results of a comparison job:

- The **When Run From** options enable you to indicate that email notifications should only be sent if a job is executed from the **Workbench**, **Command Line**, or **Both**.
- The **Job Outcome** options enable you to indicate that email notifications should be sent **Only on Success**, **Only on Error**, or **Both**.

- The **For Threshold** options enable you to specify the threshold level for which email notifications are sent, based on job results. Choose **Pass**, **Warn**, or **Fail**. 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.

The **Attachments** section enables you to specify the file format of an attached report that will display job results with the notification email. Choose **None** (does not attach a report), **CSV**, **XML**, **HTML**, **PDF**, or **RTF** to specify the file format of the report. The **Log Files** indicates when to log the notification file results to the Change Manager log file. You can choose **Always** or **Only on Error**.

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](#)

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

Click **Enable File System Notifier** to enable logging. When you select this check box, a list of notification parameters appear.

Specify the directory location of each of the output files that are produced by the notification function in the **Report Directory**, **Log Directory**, and **Script Directory**, respectively. Additionally, you can click the **Show File System Notification Preferences** link if you want to change the directory locations at the application level rather than just the job level.

The **Send Notification** options specify when job results are output to the specified file directories:

- The **When Run From** options enable you to indicate that results should only be sent to file if a job is executed from the **Workbench**, **Command Line**, or **Both**.

- The **Job Outcome** options enable you to indicate that results should be sent to file **Only on Success**, **Only on Error**, or **Both**.
- The **For Threshold** options enable you to specify the threshold level for which results are sent to file, based on job results. Choose **Pass**, **Warn**, or **Fail**. 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.

The **Attachments** section enables you to specify the file format of the output results. Choose **None** (does not produce a file), **CSV**, **XML**, **HTML**, **PDF**, or **RTF** to specify the file format. The **Log Files** indicates when to log the notification file results to the Change Manager log file. You can choose **Always** or **Only on Error**.

System Tray Notifications

When you are running jobs in the background, Change Manager can send System Tray notifications to your desktop.

Click **Enable System Tray Notifier** to enable system tray notifications. When you select this check box, a list of notification parameters appear.

Click **Show System Tray Notifications** to specify global parameters with regards to the appearance and behavior of the specific pop-up windows.

The **Send Notification** options specify when a pop-up notification will appear.

- The **When Run From** options enable you to indicate that a pop-up notification should only appear if a job is executed from the **Workbench**, **Command Line**, or **Both**.
- The **Job Outcome** options enable you to indicate that a pop-up notification should appear **Only on Success**, **Only on Error**, or **Both**.

- The **For Threshold** options enable you to specify the threshold level for which a pop-up notification will appear, based on job results. Choose **Pass**, **Warn**, or **Fail**. 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.

NOTE: The **Attachments** section enables you to specify the file format of a report that can be accessed via the pop-up notification. Choose **None** (does not produce a file), **CSV**, **XML**, **HTML**, **PDF**, or **RTF** to specify the file format of the report. The **Log Files** indicates when to log the notification to the Change Manager log file. You can choose **Always** or **Only on Error**.

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:

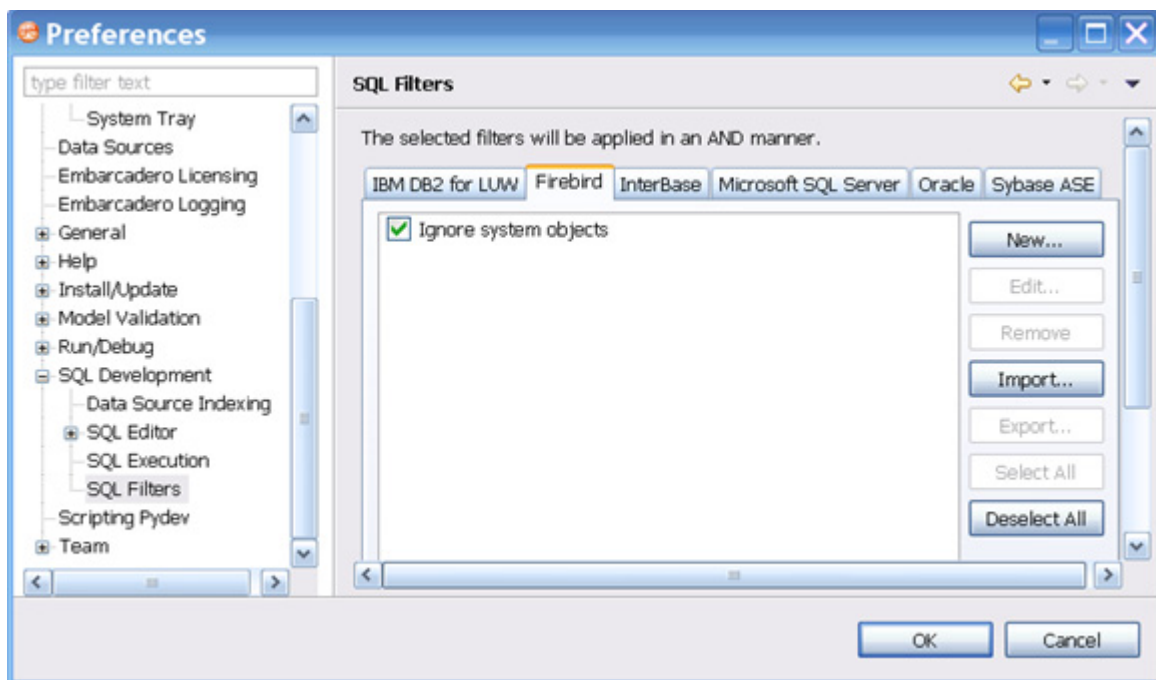
- **Global filters** affect all registered data sources.
- **Data source specific filters** affect only the specified data source.

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 Select **Window > Preferences** from the Main Menu. The **Preferences** dialog appears.
- 2 Expand the **SQL Development** node and select **SQL Filter**.

The **SQL Filter** 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.



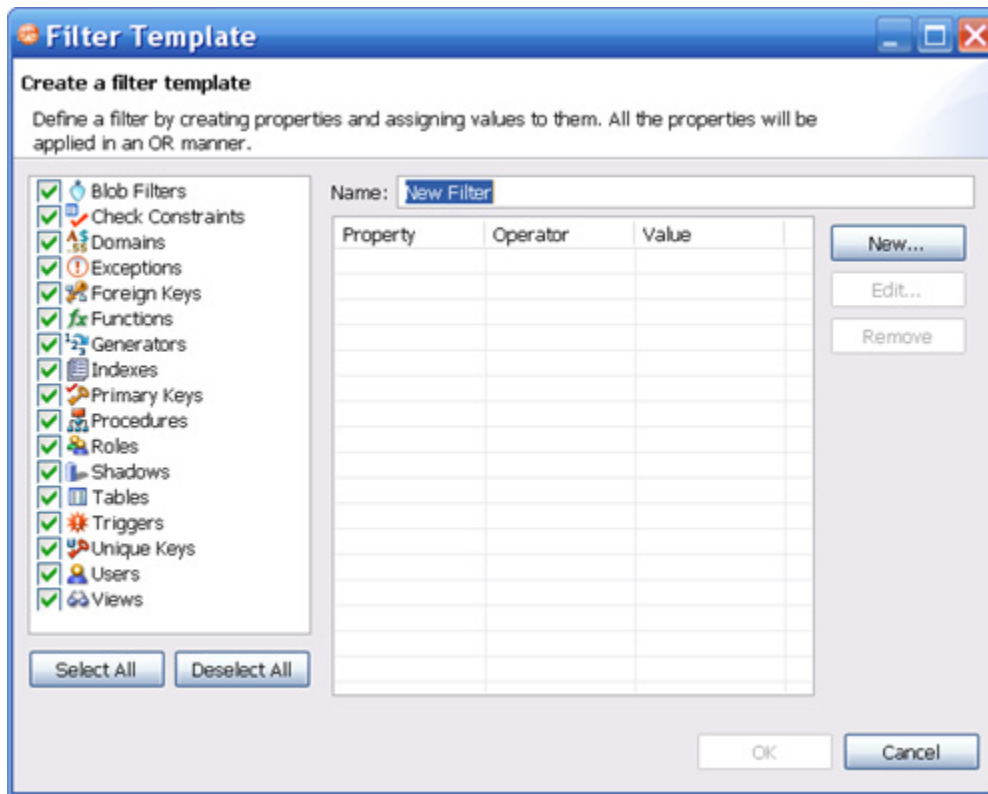
- 3 Click **New**. The **Filter Template** dialog appears. See [To create a filter template](#).

To define a data source specific filter:

- 1 In Data Source Explorer, right-click the data source and select **Properties**. The **Properties** dialog appears.
- 2 Select the **SQL Filter** node and select **Enable Data Source Specific Settings**. The other controls on the dialog become enabled.

- 3 Click **New**. The **Filter Template** dialog appears. 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 as you want it to appear in the selection window on the **SQL Filter** node.
 - 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** or **Schema**, and **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 in Data Source Explorer.
- 2 Click **OK**. Repeat the previous step to add more filter parameters.
- 3 When you have finished defining the filter template, click **OK**. The template name is added to the **SQL Filters** node. You can enable and disable it by selecting or de-selecting the check box beside its name.

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, 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.

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 the Data Source Mapping Process](#)
- 7 [Customize Database Mapping](#)
- 8 [Customize Table Mapping](#)
- 9 Optionally request email, logs, or system tray notifications; see [Setting Job Notifications](#)
- 10 [Run the Data Comparison Job](#)
- 11 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 job from the list and 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 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 screenshot shows a dialog box titled "Job Name and Description". It has two main sections: "Name" and "Notes". The "Name" field contains the text "Data Compare - Torlabsq|00_2|". The "Notes" field is empty. Below the "Name" field, there is a "Project: Change Management" label and a "Details..." button. At the bottom left, there is a checkbox labeled "Track results in Compliance Explorer" which is currently unchecked.

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 the Source and Target Data Sources

The **Data Comparison Source** and **Data Comparison Target** boxes identify what data sources are compared when the job executes. Both boxes must contain the sources before the job can be executed. When a data source is added to a job, it is displayed by **Name**, **Type**, and **Host**.

Additionally, the drop down menus below the boxes enables you to map the data source at a high level by providing a list of the databases, schemas, or owners of the specified data source, depending on the DBMS.

The screenshot shows a window titled "Job Sources" with two main panels. The left panel is titled "Data Comparison Source" and contains the following information: Name: **torlabsql00_2**, Type: Microsoft SQL Server, Host: torlabsql00_2. Below this is a "Change Datasource" button with a green checkmark icon. At the bottom is a "Databases:" label and a dropdown menu currently showing "All Databases". The right panel is titled "Data Comparison Target" and contains: Name: **TORLABSQL00_3**, Type: Microsoft SQL Server, Host: TORLABSQL00_3. It also has a "Change Datasource" button and a "Databases:" dropdown menu showing "All Databases".

To add a data source to a job:

- 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**.

Specify Execution, Mapping, Comparison, and Resolution Options

Prior to the execution of a job, parameters can be set to control aspects of the execution process itself. These options can be set prior to job execution on the **Options** tab.

[Execution Options](#)

[Mapping Options](#)

[Comparison Options](#)

[Resolution Options](#)

Execution Options

Execution Options control aspects of the job process execution:

▼ **Execution Options**

Automatically Synchronize
Clear all data on the target and replace it with all data from the source automatically. If unchecked, manual synchronization will be required, where you can review the comparison results and select rows to synchronize in either direction to generate SQL scripts.

Compression Comparison
With this option enabled data on your datasources is compressed before being transmitted to the workbench and compared. In certain situations data compression enables the job to run at a much faster rate by removing the need to transfer and compare large volumes of data.

Generate DML
Automatically output the synchronization DML resulting from job execution. This DML file can then optionally be sent with any enabled Notification.

- **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.
- The **Compression Comparison** setting enables a faster version of the execution process where the job compresses row data for faster transmission and comparison. This feature allows for a faster comparison process in general and does not affect job accuracy. This option is especially useful over 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.
- The **Generate DML** option enables you to specify that the synchronization DML used in the job execution 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 by the process:

▼ **Mapping Options**

Ignore Case
Databases, owners, tables and columns will be paired up for comparison without regards to case. For example, with this option selected the table "Product_Prices" will be paired with table "PRODUCT_PRICES".

Ignore Spaces
Pairs database objects for comparison ignoring spaces in their names. For example, with this option selected the table "Product Prices" will be paired with the table "ProductPrices".

Ignore Underscores
Pairs database objects for comparison ignoring underscores in their names. For example, with this option selected the table "Product_Prices" will be paired with the table "ProductPrices".

Include Views
Enable this option to include views in the mapping. Otherwise, views will be filtered out of the mapping and will not be compared. Note that with this option enabled, you may still select which views (if any) you wish to include in the comparison.

Exclude Computed Columns
Enable this option to exclude from mapping (and comparison) the columns whose values are computed.

- **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 the Data Source Mapping Process](#) for more information about these options.

Comparison Options

The **Comparison Options** section specifies the parameters of how the job treats comparison in terms of naming conventions.

▼ Comparison Options

Ignore Case

Data in table fields are compared without regards to case. For example, a value of "Apple" matches a corresponding value of "apple" when this option is selected. De-selected, "Apple" and "apple" will be flagged as non-matching data in the comparison results.

Trim Trailing Spaces

Spaces at the end of values in table fields are disregarded. For example, a value of "Apple " matches a corresponding value of "Apple" when this option is selected. De-selected, "Apple " and "Apple" are flagged as non-matching data in the comparison results.

Match Empty Strings with Nulls

Empty table fields are considered equal to corresponding fields with null values. If Trim Trailing Spaces is selected with this option, fields containing only space characters are considered equal to null values in addition to empty table fields.

Cache Long Datatypes Values

With this option turned on long data type (blob, clob, varchar(max), etc.) values that do not match will be cached during comparison and available for resolution. When turned off, the results view will show that the values did not match but will not allow resolution.

Check rows from source

Check rows that exist in source tables. If this option is unchecked, the rows that only exist on the source tables won't be included in the results.

Check rows from target

Check rows that exist in target tables. If this option is unchecked, the rows that only exist on the target tables won't be included in the results.

- **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: This option is not supported for Firebird and InterBase.

▼ Resolution Options

Drop and Recreate Foreign Keys

All foreign keys are dropped prior to applying changes during the synchronization process. This option should be used with **Wrap Statements in a Transaction** to ensure that foreign keys do not remain dropped in the event that the synchronization script fails. This option is currently only supported by SQL Server.

Wrap Statements in a Transaction

The synchronization process utilizes a transaction and rolls back any repository changes if it encounters an error. This option is currently not supported by DB2.

Enable Identity Column Inserts and Updates

In SQL Server and Sybase turn this option on to enable changes to identity column values. Additional statements for each table with identity columns will be included in sync scripts to temporarily allow updates and insertions into these columns.

- **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 a 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.

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 row) are compared when the job executes to determine matches and mismatches between the two sources. Mapping details are viewed on the **Mapping** tab of the Editor.

The process automatically maps data source elements based on the interpretation of both schemas. Elements that do not match are excluded from the job when it executes, unless manually paired prior to running the job.

The **Database Mapping** table displays the paired databases or owners on the **Source** or **Target**. The **% Mapped** column indicates how much of the database or owner (a percentage value of total tables and rows) is mapped. Unpaired databases appear in the table as well, but are excluded from the job unless manually mapped to a corresponding database.

Step 3: Pair databases, schemas, owners, tables, and columns between the source and target << back next >>

Database Mapping

	torlabsq00_2	TORLABSQL00_3	Last Inspected
<input checked="" type="checkbox"/>			
<input checked="" type="checkbox"/>	#cici#	[Unmapped]	
<input checked="" type="checkbox"/>	2005	[Unmapped]	
<input checked="" type="checkbox"/>	@	[Unmapped]	
<input checked="" type="checkbox"/>	a10	[Unmapped]	
<input checked="" type="checkbox"/>	a9	[Unmapped]	
<input checked="" type="checkbox"/>	ABC1	ABC1	
<input checked="" type="checkbox"/>	activation	activation	
<input checked="" type="checkbox"/>	AlexLevatoTest	[Unmapped]	
<input checked="" type="checkbox"/>	ANCAS	[Unmapped]	
<input checked="" type="checkbox"/>	AnnDh	fl Inmannedl	

Source: None Selected
Target: None Selected
Inspected: Inspect: [All](#) | [Current](#) | [Checked](#)
▶ [Refresh Databases](#)
▶ [Remap to Defaults](#)

Select: [All](#) , [None](#)

Table Mapping

Initially, Change Manager will automatically analyze and map objects based on your option selections. It pairs up matching objects in the table on the left-hand side of the screen. You can choose not to run a comparison job on a specific object or pair of objects by unchecking the box beside the appropriate objects in the left-hand column.

The **Last Inspected** column indicates if a specified pair has been analyzed by Change Manager. In order to view individual tables and view the mappings of database pairs, you need to initiate the analysis process by clicking an option in the **Inspected** section on the right-hand side of the page. **All** inspects all pairs in the table, **Current** inspects the current object mappings, and **Checked** inspects only those pairs you've specified in the table.

Once inspected, bars appear in the left-hand columns of the table that indicate how much of the selected objects matched. You can view the table mappings of a specific pair by clicking that row and choosing **Yes, retrieve the objects in this pair**. The **Table Mapping** table populates with the paired tables and views of the selected pair.

You can view the row mappings of a table or view by clicking **Edit** in the right-hand column of the appropriate row. The row pairings and corresponding values appear in a new window. Individual owners and database pairs can be examined by selecting the appropriate icon in the **Inspect Now** column.

To restore the Database Mapping table to its original settings, click **Remap to Defaults**. All settings revert to the original object mappings as determined by the job, based on the selected Mapping options.

Customize the Data Source Mapping Process

You can manually configure the way Change Manager maps data source elements and then run the process again. The automatic mapping options are specified on the **Options** tab. You can re-run the mapping process by navigating back to the **Mapping** tab and clicking **Refresh Database and Owners** once this task is completed.

▼ Mapping Options

Ignore Case
Databases, owners, tables, and columns, will be paired up for comparison without regards to case. For example, the table "Product_Prices" will be paired with the table "PRODUCT_PRICES" with this table selected.

Ignore Spaces
Pairs database objects for comparison ignoring spaces in their names. For example, with this option selected the table "Product Prices" will be paired with the table "ProductPrices".

Ignore Underscores
Pairs database objects for comparison ignoring underscores in their names. For example, with this option selected the table "Product_Prices" will be paired with the table "ProductPrices".

Include Views
Enable this option to include views in the mapping. Otherwise, views will be filtered out of the mapping and will not be compared. Note that with this option enabled, you may still select which views (if any) you wish to include in the comparison.

To configure the automatic mapping process:

- 1 Navigate to the **Options** tab and select or deselect the check boxes in the **Mapping Options** section, as needed:
 - **Ignore Case:** Pairs data source objects without regards to case. For example, the table "Product_Prices" is paired with the table "product_prices".
 - **Ignore Spaces:** Pairs data source objects and ignores extraneous spaces in object names. For example, the table "Product Prices" is paired with the table "ProductPrices".
 - **Ignore Underscores:** Pairs data source objects and ignores underscores in job names. For example, the table "Product_Prices" is paired with the table "Product Prices".
 - **Include Views:** Includes database views when performing the automatic mapping process. If this option is deselected (the default setting), views are filtered out and will not be compared. If this option is selected, you can still manually choose the views to include or exclude from the comparison.
- 2 Navigate back to the **Mapping** tab and click **Refresh Databases and Owners** to update the automatic comparison.

Customize Database Mapping

You can customize job database mapping parameters manually by choosing to exclude certain paired rows or redefining objects after the automatic mapping process is completed.

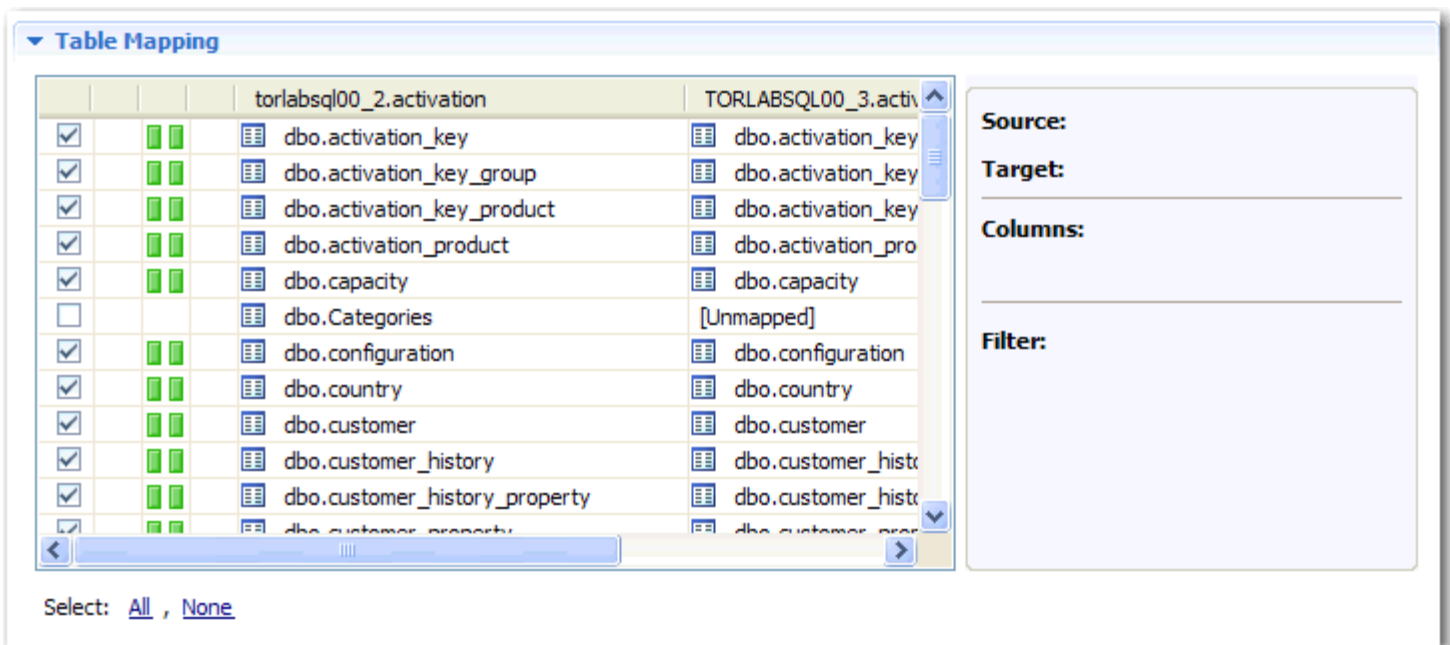
The check boxes beside each paired object displayed in the Database Mapping Source column indicate if a pair is included or excluded when the job runs.

Database pairings are redefined via the drop down menu that appears when the corresponding item in the **Target** data source column is selected. The menu contains a list of all databases or owners that reside on the **Target** data source. Databases or owners that are already paired appear in the drop down menu with an asterisk to denote them. Additionally, using the same method, unpaired databases can be assigned a matching database on the **Source**.

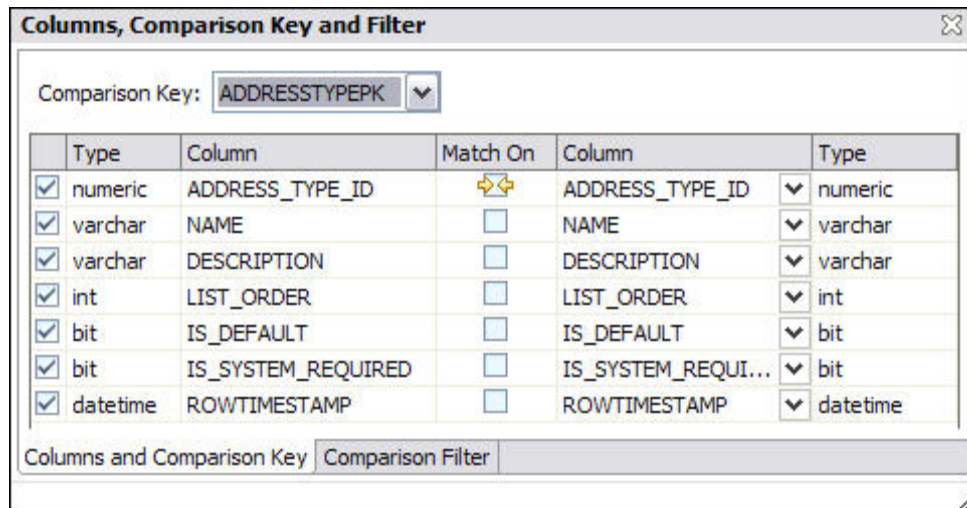
Customize Table Mapping

You can manually map databases, tables, and rows between the **Source** and **Target** data sources. Tables are displayed by name and contain mapping refinement column values and a comparison key value.

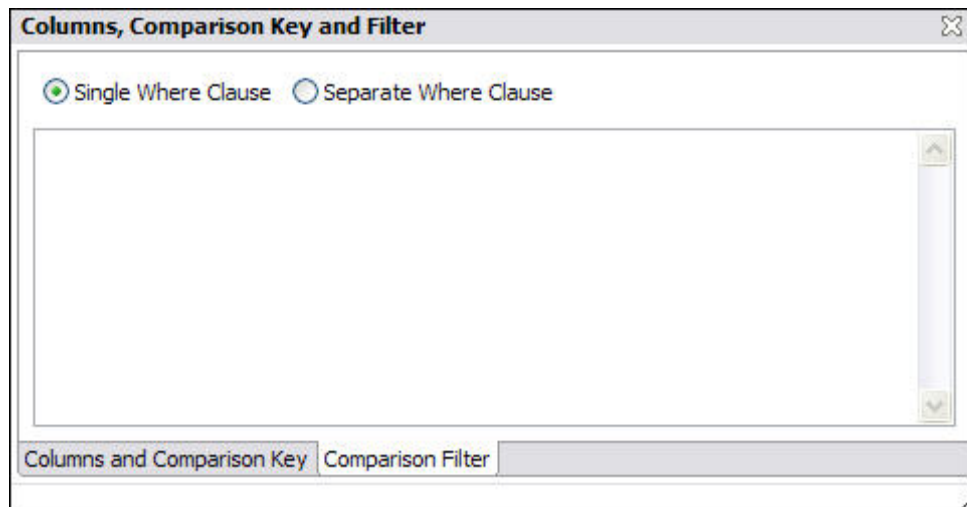
You can manually map unpaired tables between the databases by clicking the corresponding item in the **Target** data source column and define it via the drop down menu that appears, as with database mapping.



Additionally, if you click **View Column Mapping** in the appropriate selection, the **Columns**, **Comparison Key**, and **Filter** dialog enables you to further detail mapping refinements in terms of defining table roles in the execution of the job.



- The **Columns and Comparison Key** enables the mapping of columns to compare between the paired tables. Select or deselect the columns, as needed, in the dialog provided, and use the drop down menu to change the comparison key if multiple keys are available.
- The **Comparison Filter** tab enables you to manually insert a WHERE clause for execution when the job runs. Click the ellipsis button and enter the WHERE clause into the text box provided. If you want to create a different WHERE clause per table in the pair, select the **Separate WHERE Clause** radio button. The dialog adds an additional text box for a second WHERE clause.



NOTE: When entering selective WHERE clauses, **do not include the WHERE keyword** (e.g. use “AGE>12” instead of “WHERE AGE>12”).

Set Job Notification Options

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

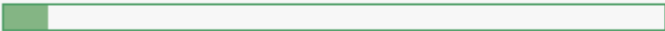
Run the Data Comparison Job

Click the **Run** icon in the upper right-hand corner of the Data Comparison Job Editor to run the job.





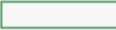
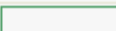
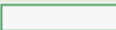
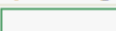
While the process runs, the **Results** tab appears and displays the progress of the job via the **Overall Progress** bar at the top of the tab.

Comparison Summary

Overall Progress
 7%

activation.property_classif - activation.property_classif
 Comparison Progress: 2,208 Rows
 Elapsed Time: 00:05 (mm:ss)

Comparison Results

torlabsql00_2	TORLABSQL00_3	Progress	Compare Index	Results	Resolution
		⚠ Nothing to Com...			
ABC1	ABC1	⚠ Nothing to Com...			
activation	activation	 69%			
capacity	capacity	 0%			
CASES	cases	⚠ Nothing to Com...			
claudiu	claudiu	 0%			
coco	coco	⚠ Nothing to Com...			
databasename_mo...	databasename_mo...	 0%			
master	master	 0%			
model	model	⚠ Nothing to Com...			
msdb	msdb	 0%			

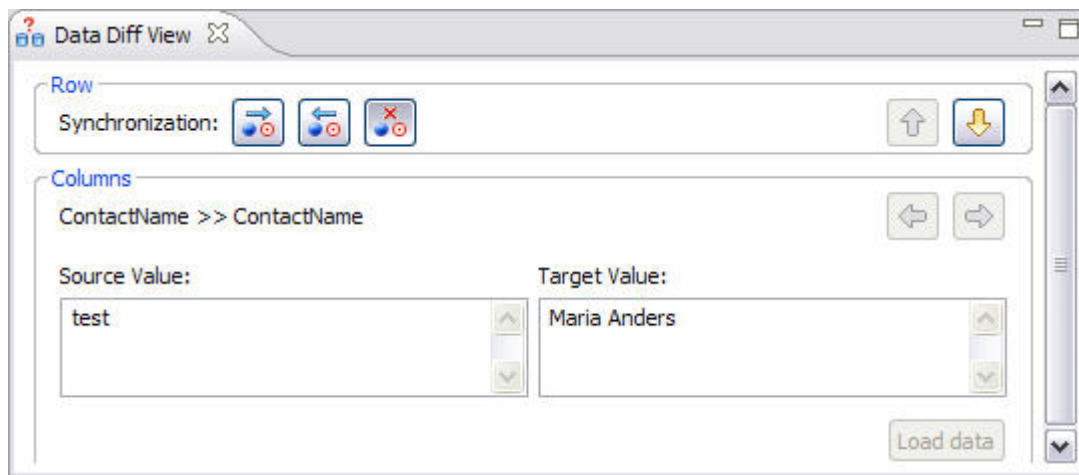
In the **Comparison Results** table, the database pairs are displayed along with **Progress**, **Compare Index**, **Results**, and **Resolution** columns.

- The **Progress** column indicates, per database pair, the progress of the job. When the job is finished comparing the database row, a result of **Done** (should the comparison be successful), **Failed** (should the comparison be unsuccessful), or **Nothing to Compare** (if the pairs were not specified) will display in the column.
- The **Compare Index** column indicates the percentage value of data per database that the job matched. Failed rows and rows in which the progress is not completed do not display results.

When the comparison process is complete, the **Results** and **Resolution** columns populate with data, as appropriate to the matched pairings.

- The **Results** column provides a link in appropriate pairings that enables you to view details about the comparison job. Click **View Results** to open the **Database Results** tab. For more information, see [Synchronizing Data Repositories](#).

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.



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 Change Manager generates UPDATE, INSERT, and DELETE statements for both databases. Then you run the scripts to synchronize the data.

- [Selecting Data to Resolve](#)
- [Running the SQL Synchronization Script](#)

Selecting Data to Resolve

On the **Results** tab, click **View** in the row you want to resolve. The **Database Results** tab appears.

Results Overview - activation

Source: activ...	Target: activation	Matched	Only in torlabsql...	Only in TORLAB...	Different	Source 1
dbo.activation_key	dbo.activation_key	0	0	0	0	0
dbo.activation_k...	dbo.activation_k...	0	0	0	0	0
dbo.activation_k...	dbo.activation_k...	0	0	0	0	0
dbo.activation_...	dbo.activation_...	0	0	0	0	0
dbo.capacity	dbo.capacity	18	0	0	0	0

Selected Table

0/0

0/0

Select All

Show rows: All Mismatched Source Only source Target Only target

The table in the **Results Overview - Activation** section displays the pairings in the selected tables.

- The **Matched** column indicates how much of the specified pairings matched between the source and target.
- The **Only in...** columns indicate where discrepancies between the two pairings are in terms of the source and target tables.

- The **Different** column indicates the number of differences Change Manager found between the pairings on the source and target.

When you select a pair from the **Database Results** table, the **Selected Table** section populates with information regarding the specified pair.

Selected Table

torlabsql00_2.activation.dbo.machine_mac_address

...	machine_id	mac_address_id
	1,000	1,002
	1,006	1,001
	1,006	1,003
	1,017	1,001
	1,017	1,010

→

←

✕

TORLABSQL00_3.activation.dbo.machine_mac_address

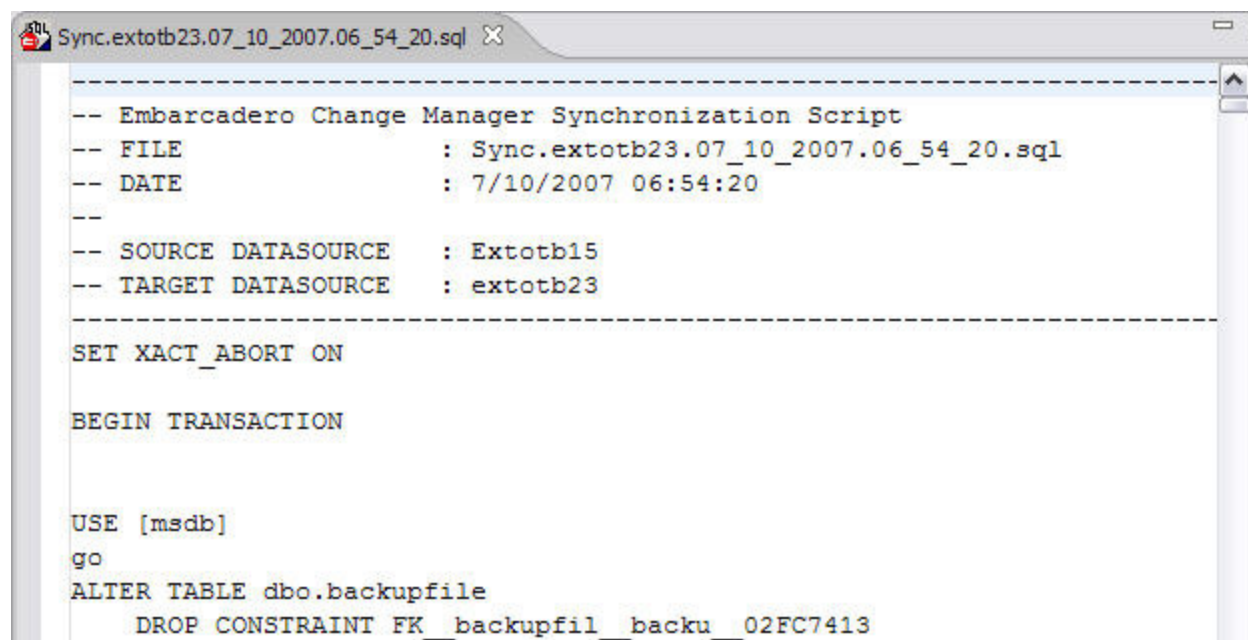
...	machine_id	mac_address_id
	1,017	1,000
	1,017	1,002

[Select All](#)

Show rows: All Mismatched Source Only source Target Only target

Specify the type of **Resolution** by clicking the **All**, **Mismatched**, **Source**, **Only Source**, **Target**, or **Only Target** icons, respectively, and then click **Generate a SQL Script**. The script appears in **SQL Editor**, and can be modified manually, as needed.

The saved script can be executed immediately, or saved and run at any time to synchronize both databases based on the results of the current data comparison job.



```

Sync.extotb23.07_10_2007.06_54_20.sql X
-----
-- Embarcadero Change Manager Synchronization Script
-- FILE           : Sync.extotb23.07_10_2007.06_54_20.sql
-- DATE           : 7/10/2007 06:54:20
--
-- SOURCE DATASOURCE : Extotb15
-- TARGET DATASOURCE : extotb23
-----

SET XACT_ABORT ON

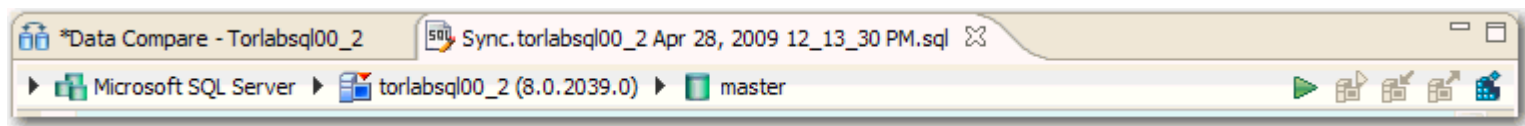
BEGIN TRANSACTION

USE [msdb]
go
ALTER TABLE dbo.backupfile
    DROP CONSTRAINT FK_backupfil__backu__02FC7413

```

Running the SQL Synchronization Script

Synchronization scripts are executed via the commands in the **Run** menu or via the command icons located along the top 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 code:

Open the SQL file you want to run, ensure it is associated with the correct database and click the **Execute** icon. 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. Change Manager executes the transaction on the data source you specified.

To commit a transaction:

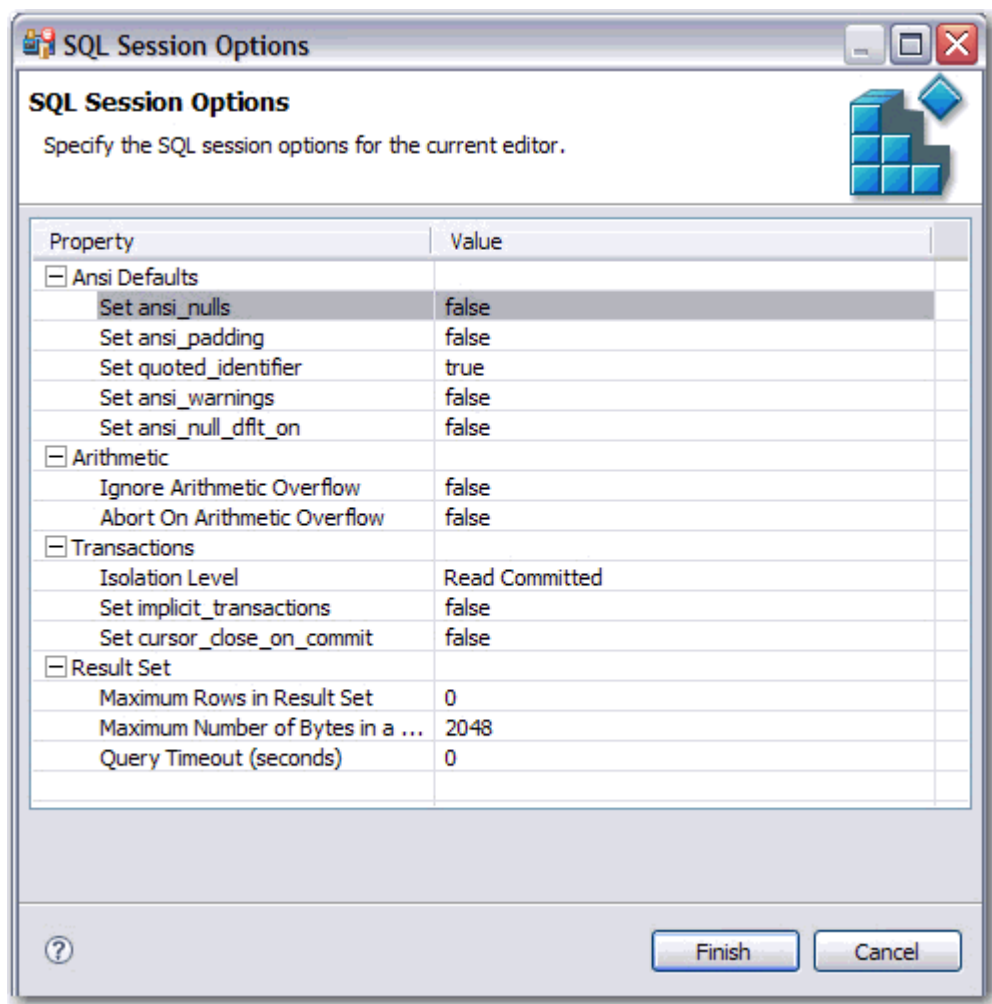
Open the transaction file you want to commit, ensure it is associated with the correct database, and then click the **Commit Transaction** icon. Change Manager commits the transaction on the data source you specified.

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**.

Configure an SQL Session

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



To modify SQL session options:

- 1 Click the SQL Session Options icon in the Toolbar.
The **SQL Session Options** dialog appears.
- 2 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 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 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 enable you to compare the configuration settings between two registered data sources. They are used to ensure consistency between the settings of two data sources in order to meet guidelines regarding the storage and retention of data.

Comparison job results are viewed as an overall “percentage matched” between two data sources. Comparison jobs also provides a list of individual configuration properties and other, more granular details regarding both data source’s configuration settings. Additionally, the job determines the total properties matched, total properties that do not match, the properties available on only one data source, and the overall total properties compared in the job.

In addition to comparing two registered data sources, you can compare a Configuration Standard, a customized list of configuration parameters, values, and comparison operators, against a data source. See:

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

You can also compare a data source or Standard against an Archive, the older configuration properties of an existing registered data source. 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 Source and Target Data Sources](#)
- 4 [Specify Job Options](#)
- 5 [Refine Property Comparisons](#)
- 6 Optionally request email, logs, or system tray notifications; see [Setting Job Notifications](#)
- 7 [Run the Comparison Job](#)
- 8 [View the Results of the Comparison](#)
- 9 Optionally generate a job report; see [Create a Job Report](#)
- 10 Optionally [Synchronize Configurations](#)

All comparison jobs, archives, and standards provide version information with regards to each data source platform. See:

- [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.

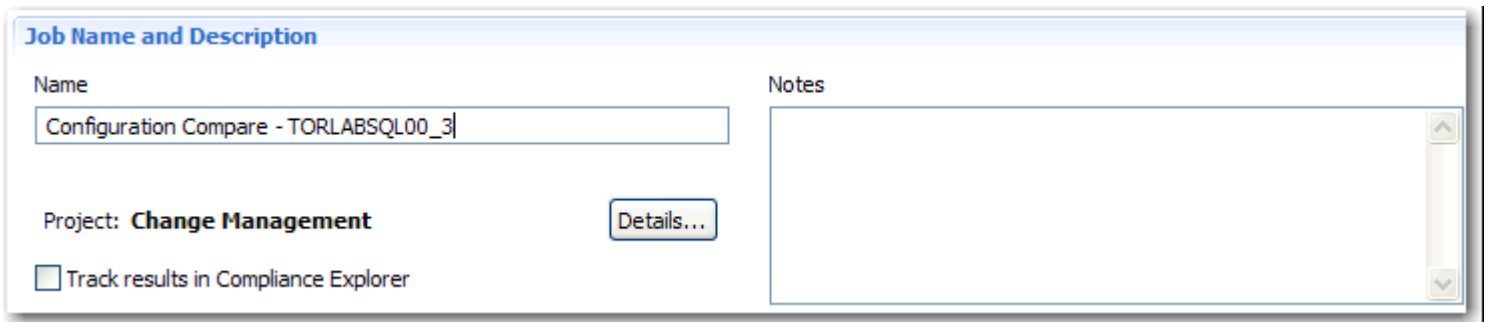
To open an existing configuration comparison job:

- Select **File > Open Job**. Select a configuration comparison job from the list and 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 screenshot shows a dialog box titled "Job Name and Description". It has two main sections: "Name" and "Notes". The "Name" section contains a text input field with the text "Configuration Compare - TORLABSQL00_3". Below the input field, it says "Project: **Change Management**". To the right of the project name is a button labeled "Details...". Below the project name is a checkbox labeled "Track results in Compliance Explorer", which is currently unchecked. The "Notes" section is a large text area on the right side of the dialog, currently empty. The dialog box has a light blue header and a white body with a thin border.

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 Source and Target Data Sources

The **Job Sources** section of the Configuration Comparison Job Editor contains two boxes labeled **Comparison Source** and **Comparison Target**. These boxes identify the data sources, archives, and standards to be used in the job.

The screenshot shows the 'Job Sources' window with two main configuration boxes:

- Configuration Comparison Source:**
 - Name: TORLABSQL00_3
 - Type: Microsoft SQL Server
 - Host: TORLABSQL00_3
 - Buttons: [Change Datasource](#) (with a green checkmark icon) and [Select Standard](#) (with a clipboard icon).
 - Radio buttons: Use Live Sources, Use an Archive.
 - Two empty text input fields with dropdown arrows and a three-dot menu icon.
- Configuration Comparison Target:**
 - Name: torlabsql00_2
 - Type: Microsoft SQL Server
 - Host: torlabsql00_2
 - Buttons: [Change Datasource](#) (with a green checkmark icon).
 - Radio buttons: Use Live Sources, Use an Archive.
 - Two empty text input fields with dropdown arrows and a three-dot menu icon.

At the bottom right of the window is a button with a hand icon and the text [Add More Targets](#).

To add a data source to the job:

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

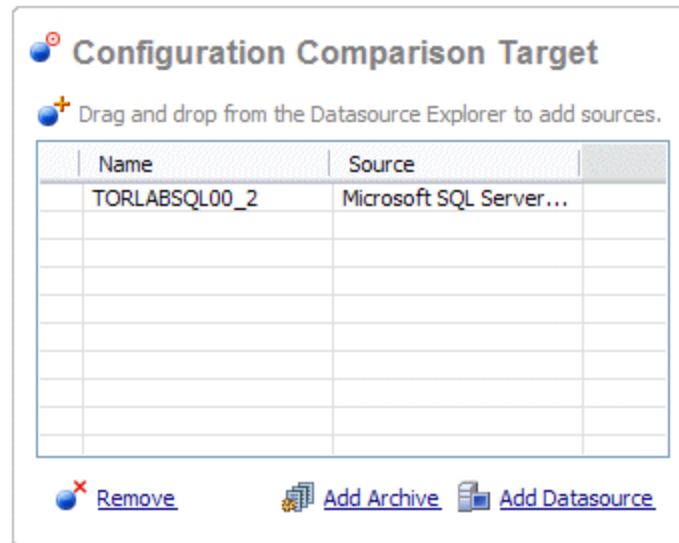
To add a Standard to the job:

- Click and drag a Standard from **Data Source Explorer** to the **Comparison Source** box, or click **Select Standard** and choose a data source from the dialog provided.

When a data source is added to a job, it is displayed by **Name**, **Type**, and **Host**. You can change a data source by clicking **Change Data Source** in the appropriate box and selecting a new data source from the dialog, or by dragging a different data source to the box from data source Explorer.

The **Configuration Comparison Source** and **Configuration Comparison Target** boxes must contain data sources, Archives, or Standards before the job can be executed.

To add multiple Comparison Targets, click **Add More Targets** to enable a **One-to-Many** comparison.



To add multiple targets to the comparison job

- 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 provided.

Additionally, you can specify whether you want to compare the current properties of the live data source, or an archived set of properties previously stored in Change Manager using the radio buttons beneath each box.

- 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 Archive and version to use in the job.

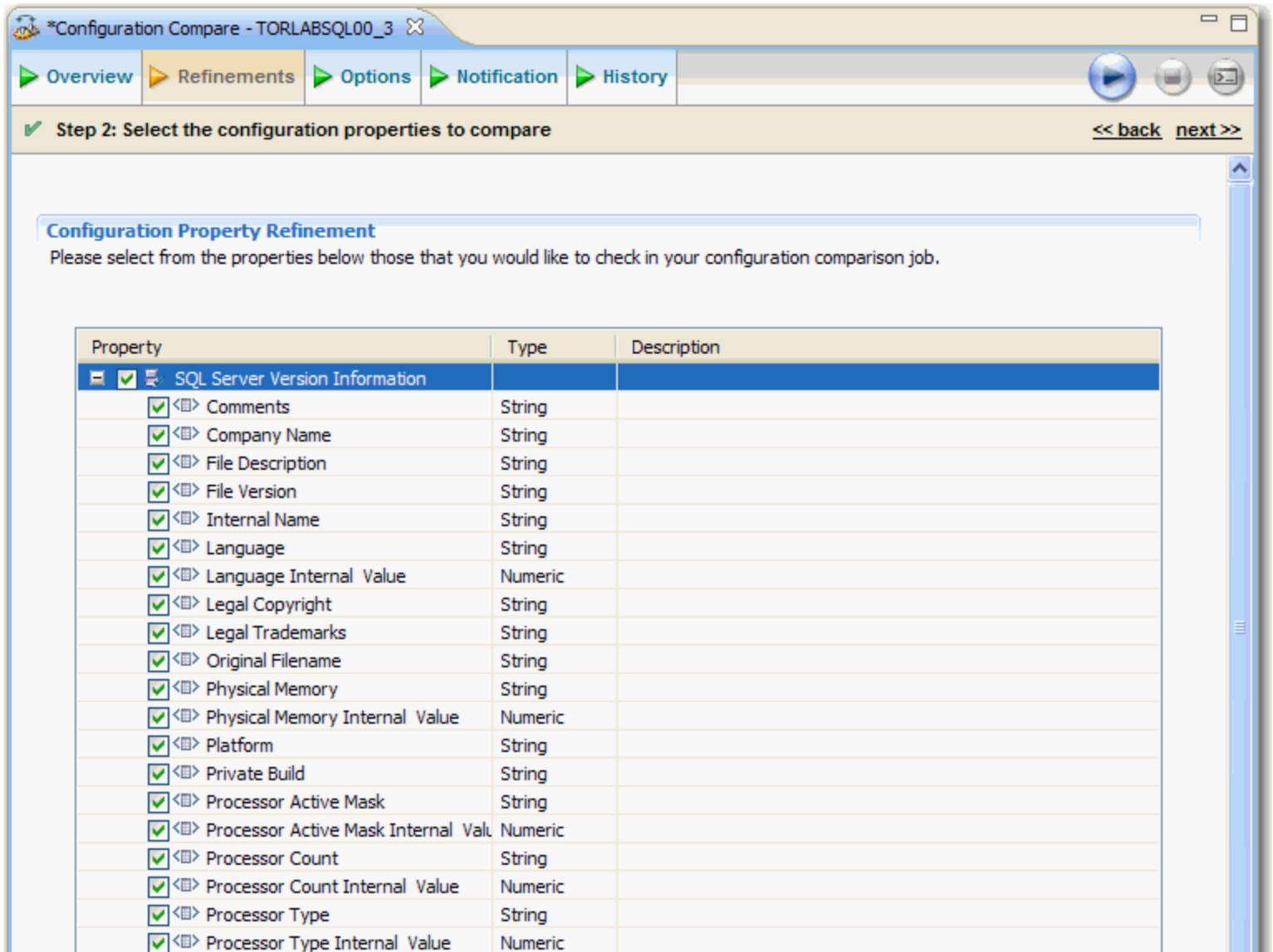
NOTE: You can drag and drop data sources, Archive Jobs, Archive Versions, and Standards into the Comparison Source and Comparison Target cells. A Standard can only be the Source of a Configuration Comparison Job.

Refine Property Comparisons

Configuration property refinement involves choosing specific configuration properties to compare and specific configuration properties to omit prior to executing a comparison job.

By default, when you specify a data source as a comparison job Source or Target, you can refine the properties to be compared prior to executing the job on the **Refinements** tab of the **Configuration Comparison Job Editor**.

Configuration properties are organized by tree and listed by **Property** name, **Description**, **Data Source Location**, and **Source**.



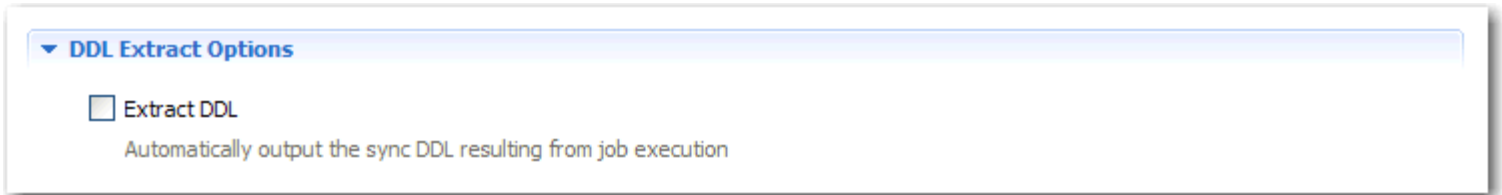
Property	Type	Description
<input checked="" type="checkbox"/> <input type="checkbox"/> SQL Server Version Information		
<input checked="" type="checkbox"/> <input type="checkbox"/> Comments	String	
<input checked="" type="checkbox"/> <input type="checkbox"/> Company Name	String	
<input checked="" type="checkbox"/> <input type="checkbox"/> File Description	String	
<input checked="" type="checkbox"/> <input type="checkbox"/> File Version	String	
<input checked="" type="checkbox"/> <input type="checkbox"/> Internal Name	String	
<input checked="" type="checkbox"/> <input type="checkbox"/> Language	String	
<input checked="" type="checkbox"/> <input type="checkbox"/> Language Internal Value	Numeric	
<input checked="" type="checkbox"/> <input type="checkbox"/> Legal Copyright	String	
<input checked="" type="checkbox"/> <input type="checkbox"/> Legal Trademarks	String	
<input checked="" type="checkbox"/> <input type="checkbox"/> Original Filename	String	
<input checked="" type="checkbox"/> <input type="checkbox"/> Physical Memory	String	
<input checked="" type="checkbox"/> <input type="checkbox"/> Physical Memory Internal Value	Numeric	
<input checked="" type="checkbox"/> <input type="checkbox"/> Platform	String	
<input checked="" type="checkbox"/> <input type="checkbox"/> Private Build	String	
<input checked="" type="checkbox"/> <input type="checkbox"/> Processor Active Mask	String	
<input checked="" type="checkbox"/> <input type="checkbox"/> Processor Active Mask Internal Value	Numeric	
<input checked="" type="checkbox"/> <input type="checkbox"/> Processor Count	String	
<input checked="" type="checkbox"/> <input type="checkbox"/> Processor Count Internal Value	Numeric	
<input checked="" type="checkbox"/> <input type="checkbox"/> Processor Type	String	
<input checked="" type="checkbox"/> <input type="checkbox"/> Processor Type Internal Value	Numeric	

To specify the properties used in the comparison job:

Navigate to the **Refinements** tab and select or deselect the configuration properties in the table, as required. Selected properties are included in the comparison job, and de-selected properties are ignored during job execution.

Specify Job Options

Prior to the execution of a job, parameters can be set to control aspects of the execution process itself. These options can be set prior to job execution on the **Options** tab.



The **Extract DDL** option enables you to indicate to Change Manager that the synchronization code produced by the job comparison is automatically output to a separate Editor window once the comparison process has executed.

Additionally, there are default commands located in the upper-right hand corner of the interface:

- Select **Make These the Defaults for New Jobs** to save your settings and retain job option selections whenever you create a new job.
- Select **Restore Defaults** to return the option page selections to the original selection format.

Set Job Notification Options

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

Run the Comparison Job

The configuration comparison job runs when you click **Run Job** on any tab in the Configuration Comparison Job Editor.



While the process runs, the **Comparison Results** tab appears and displays the progress of the job and a summary of the comparison job results.

View the Results of the Comparison

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

The screenshot shows the 'Configuration Compare - TORLABSQL00_3' application window. The 'Comparison Results' tab is active. The interface includes a navigation bar with tabs for Overview, Refinements, Options, Notification, History, and Comparison Results. A 'Close results' link is in the top right. Below the navigation bar, there are two main sections: 'Comparison Summary' and 'Resolution'. The 'Comparison Summary' section shows the source as 'TORLABSQL00_3' and the target as 'torlabsql00_2'. It displays three rows of summary data: 50 Properties Matched (76.9%), 15 Properties Did Not Match (23.1%), and 0 Properties Not Found (0.0%). The 'Resolution' section has a 'Generate a SQL Script' link. Below these sections are 'View Report' and 'Export Results...' buttons. The 'Comparison Results' section contains a table with columns: Property, Source, Op, and torlabsql00_2. The table is expanded to show 'Core SQL Properties' with various settings like 'user options', 'user connections', etc., and their corresponding values and match status.

Comparison Summary

Source: TORLABSQL00_3 Target: torlabsql00_2

✓ 50 Properties Matched 76.9%

✗ 15 Properties Did Not Match 23.1%

⚠ 0 Properties Not Found 0.0%

[Close results](#)

[Generate a SQL Script](#)

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

Comparison Results

Property	Source	Op	torlabsql00_2
Summary			
Not Found			⚠ 0.0%
Matched			✓ 76.9%
Did Not Match			✗ 23.1%
Core SQL Properties			
user options	0	EQUALS	✓ 0
user connections	0	EQUALS	✓ 0
two digit year cutoff	2049	EQUALS	✓ 2049
show advanced options	1	EQUALS	✓ 1
set working set size	0	EQUALS	✗ 1
scan for startup procs	0	EQUALS	✓ 0
remote query timeout (s)	600	EQUALS	✓ 600
remote proc trans	0	EQUALS	✓ 0
remote login timeout (s)	20	EQUALS	✓ 20
remote access	1	EQUALS	✓ 1
recovery interval (min)	567	EQUALS	✗ 0
query wait (s)	-1	EQUALS	✓ -1
query governor cost limit	0	EQUALS	✓ 0
priority boost	0	EQUALS	✗ 1
open objects	0	EQUALS	✓ 0
network packet size (B)	4096	EQUALS	✓ 4096

Job results are displayed by property with an overview of the matching and non-matching properties located at the top of the page. Individual details about each property are listed in the **Comparison Results** table.

- The column on the right-hand side of the table contains a check mark or X icon depending on whether the corresponding property value matched between the **Source** and **Target** data sources. It also displays values that represent the differences between the source and target configuration values.
- 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. (This can only be specified when a **Standard** is used in the job. Otherwise, the operator is always **equals**.)

Create a Job Report

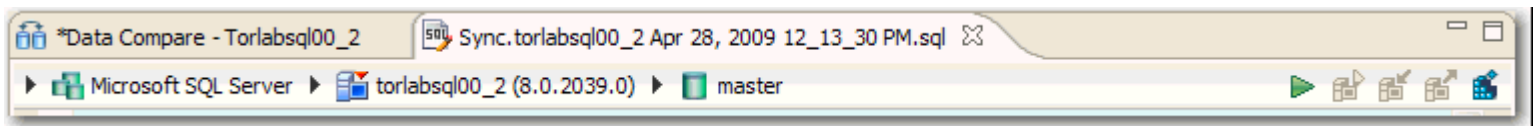
After running a comparison job, you can create a job report using the **View Report** button on the **Results** tab. See [Creating a Job Report](#) for more information.

Synchronize Configurations

When you are ready to resolve the data sources, click **Generate a SQL Script**. A synchronization script based on your modifications appears in a separate window.

Execute an SQL Synchronization Script

Synchronization scripts are executed via the commands in the **Run** menu or via the command icons located along the top 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 code:

Open the SQL file you want to run, ensure it is associated with the correct database and click the **Execute** icon. 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. 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. Change Manager commits the transaction on the data source you specified.

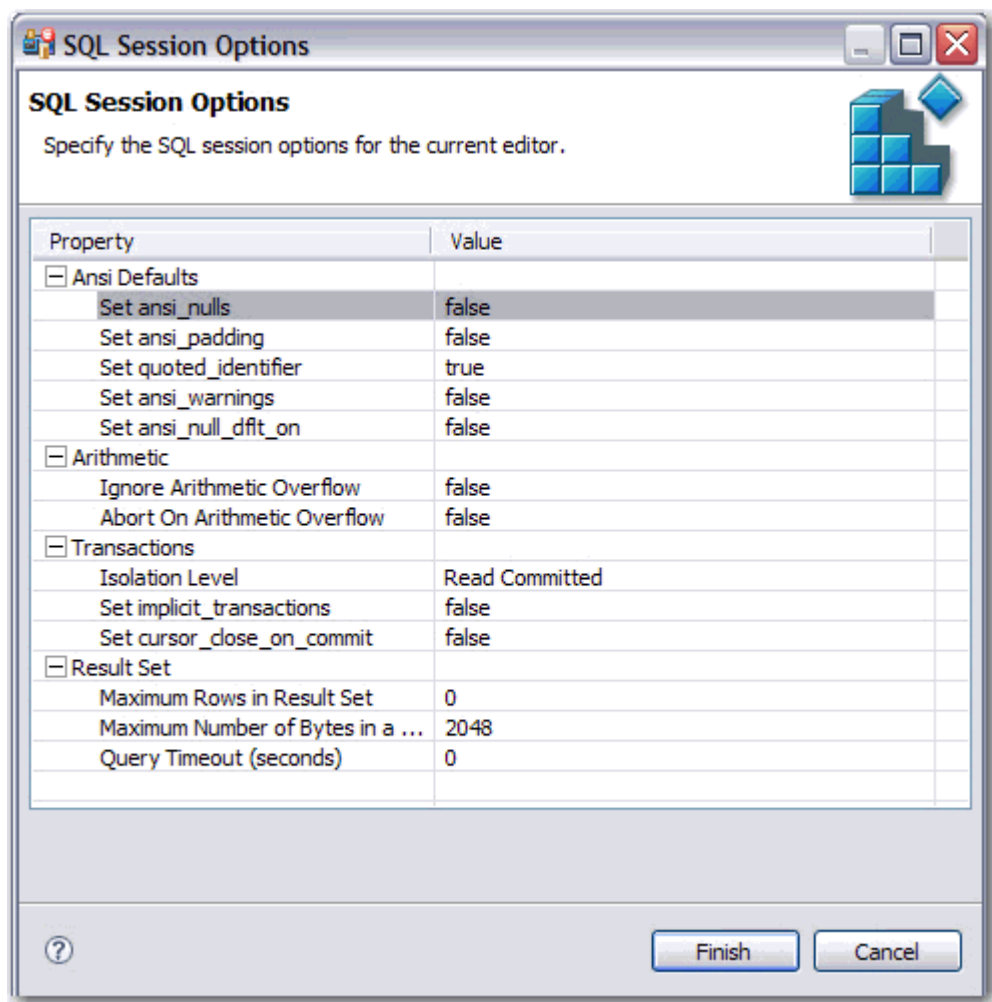
You can set transactions to auto-commit prior to execution on the **SQL Execution** node of the **Preferences** panel.

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**.

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

Configure an SQL Session

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



To modify SQL session options:

- 1 Click the SQL Session Options icon in the Toolbar.

The **SQL Session Options** dialog appears.

- 2 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 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.

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 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:

- Select **File > Open Standard**. Select a job from the list and click **OK**. The Configuration Standard Editor opens.

Describe and Save the Standard

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.

Standard Name and Description

<p>Name</p> <input style="width: 95%; border: 1px solid #ccc;" type="text" value="Untitled-Standard"/>	<p>Notes</p> <div style="border: 1px solid #ccc; height: 50px;"></div>
<p>Project: Change Management</p>	<input type="button" value="Details..."/>


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 new Standard properties will be derived.

Standard Sources

 Drag and drop from the Datasource Explorer to add a new source.

Source	Version	
TORLABSQL00_2	Microsoft SQL Server 8.0....	
Configuration Archive Job...	Microsoft SQL Server 7.0....	

Configuration Archive Job (Version 1)

Version: 1

Type: Linked

 [Add Datasource](#)
 [Add Archive Version](#)
 [Add Standard](#)

To add properties to a standard:

- 1 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.

The screenshot shows a software window titled '*Untitled-Standard' with two tabs: 'Overview' and 'Refinement'. The 'Refinement' tab is active, displaying a section titled 'Configuration Property Refinement'. At the top right of this section, there are radio buttons for 'Show: All Properties' (selected) and 'Selected Properties'. Below this is a table with four columns: 'Property', 'Source', 'Operator', and 'Value'. The 'Property' column has a tree view icon and a checkmark icon. The table lists 17 properties under the 'Core SQL Properties' group, all with 'torlabsql00_2' as the source and 'EQUALS' as the operator. The 'Value' column is currently empty for all entries.

Property	Source	Operator	Value
Core SQL Properties			
affinity mask	torlabsql00_2	EQUALS	
allow updates	torlabsql00_2	EQUALS	
awe enabled	torlabsql00_2	EQUALS	
c2 audit mode	torlabsql00_2	EQUALS	
cost threshold for parallel	torlabsql00_2	EQUALS	
Cross DB Ownership Chai	torlabsql00_2	EQUALS	
cursor threshold	torlabsql00_2	EQUALS	
default full-text language	torlabsql00_2	EQUALS	
default language	torlabsql00_2	EQUALS	
fill factor (%)	torlabsql00_2	EQUALS	
index create memory (KB)	torlabsql00_2	EQUALS	
lightweight pooling	torlabsql00_2	EQUALS	
locks	torlabsql00_2	EQUALS	
max degree of parallelism	torlabsql00_2	EQUALS	
max server memory (MB)	torlabsql00_2	EQUALS	
max text repl size (B)	torlabsql00_2	EQUALS	

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

Use a Configuration Standard as a Source in configuration comparison jobs, the same way as registered data sources or Archives. You cannot use it as a 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 Configuration Archive:

- 1 [Create or Open a Configuration Archive Job](#)
- 2 [Describe and Save the Archive Job](#)
- 3 [Specify an Archive Source](#)
- 4 [Refine Properties to be Archived](#)
- 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 a 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 Editor](#) in the [User Interface Reference](#).

To create a new configuration archive:

- Select **File > New > Configuration Archive Job**. The Configuration Archive Editor opens.

To open an existing configuration archive:

- Select **File > Open Archive**. Select a job from the list and click **OK**. The Configuration Archive Editor opens.

Describe and Save the Archive 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.

The screenshot shows a dialog box titled "Archive Name and Description". It has two main sections: "Name" and "Notes". The "Name" section contains a text input field with the text "Untitled-Config-Archive-Job-1". Below the input field, it says "Project: Change Management". To the right of the "Name" section is a "Notes" section with a large, empty text area. A "Details..." button is located between the "Name" and "Notes" sections.

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 an Archive Source

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

The screenshot shows a dialog box titled "Configuration Archive Source". It contains the following information:

- Name:** Datoedwina
- Type:** Microsoft SQL Server
- Host:** Datoedwina1

 At the bottom right of the dialog, there is a button labeled "Change Data Source".

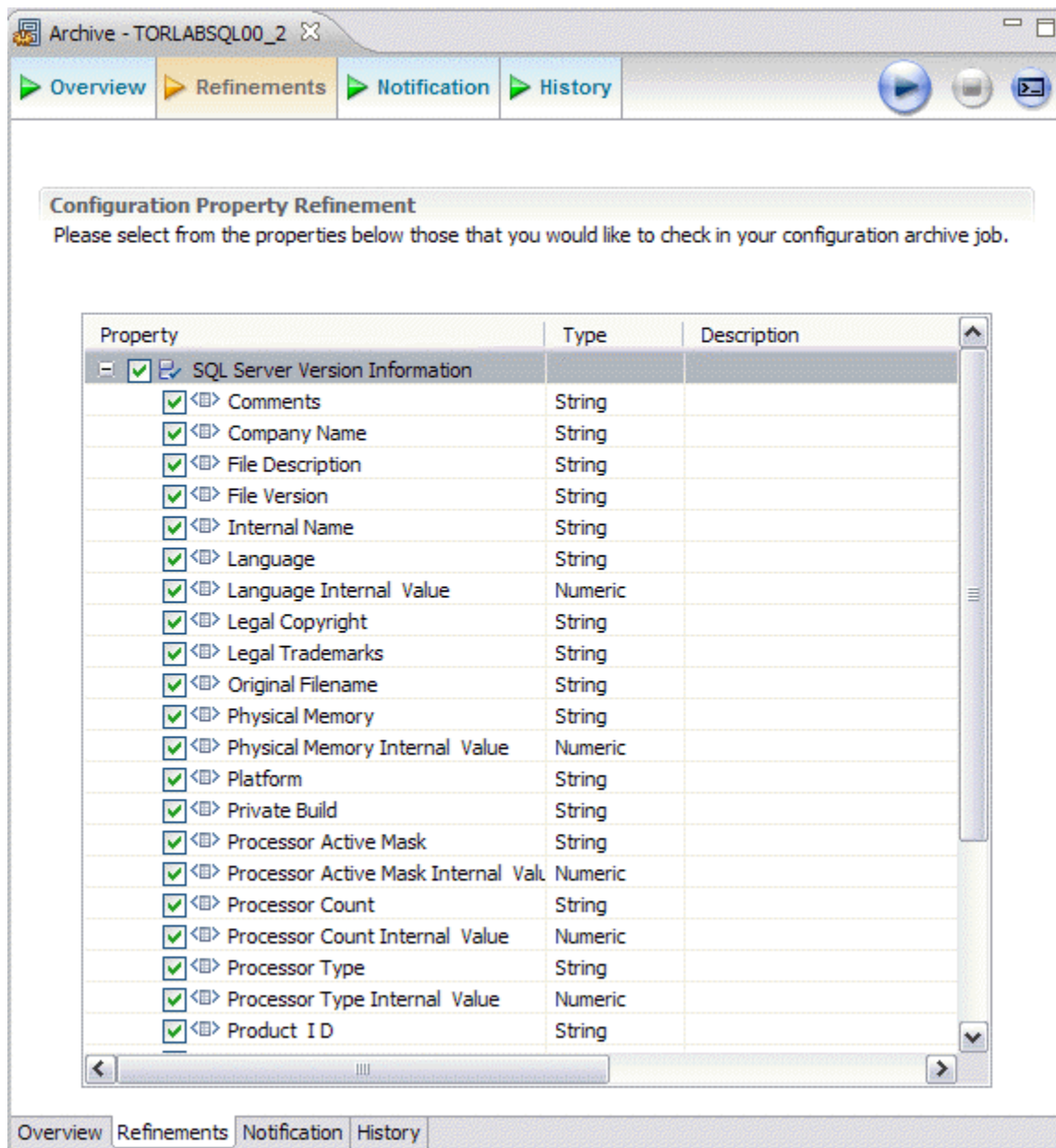
To specify a configuration archive source:

- 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, DBMS type, and host.

Refine Properties to be Archived

By default, when you create a new archive, all data source properties are included. However, you may choose to archive 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 located on the **Refinements** tab of the **Archive Job Editor**, where they are listed by name.



To refine configuration archive properties:

- Use the check boxes to select or deselect properties. Selected properties are included in the archive when comparison jobs are run, while deselected properties are ignored.

When you have finished selecting the properties to include in the archive, you will need to save the archive by creating a version. Within the same archive, properties are retained across versions unless you specify new values.

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

Set Job Notification Options

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

Create a Version and Save the Archive Job

A version stores older snapshots of the same archive for backup or tracking comparison purposes. The original archive job creates the first version. Versions are listed in the **Archive Job History** table on the **History** tab and in the Data Source Explorer under the **Configuration Archive Jobs** node.

To create a version and save a configuration archive job:

- 1 Click **Run Job** in the lower right-hand corner of any tab in the **Archive Job Editor**.

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.

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. Versions are listed on the **Overview** tab.

The screenshot displays the 'Archive Job Editor' window for 'Archive - TORLABSQL00_2'. The interface is split into two main panes. The left pane, 'Data Source Explorer', shows a tree view of 'Managed Data Sources' including 'Datoedwina (SQL Server)' and 'TORLABSQL00_2 (SQL Server)'. Under 'TORLABSQL00_2', there are 'Configuration Archive Jobs' (with 'Archive - TORLABSQL00_2' expanded to show 'Version 2' and 'Version 1') and 'Configuration Comparison Jobs', 'Data Comparison Jobs', 'Schema Archive Jobs', and 'Schema Comparison Jobs'. The right pane, 'Archive Job Editor', has tabs for 'Overview', 'Refinements', 'Notification', and 'History'. The 'History' tab is active, showing the 'Archive Job History' table. The table has columns for 'Version', 'Properties', 'Run', and a status column. It lists two versions: 'Version 2' with 65 properties, run on 7-Jul-2008 at 4:52:38 PM, and 'Version 1' with 65 properties, run on 7-Jul-2008 at 4:52:21 PM.

Version	Properties	Run	
Version 2	65	7-Jul-2008 4:52:38 PM	W
Version 1	65	7-Jul-2008 4:52:21 PM	W

To open a version:

- Double-click the version in the **Overview** tab or in the Data Source Explorer.

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

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.

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 **Comparison Source** or **Comparison 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 Job](#)
- 3 [Specify a Comparison Source and Targets](#)
- 4 [Refine Schema Comparisons](#)
- 5 [Customize Schema Mapping](#)
- 6 [Specify 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 data comparison job:

- Select **File > New > Schema Comparison Job**. The **Data Comparison Job Editor** opens.

To open an existing data 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 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.

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 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.

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


If you attempt to compare two data sources that are not compatible, the Schema Comparison Job Editor will not enable you to place the inappropriate DBMS objects in the Source and Target panes.

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.

Additionally, 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 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.


Job Sources

 **Schema Comparison Source**

Name: **Datoedwina**

Type: Microsoft SQL Server

Host: Datoedwina1


 [Change Datasource](#)

Use Live Sources

Use an Archive

▼


▼ ⋮

 **Schema Comparison Target**

Name: **TORLABSQL00_2**

Type: Microsoft SQL Server

Host: TORLABSQL00_2

 [Change Datasource](#)

Use Live Sources

Use an Archive

▼

▼ ⋮

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:

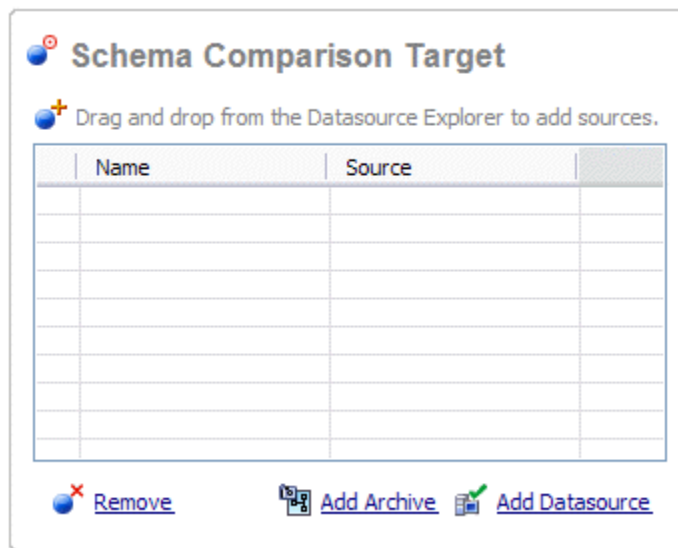
- Click and drag a schema archive from Data Source Explorer to the **Schema Comparison Source** or **Schema Comparison Target** box.

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.

Add Multiple Schema Comparison Job Targets

You can add more Targets to perform a One-to-Many schema comparison.



To add multiple targets to a schema comparison job:

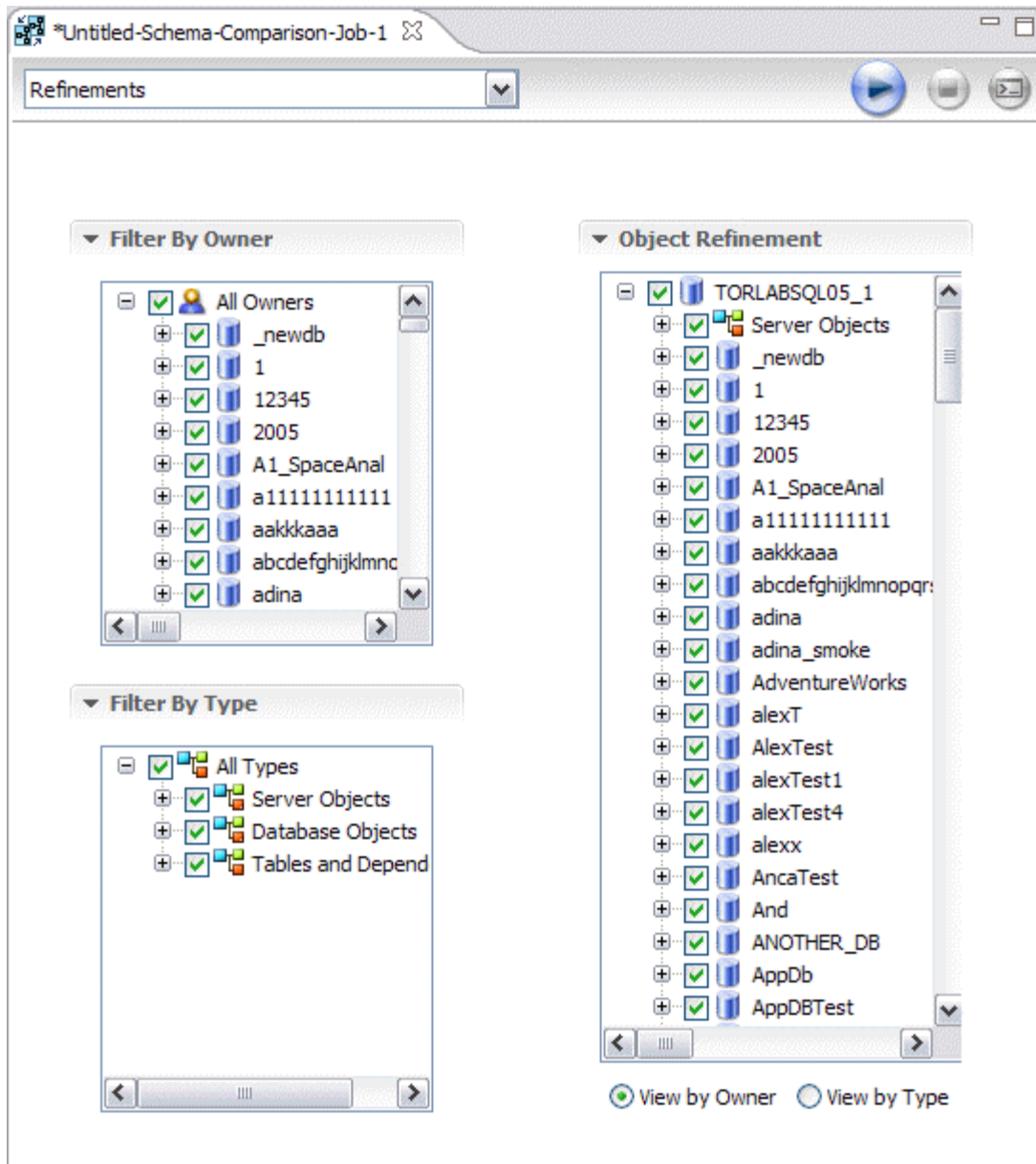
- Drag and drop additional data sources or schema archives from Data Source Explorer to the **Schema Comparison Target** box, or click the **Add Archive** or **Add Data Source** buttons beneath the table and select the appropriate target from the dialog box.

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 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. Selected objects will be included in the Schema Comparison Job, and deselected objects are ignored.

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 Editor.

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

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.

The screenshot shows the 'Refinements mapping' dialog box. At the top, there is a 'Select target:' dropdown menu set to 'TORLABSQL00_2'. To the right of this are three checkboxes: 'Ignore Case' (checked), 'Ignore Spaces' (checked), and 'Ignore Underscores' (unchecked). Below this is a table with two columns: 'Datoedwina' and 'TORLABSQL00_2'. The table lists various database objects. The 'Datoedwina' column contains object names with a plus sign icon to the left. The 'TORLABSQL00_2' column contains either the same object name with a plus sign icon (indicating a mapping) or '[Unmapped]'. Below the table are two buttons: 'Refresh Databases and Owners' (with a refresh icon) and 'Restore Default Mappings' (with a hand icon).

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 Change Manager.

Specify 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:

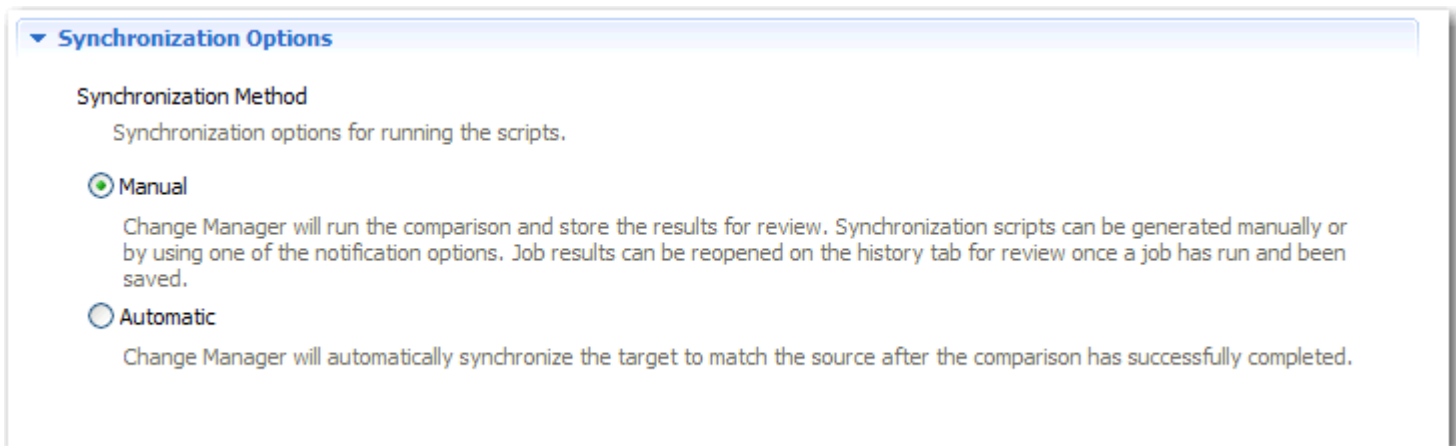
- [Synchronization Options](#)
- [Include Object Options](#)
- [Compare Options](#)
- [Identifier Options](#)
- [DDL Extract Options](#)

Options differ based 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.

Synchronization Options

Using the **Synchronization Options** pane, you specify how you want Change Manager to handle the synchronization script after it runs the comparison job.



- Select **Manual** if you want Change Manager to display the results of the comparison, but not automatically attempt to “fix” 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.
- Select **Automatic** if you want Change Manager to automatically generate and run a synchronization script immediately after the comparison job runs.

Compare Options

The **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.

▼ Compare Options

- Ignore Storage**
Ignore storage comparison
- Ignore Column Order**
Ignore table column order differences
- Ignore Text Case**
Ignore case in text object comparison -ie. view, procedure, trigger
- Ignore Text Comments**
Ignore comments in text object comparison (ie. procedures, triggers, views)
- Ignore Text Whitespace**
Ignore white space in text comparison
- Ignore Tablespace File Name**
Ignore tablespace file name
- Ignore Password Differences**
Ignore differences in object passwords (ie. group, login, role, user, etc.)
- Ignore object permissions**
Ignores the permissions on objects

Schema Comparison Options by Database Platform

Compare Options differ depending on the DBMS of the **Source** data source:

[Firebird and InterBase Schema Compare Options](#)

[IBM DB2 LUW Schema Compare Options](#)

[Oracle Schema Compare Options](#)

[SQL Server Schema Compare Options](#)

[Sybase Schema Compare Options](#)

Firebird and InterBase Schema Compare Options

Option	Description
Ignore Column Order	Ignores table column order differences when processing the job.
Ignore Text Case	Ignores syntax case in text object comparisons when processing the job. (views, procedures, trigger, etc.)
Ignore Text Comments	Ignores comments in text objects when processing the job.
Ignore Text Whitespace	Ignores white space when processing the job.
Ignore Password Differences	Ignores differences in object passwords when processing the job. (group, login, role, user, etc.)
Ignore Object Permissions	Ignores the permissions on objects when processing a job.

IBM DB2 LUW Schema Compare Options

Option	Description
Ignore Column Order	Ignores table column order differences when processing the job.
Ignore Check Constraint Quotes	Ignores quotes in check constraints when processing the job.
Ignore Table Comment	Ignores table comments when processing the job.
Ignore Data Capture	
Ignore Text Case	Ignores syntax case in text object comparisons when processing the job. (views, procedures, trigger, etc.)
Ignore Text Comments	Ignores comments in text objects when processing the job.
Ignore Text Whitespace	Ignores white space when processing the job.
Ignore Tablespace File Name	Ignores tablespace file names when processing the job.
Ignore Password Differences	Ignores differences in object passwords when processing the job. (group, login, role, user, etc.)
Ignore Tablespace	Ignores tablespaces for table, index, cluster, etc.
Ignore Object Permissions	Ignores the permissions on objects when processing a job.

Oracle Schema Compare Options

Option	Description
Ignore Storage	Ignores storage comparison when processing the job.
Ignore Partition Differences	Ignores partition differences when processing the job.
Ignore Column Order	Ignores table column order differences when processing the job.
Ignore Check Constraints Quotes	Ignores quotes in check constraints when processing the job.
Ignore Table Comment	Ignores table comments when processing the job.
Ignore Text Case	Ignores syntax case in text object comparisons when processing the job. (views, procedures, trigger, etc.)
Ignore Text Comments	Ignores comments in text objects when processing the job.
Ignore Text Whitespace	Ignores white space when processing the job.
Ignore Tablespace File Name	Ignores tablespace file names when processing the job.
Ignore Password Differences	Ignores differences in object passwords when processing the job. (group, login, role, user, etc.)
Ignore Tablespace	Ignores tablespaces for table, index, cluster, etc.
Ignore Object Permissions	Ignores the permissions on objects when processing a job.
Ignore Logging/No Logging Differences	Ignores object differences where one is set to Logging and the other is set to No Logging.
Ignore Sequence Start Value	

SQL Server Schema Compare Options

Option	Description
Ignore Storage	Ignores storage comparison when processing the job.
Ignore Column Order	Ignores column order when processing the job.
Ignore Text Case	Ignores text case when processing the job.
Ignore Text Comments	Ignores text comments when processing the job.

Option	Description
Ignore Text Whitespace	Ignores text whitespace when processing the job.
Ignore Tablespace File Name	Ignores database file names when processing the job.
Ignore Password Differences	Ignores password differences when processing the job.
Ignore Object Permissions	Ignores object permissions when processing the job.
Ignore File Group	

Sybase Schema Compare Options

Option	Description
Ignore Storage	Ignores storage comparison when processing the job.
Ignore Column Order	Ignores table column order differences when processing the job.
Ignore Check Constraint Quotes	Ignores quotes in check constraints when processing the job.
Ignore Text Case	Ignores syntax case in text object comparisons when processing the job. (views, procedures, trigger, etc.)
Ignore Text Comments	Ignores comments in text objects when processing the job.
Ignore Text Whitespace	Ignores white space when processing the job.
Ignore Tablespace File Name	Ignores the physical device name for database devices when processing the job.
Ignore Password Differences	Ignores differences in object passwords when processing the job. (group, login, role, user, etc.)
Ignore Object Permissions	Ignores the permissions on objects when processing a job.
Ignore Segment	

Include Object 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 **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 **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.

▼ Include Objects Options

- Exist in Source Only (Create)**
Include objects that only exist on the source in the comparison results
- Exist in Source and Target and Don't Match (Alter)**
Include objects that differ in the comparison results
- Force Extended Alter**
Always drop and recreate objects that do not match instead of using an alter script
- Exist in Target Only (Drop)**
Include objects that only exist on the target in the comparison results
- Generate Data Copy DDL**
Generate data copy ddl on extended alters

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.
- The **Ignore Constraint Names** option specifies that differences in table constraint names will be ignored when doing comparisons.
- The **Ignore Specific Name** option ignores SQL specific names in procedures, functions, and methods.

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

▼ **Identifier Options**

Ignore Name Case
Ignore case in object name and column name comparison

Ignore Constraint Names
Ignore table constraint name differences

Ignore Specific Name
Ignore SQL specific name in procedures, functions, and methods

DDL Extract Options

The **DDL Extract Options** section contains the **Extract DDL** parameter. When it is checked, the synchronization code is automatically sent to a separate Editor window after the comparison job runs.

▼ **DDL Extract Options**

Extract DDL
Automatically output the sync DDL resulting from job execution

Set Job Notification Options

When a job completes, 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 Editor to run the job.



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

When the job is complete, the **Comparison Results** tab displays the schema pairs along with the Compare Index, Results, and Resolution values of the job.

- The **Compare Index** column indicates the percentage of schema that matched between the source and targets of the job.
- The **Results** column provides a link that enables you to drill down and view the results between the pairs matched in the job. Click **Show Individual Results** to open the **Individual Comparison Results** tab, which provides a more detailed summarization of the schema comparison. For more information on viewing individual results, see [Viewing 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.



Windows-style window titled "Untitled-Schema-Comparison-Job-4". The top navigation bar contains tabs: Overview, Refinements, Mapping, Options, Notification, History, and Comparison Results (which is the active tab). There are also navigation icons on the right: a play button, a refresh button, and a back button.

[Close results](#)

▼ **Comparison Summary**


Source: New data source
Target: TORLABDB282_1

Start Time: Apr 28, 2009 12:54:00 PM
Elapsed Time: 02:03 (mm:ss)

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

Comparison Results

Target	Task	Progress	Compare Index
TORLABDB282_1	Compare	Done with 0 errors	43% Match

 [Generate Sync Scripts for All Targets](#)

The bottom of the window features another set of navigation tabs: Overview, Refinements, Mapping, Options, Notification, History, and Comparison Results (which is currently selected).

Viewing Individual Schema Comparison Results

To see specific schema comparison results:

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

Individual Comparison Results

View Report Export Results... Generate a Sync Script

Compared Objects

Name	Status	Action
TORLABDB282_1	<input checked="" type="checkbox"/>	

Show/Hide Objects

<input checked="" type="checkbox"/> Creates	7
<input checked="" type="checkbox"/> Drops	0
<input checked="" type="checkbox"/> Alters	2
<input checked="" type="checkbox"/> Extended Alters	0
<input checked="" type="checkbox"/> Incompatible	0
<input checked="" type="checkbox"/> Match (Unchanged)	7
<input checked="" type="checkbox"/> Error	0
<input checked="" type="checkbox"/> Do Not Sync	0
Total Object Count: 16	

Object Details

Name:
Type:
Existence:
Action:

Synchronize this Object

Open change script

View Differences

View Details

Script Source DDL Target DDL

Overview Refinements Mapping Options Notification History Comparison Results Individual Comparison Results

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**.

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

Creating a Schema Archive

An 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 an Archive Job Source](#)
- 4 [Refine Properties](#)
- 5 [Specify Extraction Options](#)
- 6 Optionally request email, logs, or system tray notifications; see [Setting Job Notifications](#)
- 7 [Create a Version and Save the Archive Job](#)
- 8 [Compare Archive Contents and Extract DDL](#)
- 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 Editor**. For more details about the editor, see [Schema Archive Editor](#) in the [User Interface Reference](#).

To create a new schema archive:

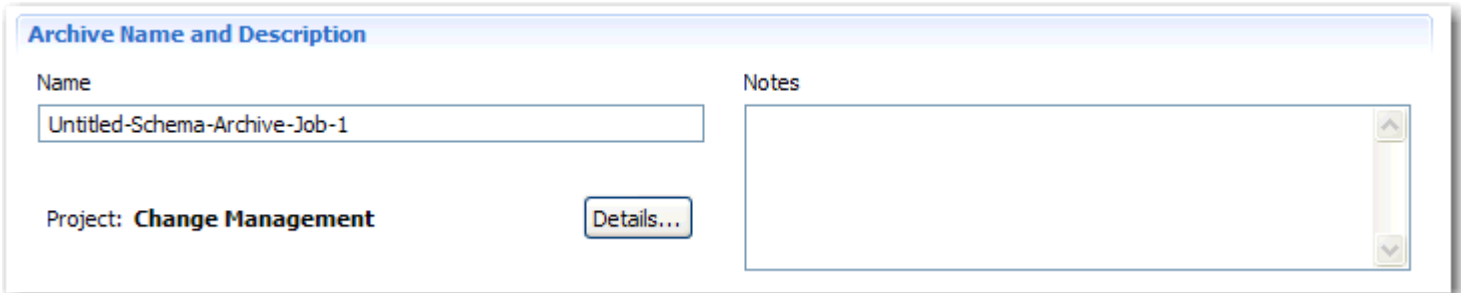
- Select **File > New > Schema Archive Job**. The **Schema Archive Editor** opens.

To open an existing schema archive:

- Select **File > Open Archive**. Select a job from the list and click **OK**. The **Schema Archive Editor** opens.

Describe and Save the Schema Archive 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



The screenshot shows a dialog box titled "Archive Name and Description". It features a "Name" text box containing "Untitled-Schema-Archive-Job-1" and a "Notes" text area. Below the "Name" field, it displays "Project: Change Management" and a "Details..." button. The "Notes" field has a vertical scrollbar on its right side.


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 an Archive Job Source

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


Archive Source

 **Schema Archive Source**

Name: **TORLABSQL00_2**

Type: Microsoft SQL Server

Host: TORLABSQL00_2

 [Change Data Source](#)

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, DBMS type, and host.

To refine schema archive properties:

- Use the check boxes to select or deselect properties. Selected properties are included in the archive when jobs are run, while deselected properties are ignored.

When you have finished selecting the properties to include in the archive, save the archive by creating a version. Within the same archive, properties are retained across versions unless you specify new values.

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

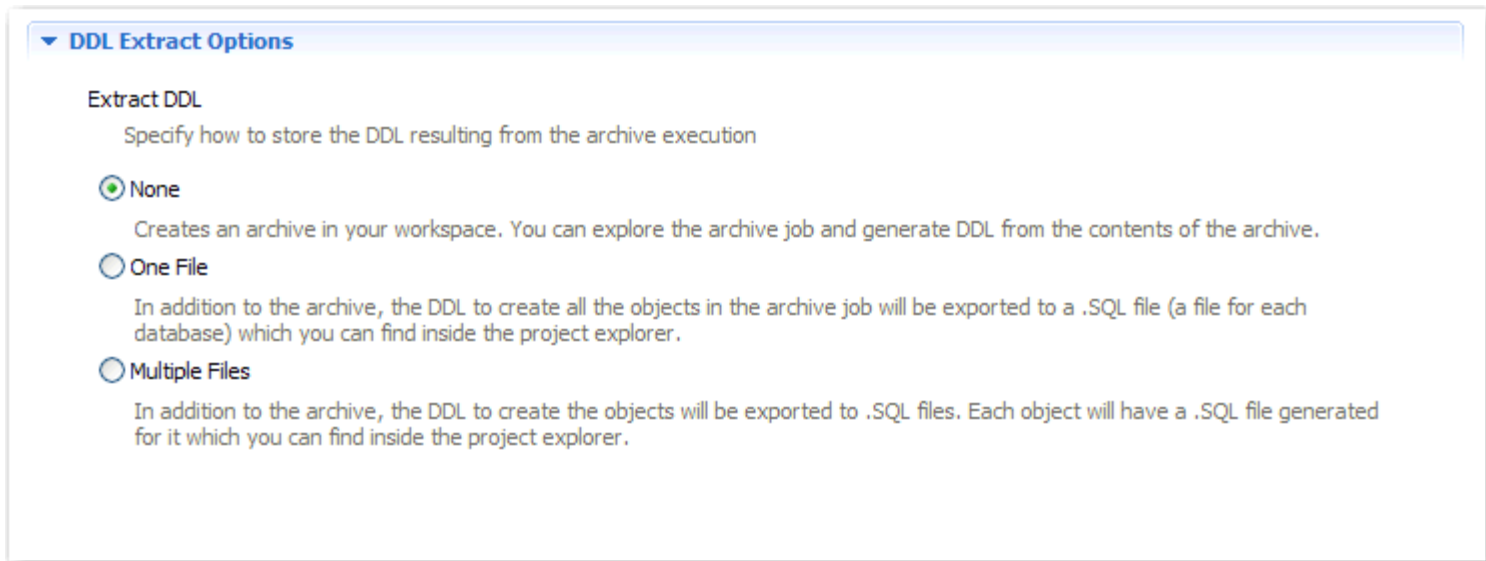
Set Job Notification Options

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

Specify Extraction Options

The **Options** tab contains the **DDL Extraction Options** setting, where you define how DDL is extracted and stored in Change Manager during the archive process. By default, this option is set to **None**, which means that the process does not retain extracted DDL when building the schema archive. You can get the DDL from the archive after the job runs.

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 define the DDL extraction process:

- Select **None**, **One File**, or **Multiple Files** to indicate how the DDL will be stored.

Create a Version and Save the Archive Job

A version stores older snapshots of the same archive for backup or tracking comparison purposes. The original archive job creates the first version. Versions are listed in the **Archive Job History** table on the **History** tab and in the **Data Source Explorer** under the **Schema Archive Jobs** node.

To create a version and save a schema archive job:

- 1 Click **Run Job** in the lower right-hand corner of any tab in the **Archive Job Editor**.

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.

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. Versions are listed on the **Overview** tab.

Compare Archive Contents and Extract DDL

To compare an archive version with the live source:

- On an open version tab, right-click on the **Archive Contents** table and select **Compare**.

The **Schema Comparison Job Editor** opens, and designates the archive version and the original source as its source and target data sources, respectively.

To extract the DDL of the archive contents:

- Select **Extract** from the **Archive Contents** menu.

The code opens in the **DDL Extract** window.

To see or edit the DDL:

- Click the **Open Editor** button.

Options to compare archive contents and extract DDL are also available in **Data Source Explorer**, under the appropriate archive node.

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.

Use an Archive in Schema Comparison Jobs

You can use a schema archive in schema comparison job.

To add a configuration archive to a schema comparison job:

- Drag and drop an archive version from **Data Source Explorer** to the **Comparison Source** or **Comparison Target** boxes in the **Schema 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 Schemas](#) for more information.

Synchronizing Schemas

This section contains these topics:

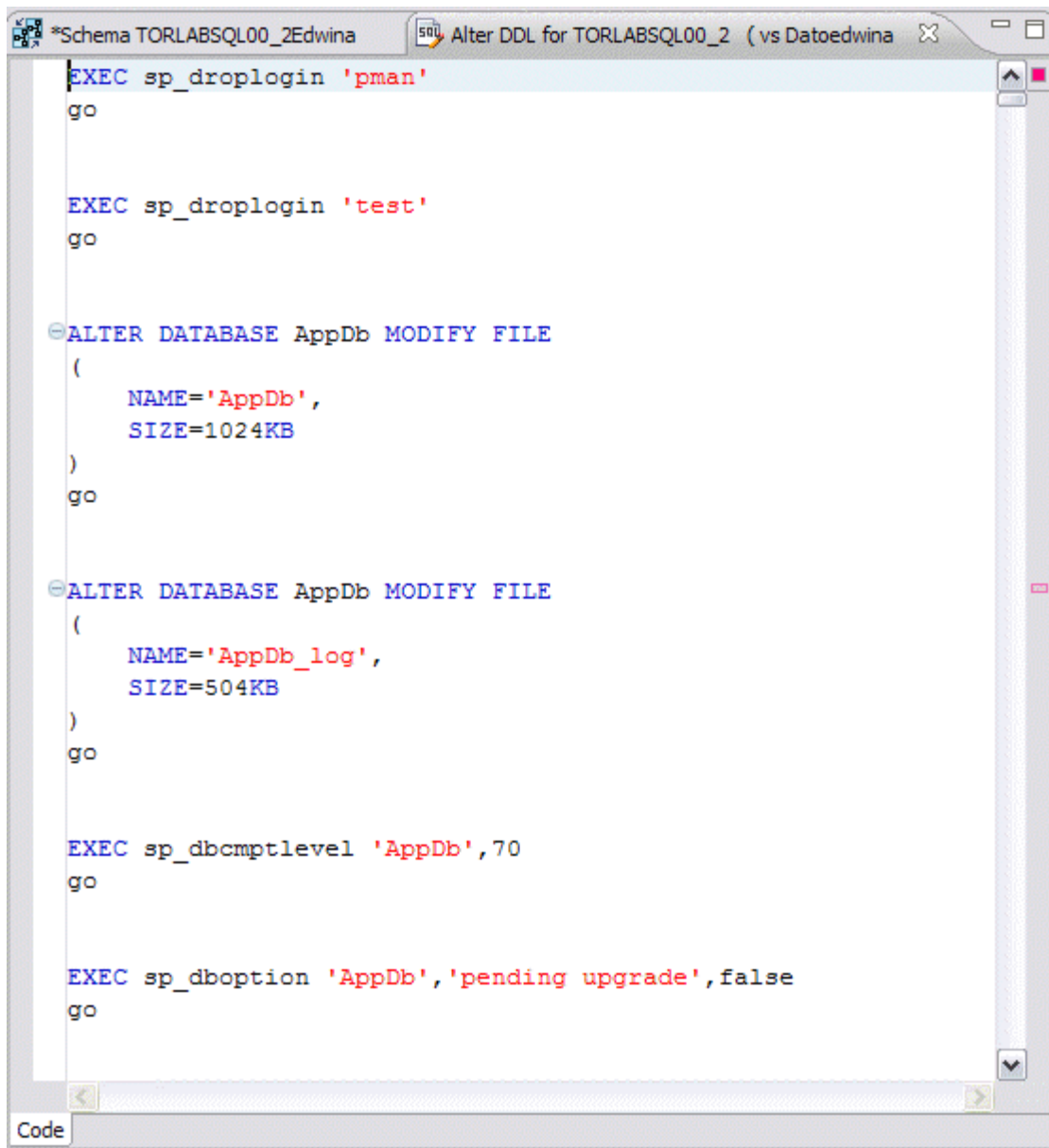
- [Resolve Schema via a Generated SQL Script](#)
- [Execute a SQL Synchronization Script](#)
- [Configure an SQL Session](#)

Resolve Schema via a Generated SQL Script

Once a schema comparison job has completed, synchronization commands are used to even the data so it matches. The synchronization process automates generation of CREATE, DROP, ALTER statements, and stored procedures on the source and target data sources, as required.

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.



```

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

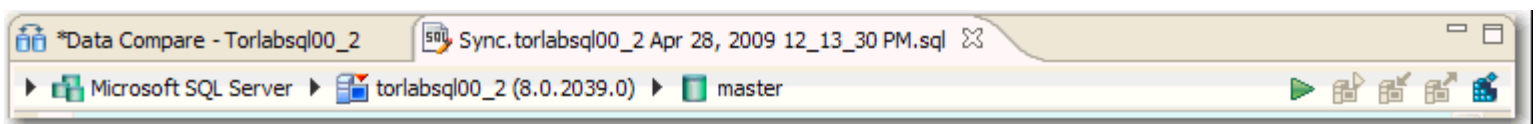
EXEC sp_dbcmplevel 'AppDb',70
go

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

```

Execute a SQL Synchronization Script

Synchronization scripts are executed via the commands in the **Run** menu or via the command icons located along the top 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 code:

Open the SQL file you want to run, ensure it is associated with the correct database and click the **Execute** icon. 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. 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. Change Manager commits the transaction on the data source you specified.

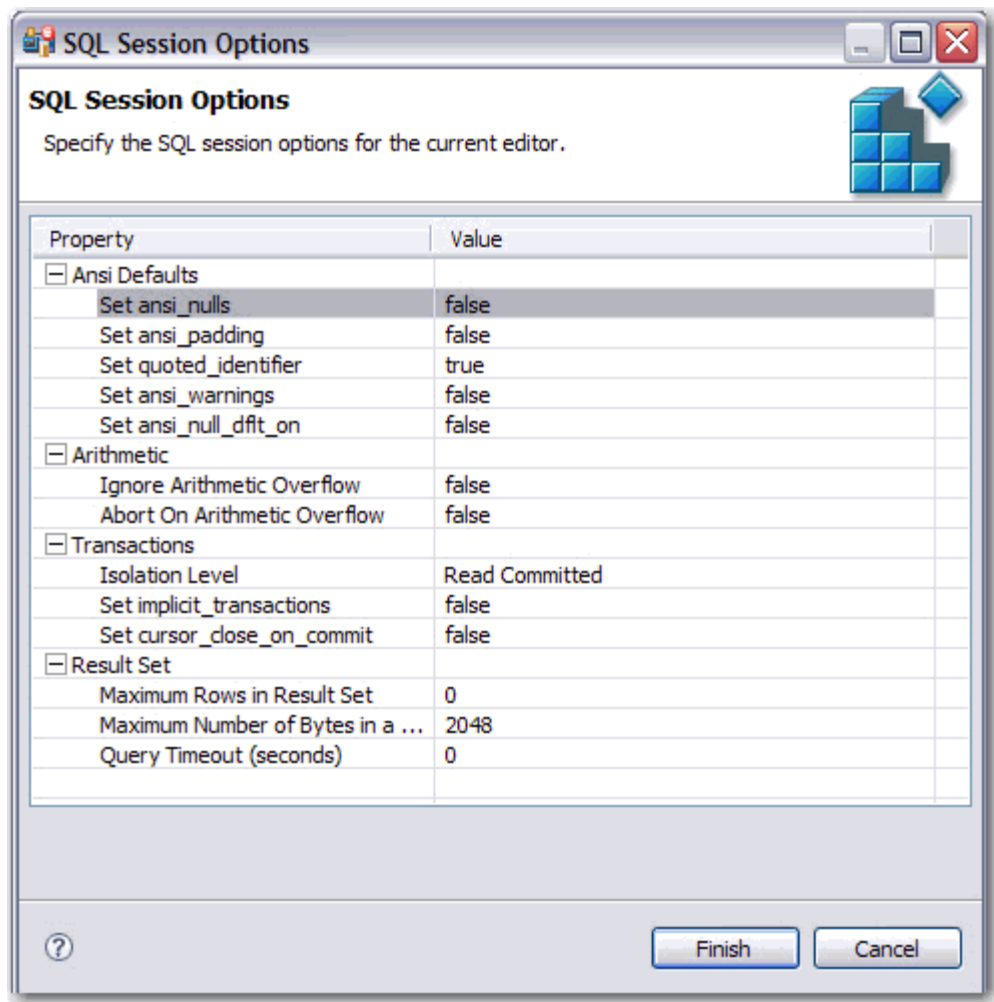
You can set transactions to auto-commit prior to execution on the **SQL Execution** node of the **Preferences** panel.

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**.

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 Change Manager how to execute code in the development environment.



To modify SQL session options:

- 1 Click the SQL Session Options icon in the Toolbar.

The **SQL Session Options** dialog appears.

- 2 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 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 **Comparison 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.

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

Untitled-Data-Comparison-Job-1

Notes

^
v

Project: **Change Management**

[Details...](#)

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:

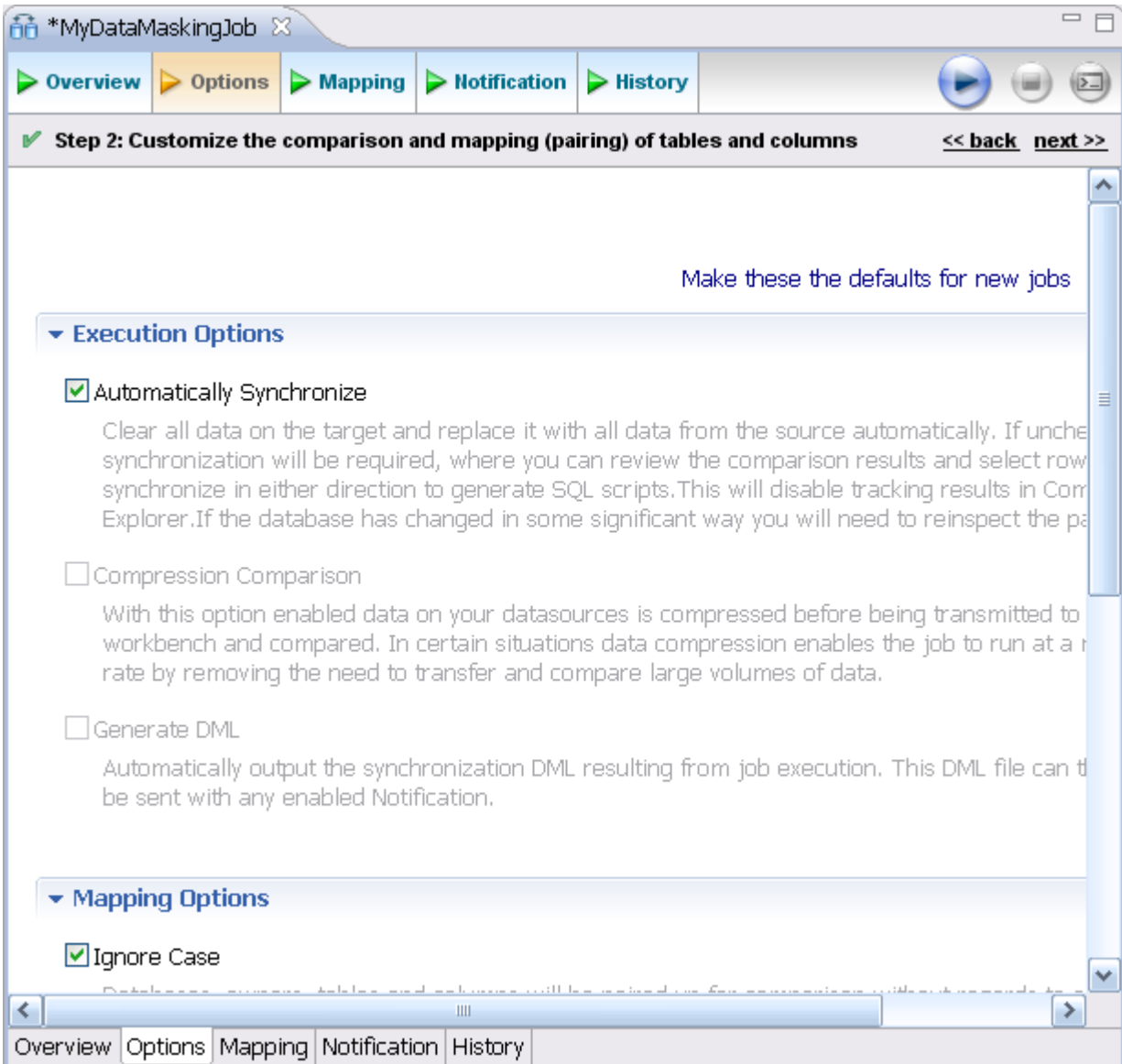
[Select Datasource](#)

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.

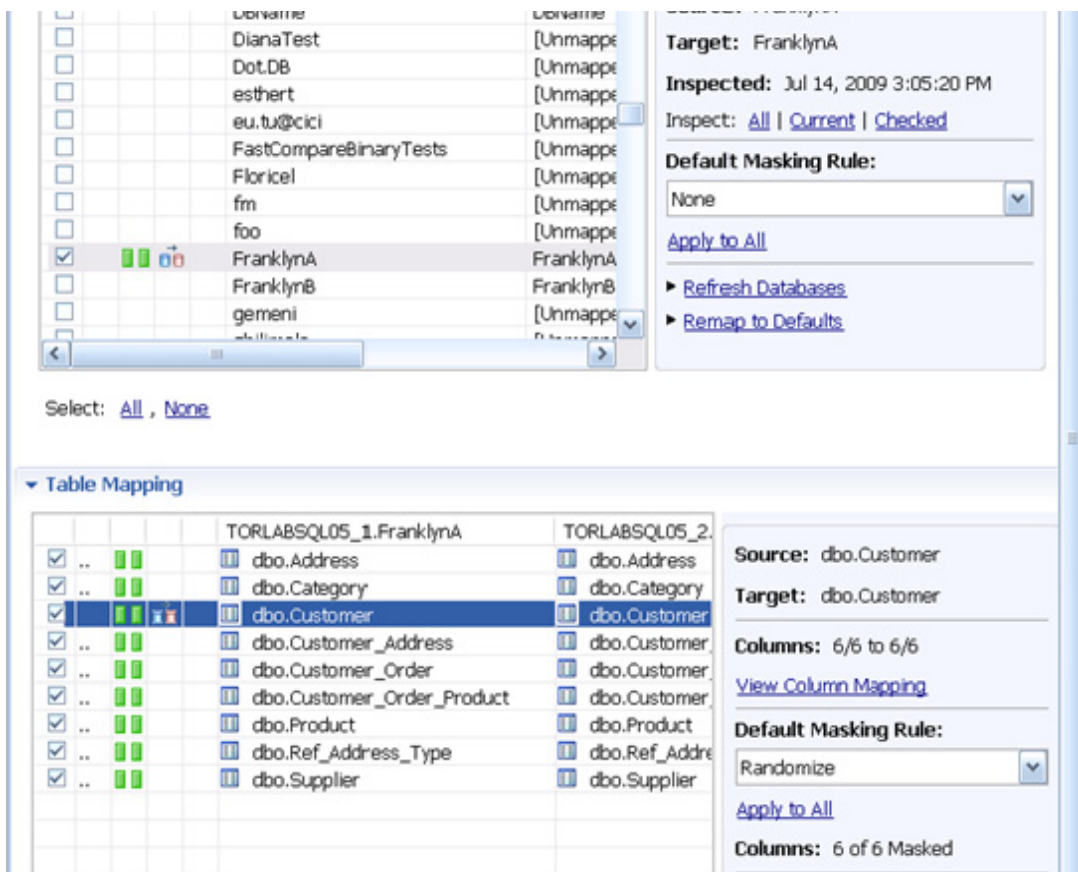
You can apply a **Default Masking Rule** to selected columns using the pulldown menus. The options are:

- **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.

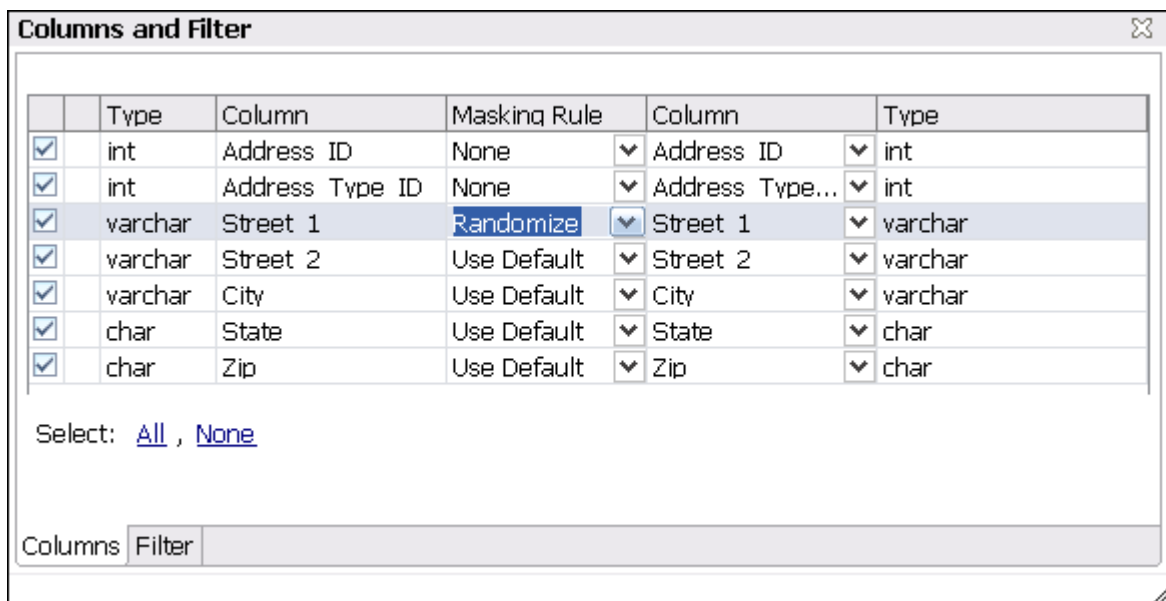


Specify Masking Rules for Individual Columns

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.

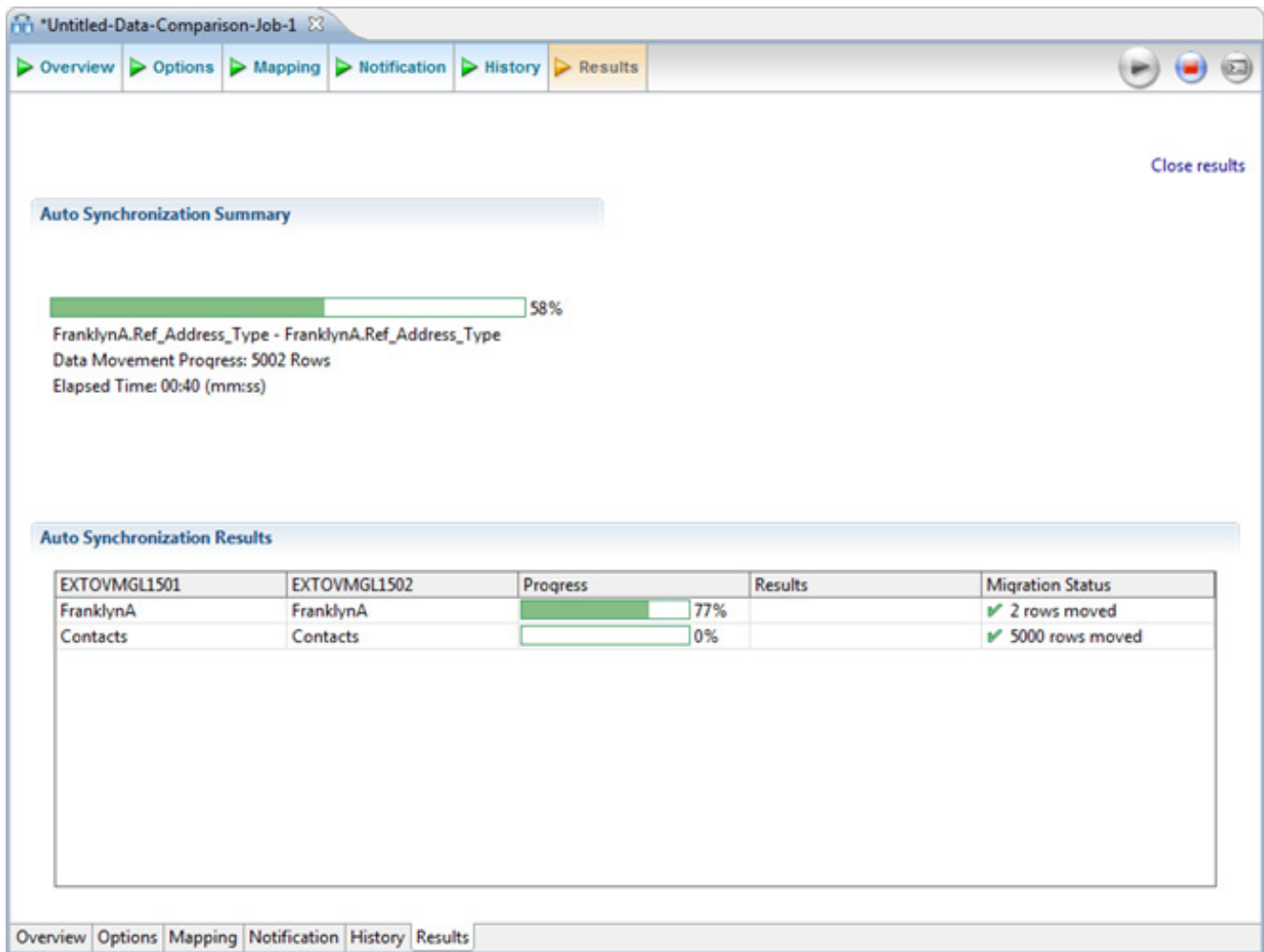


Run the Database Masking Job

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.



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 the comparison between source and target tables for 'dbo.Category' and 'dbo.Customer'. The 'Selected Table' section displays two side-by-side tables for 'TORLABSQL05_1.FranklynA.dbo.Customer' and 'TORLABSQL05_2.FranklynA.dbo.Customer', illustrating how names are randomized in the target table.

Results Overview - FranklynA

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

[View Diff Report](#) [Export Diff Results...](#)

Selected Table

...	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	Yxssfms
	735	Jaciqb	Fvwzfiq
	736	Nqel	Eaae
	737	Okeybx	Oaseqfw

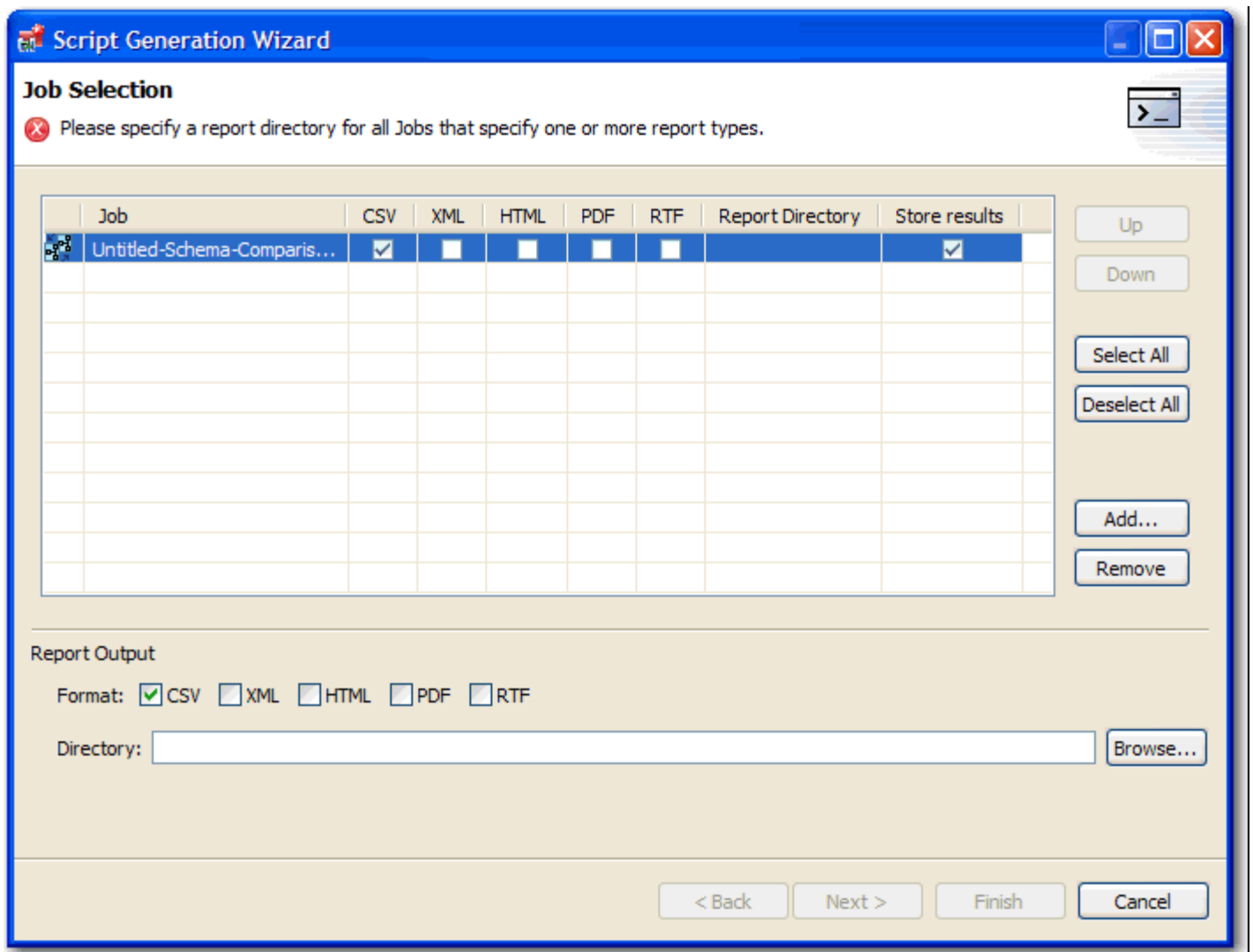
Automating Batch Scripts

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 Change Manager job scripts. It is located in the 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:

```
changecc -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:

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

To run a job from the command line:

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

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

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

Cheat Sheets

This section provides printable cheat sheets. Cheat sheets are available from the Workbench, and provide a series of step-by-step, task-based instructions on how to perform key tasks within the application.

In Change Manager, you can access cheat sheets by selecting **Help > Cheat Sheets**, and then selecting the cheat sheet you want to view.

This section of the help guide reproduces the cheat sheets provided in the application to enable hard copy printing and provide a higher level of user comprehension while working with Change Manager features and functionality.

The following cheat sheets are reproduced in this section:

Configuration Comparison Cheat Sheets

[Cheat Sheet 1A: Creating a New Configuration Comparison Job](#)

[Cheat Sheet 1B: Creating a New Configuration Archive](#)

[Cheat Sheet 1C: Creating a New Configuration Standard](#)

Data Comparison Cheat Sheets

[Cheat Sheet 2A: Creating a New Data Comparison Job](#)

Schema Comparison Cheat Sheets

[Cheat Sheet 3A: Creating a New Schema Comparison Job](#)

[Cheat Sheet 3B: Creating a New Schema Archive](#)

Cheat Sheet 1A: Creating a New Configuration Comparison

Job

Configuration comparison jobs enable you to compare the configuration settings between two registered data sources. They are used to ensure consistency between the configuration settings of one or more data sources, Standards, or Archives, in order to maintain database availability, security, and performance and to troubleshoot problems in these areas.

Use the table below to review the task and its individual steps prior to creating a new configuration comparison job:

1. Create a New Job
2. Specify a Job Name
3. Specify Source and Target Data Sources
4. Specify Job Options
5. Refine Property Comparisons
6. Execute Comparison Job
7. View Results
8. Execute an SQL Synchronization Script
9. Generate a Job Report

1. Create a New Job

Configuration comparison jobs are defined in the **Configuration Comparison Editor**.

To begin a new comparison:

- Select **File>New>Configuration Comparison Job**. The Configuration Comparison Job Editor opens.

You can now proceed to define the parameters of the new job.

2. Specify a Job Name

A job name is used to distinguish between configuration comparison jobs composed of specific sources and refinements. A name is defined when a new job is started and is used to save jobs for later use or repeated comparison of the sources.

To name a job:

- Type the name of the job in the **Name** field of the **Configuration Comparison Job Editor**.

3. Specify Source and Target Data Sources

The **Job Sources** section of the Configuration Comparison Job Editor contains two boxes labeled **Comparison Source** and **Comparison Target**. These boxes identify the data sources to be used in the job.

To add a data source to the job:

- 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.

When a data source is added to a job, it is displayed by **Name**, **Type**, and **Host**. You can change a data source by clicking **Change Data Source** in the appropriate box and selecting a new data source from the dialog, or by dragging a different data source to the box from Data Source Explorer.

The **Comparison Source** and **Comparison Target** boxes must contain data sources, archives, and standards before the job can be executed.

To add multiple comparison targets, click **Add More Targets** to enable a **One-to-Many** comparison.

Additionally, you can specify whether you want to compare the current properties of the live data source, or an archived set of properties previously stored by Change Manager using the radio buttons beneath each box:

- 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 archive and version to use in the job.

Note: You can drag and drop data sources, archive jobs, archive versions, and standards into the Comparison Source and Comparison Target boxes. A Standard can only be the Source in a Configuration Comparison Job.

4. Specify Job Options

Prior to the execution of a job, parameters can be set to control aspects of the execution process itself. These options are set on the **Options** tab.

Presently, the **Options** tab contains only the **Extract DDL** option. When selected, this function automatically outputs the synchronization code produced by the job comparison into a separate editor window once the process has executed.

To specify job options:

- Navigate to the **Options** tab and select or de-select the **Extract DDL** check box to turn this function on or off, respectively.

5. Refine Property Comparisons

Configuration property refinement involves choosing specific configuration properties to compare, and specific configuration properties to omit, prior to executing a comparison job.

By default, when you specify a data source as a comparison job Source or Target, you can refine the properties to be compared prior to executing the job on the Refinements tab of the Configuration Comparison Job Editor.

Configuration properties are organized by tree and listed by Property Name, Description, Location, and Source.

To specify the properties used in the comparison job:

- Navigate to the Refinements tab and select or de-select the configuration properties in the table, as required. Selected properties are included in the comparison job, and de-selected properties are ignored during job execution.

6. Execute Comparison Job

The configuration comparison job runs when you click the **Run Job** icon on any tab in the Configuration Comparison Job Editor. It's located in the upper right-hand corner of your screen.

When the process executes, the Comparison Results tab appears and displays the progress of the job and a summary of the comparison job results.

7. View Results

When a job is complete, results are displayed on the **Comparison Results** tab.

Job results are displayed by property with an overview of the matching and non-matching properties located at the top of the page display. Individual details regarding the data source properties are listed in the Comparison Results table:

- The Result column for each Target contains a check mark or X icon, depending on if the corresponding property value matches between the Source and Target data sources.

- The Source Value 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.

8. Execute an SQL Synchronization Script

Change Manager will build a SQL synchronization script to correct any discrepancies it discovers between the source and the target during the job execution process.

You execute a synchronization script via the **Run** menu or via the commands located along the top of the editor window where the script opened.

To execute the code:

Click the **Execute** icon. Change Manager executes the code on the specified data source. Results of the execution are displayed in the same tab or in a new tab.

9. Generate a Job Report

Once you have executed a job, you can generate a job report via the **View Report** command, located on the **Results** tab.

The report opens in a separate window where additional report options are available, such as sending the report to the printer, saving it to disk, or exporting the report in a specified file format to a directory or network location.

To generate a job report:

- Click the **View Report** command on the **Results** tab. The report opens in a new window and enables you to customize and perform other operations on it.

Cheat Sheet 1B: Creating a New Configuration Archive

A configuration archive is an object in a configuration comparison job that represents a snapshot of the configuration parameters of a data source. Archives are used in configuration comparison jobs against data sources, standards, and other archives, for the purposes of monitoring property value slippage, preserving backup settings, or otherwise enabling the use of older data source settings in configuration comparison jobs.

Use the table below to review the task and its individual steps prior to creating a new configuration archive:

1. Create a New Configuration Archive Job
2. Specify a Configuration Archive Name
3. Specify an Archive Source
4. Refine Properties
5. Specify Notifications
6. Save Configuration Archive and Manage Versioning
7. Generate a Job Report
8. Use Configuration Archive in Configuration Comparison Jobs

1. Create a New Configuration Archive Job

Archives are defined in Configuration Archive Job Editor.

To create a configuration archive:

- Right-click on the data source and select **Archive > Configuration** from the menu, or select **File > New > Configuration Archive Job**. The Configuration Archive Job Editor opens.

When the archive is first created, it is in an unsaved state and versionless. To create a version and “save” the archive, you need to issue the **Run Job** command. Unsaved archives cannot be used in comparison jobs.

2. Specify a Configuration Archive Name

A name must be defined when a new archive is created to distinguish between archives and other objects in the application environment. Ensure you specify a meaningful name that identifies the archive in other views and dialogs later on.

To name a configuration archive:

- Enter the name of the archive in the **Name** field of Configuration Archive Job Editor.

3. Specify an Archive Source

The **Configuration Archive Source** table specifies the data source from which the archive is derived. Drag and drop the data source to be archived from **Data Source Explorer** to the **Configuration Archive Source** box, where it is listed by name, DBMS type, and host.

4. Refine Properties

When a new configuration archive is created, all data source properties are included in the archive definition by default. However, you may only need to archive 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.

To refine archive properties:

Use the check boxes to select or de-select properties as needed. Selected properties are included in the archive when configuration comparison jobs run, while de-selected properties are ignored.

5. Specify Notifications

Notification options provide parameters for enabling and setting up various notifications regarding executed configuration archive comparison jobs.

You can specify email, file system, or system tray notifications on the **Notification** tab of the Configuration Archive Editor:

- Email notification options provide parameters for configuring and enabling email notification and report mailing on executed comparison jobs.
- File system notification options provide parameters for configuring and enabling logging options for executed comparison jobs.
- System tray notification parameters enable pop-up window notifications on your Desktop when Change Manager is minimized, but running jobs.

To set notifications:

- Navigate to the **Notifications** tab and modify the settings, as appropriate. Email, file system, and system tray notification options are labeled and contained in separate sections on the interface.

6. Save Configuration Archive and Manage Versioning

Once you have completed property refinement, you need to save the archive by creating a version. A version is a specific set of configuration property values associated with an archive and used to store older snapshots of the same archive for backup or tracking comparison purposes.

Versions are created by clicking the **Run Job** icon located in the lower right-hand corner of each tab on the Editor. When you define a new archive, no versions exist. This indicates the archive is not yet saved to the system, and cannot be used in comparison jobs.

To create a version and save the archive:

- Click **Run Job**. A new tab appears named **Unsaved Version** that displays the current values of the specified database.

Additionally versions are created in the same archive each time you click **Run Job**. The new version will contain the latest values for the properties, as specified on the Refinements tab.

Versions are listed on the Overview tab, (displayed in ascending order), and can be accessed by double-clicking the appropriate version in the table. The selected version opens in a new tab of the Editor. Additionally, versions are listed in the Data Source Explorer as the child nodes of the archive to which they represent.

7. Generate a Job Report

Once you have executed a job, you can generate a job report via the **View Report** command, located on the **Results** tab.

The report opens in a separate window where additional report options are available, such as sending the report to the printer, saving it to disk, or exporting the report in a specified file format to a directory or network location.

To generate a job report:

- Click the **View Report** command on the **Results** tab. The report opens in a new window and enables you to customize and perform other operations on it.

8. Use Configuration Archive in Configuration Comparison Jobs

Archives are used in configuration comparison jobs in the same manner as registered Data Sources or Standards.

To use an archive in a configuration comparison job:

- Click and drag the appropriate version from Data Source Explorer to the Source or Target box in Configuration Comparison Editor.

Cheat Sheet 1C: Creating a New Configuration Standard

A Configuration Standard is an object associated with configuration comparison feature. It provides customized data source configuration properties that are used in configuration comparison jobs, against data sources and archives to provide a specific set of values and threshold operators that “check” enterprise settings and ensure all network configuration values fall within acceptable ranges.

Use the table below to review the task and its individual steps prior to creating a new configuration standard:

1. Create a New Configuration Standard
2. Specify a Configuration Standard Name
3. Add Property Sources
4. Refine Properties
5. Save Configuration Standard
6. Use Configuration Standard in Configuration Comparison Jobs

1. Create a New Configuration Standard

Standards are defined in the Configuration Standards Editor.

To create a new configuration standard:

- Select **File > New > Configuration Standard**. The Standard Editor opens.

If you want to add the configuration properties of an existing data source or archive to the new standard, click and drag the appropriate object from Data Source Explorer or Standards Explorer to the **Standard Sources** table in the editor. The properties of the selected data source/archive populate the new standard.

2. Specify a Configuration Standard Name

A name must be defined when a new standard is created to distinguish between standards and other objects in the Change Manager environment. Ensure you specify a meaningful name that identifies the standard in other views and dialogs later on.

To name a configuration standard:

- Enter the name of the standard in the **Name** field of the Configuration Standards Editor.

3. Add Property Sources

The Standard Sources table specifies the amalgamation of data sources, archives, and other standards from which new standard properties are derived.

By default, Change manager automatically adds all properties of a selected source to the standard definition. You need to manually refine property selections on the Refinements tab once sources have been added. See the **Refine Properties** step below for more information.

To add properties to a standard:

- Drag and drop a data source, archive, or standard, from Data Source Explorer or Standards Explorer, respectively, to the Standard Sources table. Alternatively, use the Select Standard, Select Data Source, and Select Archive Version commands to add properties to the Standards definition.

4. Refine Properties

The **Standard Refinement** tab displays all sources and corresponding properties applied to the Standard. Properties are listed by Property Name, Source, Type of Property, Operator, and Type, and are organized in a tree as the children of the original source.

The Standard Refinement tab is used to customize property selections. You can choose to include or exclude properties via the check boxes located beside each property. Additionally, Operator values can be customized to set threshold or otherwise specify what constitutes a “pass” or a “fail” when a comparison job runs using the standard.

The following operators can be applied to properties:

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

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

The Type column indicates if individual properties are Linked or Static:

Linked property values refresh with the latest source values whenever the standard is used in a job. Static property values remain affixed at the value the property contained at the time it was added to the standard definition.

To refine standard properties:

- 1 Define each property as Linked or Static by clicking the appropriate row value in the Type column and choosing a selection from the drop down menu.
- 2 Define the operator used to determine the conditions of the matching result when the job runs. Click the appropriate row value in the Operator column and choose a selection from the drop down menu. Ensure you include a condition value in the corresponding Value column.

5. Save Configuration Standard

When a standard is saved, it appears in Standards Explorer as the name you defined in the **Name** field of the Editor.

To save a standard:

- Select **File > Save** or **File > Save All**. The standard appears in Project Explorer.

6. Use Configuration Standard in Configuration Comparison Jobs

Standards are used in configuration comparison jobs in the same manner as data sources and archives, with the noted exception that they may only be designated as Sources due to the unique property operator values they contain.

Cheat Sheet 2A: Creating a New Data Comparison Job

Data comparison jobs are used to compare the repositories of two data sources. Job features enable you to specify execution options and mapping refinement parameters and synchronize any discrepancies discovered by the job, once it has been executed.

Use the table below to review the task and its individual steps prior to creating a new data comparison job:

1. Create a New Job
2. Specify a Job Name
3. Specify Source and Target Data Sources
4. Specify Job Options
5. View Data Source Mapping Details
6. Customize Database Mapping
7. Customize Table Mapping
8. Specify Notification Options
9. Execute Job
10. View Results
11. Customize Data Resolution
12. Resolve Data Repositories via an SQL Script
13. Generate a Job Report

1. Create a New Job

Data comparison jobs are performed in the **Data Comparison Job Editor**. You create a new comparison job by opening the editor and defining job parameters, as needed.

To create a new comparison job:

- Select **File > New > Data Comparison Job**. The Data Comparison Job Editor opens.

Note: If any jobs already exist, they can be opened by selecting **File > Open Comparison**, and choosing the name from the list of available jobs.

2. Specify a Job Name

A job name identifies the job in the application and should be specified with this in mind.

Specify a meaningful name that clearly identifies the job in the views and dialogs of the working environment.

To name a job:

- Enter the name of the job in the **Name** field of the **Data Comparison Job Editor**.

3. Specify Source and Target Data Sources

The **Source** and **Target** boxes identify what data sources will be compared when the job executes. Both boxes must contain data sources before the job can be executed. When a data source is added to a job, it is displayed by **Name**, **Type**, and **Host**.

To add a data source:

- Click and drag a data source from **Data Source Explorer** to the Source or Target box. If no data sources have been registered in Change Manager, click **Select Data Source** in the appropriate box and then click **New Data Source** to add the data source. Data sources added in this way are automatically registered and added to Data Source Explorer.

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

4. Specify Job Options

Prior to the execution of a job, parameters can be set to control aspects of the execution process itself. These options are set on the **Options** tab.

The following option groups are available:

- Execution options control aspects of the job process execution.
- Mapping options control how the data sources in the job are mapped by the process.
- Comparison options specify the parameters of how the job treats the comparison in terms of naming conventions.
- Resolution options determine how the job resolves any discrepancies discovered during the comparison process.

Additionally, the Execution Options section contains the Compression Comparison setting.

The comparison of large repositories with many databases, tables, and rows can be a time consuming and tedious process, especially when you are required to check at the row level to ensure complete accuracy between a data source and its backup, or during migration procedures from testing to production that require optimum levels of accuracy.

The Compression Comparison option enables an algorithm during the execution process of a job. The algorithm provides Change Manager with the ability to compare at the table level instead of the row level, initially.

Each table in a pair is assigned a value and if the values match, the comparison process moves to the next pair of tables. If the values do not match, the process performs a more detailed comparison at the row level. This feature allows for a faster comparison process in general, without losing any accuracy during the process.

The Compression Comparison option is enabled on the Options tab of the Data Comparison Job Editor. It is available on SQL Server 2005, DB2, and Oracle platforms, and requires that CLR is enabled to function correctly.

To set job options:

- Navigate to the **Options** tab and modify the options, as needed. Options are grouped into execution, mapping, comparison, and resolution sections.

To enable Fast Compare:

- Before executing a job in the Data Comparison Job Editor, navigate to the Options tab and select Compression Comparison in the Execution Options section. Fast Compare is enabled and when the job is executed, the system performs the comparison, as appropriate.

5. View Data Source Mapping Details

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 executes to determine matches and mismatches between the two sources. Mapping details are viewed on the Mapping tab of the editor.

Change Manager automatically maps data source elements based on its interpretation of both schemas. Elements that do not match are excluded from the job when it executes, unless manually paired prior to running the job.

The Database Mapping table displays the paired databases or owners on the Source and Target. The % Mapped column indicates how much of the database or owner (a percentage value of total tables and rows) is mapped. Unpaired databases appear in this table as well, but are excluded from the execution process unless manually mapped to a corresponding database.

In order to view the table and view mappings of database pairs, you need to initiate the inspection process by clicking Inspect Checked Pairs. Once Change Manager has analyzed the selected pairs, to view the table mappings of a specified pair, click a row. The Table Mapping table populates with the paired tables and views of the selected pair. You can view the row mappings of a table or view pair by clicking Edit in the rightmost column of the appropriate row. The row pairings and corresponding values appear in the new window.

Individual owner and database pairs can be examined by selecting the appropriate icon in the **Inspect Now** column.

6. Customize Database Mapping

You can customize job database mapping parameters manually by choosing to exclude certain paired rows, or redefining objects after the automatic mapping process is completed.

The check boxes beside each paired object displayed in the Database Mapping Source column indicate if a pair is included or excluded when the job runs.

To include or exclude a database pair from the job:

- Select or de-select the check box beside each database pair, as needed. Use the **All** and **None** commands at the bottom of the table to select all or none of the rows in the table, respectively.

Database pairings can also be redefined via the drop down menu that appears when the corresponding item in the **Target** data source column is selected. The menu contains a list of all databases or owners that reside on the **Target** data source.

Databases or owners that are already paired appear in the drop down menu with an asterisk to denote them. Additionally, using the same method, unpaired databases can be assigned a matching database on the Source.

On Microsoft SQL Server and Sybase, you can pair up owners independently of databases by right-clicking on a row and selecting Show Database Parts, Show Source Database Parts, or Show Target Database Parts to decompose the database into its owner components. These components can be individually mapped and explored.

To recompose the database, right-click on and select Hide Database Parts. To restore the Database Mapping table to its original settings, click **Restore Default Mappings**. All settings revert to the original object mappings as determined by Change Manager based on the selected Mapping Options.

7. Customize Table Mapping

You can manually map databases, tables, and rows between the Source and Target data sources. Tables are displayed by name and contain mapping refinement column values (Columns, Comparison Key, and Filter).

You can manually map unpaired tables between the database by clicking the corresponding item in the Target data source column and define it via the drop down menu that appears, as with database mapping.

Additionally, if you click Edit in the appropriate pairing, the Columns, Comparison Key, and Filter dialog enables you to further detail mapping refinements in terms of defining table roles in the execution of the job.

- The Columns and Comparison Key tab enables the mapping of columns to compare between the paired tables. Select or de-select the columns, as needed, in the dialog provided and use the drop down menu to change the comparison key if multiple keys are available.
- The Comparison Filter enables you to manually insert a WHERE clause for execution when the job runs. Click the ellipsis button and enter the WHERE clause into the text box provided. If you want to create a different WHERE clause per table in the pair, select the Separate WHERE Clause radio button. The dialog adds an additional text box for a second WHERE clause.

Note: When entering selective WHERE clauses, do not include the WHERE keyword. (E.g. use “AGE>12” not “WHERE AGE>12”).

8. Specify Notification Options

Notification options provide parameters for enabling and setting up various notifications regarding executed jobs.

You can specify email, file system, or system tray notifications on the **Notification** tab of the Configuration Archive Editor:

- Email notification options provide parameters for configuring and enabling email notification and report mailing on executed comparison jobs.
- File system notification options provide parameters for configuring and enabling logging options for executed comparison jobs.
- System tray notification parameters enable pop-up window notifications on your Desktop when Change Manager is minimized, but running jobs.

To set notifications:

Navigate to the **Notifications** tab and modify the settings, as appropriate. Email, file system, and system tray notification options are labeled and contained in separate sections on the interface.

9. Execute Job

The job runs when you click the Run icon located in the lower right-hand corner of the Options or Mapping tab in the Data Comparison Editor.

When the process executes, the Results tab appears and displays the progress of the job via the Overall Progress bar at the top of the tab.

In the Comparison Results table, the database pairs are displayed along with Progress, Compare Index, Results, and Resolution columns.

- The Progress column indicates, per database pair, the progress of the job. When the job is finished comparing the database row, a result of Done (should the comparison be successful), Failed (should the comparison be unsuccessful), or Nothing to Compare (if pairs were not specified) will display in the column.
- The Compare Index column indicates the percentage value of data per database pair that Change Manager matched. Failed rows and rows in which the progress is not completed do not display results.

10. View Results

When a job is complete, you can view its details by clicking the View command in the Results column if the pair succeeded, or the Show Errors command if the pair failed. The Show Errors command opens a SQL error dialog that displays the error encountered by Change Manager during the job and the time that it occurred.

If a data comparison job is successful, the View command displays database or owner pair results in the Database Results tab:

- The first two columns of the Database Results are labeled with the names of the data sources that were compared. Rows are populated with the names of the tables residing on each database.
- The Matched column indicates how many row values between the specified pair of tables matched in the comparison job.
- The Only in [Source Name] column indicates the number of extra rows that were found in the specified source table.
- The Only in [Target Name] column indicates the number of extra rows that were found in the specified target table.
- The Different column indicates the number of rows different between the two tables.

When a table row is selected, the Selected Table panel populates with the rows and columns of both tables. All values that are out-of-sync between the two tables are highlighted in red. Rows that matched are not displayed.

Double-click on a mismatched row pair to open the Data Diff View. The view contains row synchronization commands and the specific data values of the selected pair.

11. Customize Data Resolution

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.

12. Resolve Data Repositories via an SQL Script

Once a data comparison job has completed, synchronization commands are used to even data in the repositories so they match.

To resolve data repositories:

- 1 On the **Results** tab, click **View** in the row you want to resolve. The **Database Results** tab appears.
- 2 The table in the **Results Overview - Activation** section displays the pairings in the selected tables, and includes the Matched, Only In, and Different columns that indicate any discrepancies between the two pairings. Select a pair from the table, the **Selected Table** populates with information regarding the specified pair.
- 3 Specify the type of **Resolution** by clicking **All**, **Mismatched**, **Source**, **Only Source**, **Target**, or **Only Target** radio buttons, respectively.
- 4 Click **Generate an SQL Script**. The script appears in SQL Editor, and can be modified manually, as needed, prior to its execution.

13. Generate a Job Report

Once you have executed a job, you can generate a job report via the **View Report** command, located on the **Results** tab.

The report opens in a separate window where additional report options are available, such as sending the report to the printer, saving it to disk, or exporting the report in a specified file format to a directory or network location.

To generate a job report:

Click the **View Report** command on the **Results** tab. The report opens in a new window and enables you to customize and perform other operations on it.

Cheat Sheet 3A: Creating a New Schema Comparison Job

Schema comparison jobs are used to capture, compare, and synchronize schema in databases and provide reports on comparison jobs that enable you to distribute information throughout your environment.

Use the table below to review the task and its individual steps prior to creating a new schema comparison job:

1. Create a New Job
2. Specify a Job Name
3. Specify Source and Target Data Sources
4. Refine Schema Comparisons
5. Customize Schema Mapping
6. Specify Job Options
7. Specify Notification Options
8. Execute the Job
9. Generate Job Report
10. Resolve Schema

1. Create a New Job

Schema comparison jobs are defined in the Schema Comparison Job Editor.

To create a new schema comparison job:

- Select File > New > Schema Comparison Job in the Menu Bar.

The Schema Comparison Job Editor opens and you can now proceed to define the parameters of the new job.

2. Specify a Job Name

A job name identifies the job in the application and should be specified with this in mind. Specify a meaningful name that clearly identifies the job in the views and dialogs of the working environment.

To name a job:

- Enter the name of the job in the Name field of the Data Comparison Job Editor.

3. Specify Source and Target Data Sources

The Schema Comparison Source and Schema Comparison Target boxes identify what data sources are compared when the job executes. Both boxes must contain the sources before the job can be executed. When a source is added to a job, it is displayed by Name, Type, and Host.

To add a data source:

- Click and drag a data source from Data Source Explorer to the Schema Comparison Source or Schema Comparison Target box. If no data sources have been registered in Change Manager, click Select Data Source in the appropriate box and then click New Data Source to add it. Data Sources added in this way are automatically registered and added to Data Source Explorer.

4. Refine Schema Comparisons

Schema comparison refinement involves selecting the schema objects that you want to include and exclude in the comparison process. All refinement is performed on the Refinements tab in the Schema Comparison Editor.

To specify the properties used in the comparison job:

- Navigate to the Refinements tab and select or de-select the objects in the table, as required. Selected objects are included in the comparison job, and de-selected properties are ignored during job execution.

5. Customize Schema Mapping

Mapping is the process of pairing objects between the source and target of the job for parallel comparison purposes. Mapping details are viewed and modified on the Mapping tab of the editor.

Change Manager automatically maps schema objects based on its interpretation of both schema. Elements that do not match are excluded initially, unless manually paired prior to running the job.

Object pairings are redefined via the drop down menu that appears when the corresponding object in the target column is selected. The menu contains a list of all schema objects that reside on the target data source.

To redefine object pairings:

- Navigate to the Mapping tab and then click the target of the pair you want to change. You can use the drop down menu to redefine the schema object pairing, as needed.

6. Specify Job Options

Prior to the execution of a job, parameters can be set to control aspects of the execution process itself. These options are set on the **Options** tab.

The following option groups are available:

- Synchronization options enables you to specify how Change manager handles the comparison synchronization script once a job has executed.
- Compare options provide settings that determine what elements of the source and target of a job are omitted when running the job.
- Include Objects options determine how objects are treated during the job execution process. For example, including or excluding specific object types.
- Identifier options determine what elements of the source and target data source will be ignored during the comparison process.

To set job options:

Navigate to the **Options** tab and modify the options, as needed. Options are grouped into execution, mapping, comparison, and resolution sections.

7. Specify Notification Options

Notification options provide parameters for enabling and setting up various notifications regarding executed jobs.

You can specify email, file system, or system tray notifications on the **Notification** tab of the Configuration Archive Editor:

- Email notification options provide parameters for configuring and enabling email notification and report mailing on executed comparison jobs.
- File system notification options provide parameters for configuring and enabling logging options for executed comparison jobs.

- System tray notification parameters enable pop-up window notifications on your Desktop when Change Manager is minimized, but running jobs.

To set notification options:

- Navigate to the **Notifications** tab and modify the settings, as appropriate. Email, file system, and system tray notification options are labeled and contained in separate sections on the interface.

8. Execute the Job

The job runs when you click the Run icon located in the upper right-hand corner of Schema Comparison Editor.

When the process executes, the Comparison Results tab appears and displays the progress of the job.

In the Comparison Results table, the schema pairs are displayed along with the Compare Index, Results, and Resolution values of the job.

- The Compare Index column indicates the percentage of schema that matched between the source and target of the job.
- The Results column provides a link that enables you to drill down and view the results between the pairs matched in the job. Click Show Individual Results to open the Individual Comparison Results tab.
- The Resolution column provides a command that produces a synchronization script to automatically synchronize any discrepancies between the compared schema.

9. Generate Job Report

Once you have executed a job, you can generate a job report via the **View Report** command, located on the **Results** tab.

The report opens in a separate window where additional report options are available, such as sending the report to the printer, saving it to disk, or exporting the report in a specified file format to a directory or network location.

To generate a job report:

- Click the **View Report** command on the **Results** tab. The report opens in a new window and enables you to customize and perform other operations on it.

10. Resolve Schema

When a schema comparison job is complete, synchronization commands are used to even the data so it matches. The synchronization process automates the generation of UPDATE, INSERT, and DELETE statements on the source and target data sources, as required.

The script can be manually modified, saved, or executed immediately via the commands in the Run menu, located on the Workbench toolbar. Save scripts are stored in Project Explorer and can be accessed at any time.

To resolve schema via a synchronization script:

- 1 On the Results tab, click Sync All in the Resolution column of the Comparison Results table. A synchronization script is produced in another window.
- 2 Modify the script as needed in the window, save the script via the File > Save command, or use the commands in the Run menu on the Workbench toolbar to execute the script, as needed.

Cheat Sheet 3B: Creating a New Schema Archive

A schema archive contains a snapshot of the schema of a registered data source, for a given point in time. Schema archives are used in schema comparison jobs against other data sources and archives as a baseline to monitor data loss and inconsistencies when moving between testing and production environments.

Use the table below to review the task and its individual steps prior to creating a new schema archive:

1. Create a New Schema Archive Job
2. Specify an Archive Name
3. Specify a Source
4. Refine Properties
5. Specify Extraction Options
6. Specify Notification Options
7. Save Schema Archive and Manage Versioning
8. Compare Archive Contents and Extract DDL
9. Generating a Job Report
10. Use Archive in Schema Comparison Jobs

1. Create a New Schema Archive Job

Archives are defined in Schema Archive Job Editor.

To create a Schema Archive:

- Select File > New > Schema Archive from the Main Menu. The Archive Job Editor opens.

When a schema archive is first created, it is in an unsaved state and versionless. To create a version, thereby “saving” the archive, you need to issue the Run Job command. Unsaved archives cannot be used in comparison jobs.

2. Specify an Archive Name

A name must be defined when a new archive is created. The name is used to distinguish between archives and other objects in Change Manager, and is used in versioning functionality later in the process.

Ensure that you specify a meaningful name that identifies the archive in other views and dialogs later.

To name a schema archive:

- Enter the name of the archive in the Name field of the Schema Archive Job Editor.

3. Specify a Source

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

To specify the archive 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, DBMS type, and host.

4. Refine Properties

By default, when you create a new archive, all schema is included. However, you may only want or need to archive a specific group of objects per archive defined. Other archives can be created, and likewise refined, for the same data source, but can maintain a different group of selected schema objects.

Archive properties are located on the Refinements tab of the Schema Archive Job Editor, where they are listed by object name, owner, and type.

To refine schema archive properties:

- Use the check boxes to select or de-select schema objects as needed. Selected objects are included in the archive when comparison jobs are run, while de-selected properties are ignored altogether.

5. Specify Extraction Options

The Options tab contains the DDL Extraction Options setting, which enables you to define how DDL is extracted and stored in Change Manager during the schema archive process. By default, the option is set to None, which indicates that the process does not retain extracted DDL when building the schema archive (although it can be extracted from the archive afterwards.)

If you choose to store extracted DDL, the directory location of the saved files can be viewed on the History tab of the editor when archive versions are created.

To define DDL extraction processes:

- Use the drop down menu to specify None, One File, or Multiple Files to indicate how DDL extraction occurs, if at all, during the archive execution process.

6. Specify Notification Options

Notification options provide parameters for enabling and setting up various notifications regarding executed jobs.

You can specify email, file system, or system tray notifications on the **Notification** tab of the Configuration Archive Editor:

- Email notification options provide parameters for configuring and enabling email notification and report mailing on executed comparison jobs.
- File system notification options provide parameters for configuring and enabling logging options for executed comparison jobs.
- System tray notification parameters enable pop-up window notifications on your Desktop when Change Manager is minimized, but running jobs.

To set notifications:

- Navigate to the **Notifications** tab and modify the settings, as appropriate. Email, file system, and system tray notification options are labeled and contained in separate sections on the interface.

7. Save Schema Archive and Manage Versioning

A version is a specific set of values associated with an archive, and is used to store older snapshots of the same archive for backup and tracking comparison purposes. Initially, you must create and save a version for the archive to become valid within Change Manager.

A version is created when you click the Run icon located in the upper right-hand corner of any tab in the Schema Job Editor.

When the process executes, the New Unsaved Version tab appears and lists information related to the archive, including its contents and any extracted DDL.

To create a version and save the archive:

- Click the Run Job icon. After saving a version, it appears in the Job History table on the History tab containing various information, such as version number, timestamp, DDL location, where the archive was run (Workbench or command line), host name, and number of errors.

Additional versions are created in the same archive each time you click Run Job. The new version contains the latest values for the archive properties specified on the Refinements tab.

8. Compare Archive Contents and Extract DDL

On an open version tab, the Schema Archive Editor enables you to compare the archive version with its live source by right-clicking the Archive Contents table and selecting Compare. When you complete this action, Schema Comparison Job Editor opens immediately and designates the archive version and the original source as its source and target data sources, respectively.

Additionally, you can select Extract from the menu to extract the DDL of the archived contents. The code opens in the DDL Extract window, and can be viewed, saved, or otherwise modified via the Open Editor command beneath the window.

9. Generating a Job Report

Once you have executed an archive job (by saving it and creating a version), you can generate a job report that summarizes the archived schema information.

Click the History tab and then right-click on the version you want to create a report for. From the menu, select View Report. A summary report of the schema archive specific to the selected version appears in a separate window.

The report can be read on screen, sent to a printer, and saved to disk via the command icons located in the port window toolbar.

Additionally, the right-click menu contains a command named Generate Report. This will produce a report file in a specified format at a directory location of your choosing.

10. Use Archive in Schema Comparison Jobs

Archives are used in Schema Comparison jobs the same way as registered data sources.

To use an archive in a schema comparison job:

- Click and drag the appropriate version from Data Source Explorer to the Source or Target box in Schema Comparison Editor prior to executing the job.

DBMS Reference

This section contains these topics:

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

DBMS Operations Permissions

When 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

Task	Permissions for Own Objects	Permissions for All Objects
Alter Schema Object	SYSDBA or Database Owner	SYSDBA or Database Owner
Configuration Settings	No special privileges are required. Change Manager queries the JDBC metadata for basic version information only for configuration content.	No special privileges are required. 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

Task	Permissions for Own Objects	Permissions for All Objects
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.
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, 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. Change Manager supplies the Product Version as “8.00.2039”, and the Product Version Internal Value as 524288.

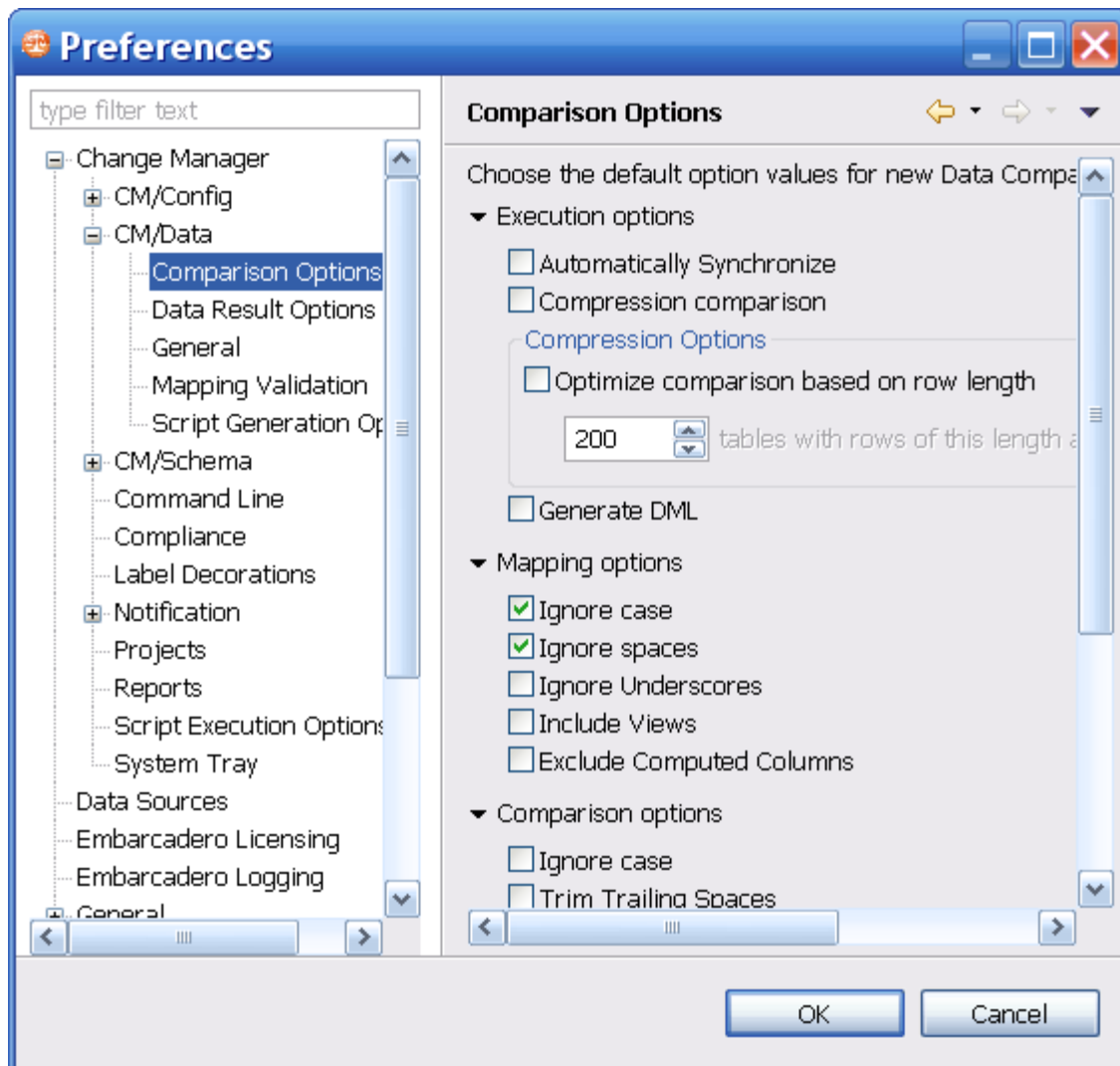
Sybase Version Information

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 Change Manager options.



The following sections describe preferences specific to Change Manager:

- [Compliance](#)
- [Comparison Options](#)
- [Data Result Options](#)
- [Script Execution Options](#)
- [Script Generation Options](#)
- [Jobs](#)

- [Command Line](#)
- [Mapping](#)
- [Notification](#)
- [Reports](#)
- [Data Sources](#)
- [SQL Editor](#)
- [Code Assist](#)
- [Code Formatter](#)
- [Results Viewer](#)
- [Syntax Coloring](#)
- [SQL Filters](#)

Configuration Comparison Options

Configuration comparison options enable you to specify the default job options in Configuration jobs. That is, when a new comparison or archive job is started, these parameters indicate if the **Extract DDL** job option is on by default or not.

- If you want to change the default **Extract DDL** job option for configuration comparison jobs, select the **Comparison Options** child node and select the **Extract DDL** check box.
- If you want to change the default **Extract DDL** job option for configuration archive jobs, select the **Archive Options** child node and select the **Extract DDL** check box.

Data Comparison Options

Data Comparison options have these nodes:

[Comparison Options](#)

[Data Result Options](#)

[General](#)

[Mapping Validation](#)

[Script Generation Options](#)

Comparison Options

The **Comparison Options** node provides a list of the default job options available when working with Data Comparison Job options. Select the check box of the appropriate job option to indicate that it is on by default.

For example, if this is your first time using Change Manager, the **Ignore Case** Mapping option is enabled by default in the **Data Comparison Job Editor**. When you navigate to the **Comparison Options** node, you will see that the **Ignore Case** option is selected. If you de-select this parameter, the next time you start a new data comparison job the **Ignore Case** mapping option will not be enabled.

The parameters in the panel correspond to the job options listed on the **Options** tab of the Data Comparison Job Editor.

Data Result Options

The **Data Result Options** node indicates where the result data of data comparison jobs are stored.

- The **Comparison Result Cache Location** parameter specifies the location of the result cache, where job data is captured. Use the **Default Location** or de-select it and enter a new directory location in the field provided.
- The **Refresh Interval** parameter sets the refresh interval of result data as it is stored to the specified location from the application, in seconds.

General

The **General** node contains two parameters that pertain to data comparison job options:

- The **Remembered Dialog Decisions** XX
- The **Don't Show Again Decisions** XX

Mapping Validation

Mapping Validation preferences enable you to specify severity levels for mapping validation on comparison and auto synchronization processes. Use the drop down menu beside each of the parameter descriptions and specify **Ignore**, **Warning**, or **Error**, as appropriate.

When Change Manager validates mapping, the value selected on the **Mapping Validation** node indicates how the application responds when the criteria is met.

- **Ignore** values are ignored by the process and Change Manager will give no indication of the specified criteria.
- **Warning** values will trigger a warning message indicating that the specified criteria has been met during the mapping validation process.
- **Error** values will issue an error if the specified criteria has been met.

Script Generation Options

Script Generation Options provide parameters that define how Change Manager handles duplicate rows when it builds a synchronization script during the data comparison job process.

- Select **Warn About Duplicate Rows** if you want the application to issue a warning whenever it encounters duplicate rows while generating the synchronization script.
- Select **Exclude Duplicate Rows** if you want the application to automatically ignore duplicate rows while generating the synchronization script.

Set Schema Comparison and Archive Options

The following child nodes are available, and provide various parameters with regards to Data jobs in Change Manager:

[Archive Options](#)

[Comparison Options](#)[General](#)[Mapping Options](#)

Archive Options

The **Archive Options** node provides a parameter that enables you to specify how the **Extract DDL** job option extracts files when enabled in the **Schema Archive Job Editor**.

- Choose **None** if you want Change Manager to build the archive in the development environment. DDL can then be extracted directly from the archive.
- Choose **One File** if you want Change Manager to export the objects of the archive job to an SQL file, in addition to building the archive in the development environment. Once created, the SQL file can be accessed via **Project Explorer**.
- Choose **Multiple Files** if you want Change manager to export the objects of the archive job into individual SQL files, in addition to building the archive in the development environment. Once created, each of the SQL files can be accessed via **Project Explorer**.

Comparison Options

The **Comparison Options** node provides a list of the default job options available when working with Schema Comparison Job options. Choose a tab and then select the check box of the appropriate job option to indicate that it is on by default.

For example, if this is your first time using Change Manager, for DB2 data sources, the **Exist in Source Only (Create)** option is enabled by default in the editor. When you navigate to the **Comparison Options** node, you will see that this option is selected. If you de-select the parameter, the next time you start a new schema comparison job, the **Exist in Source Only (Create)** option will not be enabled.

The parameters in the panel correspond to the job options listed on the **Options** tab of the Schema Comparison Job Editor.

General

The **General** node contains two parameters that pertain to schema comparison job options:

- The **Display Confirm Dialog** XX
- The **Job Launching Option** XX

Mapping Options

The **Mapping Options** node provides a list of the default mapping job options available when working with schema comparison jobs. Select the check box of the appropriate job option to indicate if it is on by default.

For example, if this is your first time using Change Manager, the **Ignore Case** option is enabled by default. When you navigate to the **Mapping Options** node, you will see that this option is selected. If you de-select the parameter, the next time you start a new job, the **Ignore Case** option will not be enabled.

The parameters in the panel correspond to the job options listed on the **Mapping** tab of the Schema Comparison Job Editor.

Command Line

Command Line parameters enable you to specify the default script output when scheduling command line automation with regards to running Change Manager jobs.

- Choose **Windows (*.bat)** if you want the command line interface to produce batch files, by default.
- Choose **Ant (*.xml)** if you want the command line interface to produce XML files, by default.

Compliance

This node is located on the **Preferences** panel, under the **Configuration Comparison** sub-node, under **Change Manager**.

The **Compliance** setting provides percentage amount parameters that specify when to **Pass**, **Warn**, or **Fail** a configuration comparison job in the [Compliance Explorer](#) view.

Adjust the **Pass**, **Warn**, and **Fail** values as needed. The symbols beside each value will appear beside a job in the view, depending on the percentage amount that results from the comparison job.

Label Decorations

Label Decorations provide extra information about an object in the Change Manager interface. The **Label Decorations** node contains a link to interface decorator parameters selections, and the **Project** parameter, which adds extra information to projects, as they are displayed in **Project Explorer**.

Notification

The following child nodes are available, and provide various parameters with regards to notification options in Change Manager jobs:

[Email Notification](#)

[File System Notifier](#)

[Popup Notification](#)

Email Notification

Email Notification parameters enable you to specify email server parameters for enabling email notifications in Change Manager jobs.

Enter the **Host** and **Port** of the email server in the fields provided, and then add the appropriate information for **Address**, **User Name**, and **Password**.

You can use the **Test Email** button once the fields have been filled in, and Change Manager will prompt you for an email address, and then send a test notification email to the address provided.

Additional child nodes are available for **Email Notification** parameters, that enable you to further define how email notifications are handled in the system:

[Contacts](#)

[Templates](#)

Contacts

Contacts enables you to add email addresses to Change Manager for use in job email notifications. When a new notification is sent, it will be sent to all of the addresses added to the contact list.

Click **Add**, and then enter an email address into the field provided. The new address is added to **Contacts**. You can also **Import** existing contact lists from XML files, or **Export** the current contact list as an XML file.

Templates

The **Templates** node enables you to create email templates that define the formatting of the emails sent via email notifications in jobs.

Click **Add** and then enter a **Name** for the template in the field provided. The **Subject** field enables you to specify the **Subject** heading of all notification emails that are sent from a job, and the **Body** field enables you to add text to the body of the email.

Use the **Template Key** icon located in the lower left-hand corner of the **Add a Template** dialog to insert variables into the template parameter fields. For example **\$Date_Time** adds a time stamp that displays the date and time that the email notification was sent.

File System Notifier

File System Notifier parameters specify the default values for file system notifications in Change Manager jobs.

Popup Notification

Popup Notification parameters specify the default values for popup notifications in Change Manager jobs.

Additionally, the following child nodes are located under **Popup Notification**

Projects

The **Projects** node contains the **Suppress Project Required Warning Dialog**, which “hides” this warning dialog from view when working in **Project Explorer**.

Comparison Options

This node is located on the **Preferences** panel, under the **Data Comparison** sub-node, under **Change Manager**.

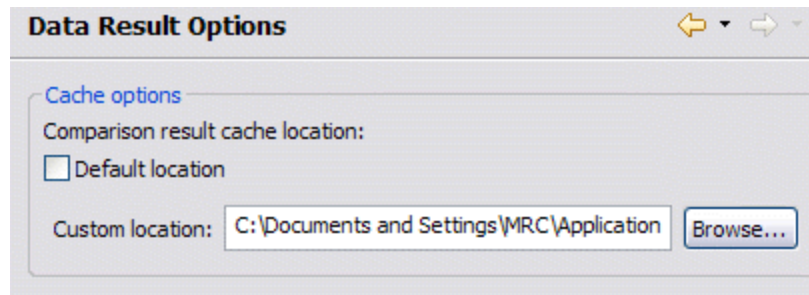
The **Comparison Options** setting enables you to specify when compression comparison is automatically initiated during a data comparison job, based on the total size of the rows in a given table.

Select **Optimize Comparison Based on Row Length** and then adjust the value (in bytes) in the box below. The value determines when compression is automatically enabled 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.

Data Result Options

This node is located on the **Preferences** panel, under the **Data Comparison** sub-node, under **Change Manager**.

The **Data Result Options** setting enables you to specify the directory location where the results of data comparison jobs are stored.



By default, results are cached locally in the system **Application** directory.

To change this location, de-select **Default Location** and enter the new directory address in the **Custom Location** field.

NOTE: This is a temporary cache, and during large data comparison jobs may use up much of the system resources.

Script Execution Options

This node is located on the **Preferences** panel, under the **Data Comparison** sub-node, under **Change Manager**.

The **Script Execution Options** setting enables you to indicate that auto commit is always applied when executing synchronization scripts for data comparison jobs.

Script Generation Options

This node is located on the **Preferences** panel, under the **Data Comparison** sub-node, under **Change Manager**.

The **Script Generation Options** setting specifies how Change Manager handles duplicate rows when generating a SQL synchronization script for data comparison jobs.

- Select **Warn About Duplicate Rows** to prompt a user when the data comparison synchronization script generation process encounters them.
- Select **Exclude Duplicate Rows** to indicate that the data comparison synchronization script generation process should ignore any duplicate rows that it encounters.

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 synchronization of data may have unexpected results.

Jobs

This node is located on the **Preferences** panel, under the **Schema Comparison** sub-node, under **Change Manager**.

The **Jobs** setting specifies how schema comparison jobs launch in the development environment.

- Select **Do Not Ask About Running the Created Jobs** if you want to ignore the system prompt when running schema comparison jobs.

- Select **Automatically Launch Jobs Created from Selected Objects** if you want the system to automatically execute a schema comparison job upon creation.

Command Line

This node is located on the **Preferences** panel, under the **Change Manager** node.

The **Command Line** node provides the **Default Script Output** setting, which enables you to define the operating system/file type by which command line scripts are stored as during the batch file creation process. See [Generating Command Line Scripts](#) for more information.

- Select **Windows (*.bat)** (for scripts to be run on Windows), **Linux (*.sh)** (for scripts to be run on Linux), or **Ant (*.xml)** as appropriate.

Mapping

This node is located on the **Preferences** panel, under the **Change Manager** node.

The **Mapping** node is used to specify how the system handles syntax nuances between objects during comparison job mapping processes.

- Select **Ignore Case**, **Ignore Spaces**, and/or **Ignore Underscores**, as needed.

Notification

This node is located on the **Preferences** panel, under the **Change Manager** node.

The **Notification** node is composed of the **Email Notification** subnode and its associated branches. These settings are used to modify and set up how Change Manager handles job notifications for its various modules.

- The **Email Notification** node contains the **Host Settings** and **User Information** fields that the system will use to send email notifications.
- The **Advanced Settings** node contains JavaMail key-value pair definitions that the system uses to send email notifications. Refer to <https://java.sun.com/products/javamail/downloads/index.html> for more information. An FAQ is also available here: <https://java.sun.com/products/javamail/FAQ.html>, and SMTP properties are listed here: <http://java.sun.com/products/javamail/javadocs/com/sun/mail/smtp/package-summary.html>.
- The **Contacts** node enables you to create a list of address contacts to whom email notifications will be sent.
- The **Templates** node enables you to define and save email notification templates, which control the format in which notification emails are sent.

NOTE: Pressing **Ctrl-Space** on the **Notification** screen opens a list of tags that Change Manager replaces with job-specific content.

Reports

This node is located on the **Preferences** panel, under the **Change Manager** node.

The **Reports** node provides settings and parameters that enable you to customize the look, feel, and format of reports issued when you generate a job report for comparison jobs.

- The **Report Style** tab enables you to change the logo and accent color of a report.

- The **Paper Setup** tab enables you to modify printer settings for paper reports.

Data Sources

This node is located on the **Preferences** panel, under the **Change Manager** node.

The **Data Sources** node contains a directory location field that specifies where registered data source definitions are stored.

SQL Editor

This node is located on the **Preferences** panel, under the **SQL Development** node.

- 1 Select **Window > Preferences > SQL Development > SQL Editor**.
- 2 Change the settings as appropriate in each section and then click **Apply**.
 - **Enable SQL Parser** indicates that the parser that provides code development features in SQL Editor is enabled. If this option is disabled, functions such as code formatting, auto completion, semantic validation, hyperlinks, etc. will not be available. The options below the check box enable you to limit (by file size) when the parser is enabled when dealing with code on a file-by-file basis.
 - **Parsing Delay** specifies the amount of time that the parser delays prior to activating features after a keystroke.
 - **Severity Level for Semantic Validation Problems** determines how semantic code errors are flagged in the editor and the **Problems** view.

Clearing **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.

Code Assist

The **Code Assist** panel is used to specify configuration parameters that determine how code completion features in SQL Editor behave.

Select **Window > Preferences > SQL Development > Code Assist**.

- **Enable Auto Activation** enables or disables code assist functionality.
- **Insert Single Proposals Automatically** specifies if only a single code completion suggestion is returned, it is inserted automatically.
- **Fully Qualified Completions Automatically** specifies if code completion results are returned specific (fully qualified), rather than the minimum required to identify the object.

Code Assist Color Options specifies the color formatting of code completion proposals. Select background or foreground options from the menu and modify them as appropriate.

Code Formatter

The **Code Formatter** pane provides configuration options for code formatting functionality in SQL Editor.

Select **Window > Preferences > SQL Development > Code Formatter**.

The panel provides a drop down list of formatting profiles and a preview window that displays how each profile formats code.

Results Viewer

The **Results Viewer** pane provides configuration options that specify how the **Results** view displays results.

Select **Window > Preferences > SQL Development > Results Viewer**.

- **Grid Refresh Interval** indicates the speed in milliseconds that the Results view refreshes.
- **Stripe the Rows of the Results Table** adds intermittent highlighted bars in the Results view.

Display Results in Separate Tab in SQL Editor opens the Results view in a separate window on the Workbench.

Syntax Coloring

The **Syntax Coloring** panel provides configuration options that change the look and feel of code syntax in SQL Editor.

Select **Window > Preferences > SQL Development > Syntax Coloring**.

Use the tree view provided in the **Element** window to select the comment type or code element you want to modify. Select the options to the right-hand side of the window to modify it. The **Preview** window shows a piece of sample code that updates according to the changes you made.

SQL Filters

See [Filtering Data Source Objects](#) for instructions.

Extensions Reference

Extending Change Manager Functionality with Eclipse Plug-Ins

Change Manager publishes two extensions that can be implemented via plug-ins to customize the application.

The two feature extensions are:

- **Notifications:** enable you to code a plug-in for Change Manager that implements any type of job execution notification required for your organization. Change Manager includes email notification by default, but other plug-ins can be coded in Java and added to Change Manager, as needed.
- **Configuration Parameters:** enable you to code a plug-in for Change Manager that adds additional configuration parameters and values for configuration standards, archive jobs, and comparisons. Change Manager includes plug-ins for standard configuration properties and version information, but other plug-ins can query data sources via Java JDBC for additional or environment-specific information.

In order to utilize the functionality of these features, you will need to build a plug-in and specify the parameters of the notification and custom configuration extensions. The following sections provide examples that demonstrate the possibilities regarding this feature.

[Creating and Implementing a Notification Plug-In](#)

[Creating and Implementing a Custom Configuration Plug-In](#)

Creating and Implementing a Notification Plug-In

The job notification plug-in is built on the Eclipse platform and some knowledge regarding java coding and Eclipse plug-in development is required.

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.

The following options and option parameters may be selected via the job notification plug-in:

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 Change Manager. Additionally, each set of notification options apply individually per job selected.

The notification plug-in must be defined in Eclipse and then activated in Change Manager.

The following tasks provide a high-level overview of defining and implementing the notification plug-in:

[Creating a Plug-In Project](#)

[Configuring the Target Platform](#)[Defining Plug-In Dependencies](#)[Implementing a Notifier Class](#)[Implementing a Viewer Contribution](#)[Specifying Setup Constants](#)[Implementing the DirectoryNotifier Class](#)[Implementing the DirectoryNotifierViewerContribution Class](#)[Registering the Extensions](#)[Deploying the Notifier Plug-In](#)

Creating a Plug-In Project

In order to start building a plug-in, you need to create a new plug-in platform in Eclipse.

To create a plug-in project

- 1 Select **File > New > Other**.

The **Select a Wizard** dialog appears.

- 2 Select **Plug-In Project** and click **Next**.

- 3 Type a name for the plug-in 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 plug-in project is created in Eclipse and is ready for plug-in development.

Configuring the Target Platform

The target platform for the plug-in development process needs to be identified in Change Manager. This will indicate to the new plug-in for what product it extends, and will grant the plug-in access to the system.

- 1 Select **Window > Preferences**.

The Preferences dialog appears.

- 2 Choose the **Plug-In Development > Target Platform** node and click **Browse**.

- 3 Navigate to the install directory for Change Manager.

- 4 Click **Apply** and **OK**.

The Preferences dialog closes and the target platform of the plug-in is now indicated.

Defining Plug-In Dependencies

The new plug-in requires a pair of dependency definitions on the Change Manager Notification Plug-In:

- **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

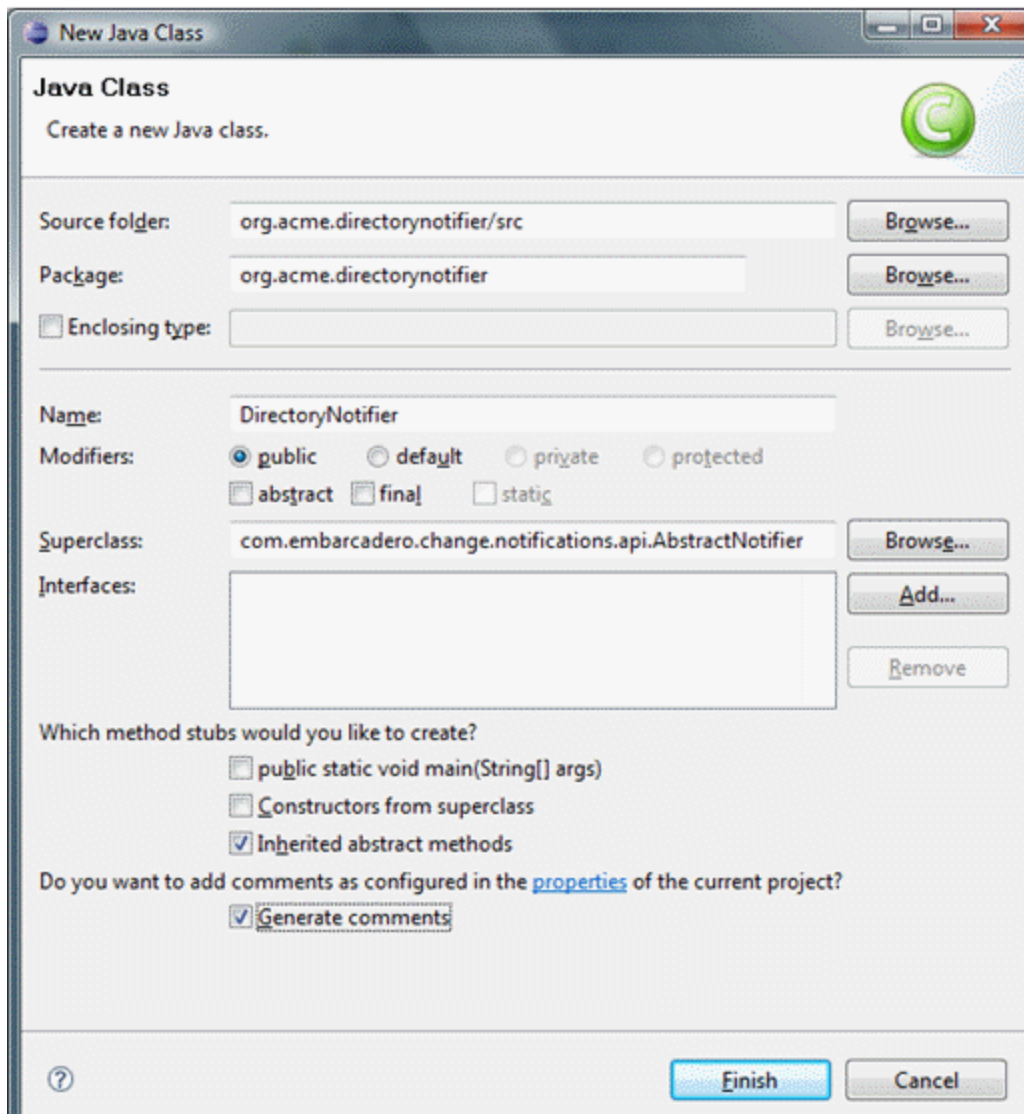
Each notification plug-in requires a Notifier class that must implement the interface **com.embarcadero.change.notifications.api.INotifier**.

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 plug-in 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 replaced 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
notifierData	INotifierData	This instance contains the configuration for the notification of the job just run.
jobMetaData	Map<String,String>	This map contains information about the job execution. The keys in the map correspond to entries in the NotificationPropertyEnum. For example, to get the data and time of the execution, you would code: "jobMetaData.get(NotificationPropertyEnum.DATE_TIME.getTag())" Note: The keys are the tag of the enum entries, so ensure that getTag() is called when accessing jobMetaData.
notificationInfo	IReportGenerator	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 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, a class needs to be written 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 lay out 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:

- Creates a new composite body, and sets it to have a grid lay out.
 - 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 Change Manager job editor to which this class applies.
- 3 Add the setData method by entering 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() );
        }
    }
}
```

The following code should be used to implement the **setEnabled** method:

```
public void setEnabled ( boolean enabled )
{
    folderField.setEnabled( enabled );
}
```

The following code should be used to implement the **dispose** method:

```
public void dispose()
{
    if ( body !=null && !body.isDisposed() )
    {
        body.dispose();
        body = null;
    }
}
```

Registering the Extensions

Once the plug-in has been coded, and the notifier and user interface implemented, definitions must be included that explain how Change Manager should interact with the plug-in.

This is performed by registering extensions that define to Change Manager the Notifier and NotifierViewerContribution. When the plug-in is added to Change Manager, the extensions are registered and Change Manager understands the purpose of the plug-in.

- 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 selecting **File>Save** from the menu.

When the `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 Change Manager.

Deploying the Notifier Plug-In

Once the plug-in has been defined, you need to deploy the plug-in in Change Manager.

- 1 Stop Change Manager if it is currently running and select **File > Export**.
The **Export** dialog appears.
- 2 Choose **Deployable Plug-Ins and Fragments** from the **Plug-In Development** group.
- 3 Click **Next**.
- 4 Select **org.acme.directorynotifier** and ensure that the target directory is attribute to the Change Manager installation folder.
- 5 Click **Finish**.
- 6 Start Change Manager and create a new Data Comparison job. Define a Source and Target data source and then navigate to the **Notification** tab.
The **Directory Notifier** has been added as a section to the tab.
- 7 Choose **Enable Directory Notifier** to enable the feature.
- 8 Specify the **Target Directory**. For example, **C:\ChangeManagerNotifications**, and choose **PDF** in the **Attached Report** section.
- 9 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 Plug-In

[Creating a Project Plug-In](#)

[Configuring the Target Platform](#)

[Defining Plug-In Dependencies](#)

[Implementing a Property Source](#)

[Determining Support](#)

[Defining Properties](#)

[Gathering Results](#)

[Synchronizing the Job](#)

[Registering Extensions](#)

[Deploying the Custom Configuration Plug-In](#)

Creating a Project Plug-In

In order to start building a plug-in, you need to create a new plug-in platform in Eclipse.

To create a plug-in project

- 1 Select **File > New > Other**.

The **Select a Wizard** dialog appears.

- 2 Select **Plug-In Project** and click **Next**.

- 3 Type a name for the plug-in 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 plug-in project is created in Eclipse and is ready for plug-in development.

Configuring the Target Platform

The target platform for the plug-in development process needs to be identified in Change Manager. This will indicate to the new plug-in for what product it extends, and will grant the plug-in access to the system.

- 1 Select **Window > Preferences**.

The Preferences dialog appears.

- 2 Choose the **Plug-In Development > Target Platform** node and click **Browse**.

- 3 Navigate to the install directory for Change Manager.

- 4 Click **Apply** and **OK**.

The Preferences dialog closes and the target platform of the plug-in is now indicated.

Defining Plug-In Dependencies

The plug-in requires a dependency for configuration comparison jobs in 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 plug-in and appears in the Dependencies list.

- 3 Press **CTRL-S**, or Select **File>Save** from the menu to save your changes.

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 plug-in 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 is indicated via 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 Change Manager, for any registered content plug-in 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 plug-in. 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 is 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 above code, the four entries that capture from JDBC the product name, version, minor version, and major version have been added to the enum.

Gathering Results

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 executed.

```

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 the `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 plug-in, you will need to register it with Change Manager. This is achieved by implementing an Eclipse extension. When the plug-in is added to Change Manager, the extensions are registered and Change Manager understands that the plug-in 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 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 Change Manager and Eclipse are informed about the property source. When the plug-in is added to Change Manager, it will add your properties to configuration comparison and Standard jobs.

Deploying the Custom Configuration Plug-In

To deploy the plug-in, perform the following steps:

- 1 Shut down Change Manager if it is running.
- 2 Choose **File>Export**.

The **Export** dialog appears.

- 3 Select **Deployable Plug-ins and Fragments** from the **Plug-In Development** group and click **Next**.
- 4 Select **org.acme.jdbcversioninformation** and ensure the target directory is the same as the Change Manager installation folder.
- 5 Click **Finish**.

The new plug-in is deployed.

Start up 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 now contain the four version properties you defined in the new plug-in.

User Interface Reference

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

The following interface components compose the Change Manager work environment:

- [The Change Manager Workbench](#)
- [Welcome Page](#)
- [Views](#)
- [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](#)

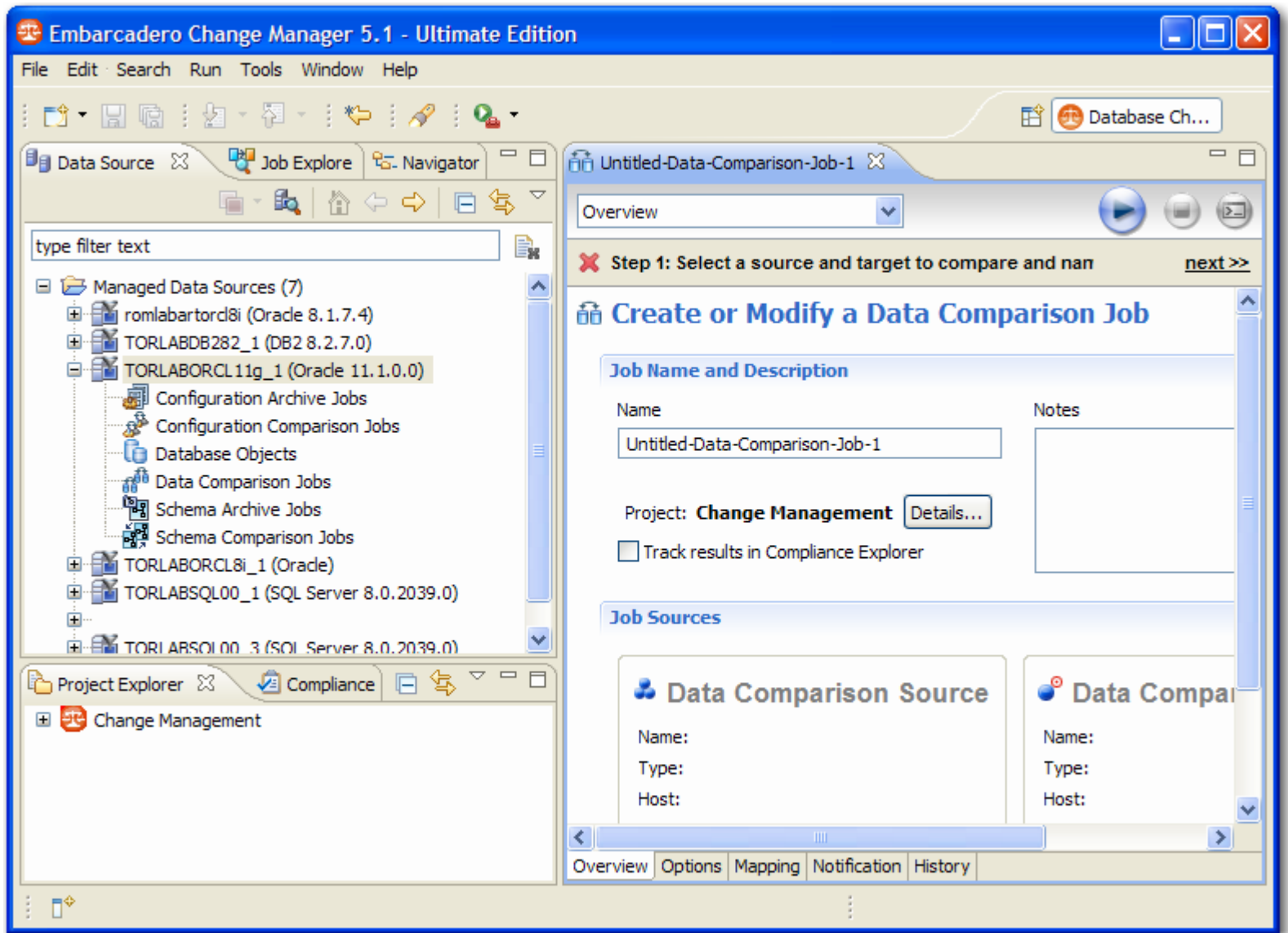
The Change Manager Workbench

The Workbench is the Change Manager development environment. It provides the interface to create, manage, and navigate 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 Change Manager. In addition, you can customize aspects of the environment to suite your development requirements.

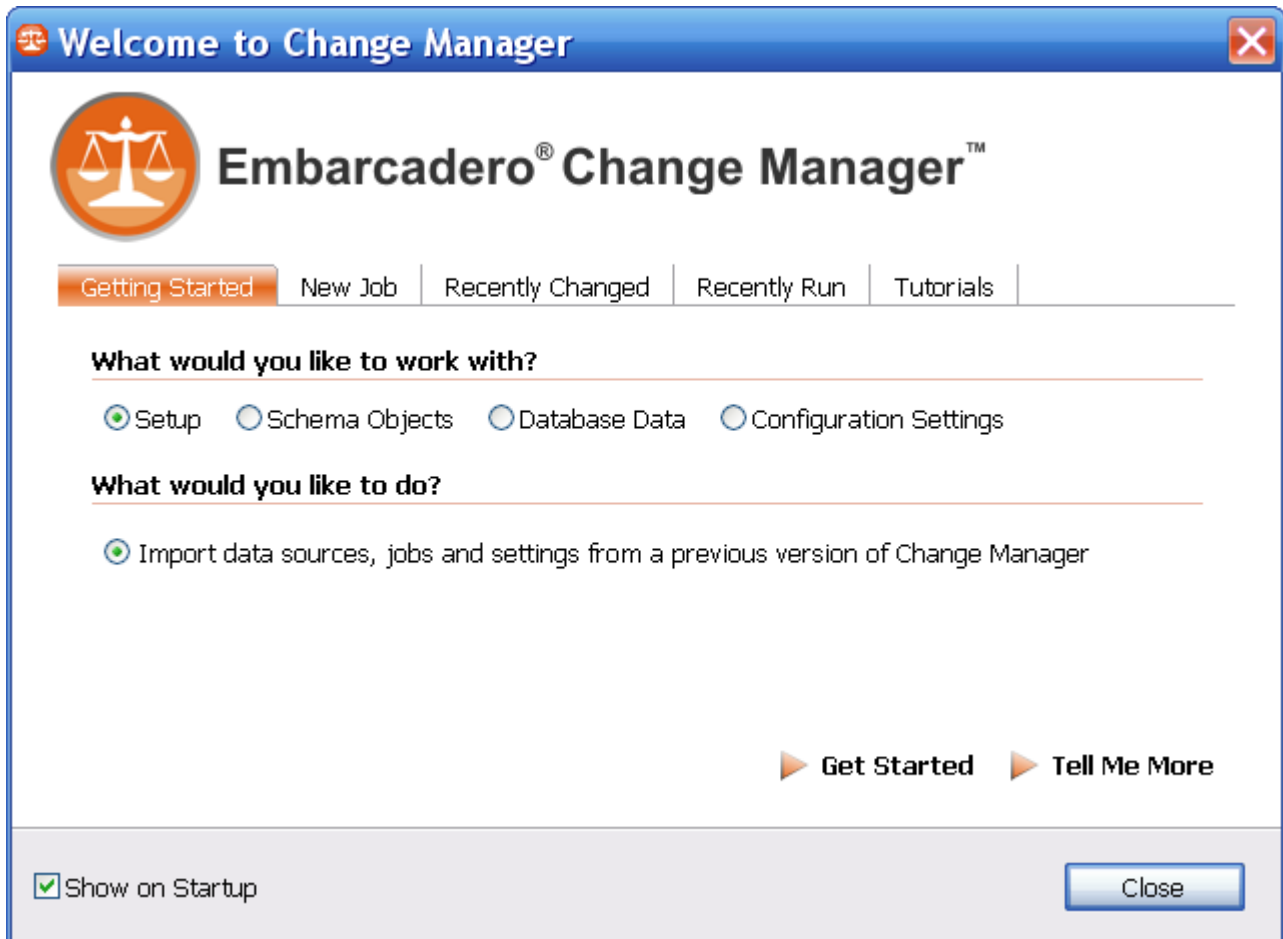
The Workbench space is composed of [Views](#) and [Editors](#), as well as the **Menu Bar** and **Command Toolbar**.

- The **Menu Bar** contains command menus that 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** provides icons that issue common **Menu Bar** commands.



Welcome Page

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

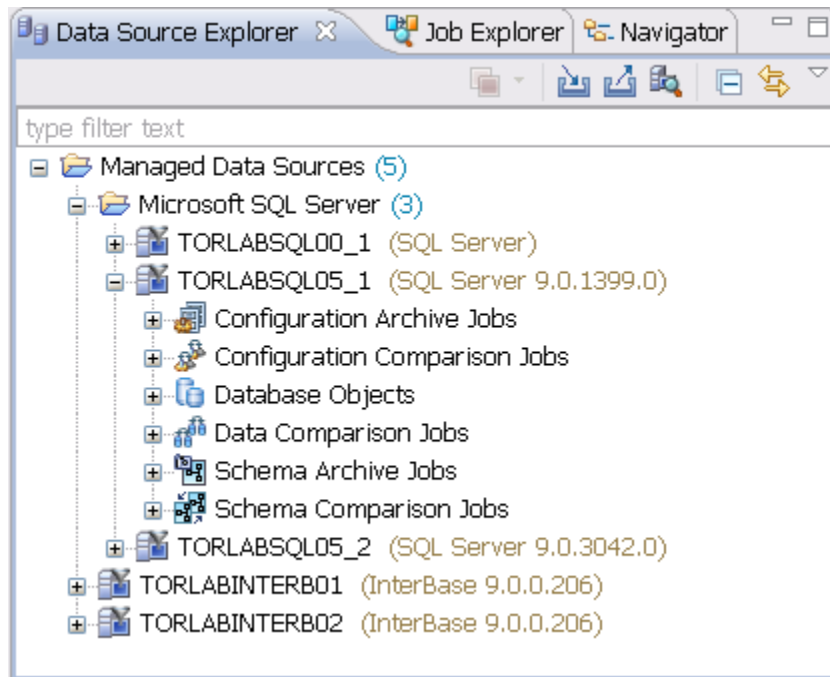


- The **Getting Started** tab presents commonly-performed tasks. Choose the options you want and click **Get Started**. A list of related instructions appears.
- The **New Job** tab provides a list of jobs that you can run. Click the job you want to start, and the appropriate editor opens.
- The **Recently 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 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 Change Manager.

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 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 contain two command menus -- one accessed by right-clicking the view tab, and the second by clicking the icon in the view toolbar. These menus will contain commands pertinent to the selected view.

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:

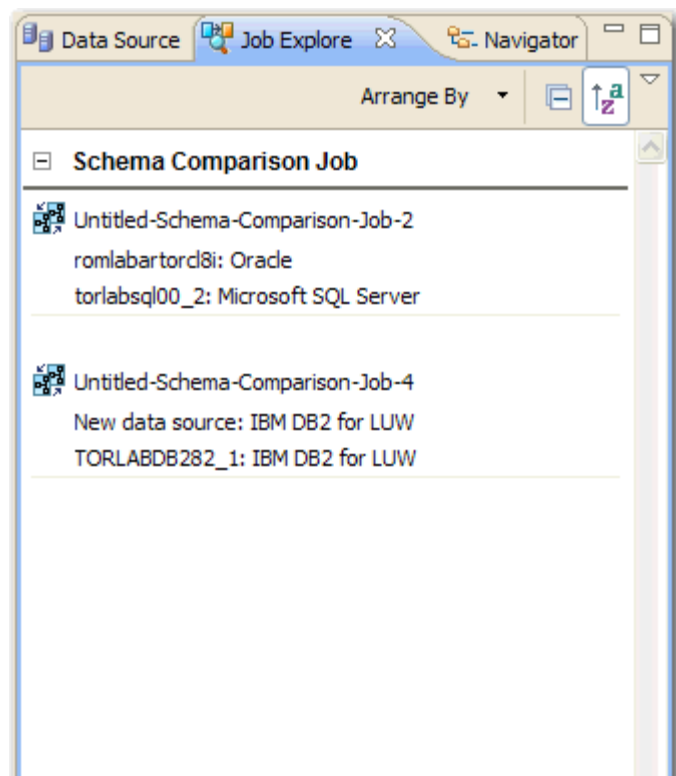
- Right-click the tab of the view and select **Detached**. The view switches to a floating window format that can be used to position it overtop of 'embedded' views in the Workbench.

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 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.

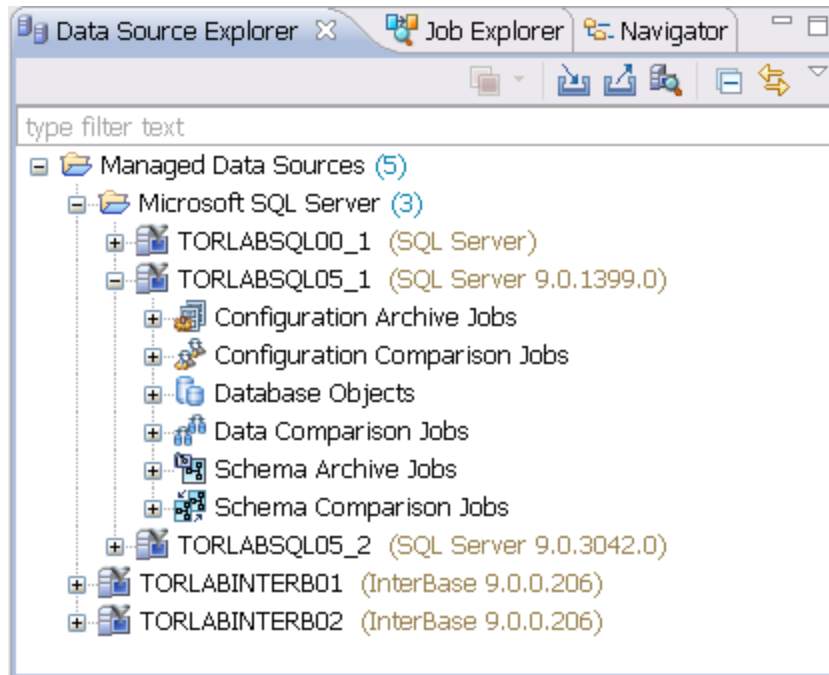
Viewing Change History

Each time a job runs, its history is recorded and in the **Job Explorer**. 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.



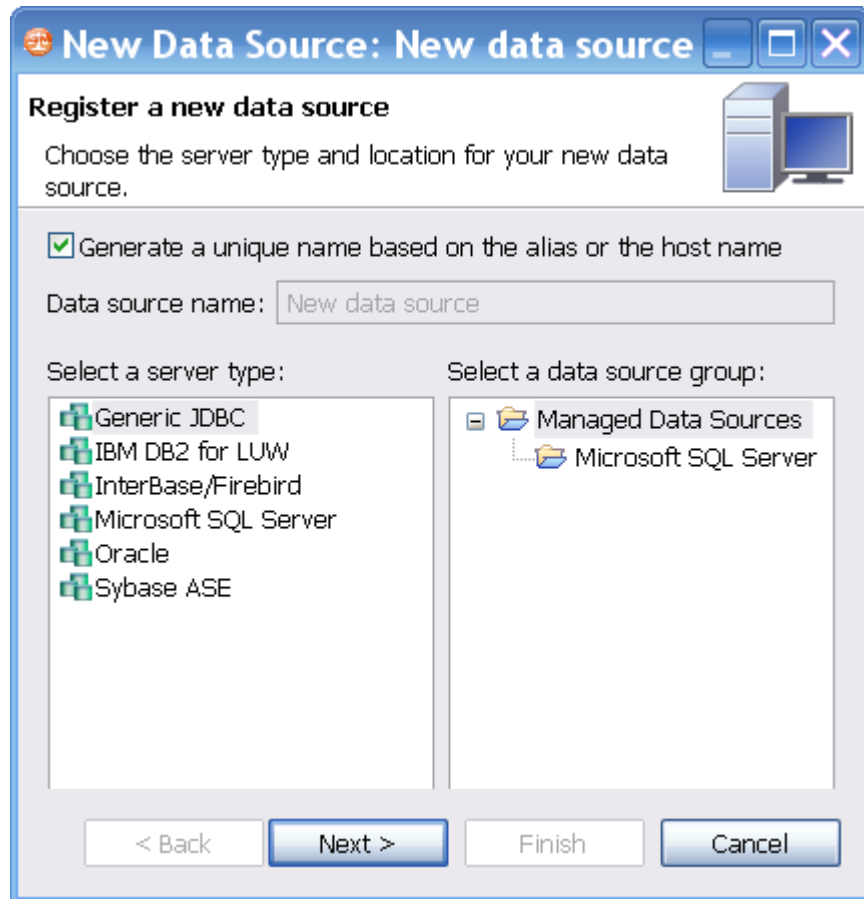
When you register a new data source, you specify its connection information. Once registered, it remains stored in a catalog and does not need to be re-registered. It can be archived and used in multiple jobs.

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:

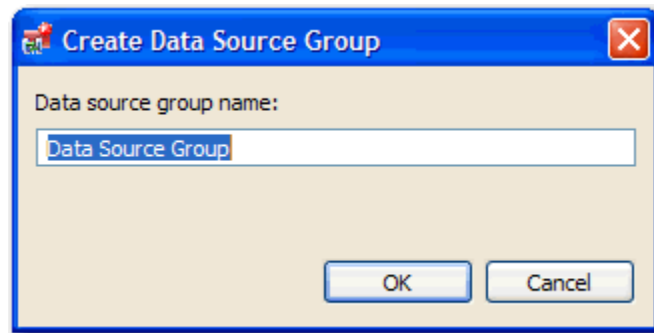
- 1 Right-click on the **Managed Data Sources** node in **Data Source Explorer** and select **New > Data Source**. The **Add Data Source** dialog appears.



- 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:

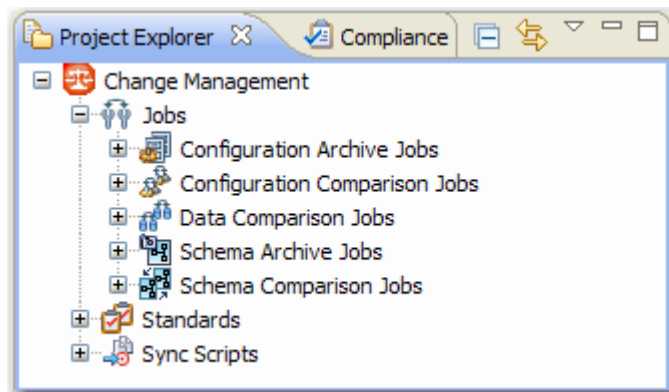
- 1 Right-click on the **Managed Data Sources** node in **Data Source Explorer** and select **New > Data Source Group**. The **Create Data Source Group** dialog appears.



- 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. Files can then be added and deleted under this folder by dragging and dropping them from the **Change Management** branch.

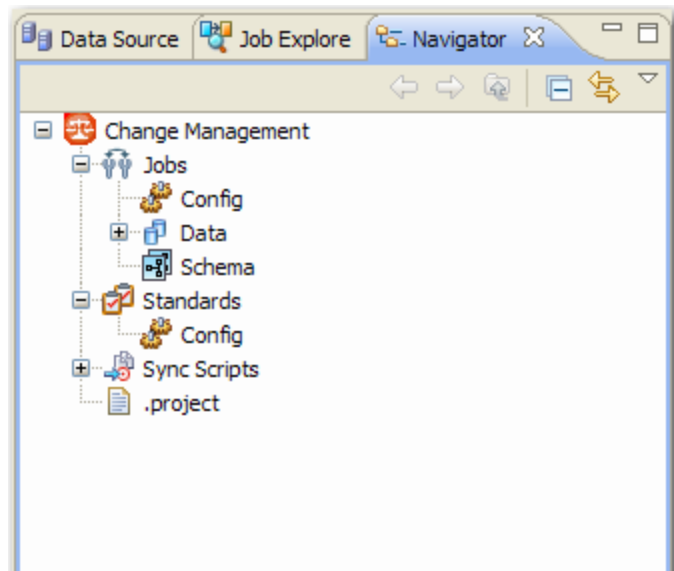
To create a new project:

- 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 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.



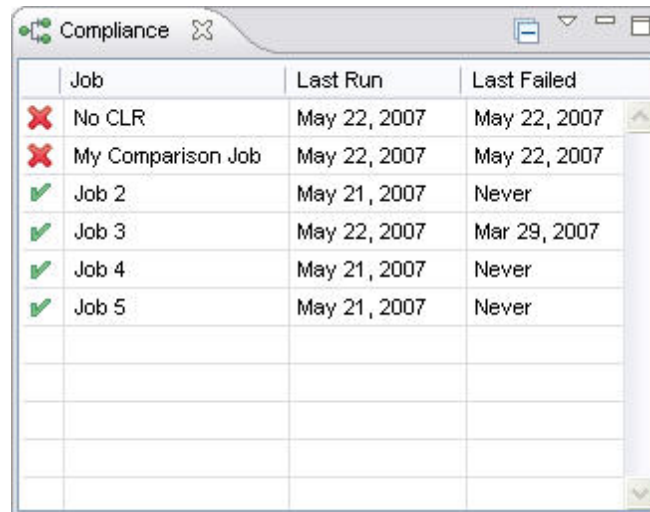
Whenever a job, standard, or synchronization script is generated by the system, its output is saved to a file that is stored in a directory and displayed in the **Navigator**. These files can then be shared in a team environment using a source control system.

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 plug-in 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:

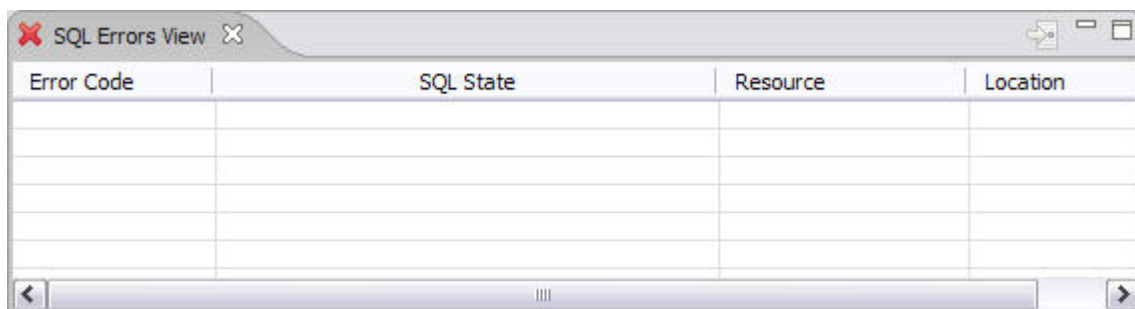
- The day the job was last run
- The day the job last failed, if applicable

By default, jobs are not listed in the **Compliance Explorer**. To list a job, 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.

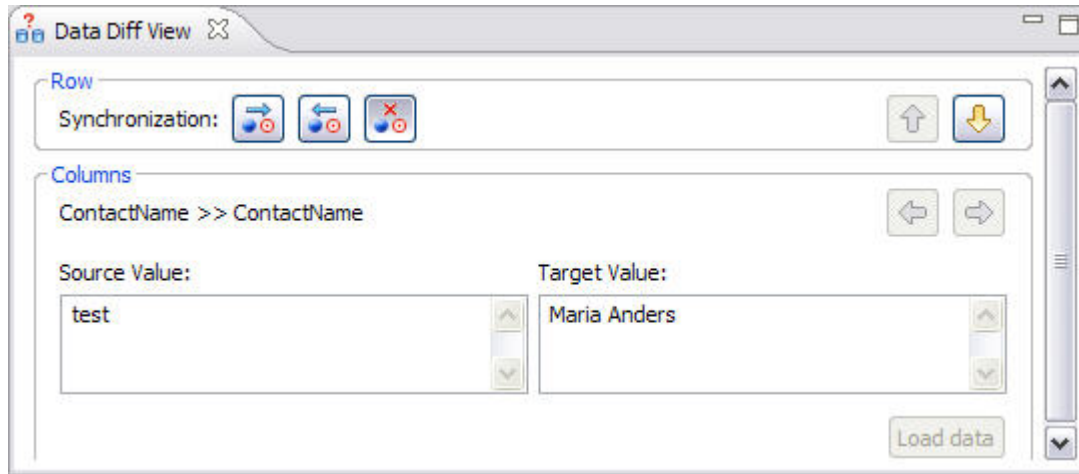


Error Code	SQL State	Resource	Location

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 enables you to specify changes without resolving the repositories of both data sources with synchronization scripts.

SQL Log

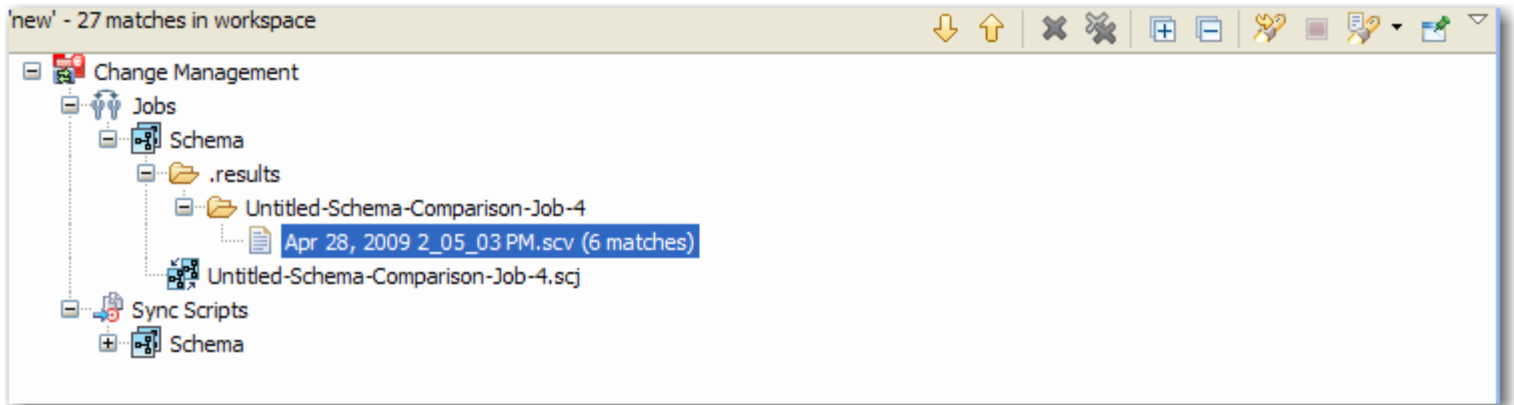
The **SQL Log** captures all SQL commands executed by **SQL Editor** and the system. **SQL Log** entries are listed by **SQL Statement** name, **Date** issued, **Host/Server**, **Service**, **User**, **Source**, and the **Time** (in milliseconds) it took to execute the command.

SQL Errors Log

The **SQL Errors** log automatically logs all SQL errors encountered when SQL commands are executed through 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.

Additionally, 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](#).

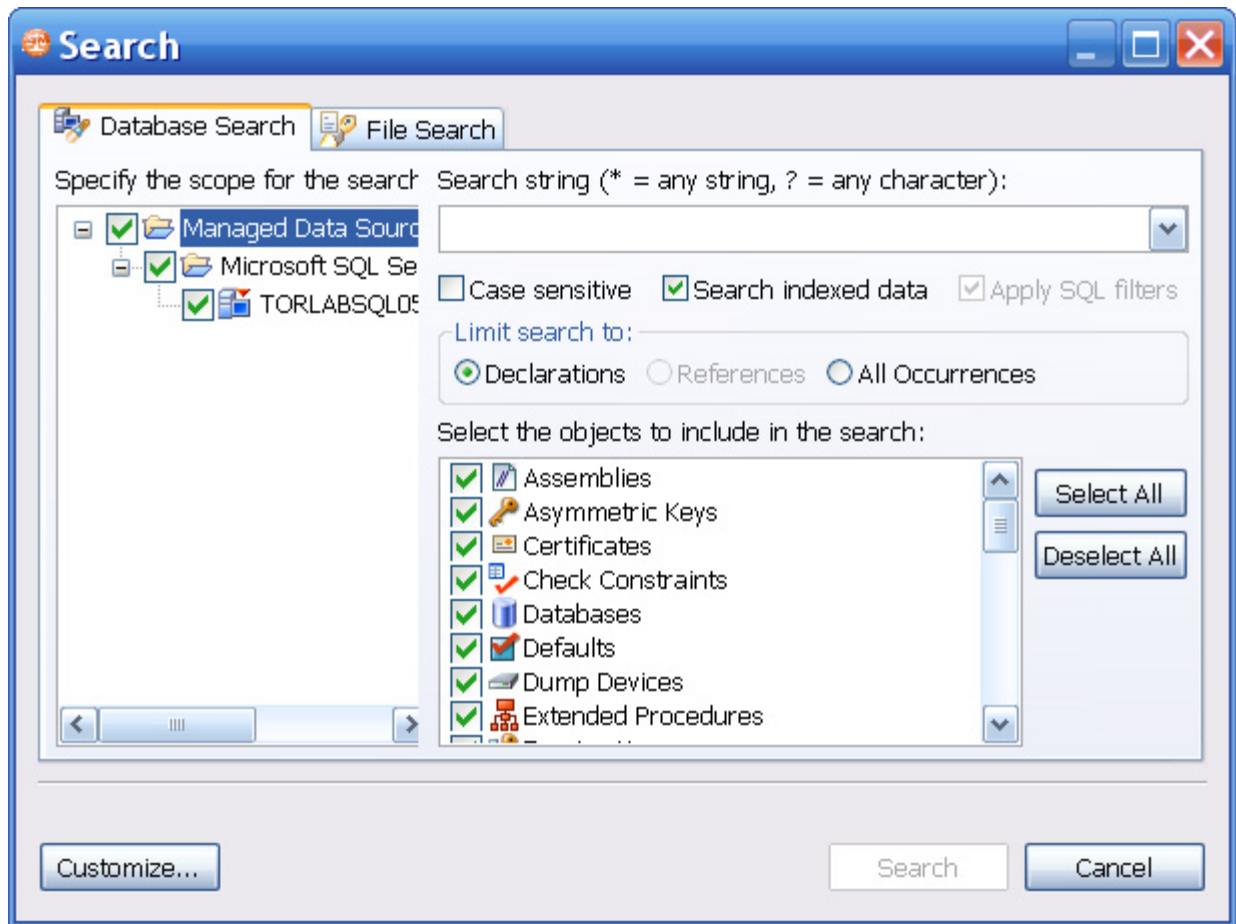
Searching for Database Objects and Synchronization Scripts

The **Search** function provides a means of searching for database objects you have registered, and synchronization scripts that have been generated by 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 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**. 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**. Change Manager returns a list of all possible script selections based on the variables you entered.

Editors

Editors are workbench interface components that enable Change Manager jobs.

- [Data Comparison Job Editor](#)
- [Configuration Comparison Job Editor](#)
- [Configuration Standard Editor](#)
- [Configuration Archive Editor](#)
- [Schema Comparison Job Editor](#)
- [Schema Archive 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. Additionally, the editor provides the ability to specify table mappings and further manipulate data via matching functions that synchronize it at the global, table, and row level, as specified by the user, and provides high-level mapping/refinement options for a job, based on data source (database, schema, or owner selection).

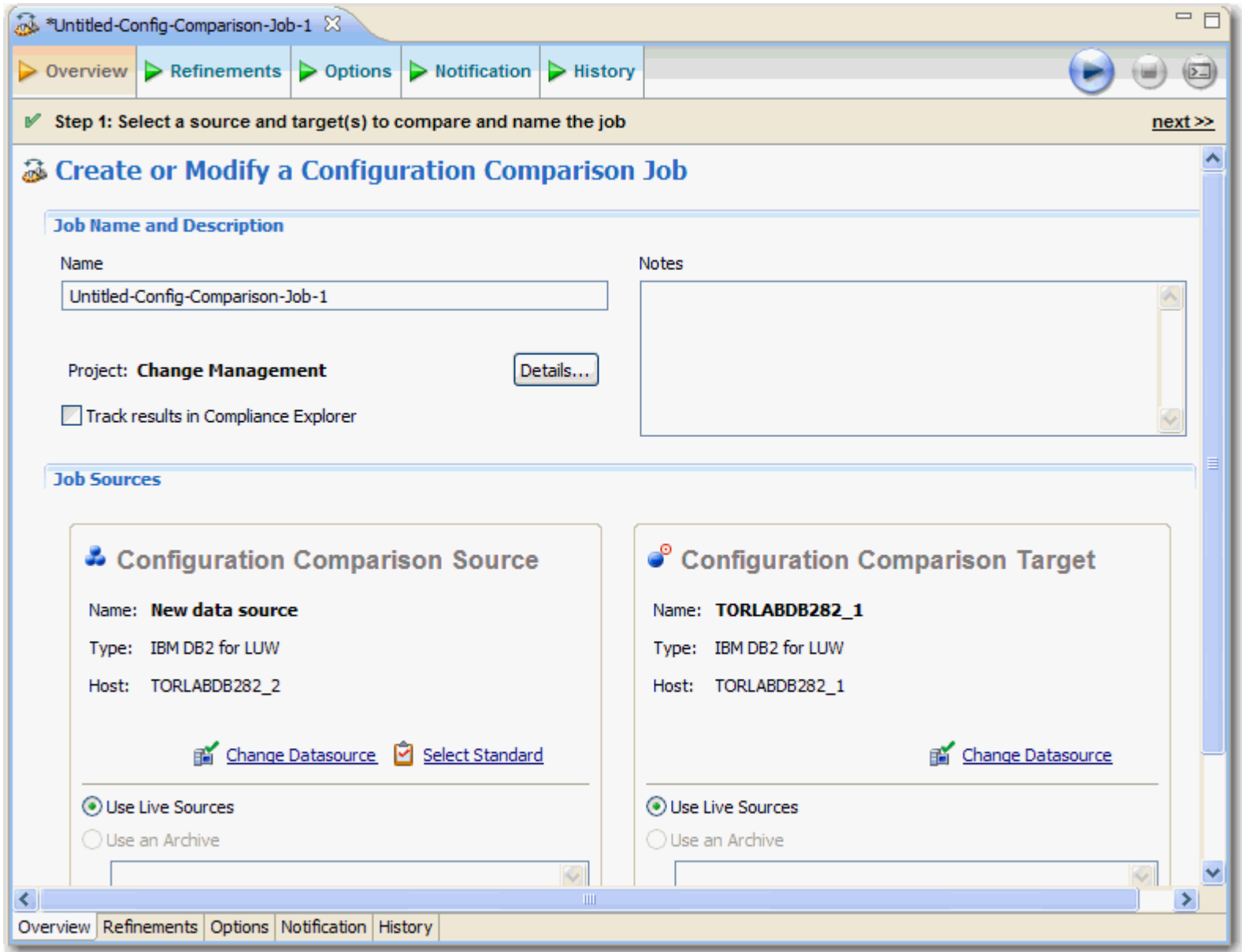
The screenshot displays the 'Data Comparison Job Editor' window for an 'Untitled-Data-Comparison-Job-1'. The interface is divided into several sections:

- Navigation:** A top bar contains tabs for 'Overview', 'Options', 'Mapping', 'Notification', and 'History'. A 'next >>' button is located on the right side of this bar.
- Step Indicator:** A green checkmark and the text 'Step 1: Select a source and target to compare and name the job' are shown on the left, with 'next >>' on the right.
- Job Name and Description:**
 - Name:** A text field containing 'Untitled-Data-Comparison-Job-1'.
 - Notes:** A large text area with a vertical scrollbar.
 - Project:** Labeled 'Change Management' with a 'Details...' button.
 - Track results in Compliance Explorer:** An unchecked checkbox.
- Job Sources:** Two side-by-side panels for configuring data sources and targets.
 - Data Comparison Source:**
 - Name: **New data source**
 - Type: IBM DB2 for LUW
 - Host: TORLABDB282_2
 - Buttons: 'Change Datasource' (with a green checkmark icon)
 - Databases: A dropdown menu.
 - Data Comparison Target:**
 - Name: **TORLABDB282_1**
 - Type: IBM DB2 for LUW
 - Host: TORLABDB282_1
 - Buttons: 'Change Datasource' (with a green checkmark icon)
 - Databases: A dropdown menu.
- Bottom Navigation:** A bar with tabs for 'Overview', 'Options', 'Mapping', 'Notification', and 'History'.

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.

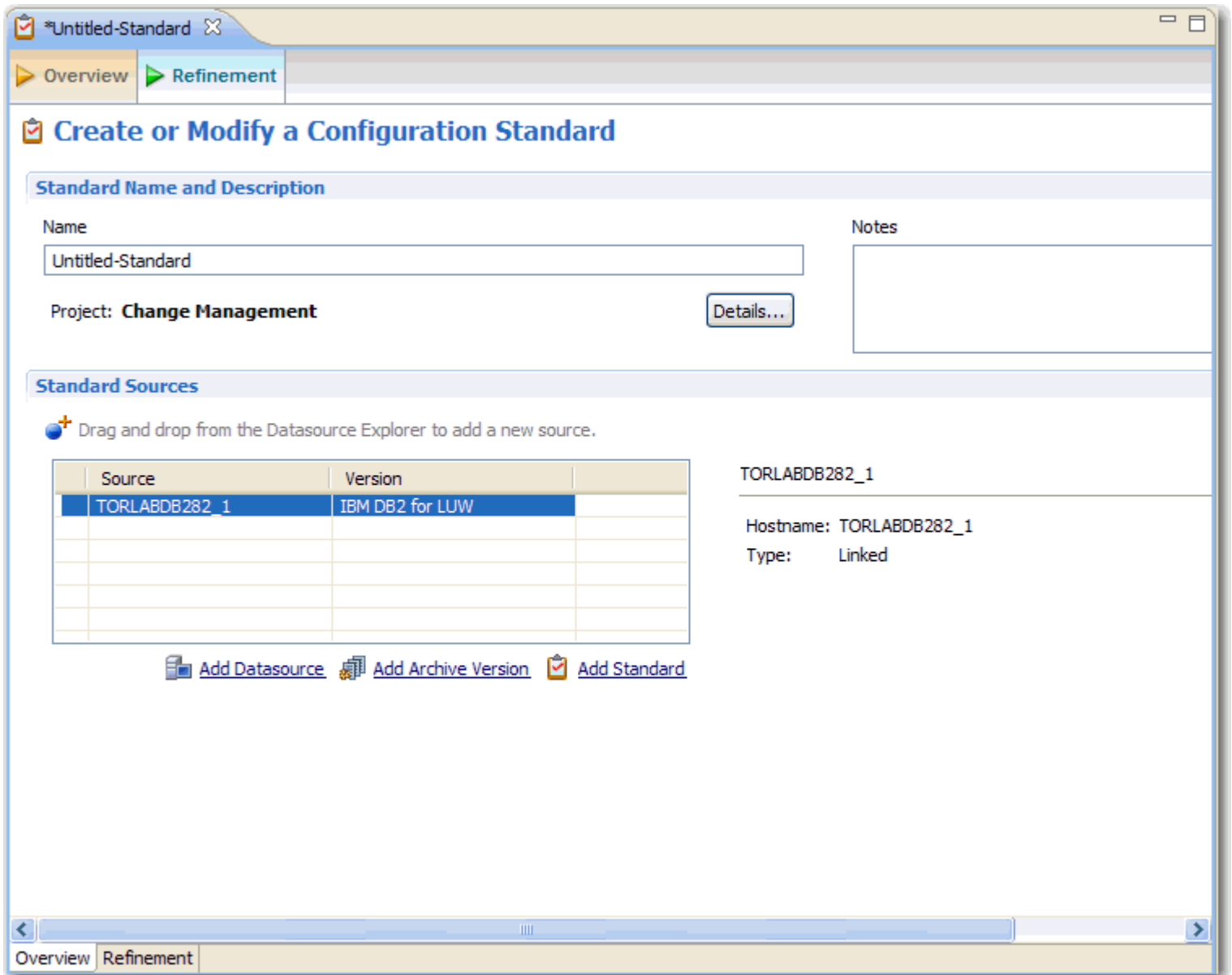


For more information on comparing the configuration settings of two specified data sources, see [Comparing Database Configuration Settings](#).

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.



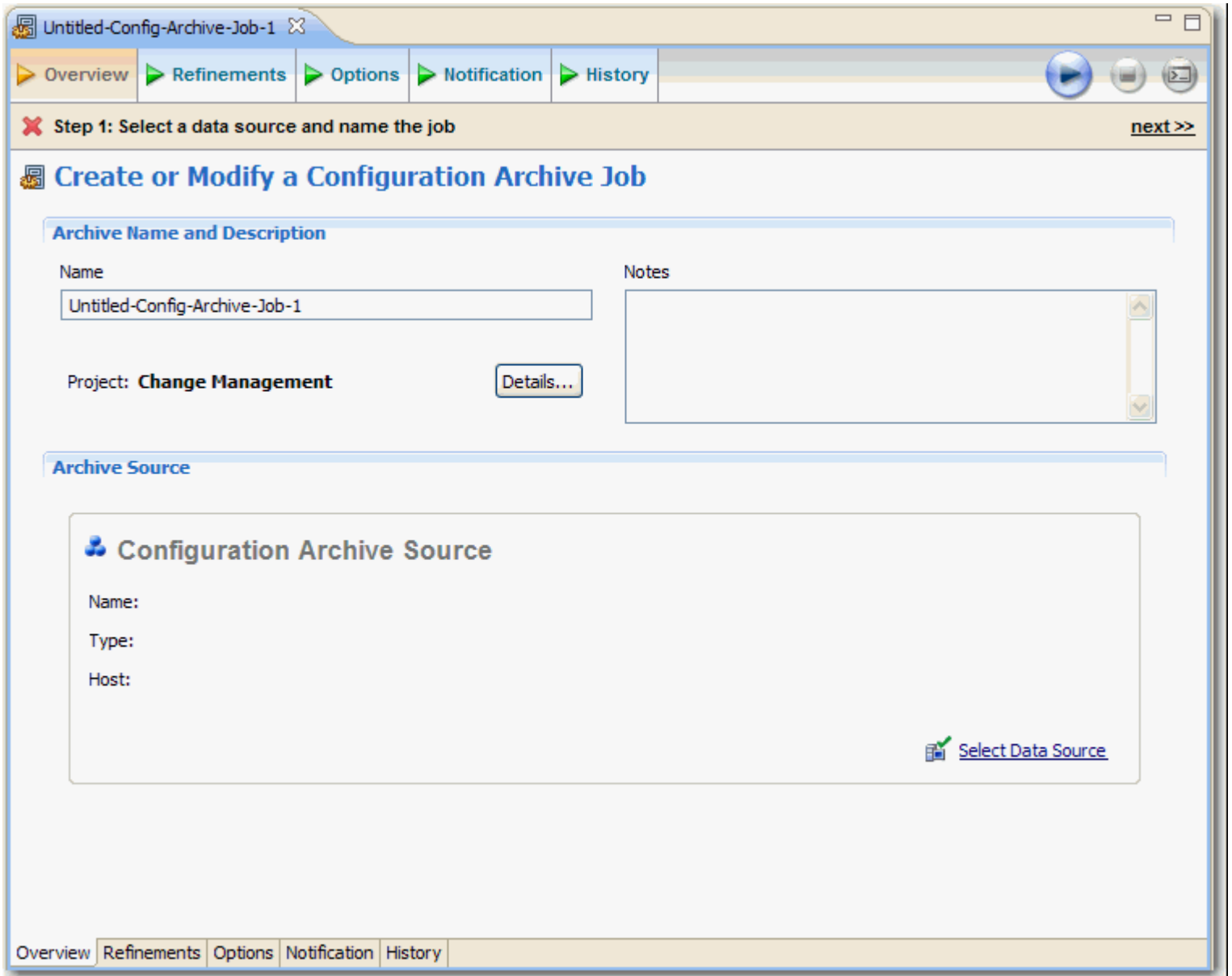
For more information on creating configuration Standards, see [Creating a Standard for Configuration Settings](#).

Configuration Archive Editor

The **Configuration Archive 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 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.

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

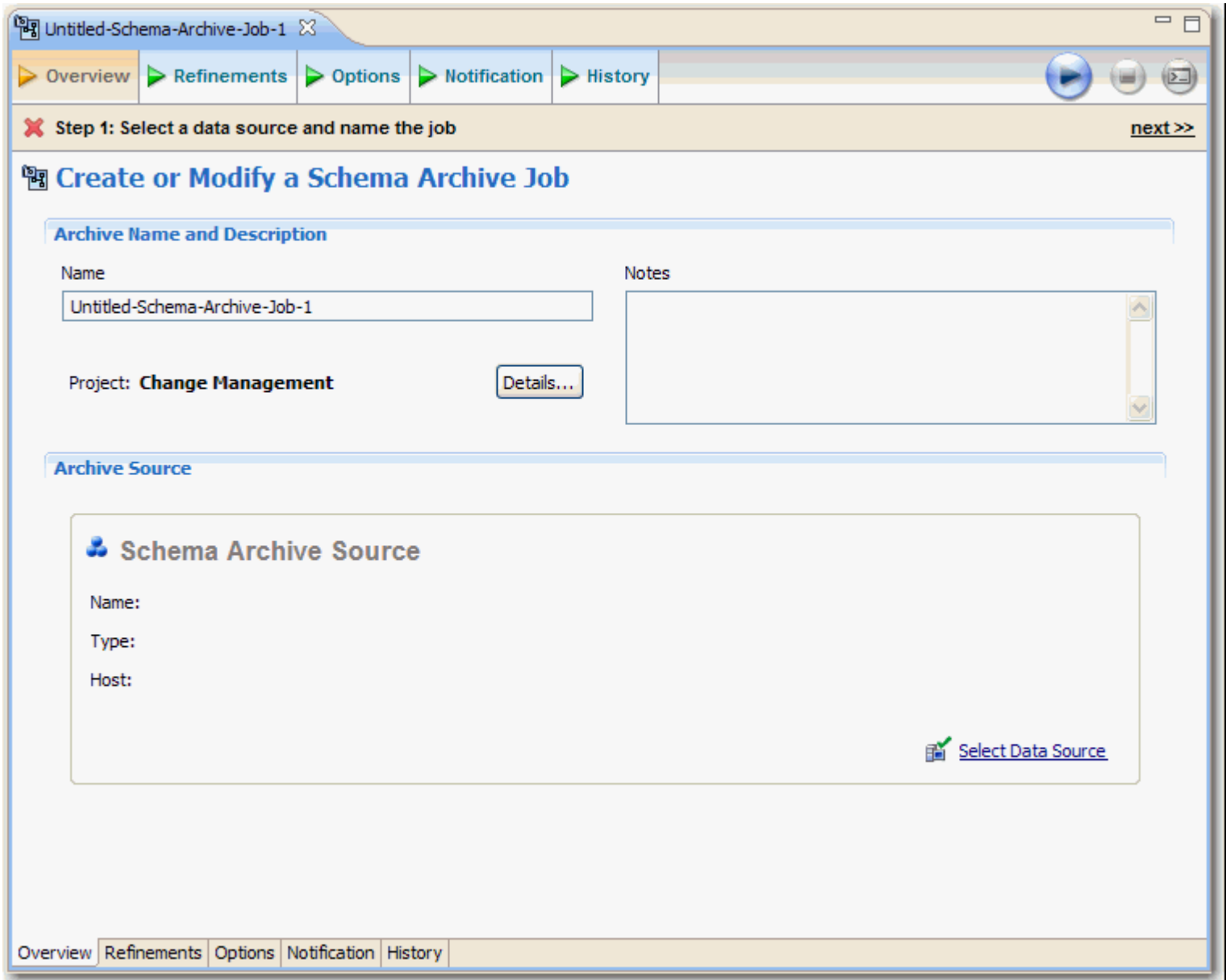
For more information on comparing the schemas of two data sources, see [Comparing Schemas](#).

Schema Archive 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.

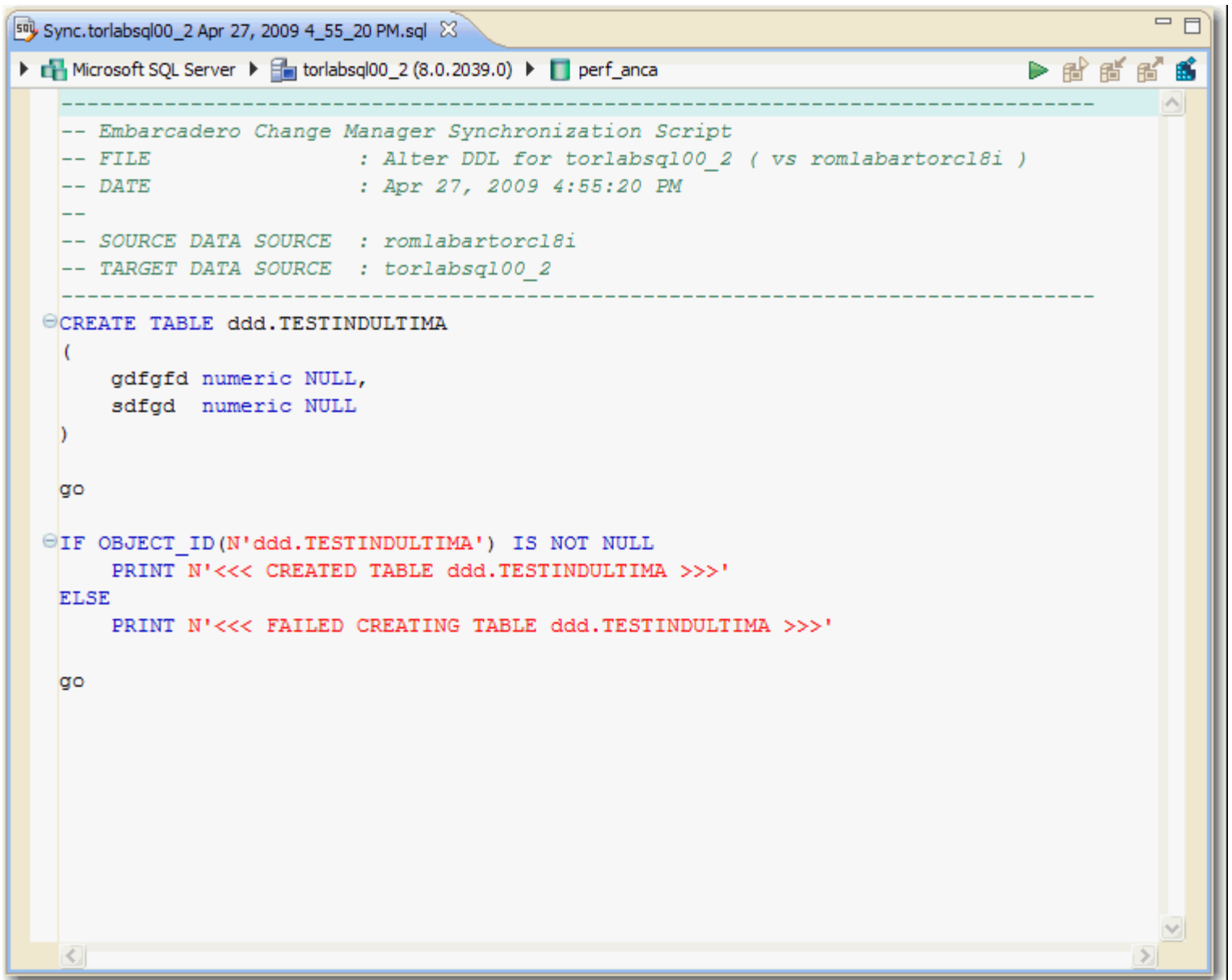
Schema archives are snapshots of existing schema values. Each data source in your environment can contain multiple archive versions. These versions can be used as baselines in schema comparison jobs against other data sources, archives, or standards.

For more information on creating [Creating a Schema Archive](#).

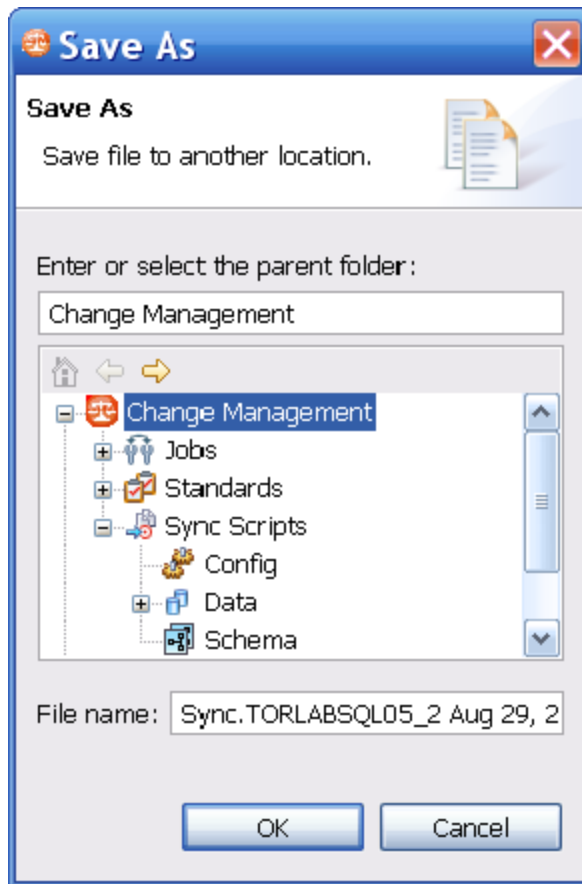


SQL Editor

In the SQL Editor you can view and modify job scripts.



All scripts, when they are generated initially, appear in **SQL Editor**. By default, new scripts are automatically saved and appear in the **Navigator** and **Project Explorer** in a default directory specified by Change Manager. However, scripts can be modified manually and saved via the **File > Save As** menu command, and stored in different directories, as required.



You can rerun scripts that are saved in Change Manager by double-clicking the script name in the **Navigator** view.

Glossary

A

Archive

A set of files, in an Embarcadero Change Manager-specific format, which represent a snapshot image of a database configuration or schema at a given point in time.

B

There are no entries for this letter.

C

Clone

A copy of a data source including all of the properties (thresholds, notifications, etc.) from the original data source.

Compliance Explorer View

A view that displays a history of all jobs that were defined with the **Track results in Compliance Explorer** option.

Configuration Archive Editor

The editor for defining a configuration archive job.

Configuration Comparison Job Editor

The editor for defining a configuration comparison job.

Configuration Standard

A set of values and threshold operators used to check database configurations and identify configuration values that fall outside the pre-defined ranges.

Configuration Standard Editor

The editor for defining a configuration standard.

D

Data Comparison Job Editor

The editor for defining a data comparison job.

Data Diff View

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

Data Source

A reusable connection to a specific database on a specific database server. A data source provides connection information, a user name, and a password to the database environment.

Data Source Explorer

Data Source Explorer lists registered data sources and their associated archives and jobs.

E

Editor

A Workbench interface where you define Change Manager jobs.

F

There are no entries for this letter.

G

Group

A folder that organizes data sources in **Data Source Explorer**. Groups help you organize many data sources.

H

There are no entries for this letter.

I

Imported Archives

Archives created on a different machine (and possibly by a different user) that you have copied into your Change Manager environment.

J

Job

A Change Manager process. Different types of jobs are supported, such as data comparison jobs that compare two or more data repositories. You define jobs, save them, run them, and reuse them in Change Manager.

Job Explorer

A view that lists all jobs defined and saved in Change Manager.

K

There are no entries for this letter.

L

There are no entries for this letter.

M

There are no entries for this letter.

N

Navigator View

A view that lists all files in each of your projects.

Notification

Notifications are messages containing information about a completed job.

O

There are no entries for this letter.

P

There are no entries for this letter.

Q

There are no entries for this letter.

R

There are no entries for this letter.

S

Schema

The definition or structure of data or database objects.

Shell Data Source

An empty data source with only the name and DBMS type defined. A shell data source is created when importing an archive whose data source is not connected. The connection information, user name, and password is intentionally missing from the shell data source for security and data integrity reasons.

SQL Editor

An editor for viewing and modifying SQL synchronization scripts generated by Change Manager.

SQL Errors View

A view that displays any errors encountered while running synchronization scripts.

Synchronization Script

Scripts generated by Change Manager to synchronize the data or schema of two or more data sources.

Synchronization Script Navigator

A view that lists all synchronization scripts generated by comparison jobs. You can modify and run scripts from the Synchronization Script Navigator.

T

There are no entries for this letter.

U

There are no entries for this letter.

V**View**

A Workbench interface for navigating a hierarchy of information, opening editors, or displaying properties.

W**Welcome page**

A window providing access to tutorials, cheat sheets, and other orientation information. By default, the **Welcome page** opens when you start the program.

Wizard

A series of windows that lead you through a process or series of task.

Workbench

The Change Manager desktop development environment. It provides the interface to create, manage, and navigate comparison jobs via an array of views, editors, and menus. The Workbench environment is defined by the Eclipse Foundation.

X

There are no entries for this letter.

Y

There are no entries for this letter.

Z

There are no entries for this letter.

Index

A

- Additional resources 10
- Archive 68
 - New 68
 - Property Refinement 70, 93
 - Save 71, 95
 - Use 72

C

- Change history 170
- Configuration Comparison Job
 - Create 53
- Configuration Standard 63
 - Create 63

D

- Data Comparison Job 35
- Datasource group 20
- Datasources
 - Register 15, 148
- Discussion Groups 10
- Documentation 10

E

- Editors 180

F

- FAQs 10
- Forums 10

J

- Job Execution Options 37
- Job Name 35, 54, 64, 69, 74, 91

K

- Knowledge Base 10

M

- Mapping
 - Concepts 40
 - Customization 42
 - Database 42
 - Table 43

O

- OS 10

R

- Refine Properties 57
- Requirements 9
 - Hardware 9
 - Operating System 10
- Run Job 45, 58

S

- Scripts 177
- Source 36, 55
- Standard
 - Property Refinement 65
 - Property Sources 64
 - Using 66

T

- Target 36, 55

V

- Versioning 71, 95
- View Results 59
- Views 169
 - Move 169
 - Toggle 169

W

- Welcome Page 168
- Workbench 167
- Workbench Space 169