

DB Change Manager 18.0

User Guide

Exported on 09/17/2021

I D E R A

Table of Contents

Getting started	11
Data comparison and synchronization	11
Schema comparison and synchronization	11
Database configuration management	11
Reports, notifications, and change history	11
Data masking.....	11
Source control system integration.....	12
Product requirements and database support.....	12
Eclipse plugin support	13
Upgrade notes.....	13
Upgrading from Version 5.0, 5.1.x, 5.5.x or 5.6 to DB Change Manager 2016 (6.3.3), 2016+ (16.5), 16.6, 17.0, 17.1, or 18.0	14
Upgrading from Version CMSchema 4.0 or CMDData/Config 4.0 to DB Change Manager 18.0	14
Installing DB Change Manager	15
Installing DB Change Manager in an AppWave environment	15
Licensing DB Change Manager	15
Registering a Trial or Workstation license at application startup	15
Registering a Workstation license after application startup	16
Registering by phone	16
Key concepts of DB Change Manager.....	17
Understanding jobs and scripts in DB Change Manager	17
Understanding the DB Change Manager User Interface	17
Comparing and synchronizing data	19
Registering and organizing data sources.....	19
Registering data sources on start up	19
Scanning for and registering data sources	19
Registering a new data source one at a time.....	21
Grouping data sources.....	23
Importing and exporting data sources	23
Filtering data source objects	25
Comparing data	27

Create a Data Comparison Job.....	28
Describe and Save the Job	28
Specify the Source and Target Data Sources.....	29
Specify Execution, Mapping, Comparison, and Resolution Options.....	29
Understand the Data Source Mapping Process.....	30
Customize Database Mapping.....	31
Inspect the Databases.....	31
Customize Table Mapping	32
Reconfigure the Data Source Mapping	33
Set Job Notification Options	33
Run the Data Comparison Job	33
Review Comparison Results	34
Synchronizing data repositories	35
Select Data to Resolve	35
Run the SQL Synchronization Script.....	36
Comparing database configuration settings.....	36
Create or Open a Configuration Comparison Job	37
Describe and Save the Job	37
Specify Configuration Comparison Source and Targets.....	37
Refine property comparisons.....	39
Specify Configuration Comparison Job Options.....	40
Set Configuration Comparison Job Notifications	40
Run the Configuration Comparison Job	40
View the Results of the Configuration Comparison	40
Create a Job Report	41
Synchronize Configurations	41
Execute an SQL Synchronization Script.....	41
Creating a standard for configuration settings	42
Create or open a configuration standard	42
Describe and Save the Standard	42
Add Standard Sources	42
Define Configuration Standard Properties	43
Use a Configuration Standard in Configuration Comparison Jobs.....	44
Creating a configuration archive.....	44
Create or Open a Configuration Archive Job	44

Describe the Configuration Archive Job	45
Specify a Configuration Archive Source.....	45
Refine Configuration Archive Properties	45
Set Configuration Archive Job Notifications	46
Specify Configuration Archive Job Options.....	46
Create a Version and Save the Archive Job.....	47
Create an Archive Job Report.....	48
Use an Archive in a configuration Comparison Job	48
Masking data	48
Create a New Masking Job.....	48
Specify a Masking Job Name	49
Specify Source and Target Data Sources	49
Set to Automatically Synchronize	49
Specify Mapping Options.....	50
Specify Masking Rules for Individual Columns.....	51
Run the Database Masking Job	51
See the Masked Data.....	52
Comparing and synchronizing schemas.....	54
Comparing schemas	54
Create or Open a Schema Comparison Job.....	54
Describe and Save the Schema Comparison Job.....	54
Specify a Schema Comparison Source and Targets.....	55
Cross-Platform Schema Comparisons Supported by DB Change Manager	55
Add Multiple Schema Comparison Job Targets	55
Refine Schema Comparisons.....	56
Customize Schema Mapping.....	57
Specify Schema Comparison Job Options	58
Select Schema Properties to Ignore During a Comparison	59
Set Schema Comparison Job Notifications.....	60
Run the Schema Comparison Job.....	60
View Individual Schema Comparison Results	62
Create a Schema Comparison Job Report.....	64
Cloning a schema comparison job.....	64
Creating a schema archive	66

Create or Open a Schema Archive Job.....	67
Describe and Save the Schema Archive Job.....	67
Specify a Schema Archive Job Source	67
Refine Schema Archive Properties	67
Specify Schema Archive Job Options	68
Include Objects Options	68
DDL Extraction Options	68
Automatic Purge Options	69
Set Schema Archive Job Notifications	69
Create a Version and Save the Schema Archive Job	69
Browse and Modify Schema Archive Contents	69
Review the History of Archived Objects	71
Create a Job Report	72
Use an Archive in Schema Comparison Jobs.....	72
Synchronizing schemas	73
Resolve Schema via a Generated SQL Script.....	73
Execute a SQL Synchronization Script.....	74
Configure an SQL Session.....	75
Working with jobs	77
Creating a job report.....	77
Scheduling a job.....	79
Setting job notifications	80
Email Notifications.....	81
File System notification for Log Files	82
System Tray Notifications.....	82
Send Notification and Attachment Options	82
Exporting and importing jobs.....	83
Export a Job	83
Import a Job	85
Automating batch scripts	86
Generating command line scripts	86
Using the changec Command Line Program	87
Database management system reference	89
DBMS Operations Permissions	89

Permissions Required for DB2 for LUW.....	89
Permissions Required for Oracle.....	90
Permissions Required for SQL Server	90
Permissions Required for Sybase.....	91
Version Information Reference	92
Version Information for DB2 for LUW	92
Version Information for Oracle.....	93
Version Information for SQL Server	93
Version Information for Sybase.....	94
Preferences reference.....	95
DB Change Manager Preferences	95
CM/Config > Archive Options.....	96
CM/Config > Comparison Options.....	96
CM/Data > Comparison Options.....	96
Execution Options.....	97
Mapping Options.....	97
Comparison Options.....	97
Resolution Options	98
CM/Data > Data Result Options	98
CM/Data > General	98
CM/Data > Mapping Validation	99
CM/Data > Script Generation.....	100
CM/Schema > Archive Options	100
CM/Schema > Comparison Options	100
CM/Schema > Comparison Options > Include Objects Options	101
CM/Schema > Comparison Options > Compare Options	101
CM/Schema > Comparison Options > Decorator Options.....	102
CM/Schema > Comparison Options > Identifier Options	102
CM/Schema > Comparison Options > DDL Extract Options.....	102
CM/Schema > Comparison Options > Additional Options	103
CM/Schema > General.....	103
CM/Schema > Mapping Options	103
CM/Schema > Refinements Options	103
Command Line.....	103

Compliance	104
Label Decorations	104
Notification > Email Notification	104
Notification > Email Notification > Advance Settings	104
Notification > Email Notification > Contacts	105
Notification > Email Notification > Templates.....	105
Notification > File System Notifier	105
Notification > Popup Notification	105
Notification > Popup Notification > Advanced Settings.....	106
Projects.....	106
Reports	106
Script Execution Options	107
System Tray.....	107
Task Scheduler.....	107
Data Sources	108
Eclipse SQL Development Preferences	108
SQL Development > SQL Editor.....	108
SQL Development > SQL Editor > Code Assist	108
SQL Development > SQL Editor > Code Formatter.....	109
SQL Development > SQL Editor > Results Viewer	109
SQL Development > SQL Editor > Syntax Coloring.....	110
SQL Development > SQL Filters.....	110
User interface reference	111
DB Change Manager Workbench.....	111
License Selection Dialog.....	113
Welcome Page.....	113
Workspace	113
Views	114
Job explorer	115
Viewing change history.....	115
Data Source Explorer	115
Data Source Properties.....	117
Project Explorer.....	119
Navigator view	120

Compliance Explorer	121
SQL Errors view	121
Data Diff view	122
Extract DDL Scripts.....	122
SQL Log.....	123
SQL Errors Log.....	123
Search view	123
Searching for database objects and synchronization scripts.....	124
Editors.....	125
Data Comparison Job Editor	125
Configuration Comparison Job Editor	126
Configuration Standard Editor	127
Configuration Archive Job Editor	128
Schema Comparison Job Editor.....	129
Schema Archive Job Editor.....	130
SQL Editor.....	131
SQL Session Options.....	133
Eclipse plugin extensions reference.....	135
Creating and implementing a notification plugin	135
Create a plugin project	136
Configure the target platform	136
Define dependencies	136
Implement a notifier class	137
Implement a viewer contribution	138
Specify setup constants.....	139
Implement the DirectoryNotifier class.....	139
Implement the DirectoryNotifierContribution class	141
Register the extensions.....	143
Deploy the Notifier plugin	143
Creating and implementing a custom configuration plugin	144
Create a project plugin	144
Configure the target platform	144
Define dependencies	144
Implement a property source.....	145

Determine support.....	145
Define properties	145
Gather results.....	148
Synchronize the job	148
Register extensions.....	149
Deploy the custom configuration plugin	149

Use DB Change Manager to:

- Quickly roll out and reconcile database changes
- Reveal, track, and report on database changes
- Comply with database audit and reporting requirements
- Protect data privacy within the database environment
- Track changes from multiple major database platforms

Getting started

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

- [Data comparison and synchronization](#)
- [Schema comparison and synchronization](#)
- [Database configuration management](#)
- [Reports, notifications, and change history](#)
- [Data masking](#)
- [Source control system integration](#)

Data comparison and synchronization

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

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

Schema comparison and synchronization

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

Database configuration management

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

Reports, notifications, and change history

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

Data masking

There are specific scenarios when you need to hide the true values of the data in a database. For example, during development you need to protect personal or proprietary data, yet you still need to work with data that accurately

represents the source. By combining data synchronization with data masking features, such as randomization or shuffling, you can duplicate source data and use the masked copies in non-production or non-secure environments.

Source control system integration


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

Product requirements and database support

Type	Platform	Version
CPU		<ul style="list-style-type: none"> • 1+ GHz
Random Access Memory (RAM)		<ul style="list-style-type: none"> • 3 GB
Storage Disk Space		<ul style="list-style-type: none"> • 1 GB
Operating System	Microsoft Windows	<ul style="list-style-type: none"> • 10 • 8.1 • 8 • 7
	Microsoft Windows Server	<ul style="list-style-type: none"> • 2019 • 2016 • 2012 R2 • 2012 • 2008 SP1 • 2008 R2
Cloud	<ul style="list-style-type: none"> • Amazon Elastic Compute Cloud (EC2) • Microsoft Azure Virtual Machine (VM) 	
Database	IBM Db2 LUW and Z/OS	<ul style="list-style-type: none"> • 11.x* • 10.x

Type	Platform	Version
	Microsoft SQL Server	<ul style="list-style-type: none"> • Azure* • 2019* • 2017* • 2016* • 2014 • 2012 SP3 • 2008
	Oracle	<ul style="list-style-type: none"> • 21c* • 19c* • 18c* • 12c R2* • 11g R2* • 10g*
	Sybase ASE	<ul style="list-style-type: none"> • 16.x* • 15.x

* DB Change Manager supports only a subset of features and functions for this database version.

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

Eclipse plugin support


If you choose to install the Eclipse plugin version, you also need the following items:

- Eclipse 3.6+
- Sun Java Standard Edition 6, Update 45+

Upgrade notes

- DB Change Manager 17.1 should be installed in a different directory than 4.x, 5.x, 6.0.x, 6.1, 6.2, 6.3.x, 2016 (6.3.3), 2016+ (16.5), 16.6, and 17.0 versions. If DB Change Manager is installed in the same directory, the following error will result at application start up: "Unable to read workbench state. Workbench UI layout will be reset."
- If one of the conflicting versions is already installed on your machine, either change the default Workspace or delete all files from the Workspace directory. The Workspace directory location appears whenever you start DB Change Manager and can be redefined from this dialog box. If the same Workspace is indicated for the new version, as was indicated previously, the following error will result: "Cannot restore workbench layout."
- If DB Change Manager is already running, you can also modify the Workspace directory by choosing **File > Switch Workspace > Other** from the Main menu.
- The format in which schema comparisons are saved has changed in 6.x. Use the import feature to import pre-6.0 schema comparison jobs.

- Beginning with version 17.0, DB Change Manager provides an upgrade process for moving jobs and other information from an existing DB Change Manager 4.0, 5.0, 5.1.x, 5.5.x or 5.6 installation. No upgrade process is required to move from 6.x installations to DB Change Manager 17.1.

 You do not have to upgrade to 5.6, and then to DB Change Manager 17.1. The import procedure is the same.

Upgrading from Version 5.0, 5.1.x, 5.5.x or 5.6 to DB Change Manager 2016 (6.3.3), 2016+ (16.5), 16.6, 17.0, 17.1, or 18.0

All objects can be imported from version 5.0, 5.1.x, 5.5.x or 5.6 to DB Change Manager 2016 (6.3.3), 2016+ (16.5), 16.6, 17.0, 17.1, or 18.0.

To upgrade from 5.0, 5.1.x, 5.5.x or 5.6:

- Make sure that the earlier product is not running and that no command line jobs are being processed.
- Install and run DB Change Manager 18.0 in a different directory than the previous installation.
- The Welcome to DB Change Manager dialog opens with the Setup and Import data sources, jobs, and settings from a previous version import options selected.
- Click **Get Started** to launch the Import Wizard. Alternatively, you can launch the Import Wizard from the main menu using the following steps:
 - Select **File > Import**.
 - Select **Change Manager Workspace** from the IDERA group, and then click **Next**.
- Select the version of DB Change Manager from which to import.
- Select the location of your DB Change Manager workspace. The default value is the default workspace, so you need to change this value only if you configured DB Change Manager to store the workspace in a custom location. The wizard imports the items from the previous DB Change Manager version.
- Click **Next** to continue through each step of the wizard.
- Click **Finish**. The jobs are imported to the latest version of DB Change Manager.


Upgrading from Version CMSchema 4.0 or CMData/Config 4.0 to DB Change Manager 18.0

The following objects can be imported from version 4.0:

- data sources
- all comparison jobs (data, configuration, and schema)
- configuration archives
- configuration standards

The following objects cannot be imported from version 4.0:

- schema archive versions
- synchronization scripts
- reports
- command line automation (scheduled jobs, batch files, etc.)

 4.0 should remain installed to access any archives and reports.

To upgrade from CM Schema 4.0:

- To import schema comparison jobs from 4.0, obtain and install *cmschema_40_932_upgrade.exe* from your IDERA Technical Support representative.
- Run *cmschema_40_932_upgrade.exe* to convert the schema component jobs from version 4.0 to 5.0, which uses an XML-based format.

3. Follow the steps as described in the previous section, Upgrading from Version 5.0, 5.1.x, 5.5.x or 5.6 to DB Change Manager 2016 (6.3.3), 2016+ (16.5), 16.6, 17.0, 17.1, or 18.0.

Installing DB Change Manager

To install DB Change Manager on a workstation, run the installer and follow the prompts provided by the Installation Wizard. When the installation is complete, see the ReadMe file for known issues or advisory notes that could affect your use of the product.

Installing DB Change Manager in an AppWave environment

If you are a DB Change Manager user and your System Administrator provides you with AppWave, then your System Administrator enables DB Change Manager in your AppWave Browser. Consult your System Administrator and the AppWave Browser online help for details.

Licensing DB Change Manager

DB Change Manager supports the following types of licenses:

- **Trial licenses.** A license for a time-limited, full-featured trial version of the product. You must register the trial license before you can use the product.
- **Workstation licenses.** A license or set of licenses that is tied to a particular workstation. The product can only be used on that workstation.
- **AppWave licenses.** AppWave licenses are normally concurrent licenses managed by your System Administrator. See your System Administrator for details.

Registering a Trial or Workstation license at application startup

Shortly after downloading a trial version of an IDERA product, you should receive an email from IDERA containing a serial number, which must be registered. These instructions also apply if you have purchased a product and received a serial number prior to running the application for the first time, or if the trial period expired.

To register a serial number at application startup

1. Start the application. The License Registration dialog appears.
2. Copy the serial number from the email, and then paste it into **Serial Number**.
3. Enter your Developer Network (EDN) account credentials in the **Login**, or in the **Email** and **Password** fields. If you have not previously created an EDN account, or have forgotten your password, click **I need to create ... or I've lost my password**.
4. Click **Register**.


The activation file should automatically download and install. If this does not happen, click the **Trouble Connecting? Try Web Registration** link and follow the prompts.

Registering a Workstation license after application startup

The following instructions assume that you have received a workstation license by email and that you currently have a valid trial license. If you did not install a trial version or if the trial period has expired, follow the instructions in [Registering a Trial or Workstation License at Application Startup](#) instead.

To register a serial number after application startup

1. Click **Help > Launch License Manager**.
The License Manager dialog appears.
2. In the License Manager dialog, click **Serial > Add**.
3. Copy the serial number from the email, and then paste it into the Add Serial Number dialog.
4. Click **OK**.
5. Expand Unregistered Serial Numbers, right-click the serial number just added, and then click **Register**.
A registration dialog appears.

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

6. Ensure that **Register using Online Registration** is selected.
7. Provide Developer Network credentials in the **Login** field, or into the **Email** and **Password** fields.
8. Click **Register**.
The activation file should download and install automatically. If this does not happen, click the **Trouble Connecting? Try Web Registration** link and follow the prompts.

Registering by phone

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

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

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

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

After phoning in, you will receive an email containing an activation file to import into DB Change Manager.

To import an activation file

1. Save the activation file from the email to the desktop or to a scratch directory such as c:\temp.
2. From within the application, click **Help > IDERA Licensing > License Registration**.
3. Click **License > Import**.
4. Navigate to the directory where you saved the activation file and then double-click the activation
5. Click the **Import** button to import the activation
6. Click **Finish**.

Key concepts of DB Change Manager

- [Understanding jobs and scripts in DB Change Manager](#)
- [Understanding the DB Change Manager User Interface](#)

Understanding jobs and scripts in DB Change Manager

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

- **Data source.** A data source is a data repository or database.
- **Configuration standard.** A standard is a customized set of properties with threshold operators that help you create optimal configurations against which you can compare other data source configurations.
- **Schema or configuration archive.** An archive is a snapshot of a configuration or schema.

You can use this archive to compare against other schemas or configurations.

There are two types of jobs in DB Change Manager:

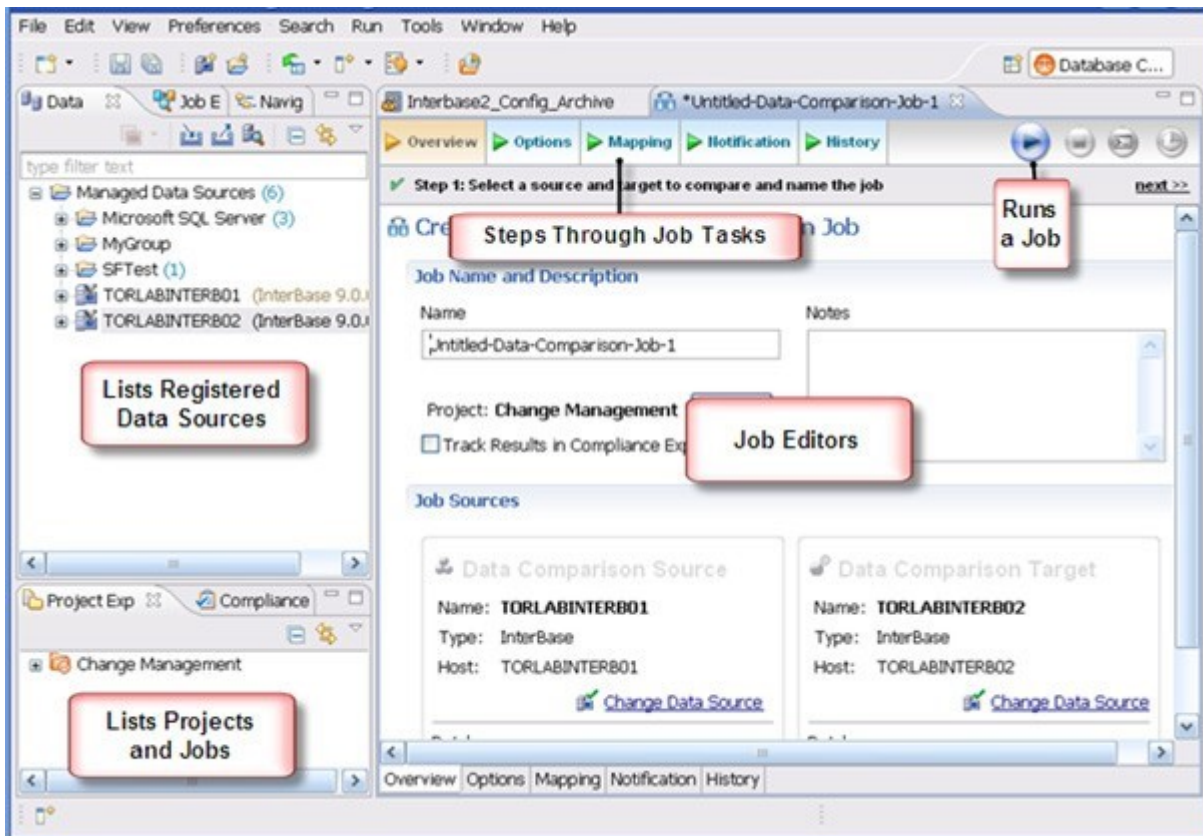
- **Object creation jobs.** Some jobs create objects that are used in other jobs. You use a job to create configuration standards and archives.
- **Comparison jobs.** Processes source and target objects and provides information describing their similarities and differences. This comparison is then used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

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

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

Understanding the DB Change Manager User Interface

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



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

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

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

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

Comparing and synchronizing data

- [Registering and organizing data sources](#)
- [Filtering data source objects](#)
- [Comparing data](#)
- [Synchronizing data repositories](#)
- [Comparing database configuration settings](#)
- [Creating a standard for configuration settings](#)
- [Creating a configuration archive](#)
- [Masking data](#)

Registering and organizing data sources

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

Once registered, the data source appears in the Data Source Explorer view. If you have created more than one workspace, they all share the same data source catalog. You can organize data sources into groups and subgroups. Once a data source has been registered, the connection parameters are stored locally. In some cases, a user ID and password are required to connect to a registered data source. DB Change Manager can encrypt and save user IDs and passwords to connect automatically.

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

- [Registering data sources on start up](#)
- [Scanning for and registering data sources](#)
- [Registering a new data source one at a time](#)
- [Grouping data sources](#)
- [Importing and exporting data sources](#)

Registering data sources on start up

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

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

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

Scanning for and registering data sources

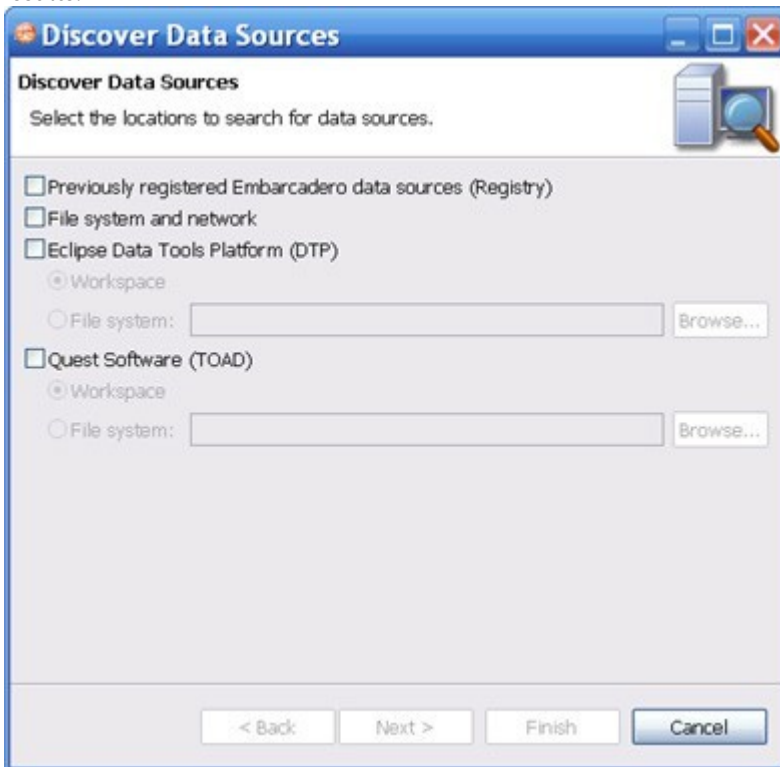
DB Change Manager can register data sources by scanning your systems.

To initiate a scan for data sources

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



2. In the Discover Data Sources dialog, choose the type of data sources that you want returned in the search results.

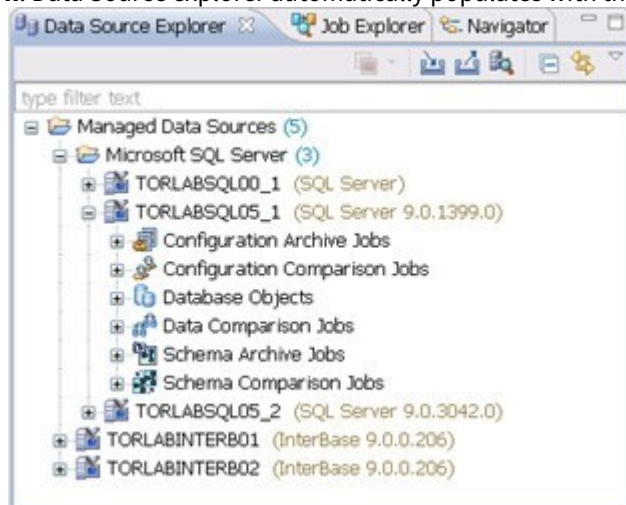


3. Click **Next**.

The search returns a list of the discovered data sources.



4. Choose the data source(s) that you want to add to the DB Change Manager environment, and then click **Finish**. Data Source explorer automatically populates with the new data source selections.

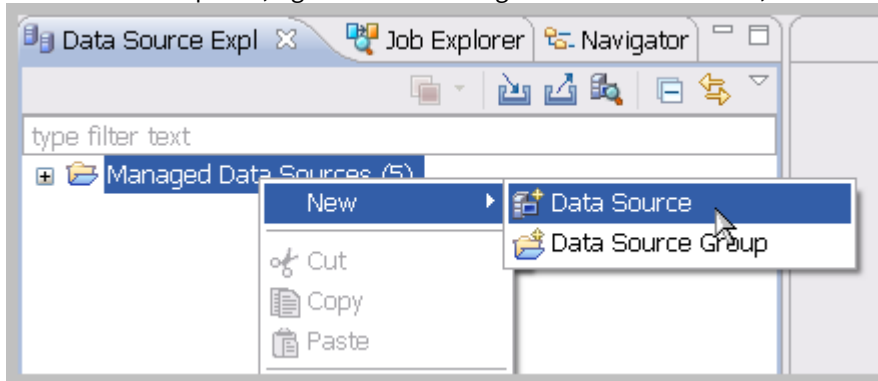


Registering a new data source one at a time

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

To register a new data source

1. In Data Source explorer, right-click the Managed Data Sources folder, and then 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 and want to add this data source to that group, select the appropriate group in this window. Otherwise, leave the default selection, **Managed Data Sources**.
4. Click **Next**.

5. Type the connectivity details and login credentials, and then click **Test Connection** to confirm that the configuration information is correct.



6. Click **Finish**. Data Source explorer now lists the new data source.

Grouping data sources

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

To create a data source group

1. In Data Source Explorer, right-click the **Managed Data Sources** folder and select **New > Data Source Group**.
2. Type a descriptive name for the new group in the **Name** field, and then 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. Type a descriptive name for the new group in the **Name** field, and then click **OK**.

Importing and exporting data sources

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

- Eclipse Data Tools Platform (DTP)
- Previously-registered IDERA Data Sources (File)
- Previously-registered IDERA Data Sources (Registry)
- Quest Software (TOAD)

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

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

To import data sources

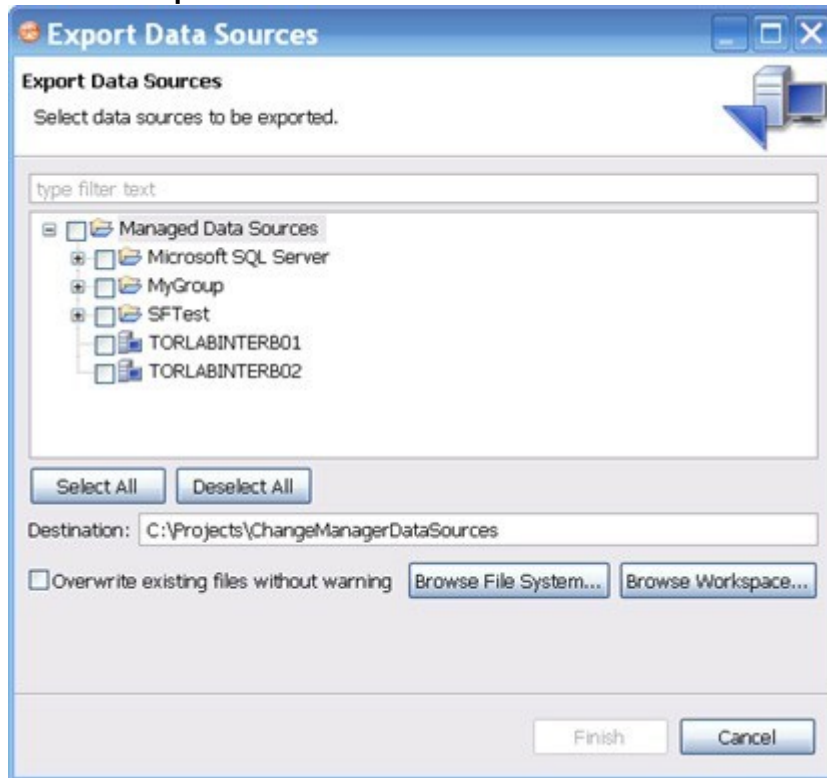
1. Select **File > Import > Data Sources**.



2. Choose a source from which you want to import the data sources, and then click **Next**.
3. Specify the details about the location of the import source.
4. Click **Finish**. Data Source Explorer is automatically populated with the new data sources.

To export data sources

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



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

Filtering data source objects

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

There are two types of data source filters:

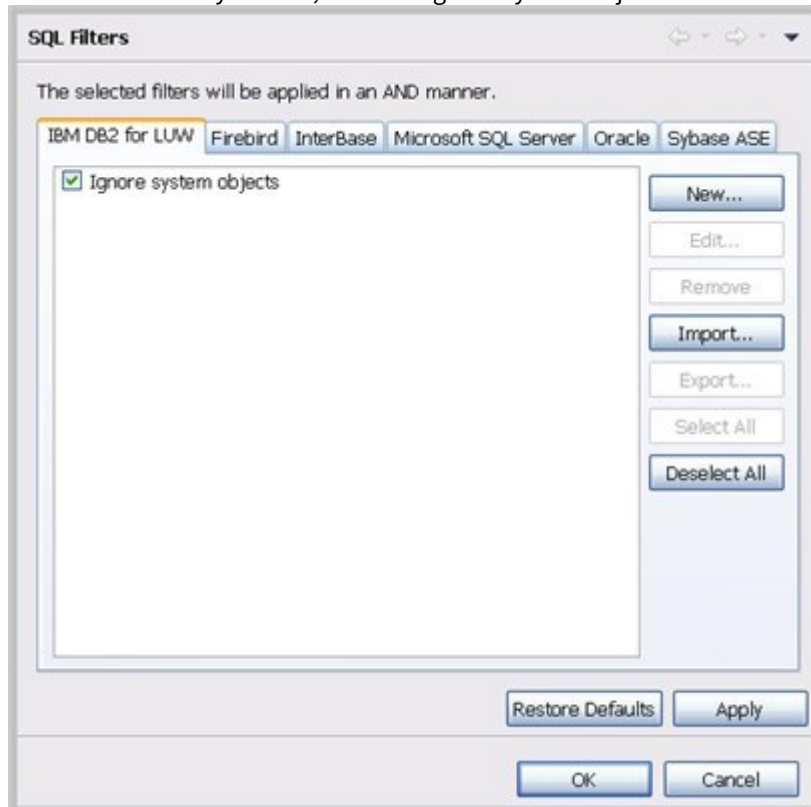
- Global filters affect all registered data sources.
- Data source specific filters affect only the specified data source.

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

To define a global filter

1. From the Main menu, select **Preferences > SQL Filters**.
The SQL Filters pane displays global filter templates in tabs organized by platform. You can create and apply a global filter to a database platform such as SQL Server, and all databases associated with that platform

will use the filter. By default, all filters ignore system objects.

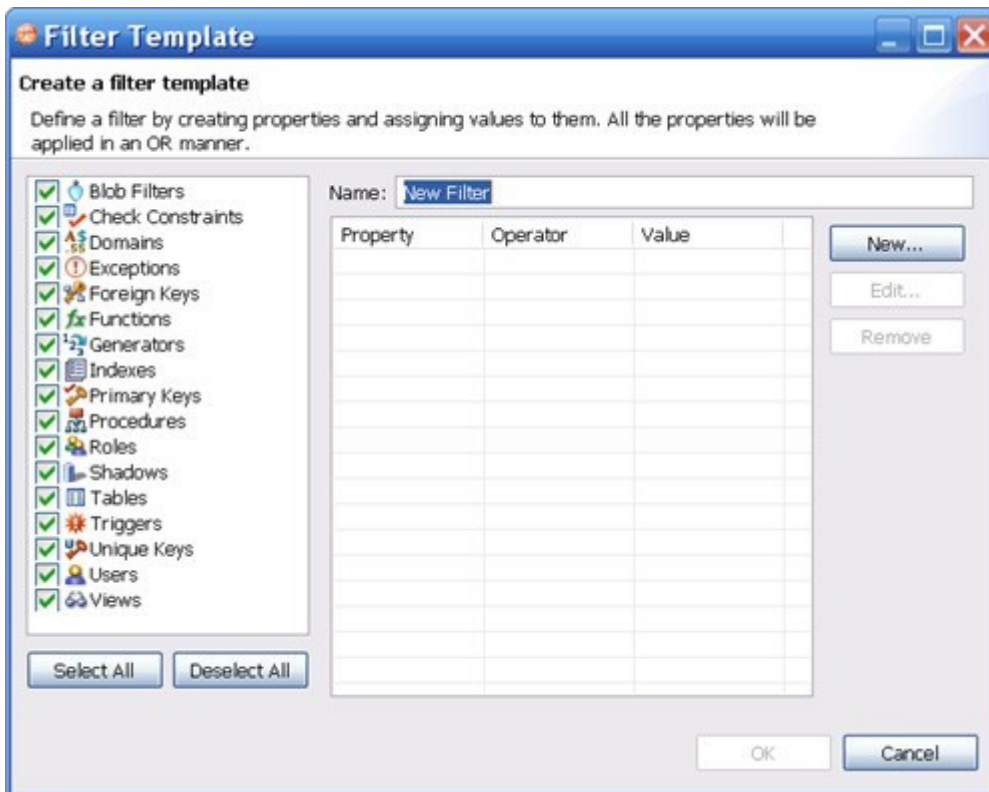


2. Click **New**. The Filter Template dialog appears.

To define a data source specific filter

1. In Data Source Explorer, right-click the data source, and then select **Properties**.
2. Select the **SQL Filter** node.
3. Select **Enable data source specific settings**. The other controls on the dialog become enabled.
4. Click **New**. The Filter Template dialog appears.

To create a filter template



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

Comparing data

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

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

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

- [Create a Data Comparison Job](#)
- [Describe and Save the Job](#)
- [Specify the Source and Target Data Sources](#)
- [Specify Execution, Mapping, Comparison, and Resolution Options](#)
- [Understand the Data Source Mapping Process](#)
- [Customize Database Mapping](#)
- [Inspect the Databases](#)
- [Customize Table Mapping](#)
- [Reconfigure the Data Source Mapping](#)
- [Set Job Notification Options](#)
- [Run the Data Comparison Job](#)
- [Review Comparison Results](#)

After comparing the data, you can synchronize it to match. For more information about this feature, 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](#).

To create a new data comparison job

Select **File > New > Data Comparison Job** and the Data Comparison Job editor opens.

To open an existing data comparison job

1. Select **File > Open Job**.
2. Select a data comparison job from the list.
3. Click **OK**.
The Data Comparison Job editor opens.

Describe and Save the Job

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

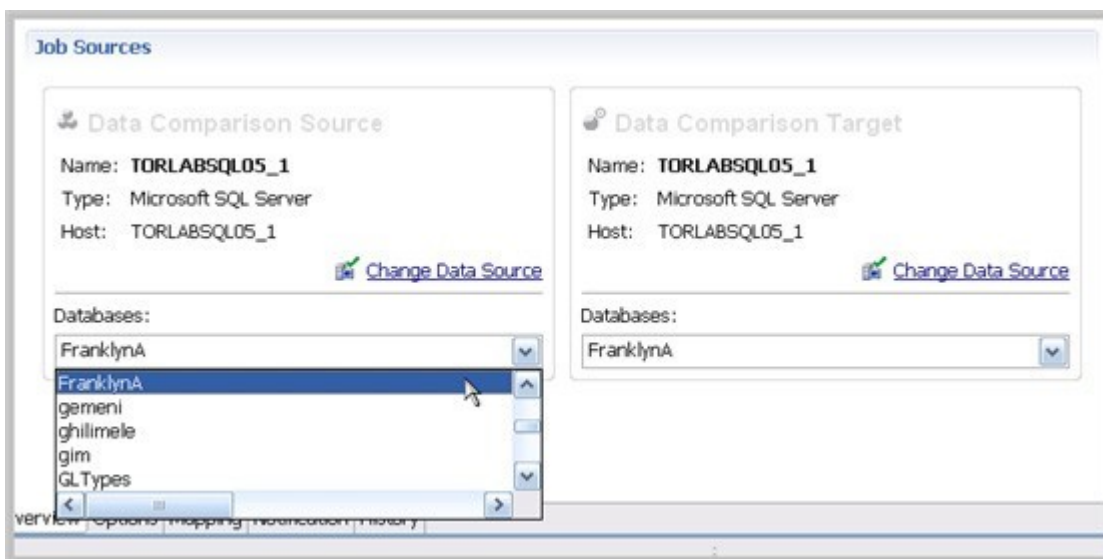
If the job will be used to monitor compliance against company standards or policies, you can add it to the [Compliance Explorer](#) list. This list lets you quickly see all the jobs that have recently passed or failed the compliance criteria you set.

To describe and save a job

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

Specify the Source and Target Data Sources

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



To add a data source to a job

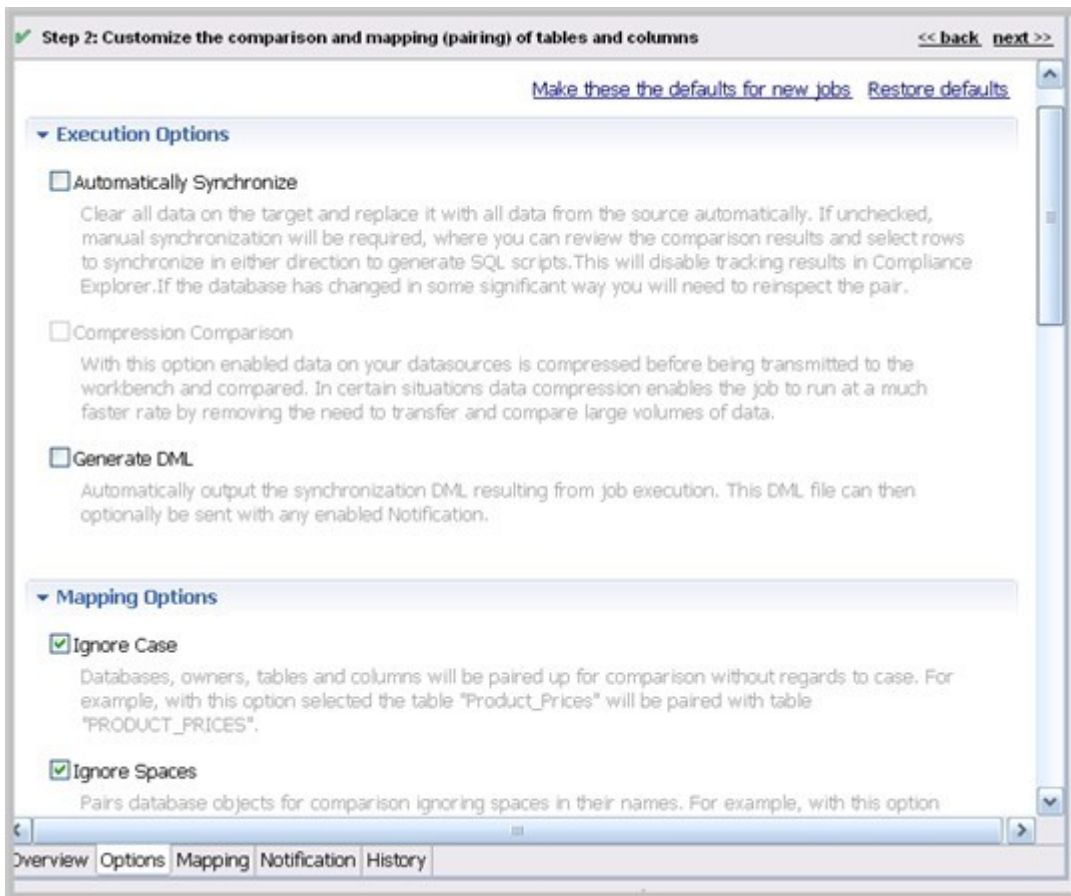
1. Click and drag a data source from Data Source Explorer to the **Data Comparison Source** or **Data Comparison Target** box, or click **Select Data Source** in the appropriate box and choose a data source from the dialog.

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

2. Select the databases, or tables and views (depending on the type of database), from the **Databases** pulldown menu.
3. After you name the job and specify the data sources, click **Options**.

Specify Execution, Mapping, Comparison, and Resolution Options

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



To set data comparison job options

1. Scroll down to see all the options and their descriptions, and select or deselect them as needed.
2. If an option is not available for the data source, the check box is greyed out. The options are described in the [DB Change Manager Preferences](#):
 - [Execution Options](#)
 - [Mapping Options](#)
 - [Comparison Options](#)
 - [Resolution Options](#)
3. When you are satisfied with the options, click **Mapping**.

To save options as defaults for future jobs

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

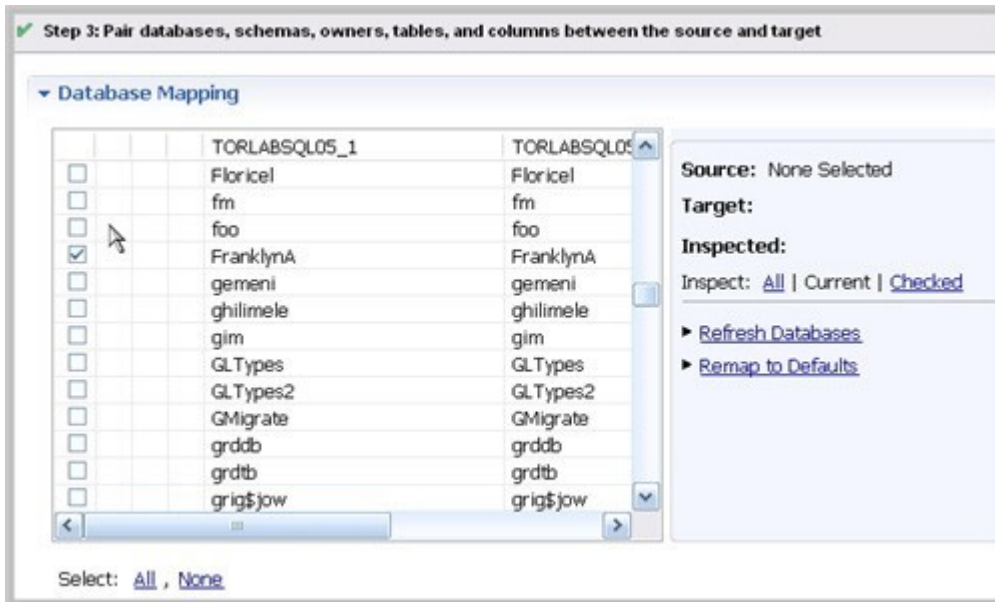
Understand the Data Source Mapping Process

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

Mapping details are shown on the Mapping tab of the editor.

Customize Database Mapping

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



To select databases to compare

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

To change the default target database

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


Inspect the Databases

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

To run the database comparison inspection

On the right side of the Database Mapping pane, click one of the following Inspect options:

- **All** inspects all pairs in the table. This is not recommended if you have a large number of databases.
- **Current** inspects the current object mappings.
- **Checked** inspects only those pairs selected in the table.

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

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

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

To restore the Database Mapping table to its original settings

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

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

Customize Table Mapping

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

The screenshot shows a 'Table Mapping' window with two columns of tables: 'TORLABSQL05_1.FranklynA' and 'TORLABSQL05_1.Fran...'. The first column lists tables like 'dbo.Address', 'dbo.Category', 'dbo.Customer', etc. The second column lists corresponding tables in the target database. The 'dbo.foo2' row is highlighted in red and has a red 'X' icon. A tooltip over this row reads: 'The following errors have been found: Table or View can not be compared if there is no column selected as comparison key'. On the right, a pane shows 'Source: dbo.foo2' and 'Target: dbo.foo2' with 'Columns: 2/2 to 2/2' and a 'View Column Mapping' link. At the bottom, there is a 'Filter:' field with an 'edit' link and a 'Select: All, None' option.

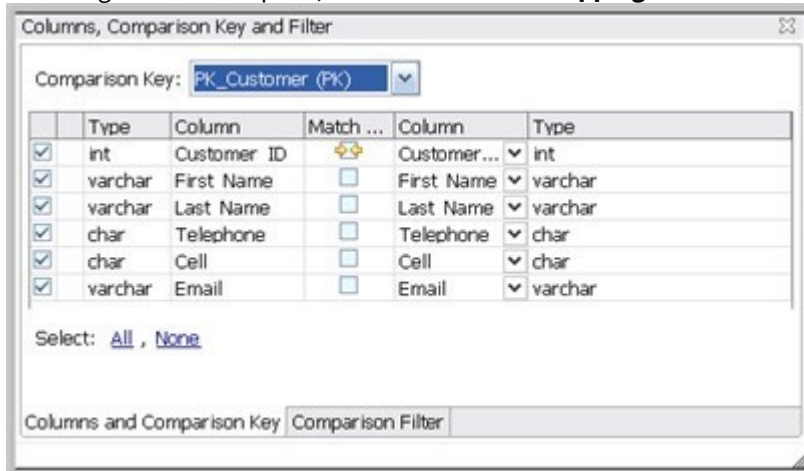
To change the mapping to a new target column

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

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

1. Select a row by highlighting it with the cursor. (The check box is irrelevant for this task.)

2. On the right side of the pane, click **View Column Mapping**.



3. In the Columns and Comparison Key pane, you can:
 - Select or deselect columns to use in the comparison job. Because this could affect referential integrity, review [Resolution Options](#).
 - Change or add a comparison key if multiple keys are available. In the Comparison Filter pane, you can:
 - Write a WHERE clause to use during the comparison.
 - Create a different WHERE clause per table in the pair. Select **Separate Where Clause** and an additional text box is added for a second WHERE clause.

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

Reconfigure the Data Source Mapping

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

To reconfigure the automatic mapping process

1. In the Options tab, select or deselect **Mapping Options**. The options are also described in [Mapping Options](#).

i If you change the mapping options in **Preferences**, they will become the defaults for all new jobs.
2. In the Mapping tab, click **Refresh Databases and Owners** to update the automatic comparison.

Set Job Notification Options

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

Run the Data Comparison Job

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

To run a job

1. Click **File > Save** to save the job before you run it.

- In the upper right of the Data Comparison Job editor, click the **Run** icon.



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

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

Review Comparison Results

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

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

The Comparison Results table lists:

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

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

Synchronizing data repositories

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

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

Select Data to Resolve

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

To view columns and rows to resolve

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

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



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

View Diff Report Export Diff Results...

The Results Overview - Activation pane includes:

- Source and target database columns
- Number of matched rows between the source and target.
- Rows that exist only in the source database.
- Rows that exist only in the target database.
- Number of differences between the two databases.

To view data in the columns

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

To specify how the data synchronization is resolved

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

To generate an SQL script

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

Run the SQL Synchronization Script

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

To run the SQL synchronization script

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

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

Comparing database configuration settings

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

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

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

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

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

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

- [Create or Open a Configuration Comparison Job](#)
- [Describe and Save the Job](#)
- [Specify Configuration Comparison Source and Targets](#)
- [Refine property comparisons](#)
- [Specify Configuration Comparison Job Options](#)
- [Set Configuration Comparison Job Notifications](#)
- [Run the Configuration Comparison Job](#)
- [View the Results of the Configuration Comparison](#)
- [Create a Job Report](#)
- [Synchronize Configurations](#)
- [Execute an SQL Synchronization Script](#)

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

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:

1. Select **File > Open Job**.
2. Select a configuration comparison job from the list.
3. 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 you want the job to monitor compliance against company standards or policies, you can add it to the [Compliance Explorer](#) list. This list lets you quickly see all the jobs that have recently passed or failed the compliance criteria you set.

To describe and save a job

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

Specify Configuration Comparison Source and Targets

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

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

You can also compare a single source against multiple targets.

To add a data source to a job

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

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

To add a configuration standard to the job

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

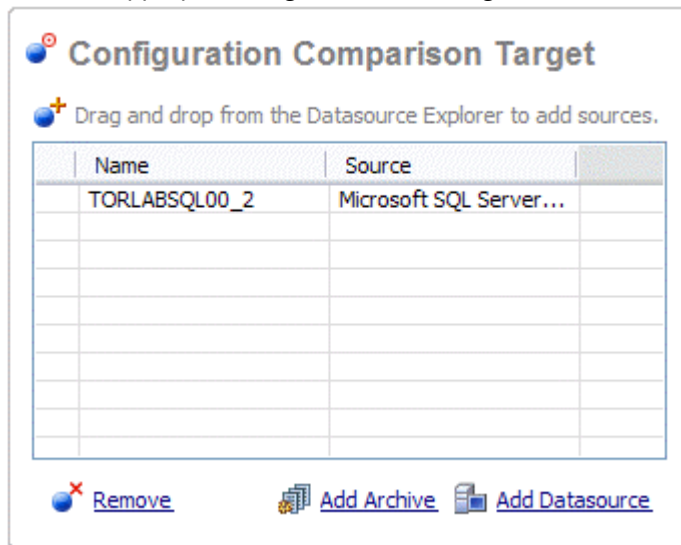
To add a configuration archive to the job

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

To add multiple targets to the comparison job

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

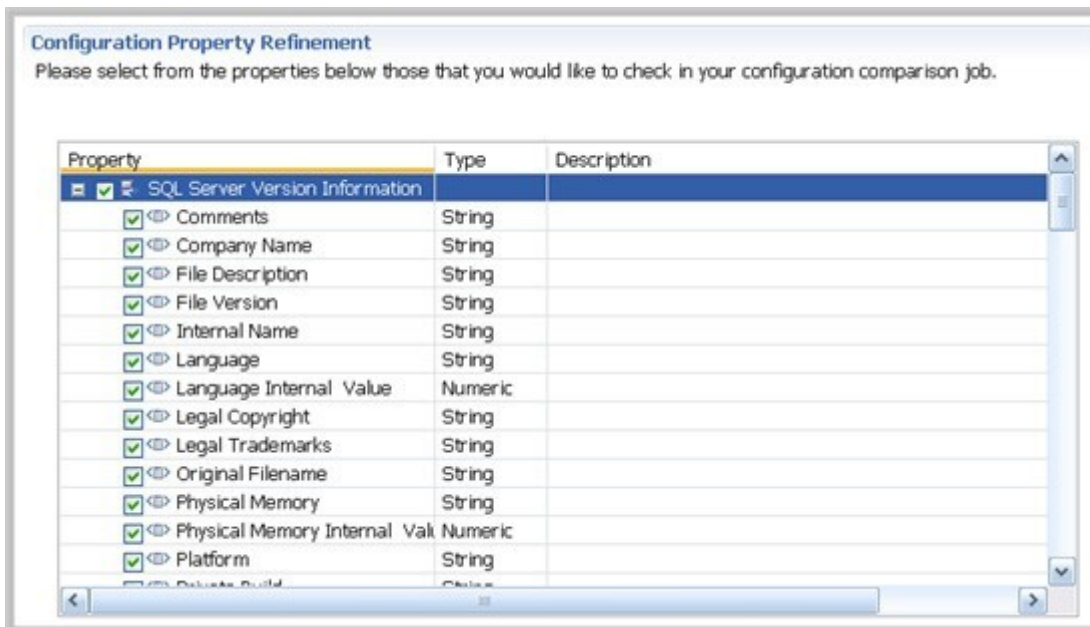
select the appropriate target from the dialog.



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

Refine property comparisons

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



To specify the properties used in the comparison job

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

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

Specify Configuration Comparison Job Options

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

Set Configuration Comparison Job Notifications

When a job completes, DB Change Manager provides various ways of notifying you. For more information, see [Setting job notifications](#).

Run the Configuration Comparison Job

The configuration comparison job runs when you click the **Run Job** icon in the [Configuration Comparison Job Editor](#).



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

View the Results of the Configuration Comparison

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

The screenshot displays the 'Comparison Results' window. At the top right is a 'Close results' link. Below it are two tabs: 'Comparison Summary' and 'Resolution'. Under 'Comparison Summary', the source is 'TORLABSQL00_1' and the target is 'TORLABSQL05_1'. A 'Generate a SQL Script' link is also present. The summary shows: 48 Properties Matched (53.3%), 17 Properties Did Not Match (18.9%), and 25 Properties Not Found (27.8%). Below this are 'View Report' and 'Export Results...' links. The 'Comparison Results' section contains a table with the following data:

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

The Comparison Summary shows:

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

The Comparison Results table lists the details further down in the window.

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

Comparison Results includes:

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

Create a Job Report

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

Synchronize Configurations

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

To create a SQL synchronization script

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

Execute an SQL Synchronization Script

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

To execute the SQL synchronization script

In the SQL Editor, click the **Execute SQL** icon. DB Change Manager executes the code on the target data source and displays the results.

Creating a standard for configuration settings

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

A configuration standard has a list of properties to which you assign values. Properties are derived from data sources, archives, or other standards. You assign each property value a range or set an operator to be used in comparisons. For more information, see [Comparing database configuration settings](#).

The following tasks step you through creating a configuration standard:

- [Create or open a configuration standard](#)
- [Describe and Save the Standard](#)
- [Define Configuration Standard Properties](#)
- [Use a Configuration Standard in Configuration Comparison Jobs](#)

Create or open a configuration standard

You create or modify a Configuration Standard in the [Configuration Standard Editor](#).

To create a new configuration standard

Select **File > New > Configuration Standard**. The Configuration Standard editor opens.

To open an existing configuration standard

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

Describe and Save the Standard

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

To describe and save a configuration standard

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

Add Standard Sources

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

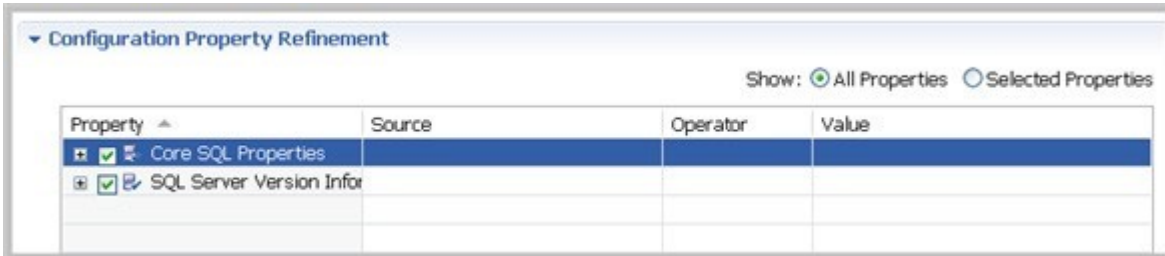
To add properties to a standard

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

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

Define Configuration Standard Properties

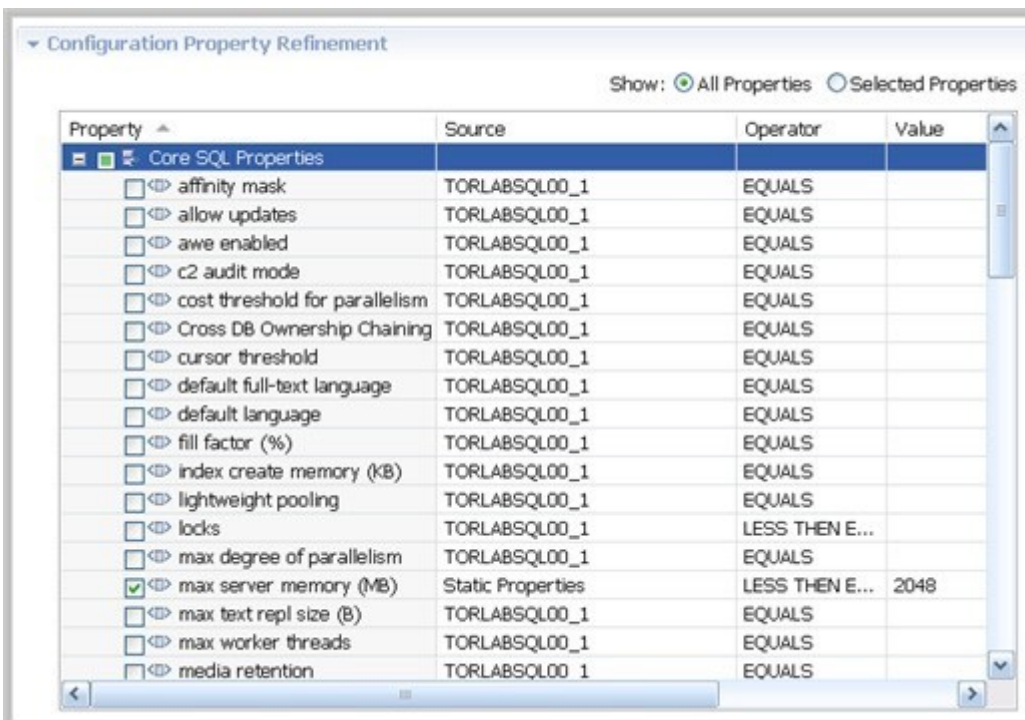
The Configuration Property Refinement tab displays properties and corresponding sources applied to the standard.



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

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

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



To define the properties of a configuration standard

1. Deselect the properties you do not want to consider using the check boxes.
2. Click **Show: Selected Properties** to narrow down the list in the window.
3. For a property you want to use in the standard, click its appropriate row value in the **Operator** column, and then 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 then press *Ctrl+S*.

Use a Configuration Standard in Configuration Comparison Jobs

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

To add a standard to a configuration comparison job

1. Drag and drop the standard from the **Project Explorer** to the comparison source box in the [Configuration Comparison Job Editor](#).
2. Continue to define and run the job. For more information, see [Comparing database configuration settings](#).

Creating a configuration archive

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

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

Archives are listed in Data Source Explorer under their corresponding data source. The following tasks step you through creating a configuration archive:

- [Create or Open a Configuration Archive Job](#)
- [Describe the Configuration Archive Job](#)
- [Specify a Configuration Archive Source](#)
- [Refine Configuration Archive Properties](#)
- [Set Configuration Archive Job Notifications](#)
- [Specify Configuration Archive Job Options](#)
- [Create a Version and Save the Archive Job](#)
- [Create an Archive Job Report](#)
- [Use an Archive in a configuration Comparison Job](#)

Create or Open a Configuration Archive Job

You create or modify a configuration archive in the Configuration Archive editor. For more details about the editor, see Configuration Archive Job Editor.

To create a new configuration archive job

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

To open an existing configuration archive job

1. Select **File > Open Job**.
2. Select a configuration job from the list.
3. Click **OK**.
The Configuration Archive editor opens.

Describe the Configuration Archive Job

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

To describe a configuration archive job

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

Specify a Configuration Archive Source

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

To specify and save a configuration archive source

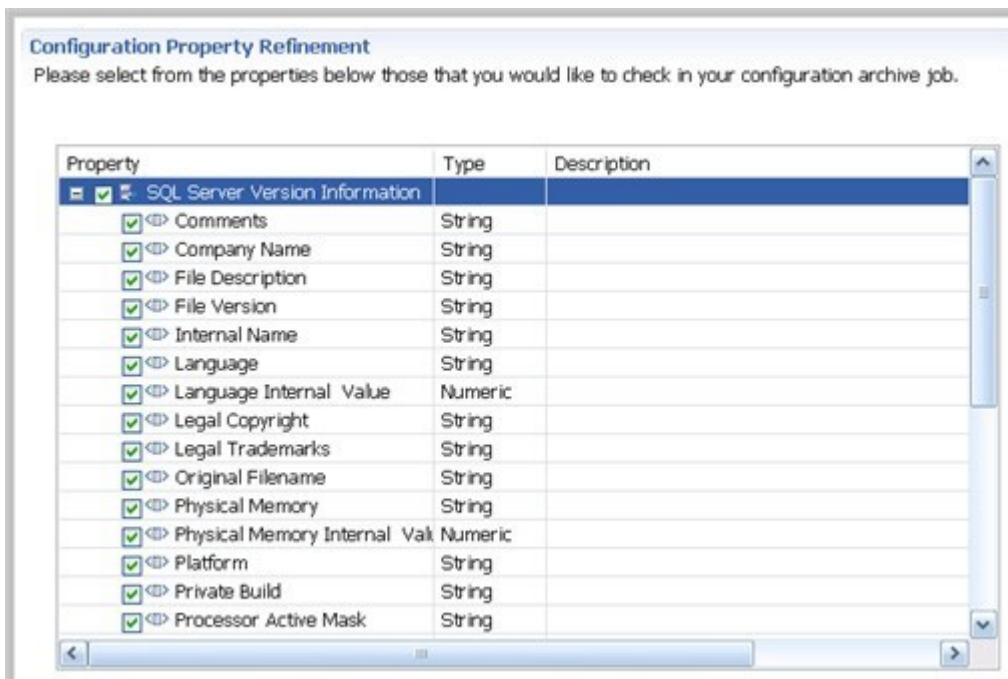
1. Drag and drop the data source to be archived from **Data Source Explorer** to the **Configuration Archive Source** box. The data source is listed by name, database type, and host.
2. Click **File > Save** or **File > Save All**, or highlight the editor window and press **Ctrl+S**.

Refine Configuration Archive Properties

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

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

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



To specify configuration archive properties

- Use the check boxes to select or deselect properties.

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

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

Set Configuration Archive Job Notifications

When an archive job completes, DB Change Manager provides various ways of notifying you. For more information, see [Setting job notifications](#).

Specify Configuration Archive Job Options

The Options tab in the Configuration Archive Job editor contains options for handling the DDL and purging older archives.

If you select the Extract DDL option, the synchronization code produced by the job comparison will appear in a separate editor window when the comparison is done.

The Automatic Purge Options delete archive files after a number of days, number of versions, and/or when the size of the files reaches a certain limit. The purge conditions are checked and enforced when an archive job is saved or when a new version is created.

To specify the conditions for automatically deleting archives

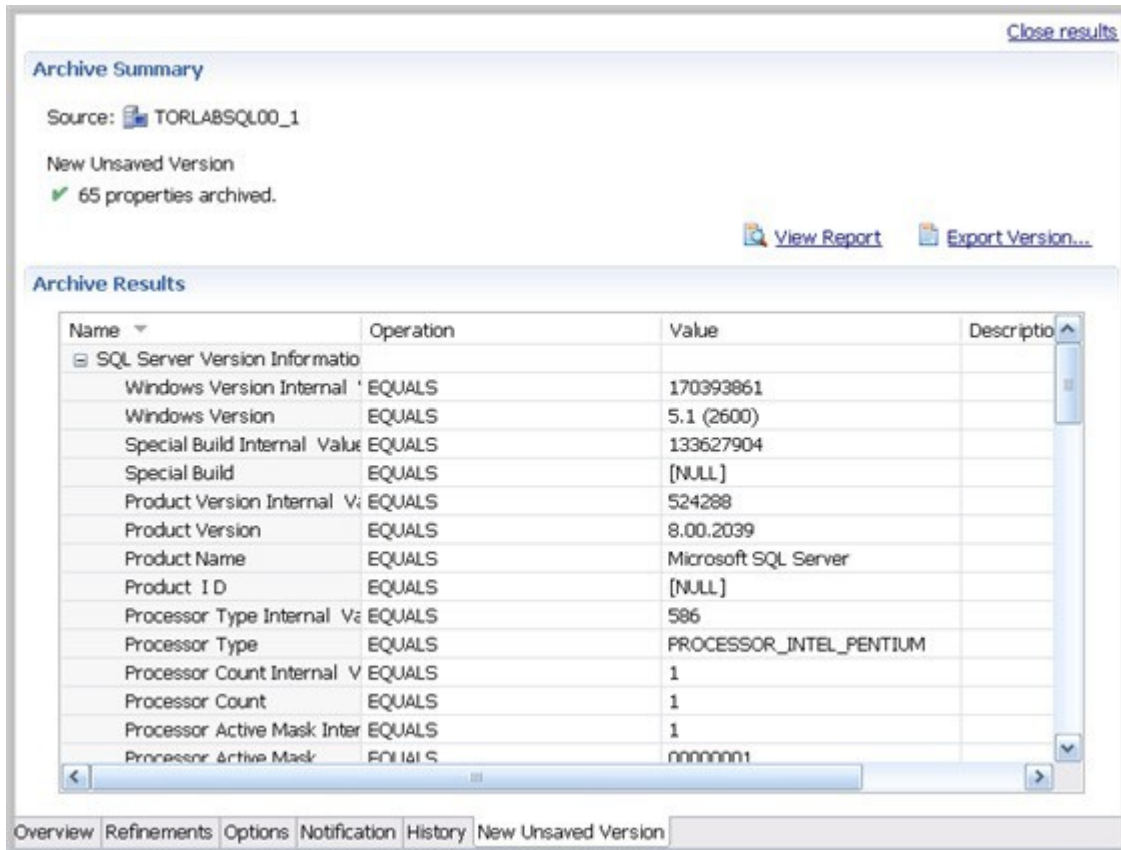
- In the Options tab, fill in one, two, or all of the options:
 - The number of Days after which archives will be deleted.
 - The number of Versions stored, after which the oldest version will be deleted.
 - The number of MB total size of the version files, after which the oldest version will be deleted.

Create a Version and Save the Archive Job

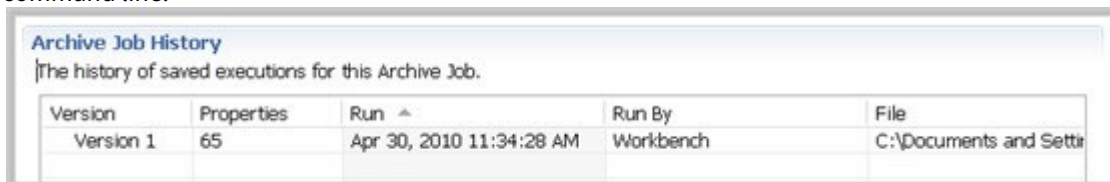
When you first run an archive job, DB Change Manager creates the first version of the archive. A version stores snapshots of the same archive job for backup or tracking comparison purposes. Versions are listed in the Archive Job History table on the History tab and in the Data Source Explorer.

To create a version and save a configuration archive job

1. In the Configuration Archive Job editor, click **Run Job**. The New Unsavd Version tab lists the properties of the archive.



2. Highlight the tab, and then select **File > Save** or **File > Save All**. The tab shows that the version is saved.
3. Click the History tab. The **Archive Job History** table lists the archive version, the number of properties retained by the version, a time stamp, and whether the archive job was run from the workbench or the command line.



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

To open an archive version

- In the History tab or in the Job Explorer, double-click the version. The selected version opens in a new tab of the Archive Job editor.

Create an Archive Job Report

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

Use an Archive in a configuration Comparison Job

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

To add a configuration archive to a configuration comparison job

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

For more information, see [Comparing database configuration settings](#).

Masking data

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

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

The following tasks describe how to create masked data:

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

Create a New Masking Job

Use the [Data Comparison Job Editor](#) to define a masking job.

To start creating a new Data Masking Job:

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

Specify a Masking Job Name

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

To name a masked data job

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

Job Name and Description

Name:

Notes:

Project: **Change Management**

Track results in Compliance Explorer

Specify Source and Target Data Sources

The **Job Sources** section of the Data Comparison Job Editor contains the **Data Comparison Source** box, which identifies the original data source, and **Target Data Source** box, which identifies the data source where the data is 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 is 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:

Data Comparison Target

Name:

Type:

Host:

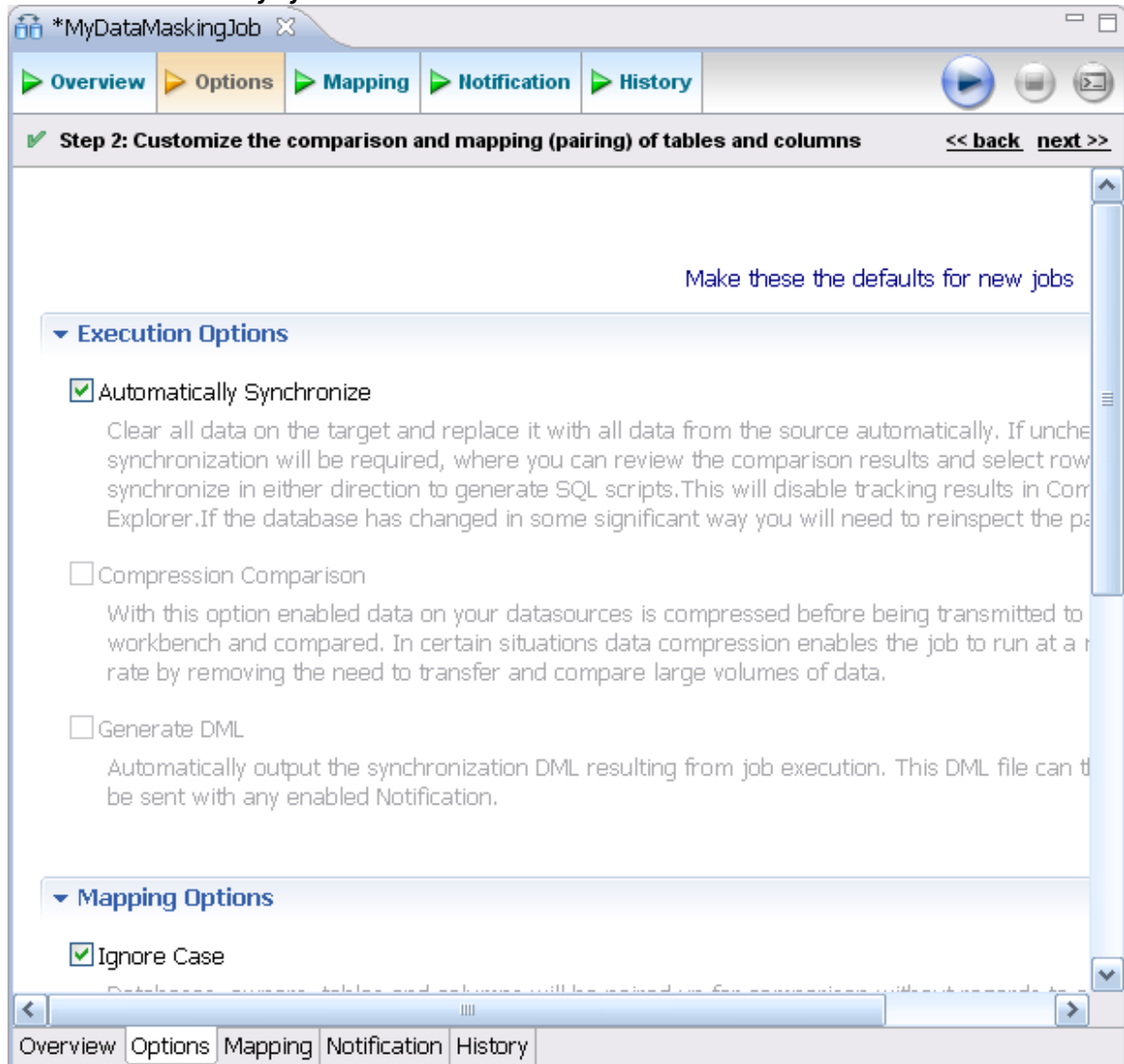
Set to Automatically Synchronize

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 is 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

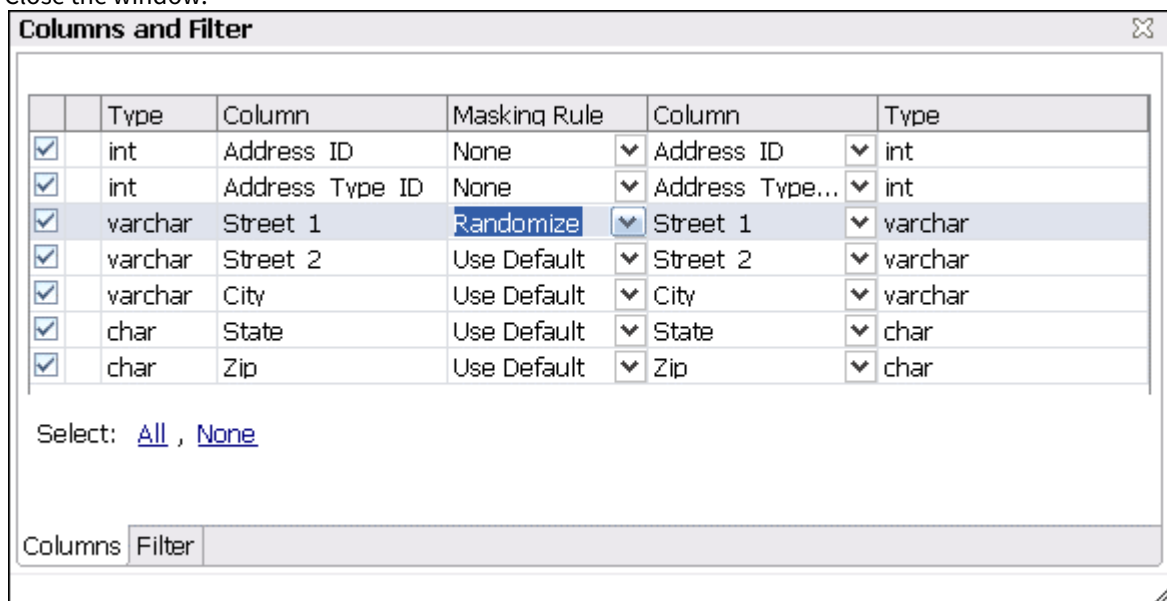
1. Click on the Mappings tab.
2. Select check box of the database you want to mask.
3. Click on the **Yes, retrieve the objects in the pair** button to inspect the tables and columns. The Table Mapping section opens.
4. Select a **Default Masking Rule** for the in the **Database Mapping** box.
5. Select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

Specify Masking Rules for Individual Columns

You can specify different masking rules for different columns.

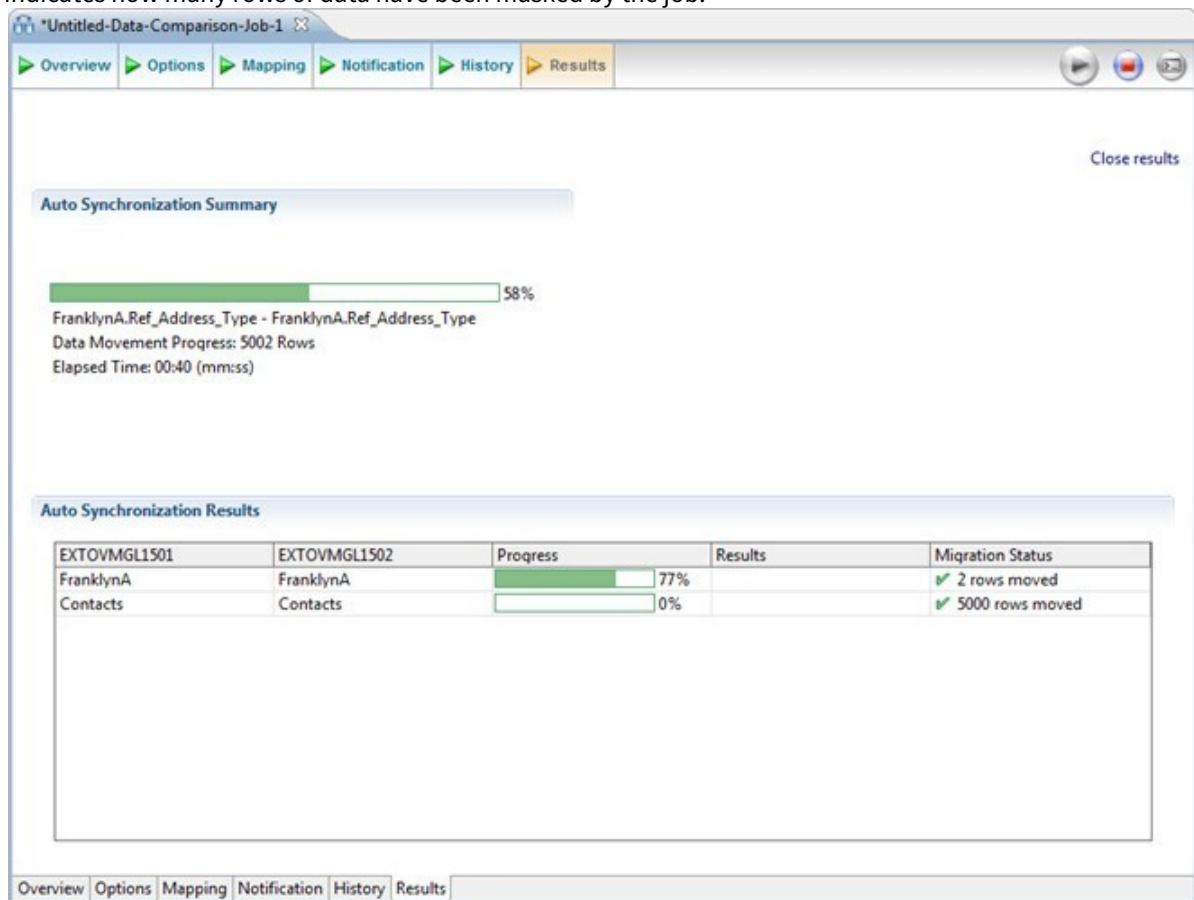
To specify a masking rule for a column

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 is 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. For more information, see [Comparing data](#).

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 displays the 'Results' tab of the IDERA DB Change Manager. It shows a comparison between two data sources for the 'FranklynA' schema. The 'Results Overview' section provides a summary of the comparison for 'dbo.Category' and 'dbo.Customer' tables. Below this, the 'Selected Table' section shows a side-by-side comparison of the 'Customer' table from two different instances, highlighting differences in the 'First Name' and 'Last Name' columns.

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

TORLABSQL05_1.FranklynA.dbo.Customer				TORLABSQL05_2.FranklynA.dbo.Customer			
...	Customer ID	First Name	Last Name	...	Customer ID	First Name	Last Name
	734	James	Battiste		734	Gxibk	Yxssfmsl
	735	Samual	Philpott		735	Jqciqb	Fvwzfiac
	736	Noel	Arra		736	Nqel	Eaee
	737	Gaston	Amesqui		737	Okeybx	Oaseafw

Comparing and synchronizing schemas

- [Comparing schemas](#)
- [Creating a schema archive](#)
- [Synchronizing schemas](#)

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. For more information, see [Creating a schema archive](#).

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

- [Create or Open a Schema Comparison Job](#)
- [Describe and Save the Schema Comparison Job](#)
- [Specify a Schema Comparison Source and Targets](#)
- [Refine Schema Comparisons](#)
- [Customize Schema Mapping](#)
- [Specify Schema Comparison Job Options](#)
- [Set Schema Comparison Job Notifications](#)
- [Run the Schema Comparison Job](#)
- [Create a Schema Comparison Job Report](#)
- [Cloning a schema comparison job](#)

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](#).

To create a new schema comparison job

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

To open an existing schema comparison job

- Select **File > Open Job**. Select a job from the list, and then click **OK**. The Schema Comparison Job editor opens.

Describe and Save the Schema Comparison Job

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

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

To describe and save a job

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

2. Add relevant information in the **Notes** field.
3. Click **Track Results** in Compliance Explorer if you want the job to be listed in the [Compliance Explorer](#).
4. Click **File > Save** or **File > Save All**, or highlight the editor window, and then press *Ctrl+S*.

Specify a Schema Comparison Source and Targets

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

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

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

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

To add a data source to a schema comparison job

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

Cross-Platform Schema Comparisons Supported by DB Change Manager

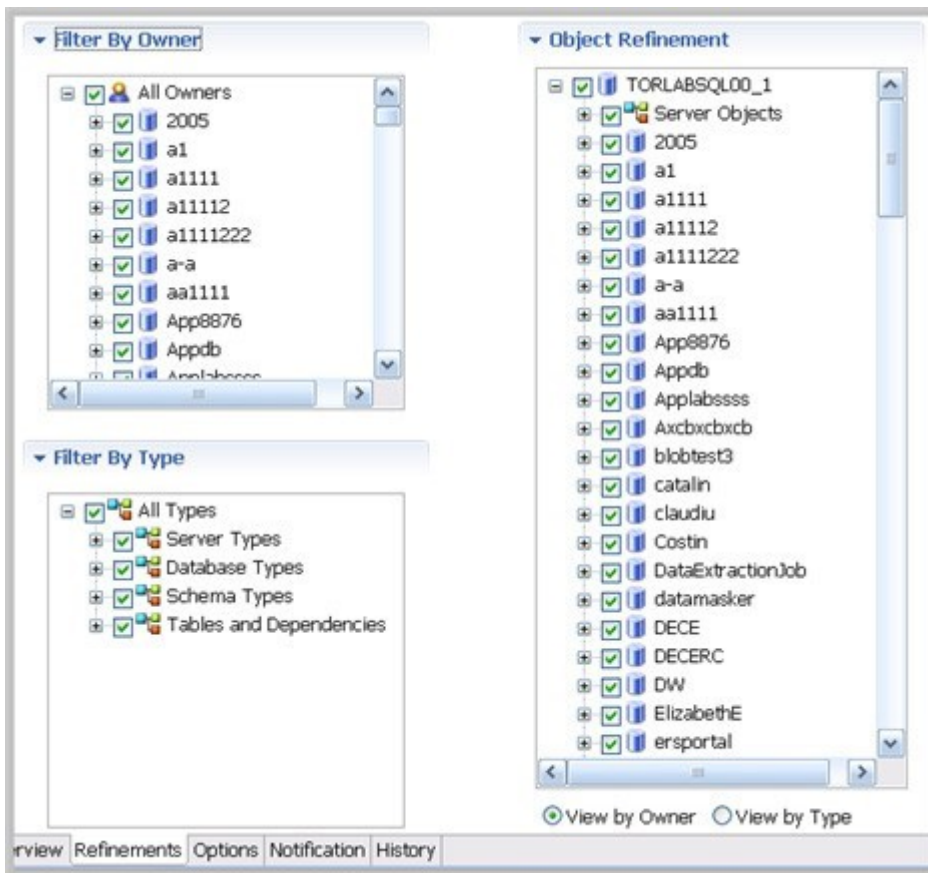
DB Change Manager supports cross-platform schema comparisons. However, such jobs are limited to the following DBMS combinations and data source objects:

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

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

Add Multiple Schema Comparison Job Targets

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



To specify the objects used in a schema comparison job

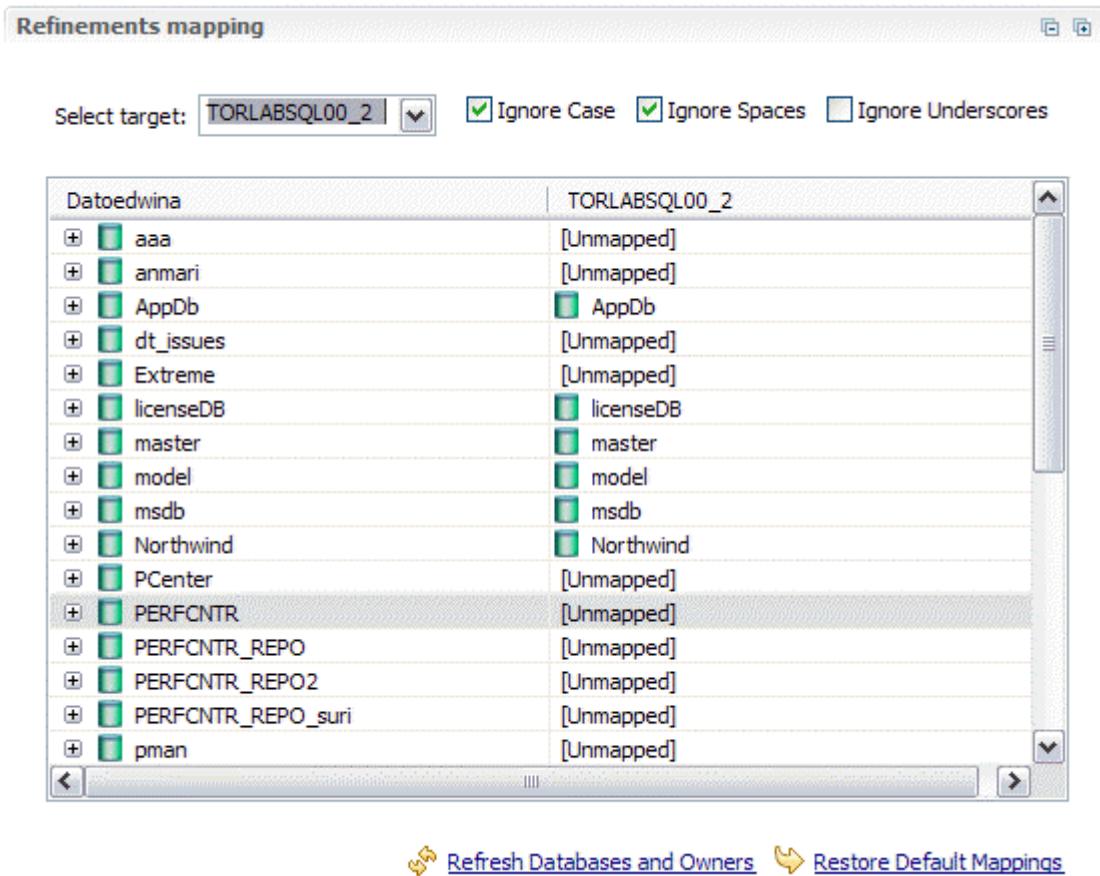
1. In the Refinements tab, select or deselect objects in the **Filter by Owner** and/or **Filter by Type** panes to add or remove objects from the **Object Refinement** pane.
2. Select or deselect the objects in the **Object Refinement** pane.

Customize Schema Mapping

The mapping process pairs objects between the job source and target so that the objects can be compared. You can see and modify pairs using the Mapping tab of the [Schema Comparison Job editor](#).

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

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



By default, new objects are included in this list. For more information, see [CM/Schema > Refinements Options](#).

When comparing master catalogs or user mappings, you can select multiple targets against a single source.

To redefine schema object mappings

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

To restore original schema object mappings

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

Specify Schema Comparison Job Options

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

The list of options differs depending on the DMBS of the source and target data sources. Options differ according to the objects included in the job. For example, Ignore Constraint Name Differences do not appear if you exclude constants from the job.

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

Schema comparison job options include:

- **Synchronization Options.** These options specify how you want DB Change Manager to handle the synchronization script after it runs the comparison job, either:
 - **Manual** causes DB Change Manager to display the results of the comparison, but it does not automatically attempt to repair any synchronization issues. If you choose this option, you will still be able to manually generate and run a synchronization script after the job runs.
 - **Automatic** causes DB Change Manager to automatically generate and run a synchronization script immediately after the comparison job runs.
- **Include Object Options.** These settings determine what elements of the job source and target will be omitted when running a schema comparison job and synchronization. For more information, see [CM/Schema > Comparison Options > Include Objects Options](#).
- **Compare Options.** These settings affect how objects are treated in the comparison process. You can include or exclude specific object types in the job, and specify the behavior of the comparison job and synchronization scripts. For more information, see [CM/Schema > Comparison Options > Compare Options](#).
- **Decorator Options.** These settings affect particular database objects. For more information, see [CM/Schema > Comparison Options > Decorator Options](#).
- **Identifier Options.** These settings determine what elements of the source and target data source will be ignored during the schema comparison and synchronization process. For more information, see [CM/Schema > Comparison Options > Identifier Options](#).
- **DDL Extract Option.** This setting causes the DDL code to appear in a separate editor window after the comparison job runs. For more information, see [CM/Schema > Comparison Options > DDL Extract Options](#).
- **Custom Options.** This section lists options that you defined in the Create New Schema Custom Compare Options dialog. For more information, see [Select Schema Properties to Ignore During a Comparison](#).

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

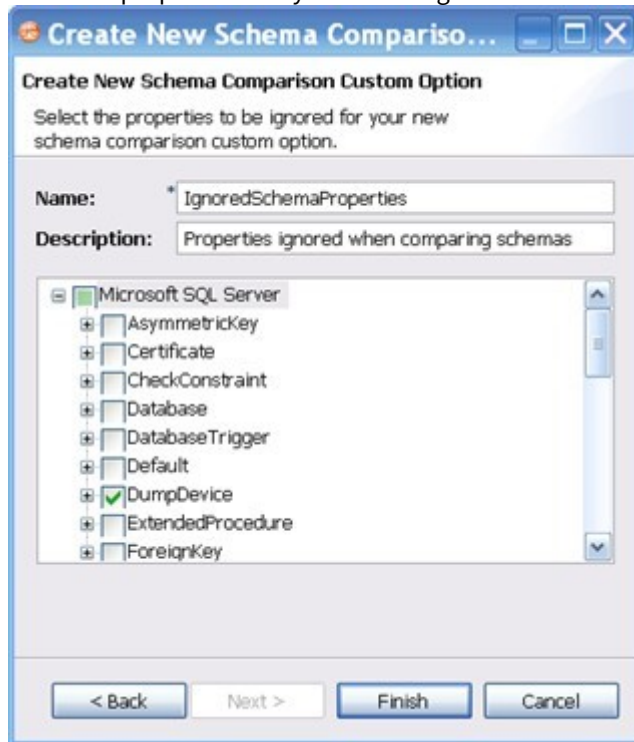
Select Schema Properties to Ignore During a Comparison

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

To define schema properties to ignore

1. Click **File > New > Custom Schema Comparison Option**.
2. Name the custom properties and add a description.

3. Select the properties that you want to ignore.



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

Set Schema Comparison Job Notifications

When a job completes, DB Change Manager provides various ways of notifying you. For more information, see [Setting job notifications](#).

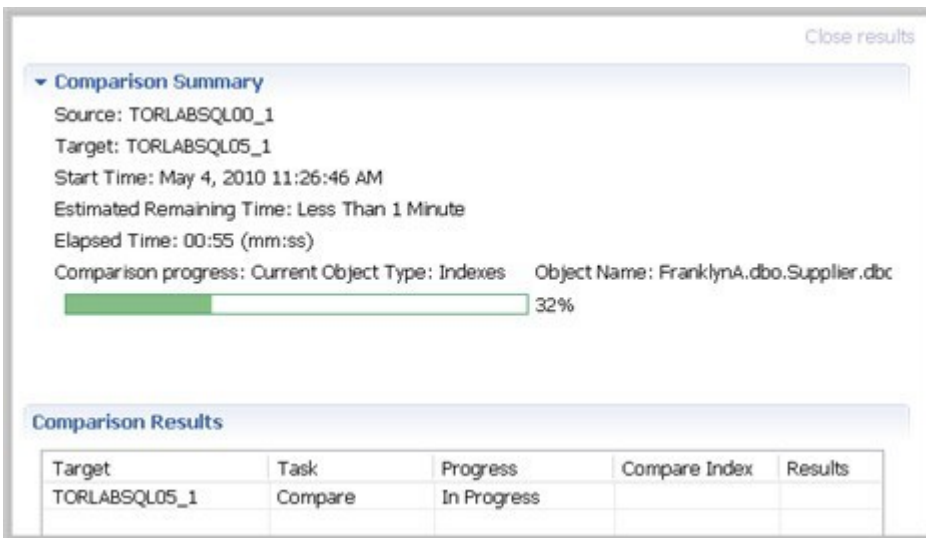
Run the Schema Comparison Job

To run the schema comparison job:

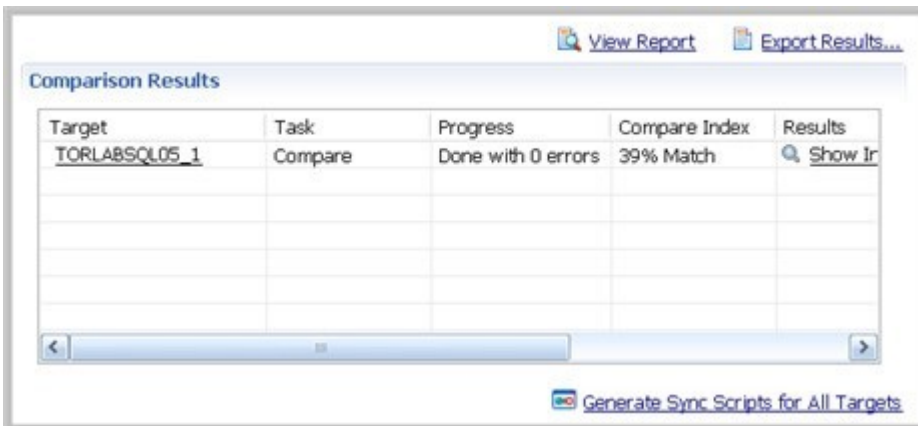
- Click the **Run** icon located in the upper right-hand corner of the Schema Comparison Job editor.



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



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



The Comparison Results pane includes these columns:

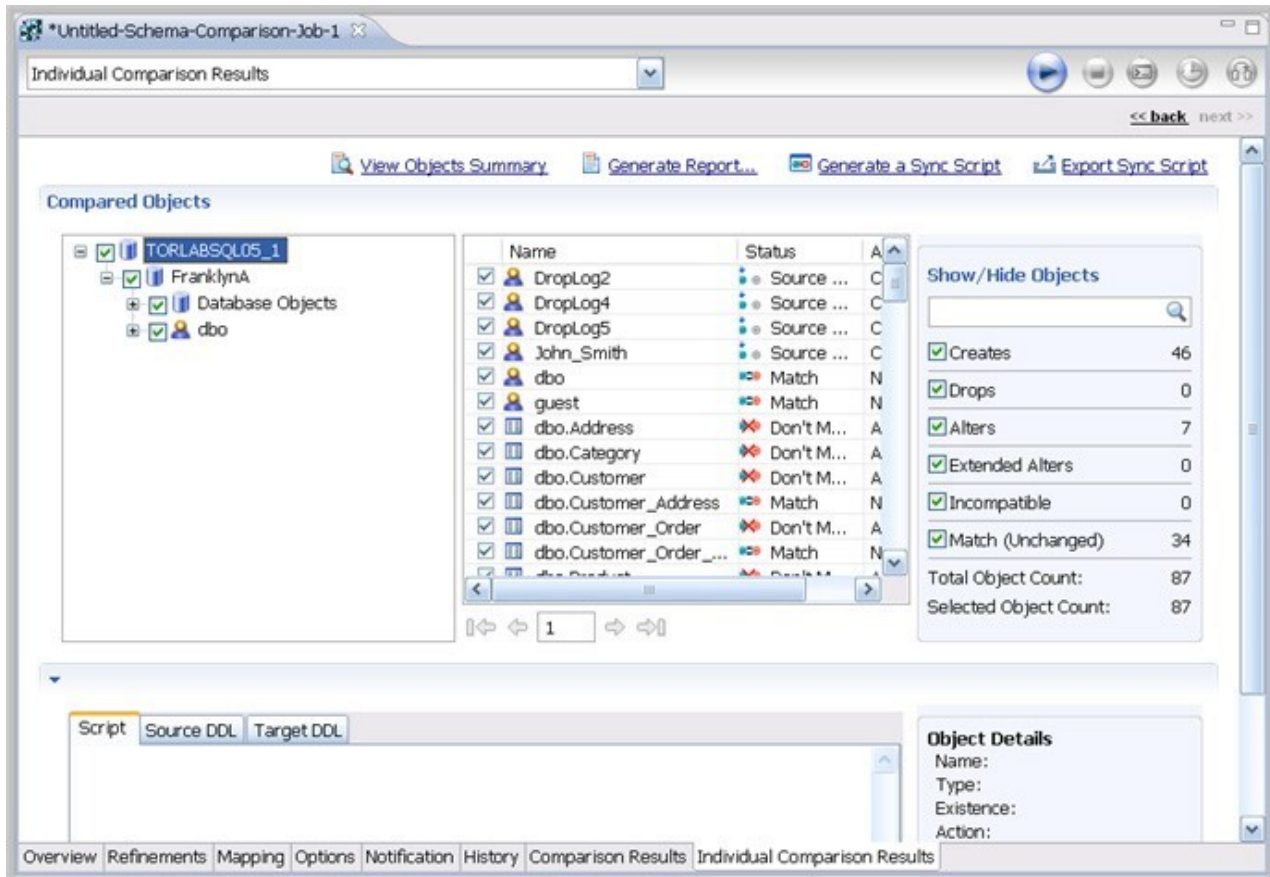
- The **Target** column shows the name of the target, and provides a link to open the Individual Comparison Results tab, which provides more details. For more information on viewing individual results, see [View Individual Schema Comparison Results](#).
- The **Task** column shows what kind of job was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

View Individual Schema Comparison Results

To see specific schema comparison results

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

The Individual Comparison Results window appears.

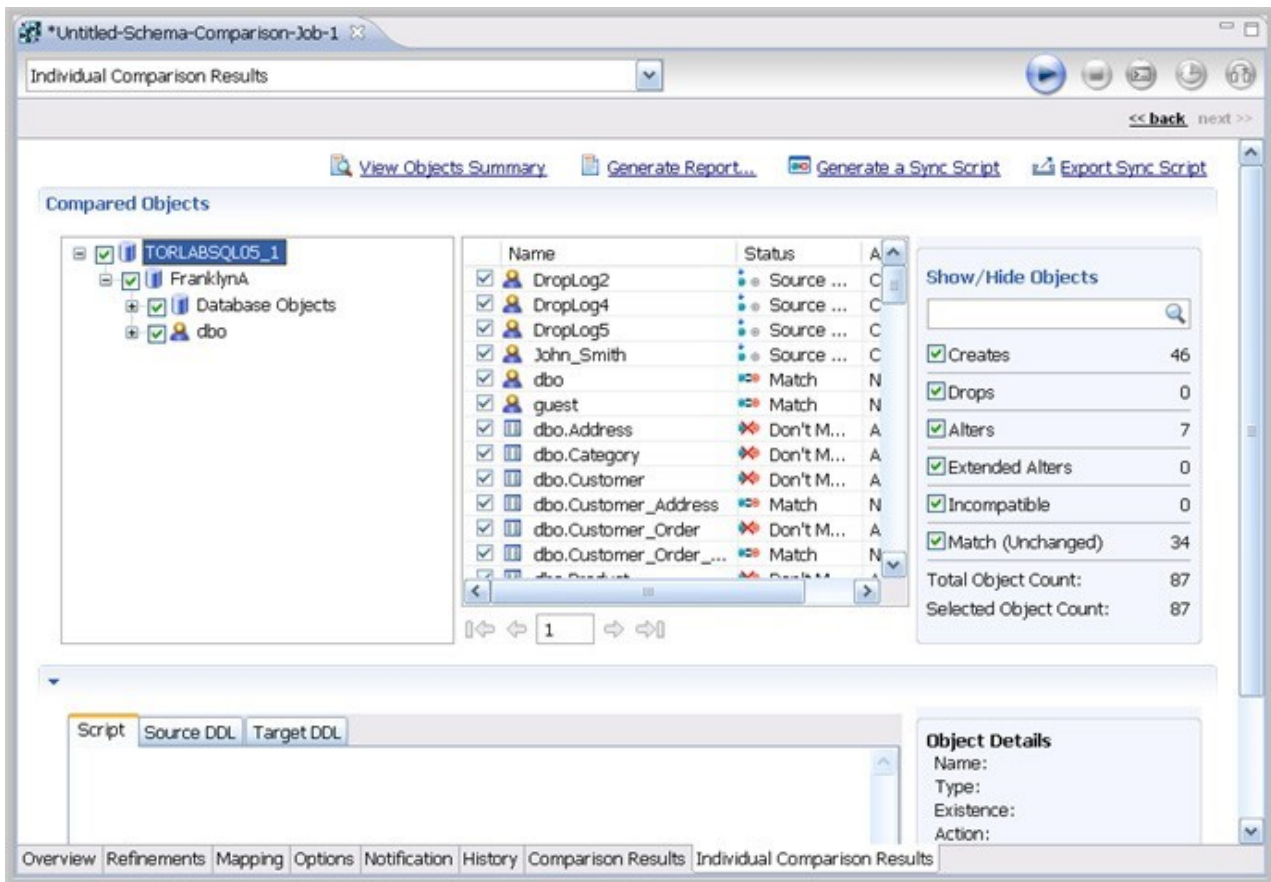


To browse the objects

- Select or deselect the check boxes in the left-panel list.
- Right-click and use the menu to select or deselect objects at the global or page levels.
- Click the right and left arrows underneath the middle panel.
- Enter a page number in the box underneath the middle panel.

To filter the objects

Type a search string into the **Show/Hide Objects** search box.



To see a summary report

- Click **Generate Report**.



To view the DDL between two schema comparison results

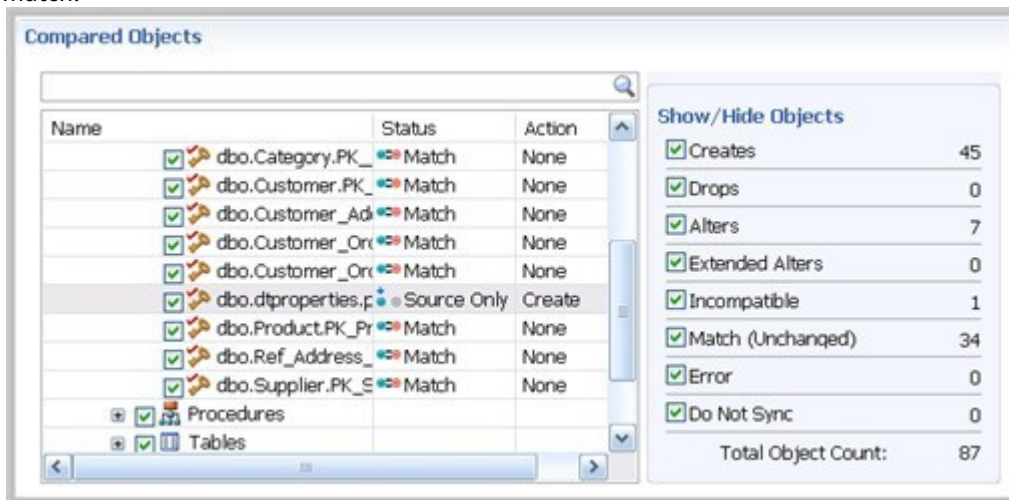
- Click **View Details**.

To save individual schema comparison results data to a text file

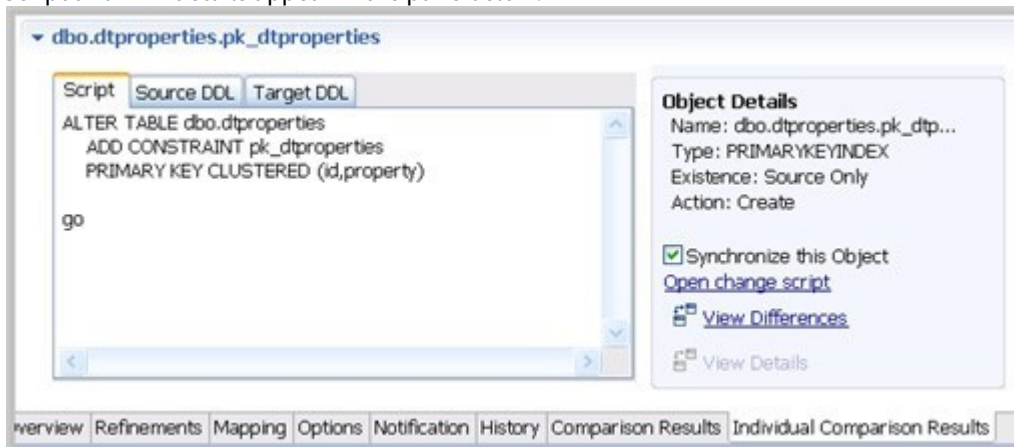
- Click **Export to File**.

To view the synchronization script and DDL between two individual objects

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



Script and DDL details appear in the pane below.



2. In the Script tab, view the SQL that synchronizes the objects.

Create a Schema Comparison Job Report

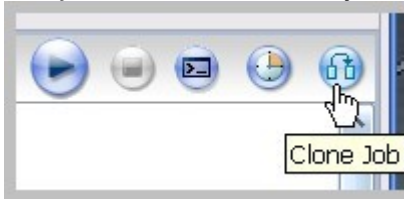
After running a comparison job, you can create a job report using the **View Report** button in the Results tab. For more information, see [Creating a job report](#).

Cloning a schema comparison job

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

To clone a schema comparison job

1. After you have saved and run a job, the **Clone Job** button is enabled in the Schema Comparison Job Editor.

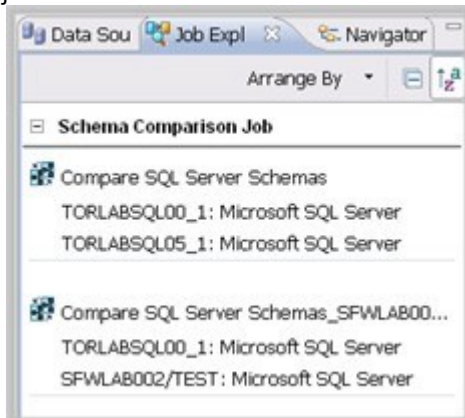


2. Click **Clone Job**.



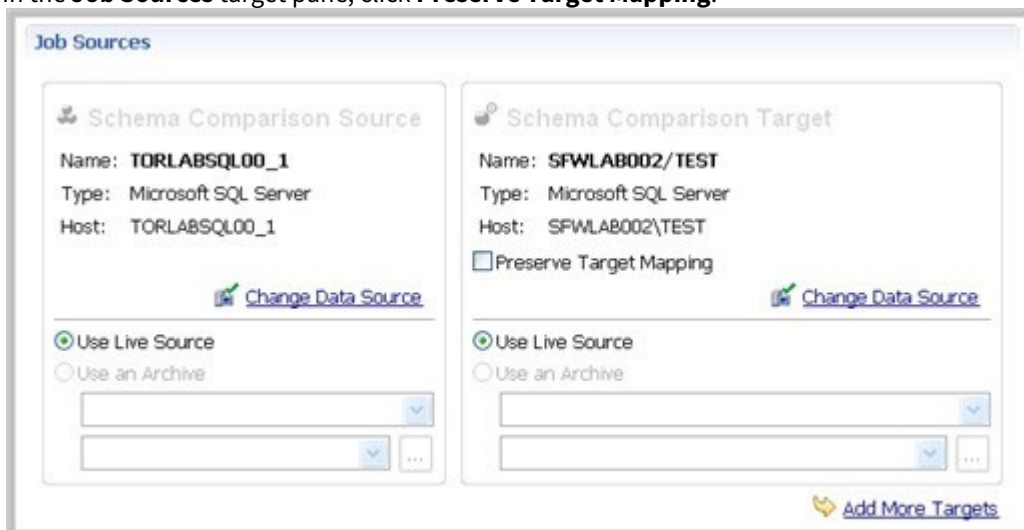
3. In the **Data Source Selection** list, select the new target for the cloned job. If you need to add a new data source to the list, click **New Data Source**.
4. Click **OK**.
DB Change Manager lists the new, cloned job in the Job Explorer. You can modify and run the cloned job as just as any other schema comparison job. In addition, you can preserve the target mapping from the original

job.



To preserve target mapping in a cloned schema comparison job

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



Creating a schema archive

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

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

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

The following tasks step you through creating a schema archive:

- [Create or Open a Schema Archive Job](#)
- [Describe and Save the Schema Archive Job](#)
- [Refine Schema Archive Properties](#)

- [Specify Schema Archive Job Options](#)
- [Set Schema Archive Job Notifications](#)
- [Create a Version and Save the Schema Archive Job](#)
- [Browse and Modify Schema Archive Contents](#)
- [Create a Job Report](#)
- [Use an Archive in Schema Comparison Jobs](#)

Create or Open a Schema Archive Job

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

To create a new schema archive

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

To open an existing schema archive

1. Select **File > Open Archive**.
2. Select a job from the list, and then click **OK**.
The Schema Archive Job editor opens.

Describe and Save the Schema Archive Job

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

To describe and save a configuration archive job

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

Specify a Schema Archive Job Source

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

To specify a schema archive job source

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

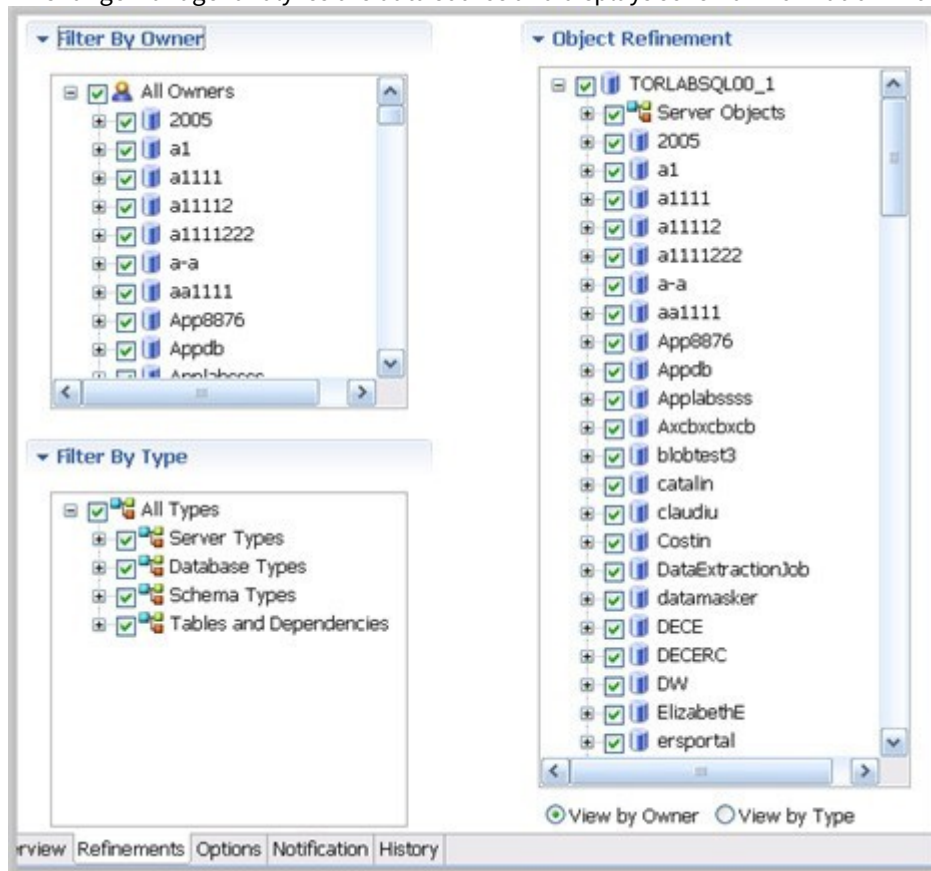
Refine Schema Archive Properties

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


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

To refine schema archive properties

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



2. Use the check boxes to select only the data source objects from which you want to create a schema archive. Within the same archive, properties are retained across versions unless you specify new values.

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

Specify Schema Archive Job Options

The Options tab contains the **Include Objects Options**, **DDL Extraction Options**, and the **Automatic Purge Options** settings.

Include Objects Options

The **Include Table Dependencies** option includes table dependent objects in the results of the archive job.

DDL Extraction Options

You define whether DDL is extracted and stored in files during the archive process using the DDL Extraction Options. You can also obtain the DDL from the archive after the job runs.

These options include:

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

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

To specify the DDL extraction process

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

Automatic Purge Options

The Automatic Purge Options delete archive files after a number of days, number of versions, and/or when the size of the files reaches a certain limit. The purge conditions are checked and enforced when an archive job is saved or when a new version is created.

To specify the conditions for automatically deleting archives

- In the Options tab, fill in one, two, or all of the options:
 - The number of **Days** after which archives are deleted.
 - The number of **Versions** stored, after which the oldest version is deleted.
 - The number of **MB** total size of the version files, after which the oldest version is deleted.

Set Schema Archive Job Notifications

When a job completes, DB Change Manager provides various ways of notifying you. For more information, see [Setting job notifications](#).

Create a Version and Save the Schema Archive Job

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

To create a version and save a schema archive job

1. In the Archive Job Editor, click **Run Job**.
The New Unsaved Version tab lists the properties of the archive.
2. Highlight the tab, and then 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.

Browse and Modify Schema Archive Contents

After you run a schema archive job, you can look at its contents and, if needed, modify the list of objects to be included in the job. A View Objects Summary report shows you what objects are selected and are included in the DDL Extract script.

To browse the contents of the archive

- In the **Archive Contents** list, use the scroll bar and page arrow buttons at the bottom to see all the archive contents.
- At the top of the **Archive Contents** list, use the search box to filter and find objects.

To modify the contents of the archive

1. In the **Archive Contents** list, right-click an object to select or deselect multiple objects.
2. Check or uncheck the boxes to select or deselect single objects.

To see a list of objects used in the Extract DDL script

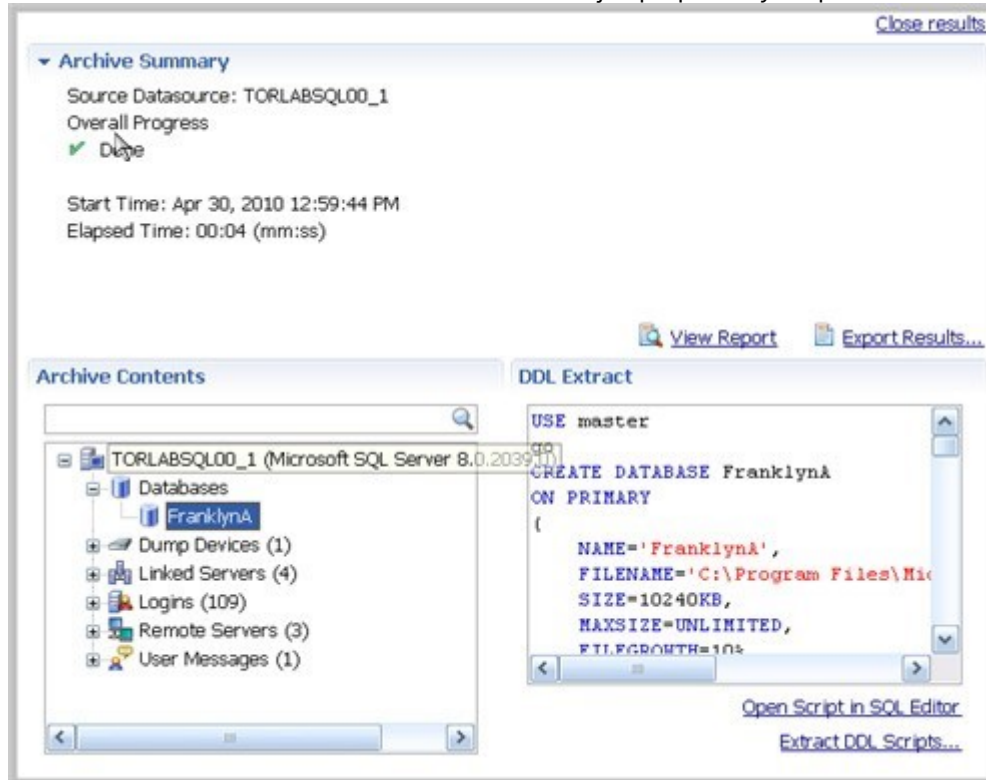
- At the top of the archive Results tab, click **View Objects Summary**. The report lists details about the types and number of objects in the job.



To see the DDL

- In the **Archive Contents** list, right-click the object, and then select **Extract**. The DDL appears in the DDL Extract pane. You create additional versions within the same archive job each time you click **Run Job**. Each

new version contains the latest values for the archive job properties you specified on the Refinements tab.



To edit the DDL

1. Open the Version tab of the schema archive.
2. At the bottom of the tab, click the **Open Script in SQL Editor** button.
The SQL Editor appears. For more information, see [SQL Editor](#).

To extract the DDL of the archive contents to a file

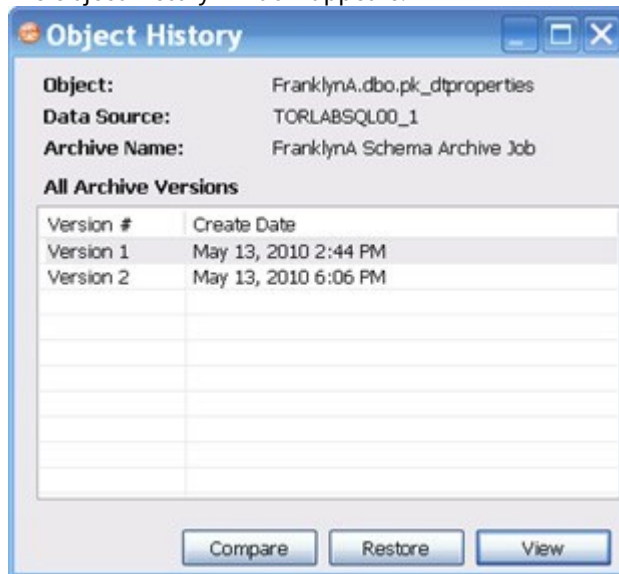
1. Open the Version tab of the schema archive.
2. In the Archive Contents list, right-click the object, and then select **Extract DLL Scripts**.
The Extract DLL Scripts window opens. For more information, see [Extract DDL Scripts](#).

Review the History of Archived Objects

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

To see the history of an archived object

1. In the **Archive Contents** list, right-click the object, and then select **History**. The Object History window appears.



2. Select the two versions that you want to compare.
3. Click **Compare**.
DB Change Manager compares the two and reports on the differences, if any.

To restore an archived object to its previous version

1. In the Object History window, select the version you want to replace and the version with which you want to replace it.
2. Click **Restore**.
The newer version is replaced by a copy of the older version.

Create a Job Report

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

Use an Archive in Schema Comparison Jobs

You can use a schema archive in schema comparison job.

To add a schema archive to a schema comparison job

- In the [Schema Comparison Job editor](#), drag and drop an archive version from **Data Source Explorer** to the **Comparison Source** or **Comparison Target** boxes.
- Alternatively, if a data source is selected as the source or target, select **Use an Archive** in the respective box, and then use the drop-down menu to choose an archive.

For more information, see [Comparing schemas](#).

Synchronizing schemas

This topic contains the following subtopics:

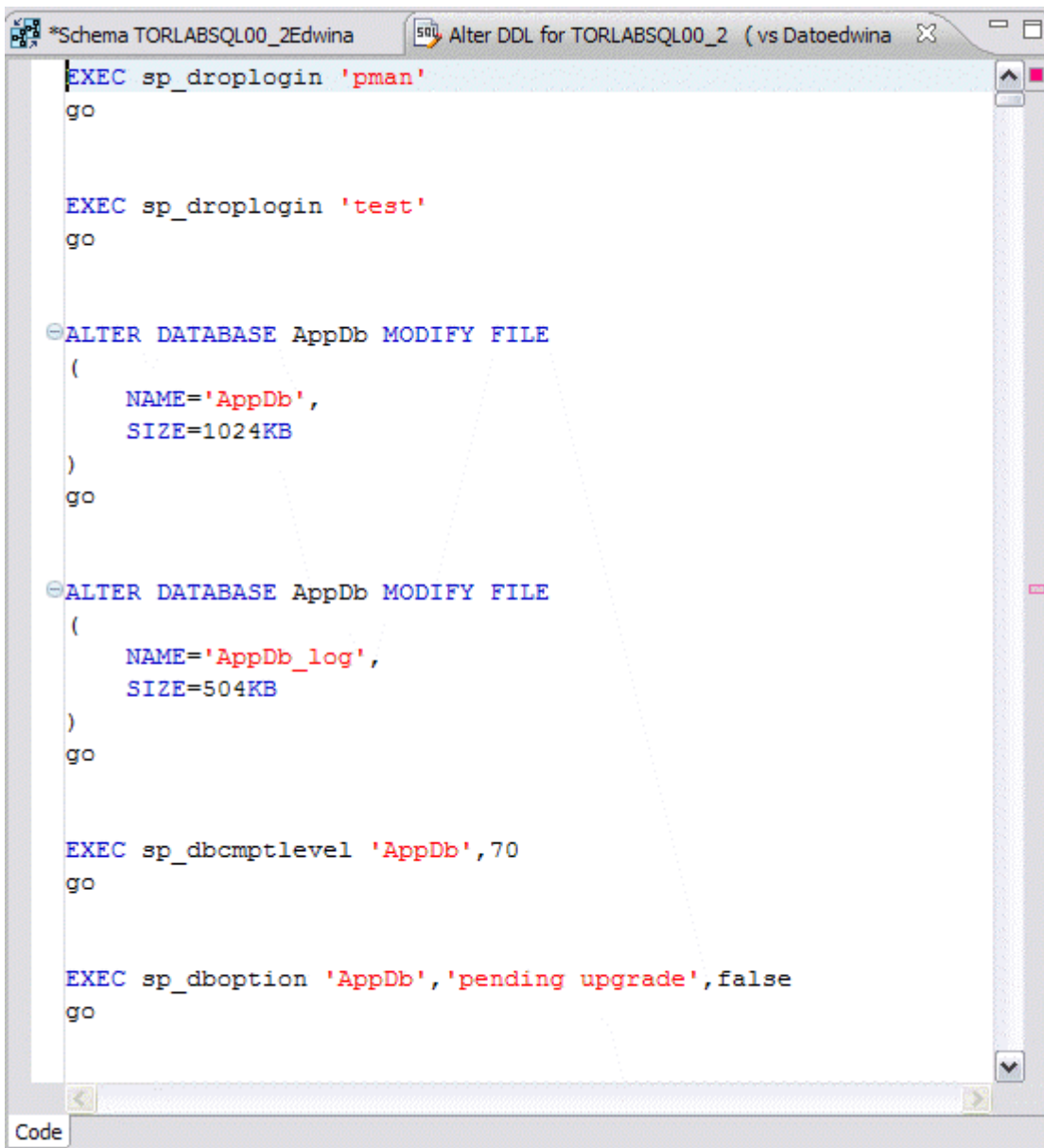
- [Resolve Schema via a Generated SQL Script](#)
- [Execute a SQL Synchronization Script](#)
- [Configure an SQL Session](#)

Resolve Schema via a Generated SQL Script

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

To generate a SQL synchronization script

- On the Results tab, click **Sync All** in the **Resolution** column of the **Comparison Results** table. A synchronizing script is produced in another window. The script can be modified manually, executed as needed, and saved at any time.



```

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

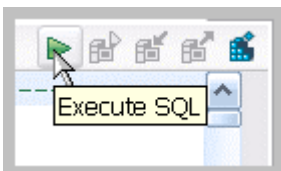
EXEC sp_dboption 'AppDb','pending upgrade',false
go

```

Code

Execute a SQL Synchronization Script

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



When an SQL file is open in the workspace, select it and choose a database and an associated catalog on which you want to execute file using the drop down menus in SQL Editor. You can click the execute icon to execute the code on

the specified database and catalog, start a transaction or commit a transaction, or modify the SQL session options prior to execution.

To execute SQL code

1. Open the SQL file you want to run. Make sure that it is associated with the correct database.
2. Click the **Execute SQL** icon.
DB Change Manager executes the code on the data source you specified and displays the results in the same tab or in a new tab.


To execute a transaction

1. Open the transaction file you want to run. Make sure that it is associated with the correct database.
2. Click the **Start Transaction** icon.
DB Change Manager executes the transaction on the data source you specified.

To commit a transaction

1. Open the transaction file you want to commit. Make sure that it is associated with the correct database.
2. Click the **Commit Transaction** icon.
DB Change Manager commits the transaction on the data source you specified.

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

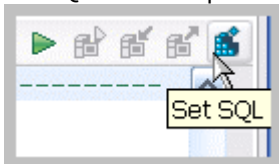
 The synchronization script generation process does not support all object parameters. You should add the following object parameters, as needed, after a synchronization script is generated: fillfactor, consumers, allow_dup_row, and statistics.

Configure an SQL Session

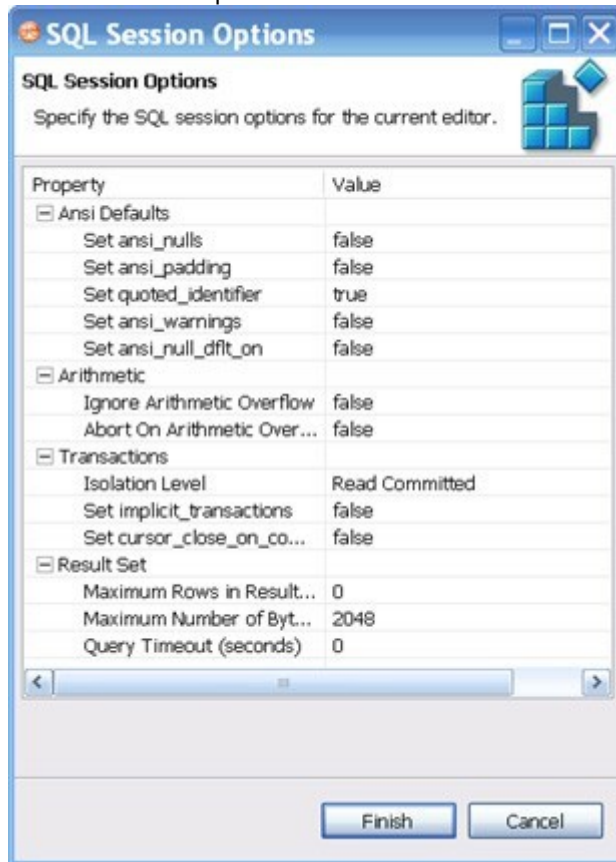
The SQL Session Options dialog provides configuration parameters that indicate to DB Change Manager how to execute code in the development environment.

To modify SQL session options

1. In the SQL Editor, click the **Set SQL Session Options** icon.
The SQL Session Options dialog appears.



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



- Click **Finish**.
The session options change and DB Change Manager executes the code as specified when you launch the script.

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

Working with jobs

- [Creating a job report](#)
- [Scheduling a job](#)
- [Setting job notifications](#)
- [Exporting and importing jobs](#)
- [Automating batch scripts](#)

Creating a job report

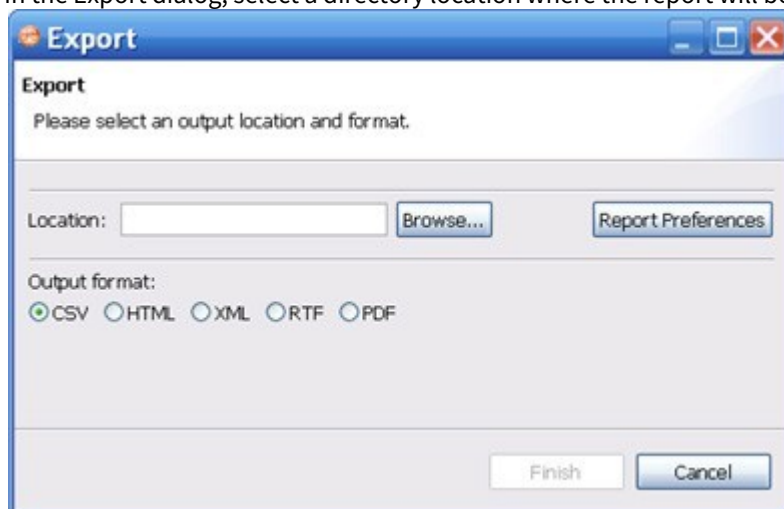
After you run a job, you can see its default report, and you can export a report in various formats.

To view or print a default job report

- On the Results tab, click the **View Report** button.
The report opens in a separate window and can be read on screen, sent to a printer, and saved to disk.

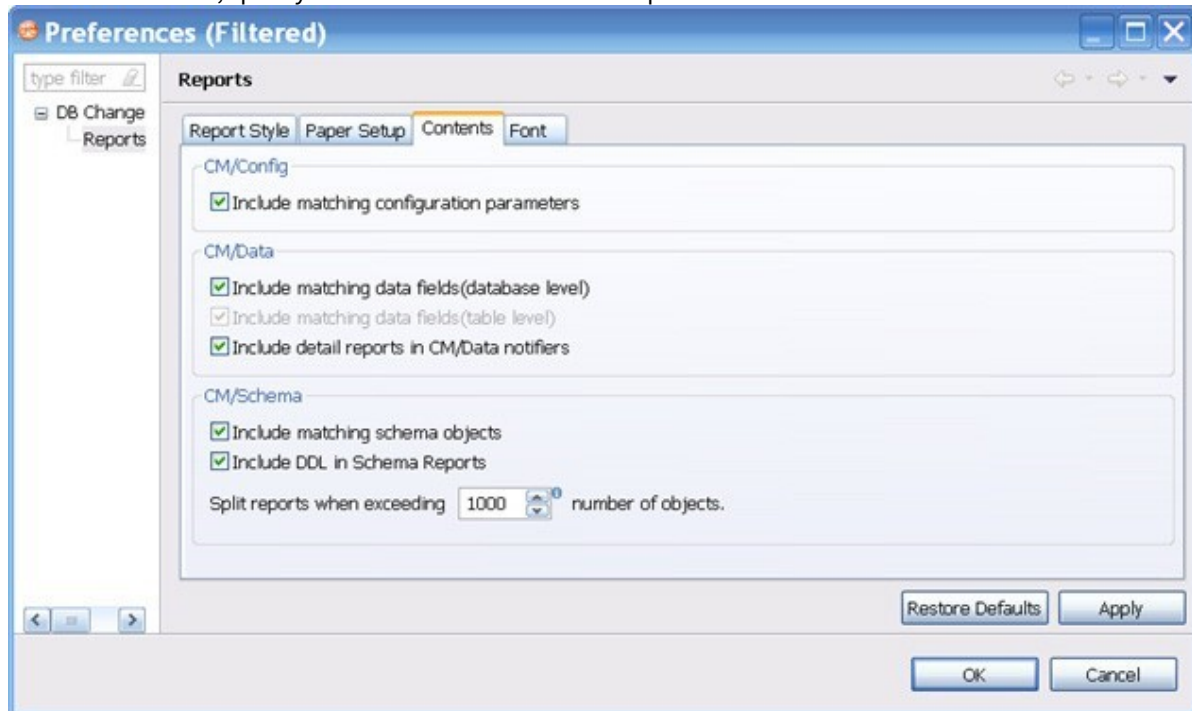
To save a report in a particular file format

1. On the Results tab, click the **Export Results** button.
2. In the Export dialog, select a directory location where the report will be saved.




3. Click an output format.
4. Click **Report Preferences**.
5. In the Report Preferences tabs, select the visual properties of the report including style, paper size, and font.
For details about the preferences, see [Reports](#).


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



- For configuration comparison jobs:
 - Select **Include matching configuration parameters** to include configuration parameters and values that matched when the job was run.
- For data comparison jobs:
 - Select **Include matching data fields (database level)** to include tables and/or views that matched in a data comparison job. If this option is not checked, only comparison tables and/or views that did not match will be included.
 - Select **Include matching data fields (table level)**, after deselecting the database level option, to include data fields that matched in rows, for which one value did not match. If this option is not checked, only comparison key values and values that did not match are included.
 - Select **Include detail reports in CM/Data notifiers** to include information for all data fields selected in a data comparison job.
- For schema comparison jobs:
 - Select **Include matching schema objects** to include information about the objects that matched in a schema comparison job.
 - Select **Include DDL in Schema Reports** to include information about the DDL.

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

- Use the Split reports when exceeding number of objects option to limit the number of objects per section.

 If your schema comparison reports are very large and use a lot of memory, you can split them into smaller sections for better performance.

Scheduling a job

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

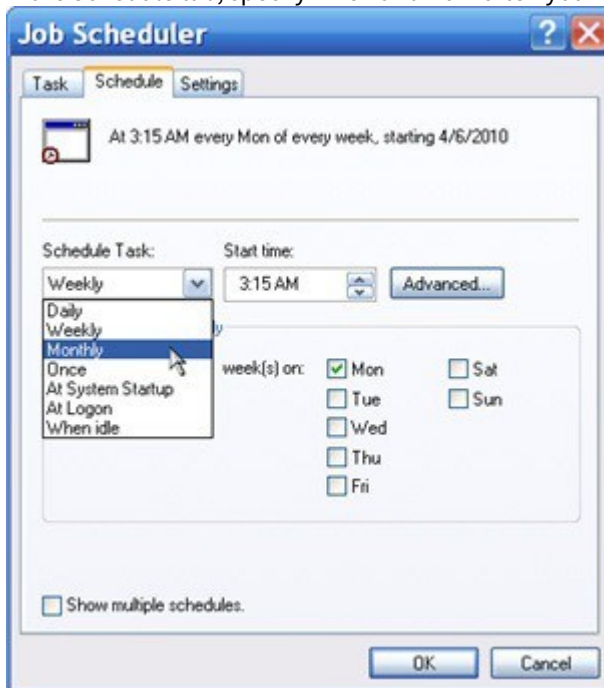
To schedule a job

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



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

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

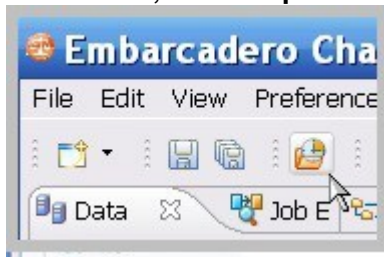


4. In the dialog, verify your user name and type your Windows password, and then click **OK**. The Job Scheduler gives you additional options on the Settings tab.



To modify an existing job schedule

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



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

Setting job notifications

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

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

Email Notifications

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

To set up email notifications

1. Click **Enable Email Notifier** to enable email notification. When you select this check box, a list of notification parameters appears.
If you have not set your email notification preferences, a message will alert you to do so. For more information, see [Notification > Email Notification](#).
2. Enter the email addresses to which the notification will be sent in the **To** field.
3. Select an email format template in the **Template** field.

To specify when to attach log files

Select a **Log Files option** to set whether DB Change Manager sends a log file every time or only when there are errors.

To specify when notifications are sent

Select the **Send Notification** options, including:

- The **When Run From** option specifies that email notifications are sent depending on where the job is initiated. By default, notifications are sent when you use either the **Workbench** or **Command Line**.
- The **Job Outcome** option specifies that email notifications are sent depending on whether the job is successful. By default, notifications are sent every time.

For comparison jobs, the **For Threshold** option specifies the threshold level for which email notifications are sent, based on job results.

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

To specify where to put notification reports

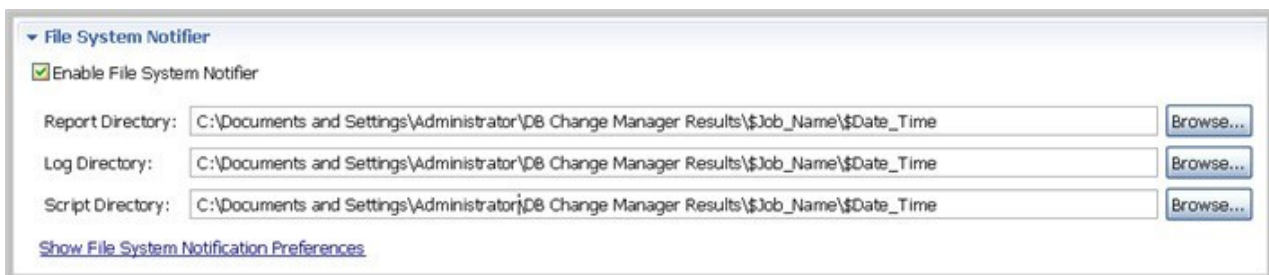
- Select **Attach Report to Email** to send it as an email attachment.
- Select **Publish Report on Web Server** to post the file to a web server. Be sure to specify the web server preferences. For more information, see [Notification > Email Notification](#).

To specify the format of email report files

Select the **Report** option, and then select the preferred file format for the attached report.

File System notification 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.

**To set up file system notifications**

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

System Tray Notifications

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

**To set up system tray notifications**

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

Send Notification and Attachment Options

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



To specify when notifications are sent

Select the Send Notification options, including:

- The **When Run From** option specifies that email notifications will be sent depending on where the job is initiated. By default, notifications are sent when you use either the **Workbench** or **Command Line**.
- The **Job Outcome** option specifies that email notifications will be sent depending on whether or not the job is successful. By default, notifications are sent every time.

For comparison jobs, the **For Threshold** option specifies the threshold level for which email notifications are sent, based on job results.

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

To specify the format of notification files

Select the **Report** option, and then select the preferred file format for the attached report.

Exporting and importing jobs

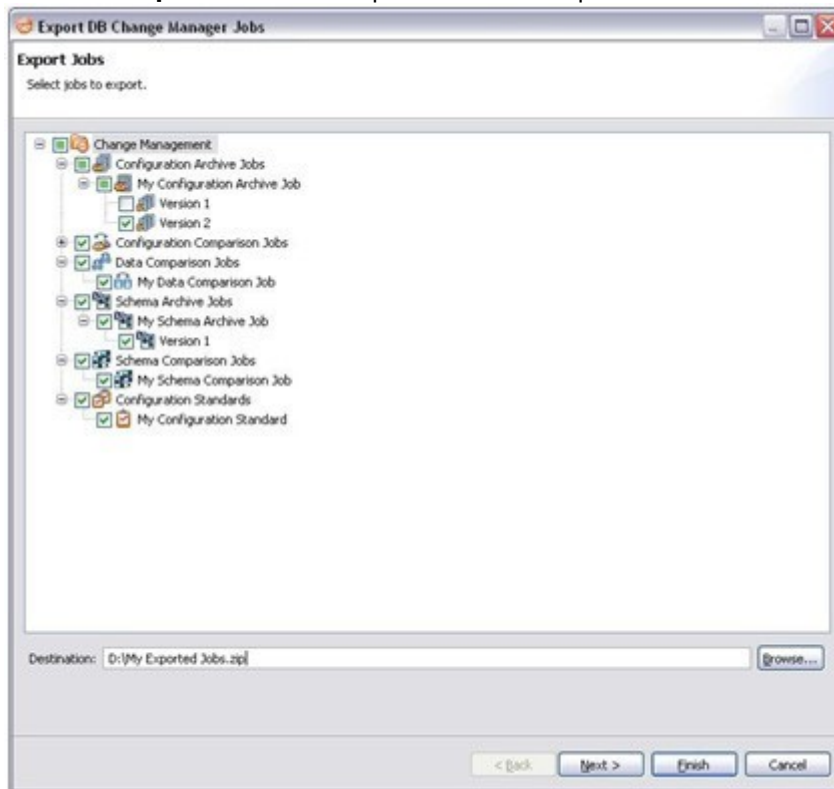
You can use the Jobs wizard to export or import a job or jobs stored in a zip file.

- [Export a Job](#)
- [Import a Job](#)

Export a Job

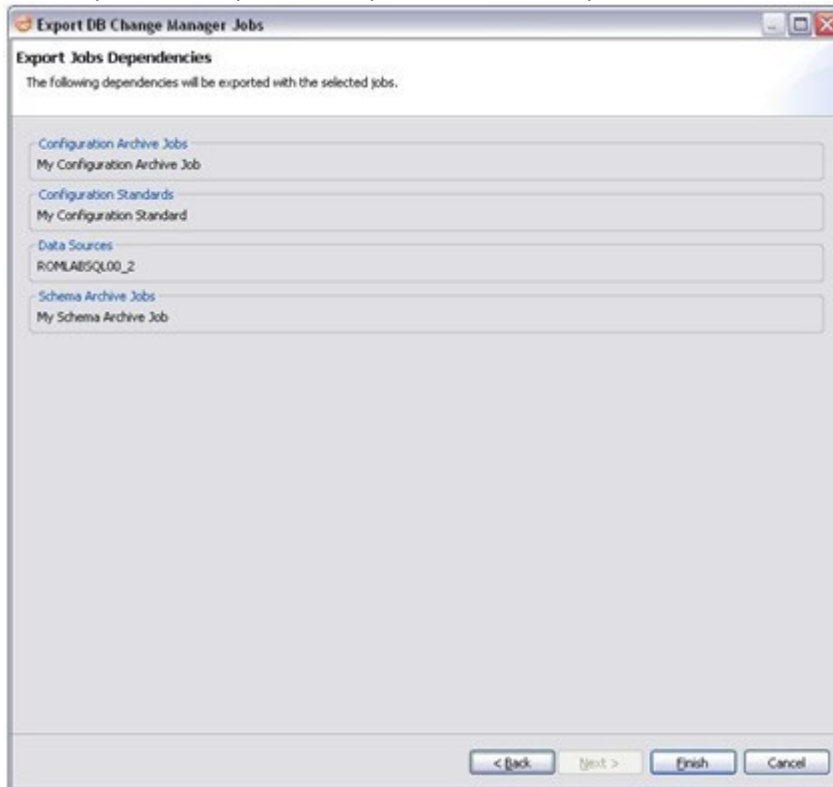
To export a job or jobs

1. Click **File > Export > Jobs**. The Export Jobs wizard opens.



2. Select the job or job version that you want to export. You can select more than one.
3. Click **Browse** and set the location of the zip file, and then click **Next**.

4. In the Export Jobs Dependencies pane, review the dependencies that will be exported with the job.



5. Click **Finish**. DB Change Manager will generate the zip file and save it to the specified location.

Import a Job

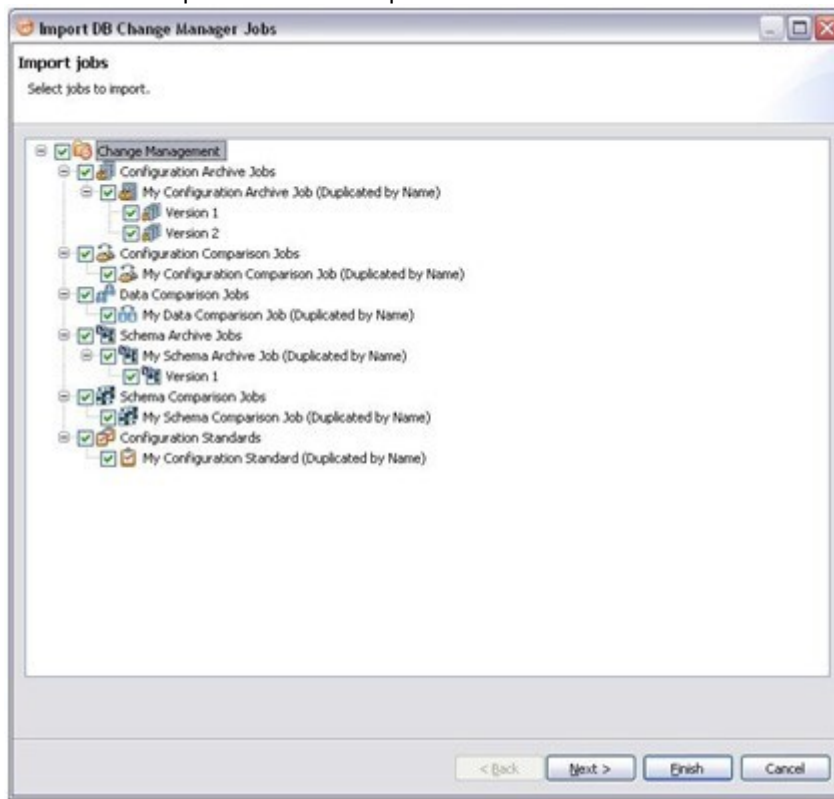
To import a job or jobs

1. Click **File > Import > Jobs**. The Workspace Selection wizard opens.



2. Select **Zip file**.

3. Click **Browse** and specify the zip file to import.
4. Click **OK**. The Import Jobs wizard opens.



5. Select the job or job version that you want to import. You can select more than one.
6. Click **Next**. The Import Jobs Dependencies pane shows the dependent jobs and data sources that will be imported.
7. Click **Finish**. DB Change Manager imports the zip file job or jobs.

Automating batch scripts

This topic contains the following subtopics:

- [Generating command line scripts](#)
- [Using the change 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.

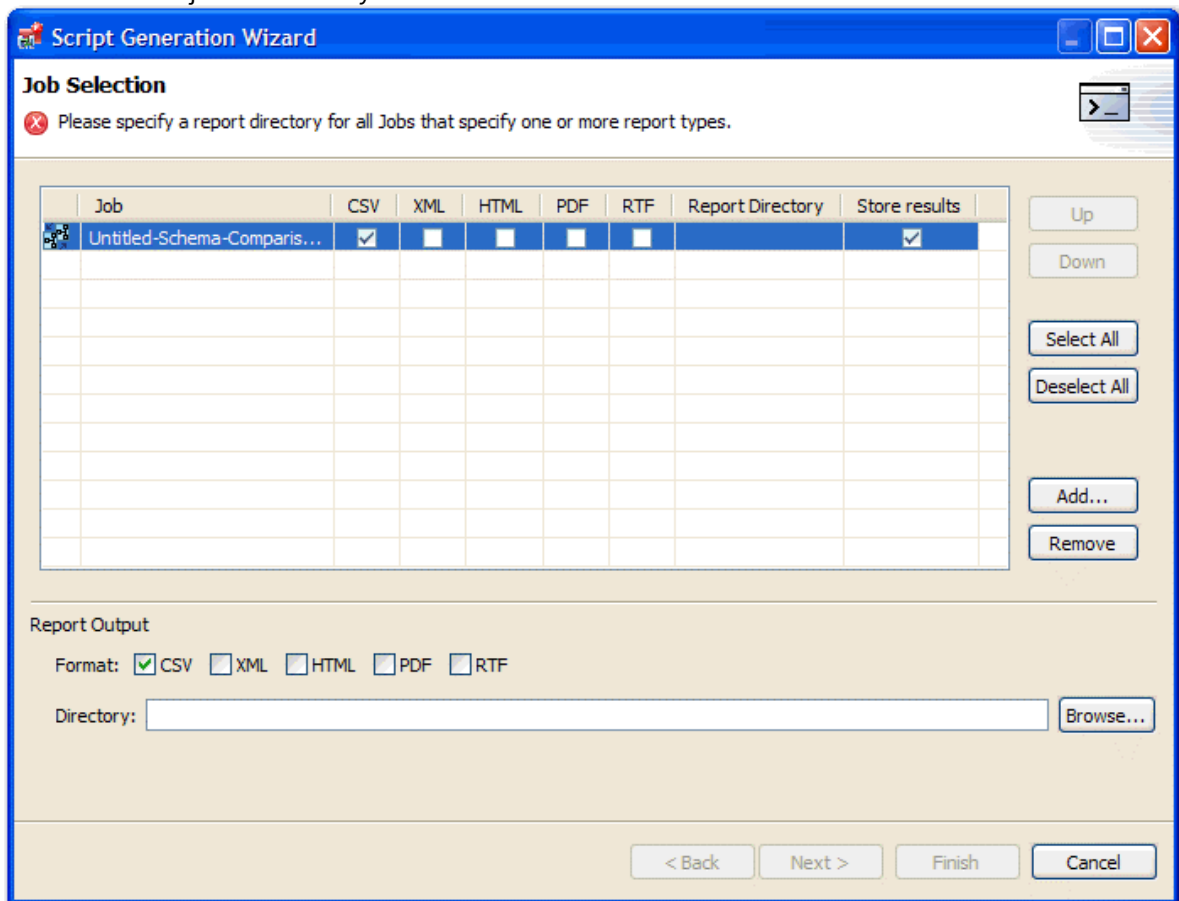
The **Report Directory** column indicates the location where reports are created.

The **Store Results** column indicates whether the job results are saved once the job has executed.

 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 you want to store the report file.
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.
5. Click **Next**.
6. Click and specify the location where you want to store the batch script file.
7. Select the script format, **.bat** or **.xml**.
8. Click **Finish**. The job immediately runs.



Using the changec Command Line Program

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

changec 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 are not stored.

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

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

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

```
changec -bulk bulk_file_name
```

Where `bulk_file_name` is a file where each row has a `-job` option. The `-job` option accepts these regular expressions:

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

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

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

To run a job from the command line:

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

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

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

Database management system reference

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

- [DBMS Operations Permissions](#)
- [Version Information Reference](#)
- [Version Information for DB2 for LUW](#)
- [Version Information for Oracle](#)
- [Version Information for SQL Server](#)
- [Version Information for Sybase](#)

DBMS Operations Permissions

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

- [Permissions Required for DB2 for LUW](#)
- [Permissions Required for Oracle](#)
- [Permissions Required for SQL Server](#)
- [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	<p>Select privilege on the relevant tables and views.</p> <p>Schema archive permissions on the database or schema objects used in the job.</p>	<p>Select privilege on the relevant tables and views.</p> <p>Schema archive permissions on the database or schema objects used in the job.</p>
Synchronize/Mask/Move Data	<p>Insert/Update/Delete privileges on the relevant tables and views.</p> <p>Schema archive permissions on the database or schema objects used in the job.</p>	<p>Insert/Update/Delete privileges on the relevant tables and views.</p> <p>Schema archive permissions on the database or schema objects used in the job.</p>

Permissions Required for Oracle

Task	Permissions for Own Objects	Permissions for All Objects
Archive and Compare Schemas	No special privileges required.	SELECT_CATALOG_ROLE
Alter Schema Object	No special privileges 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 SQL Server

Task	Permissions for Own Objects	Permissions for All Objects
Archive and Compare Schemas	No special privileges required. Encrypted stored procedures cannot be viewed.	No special privileges 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 to allow updates, and have permission to connect to the MSDB database.
Configuration Settings	Not applicable	sysadmin

Task	Permissions for Own Objects	Permissions for All Objects
Compare Data	<p>Select privilege on the relevant tables and views.</p> <p>Schema archive permissions on the database or schema objects used in the job.</p> <p>Cannot view encrypted stored procedures.</p>	<p>Select privilege on the relevant tables and views.</p> <p>Schema archive permissions on the database or schema objects used in the job.</p> <p>Cannot view encrypted stored procedures.</p>
Synchronize/Mask/Move Data	<p>Insert/Update/Delete privileges on the relevant tables and views.</p> <p>Schema archive permissions on the database or schema objects used in the job.</p>	<p>Insert/Update/Delete privileges on the relevant tables and views.</p> <p>Schema archive permissions on the database or schema objects used in the job.</p>

Permissions Required for Sybase

Task	Permissions for Own Objects	Permissions for All Objects
Archive and Compare Schemas	<p>No special privileges required.</p> <p>Stored procedures cannot be hidden.</p> <p>syscomment . text read must be allowed in order to compare or archive stored procedures.</p>	<p>No special privileges required.</p> <p>Stored procedures cannot be hidden.</p> <p>syscomment . text read must be allowed in order to compare or archive stored procedures.</p>
Alter Schema Object	<p>No special privileges required.</p>	<p>sa_role</p> <p>Enable sp_configure to allow updates</p>
Configuration Settings	<p>Not applicable</p>	<p>sa_role</p>

Task	Permissions for Own Objects	Permissions for All Objects
Compare Data	<p>Select privilege on the relevant tables and views.</p> <p>Schema archive permissions on the database or schema objects used in the job.</p> <p>The compression comparison function is unavailable for Sybase at this time.</p>	<p>Select privilege on the relevant tables and views.</p> <p>Schema archive permissions on the database or schema objects used in the job.</p> <p>The compression comparison function is unavailable for Sybase at this time.</p>
Synchronize/Mask/Move Data	<p>Insert/Update/Delete privileges on the relevant tables and views.</p> <p>Schema archive permissions on the database or schema objects used in the job.</p>	<p>Insert/Update/Delete privileges on the relevant tables and views.</p> <p>Schema archive permissions on the database or schema objects used in the job.</p>

Version Information Reference

The following subtopics describe the version information used when comparing, archiving, or creating a configuration standard:

- [Version Information for DB2 for LUW](#)
- [Version Information for Oracle](#)
- [Version Information for SQL Server](#)
- [Version Information for Sybase](#)

Version Information for DB2 for LUW

Parameter	Description
INST_NAME	Name of the current instance.
IS_INST_PARTITIONABLE	<p>Indicates whether the current instance is a partitionable database server instance.</p> <p>Possible return values are 0, if it is NOT a partitionable database server instance, and 1 if it IS a partitionable database server instance.</p>
NUM_DBPARTITIONS	<p>The number of database partitions.</p> <p>If it is NOT a partitioned database environment, this value is 1.</p>

Parameter	Description
INST_PTR_SIZE	The bit size of the current instance. Values are 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 v8.1.80.
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.


Version Information for Oracle

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

Version Information for SQL Server

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

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

Version Information for Sybase

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

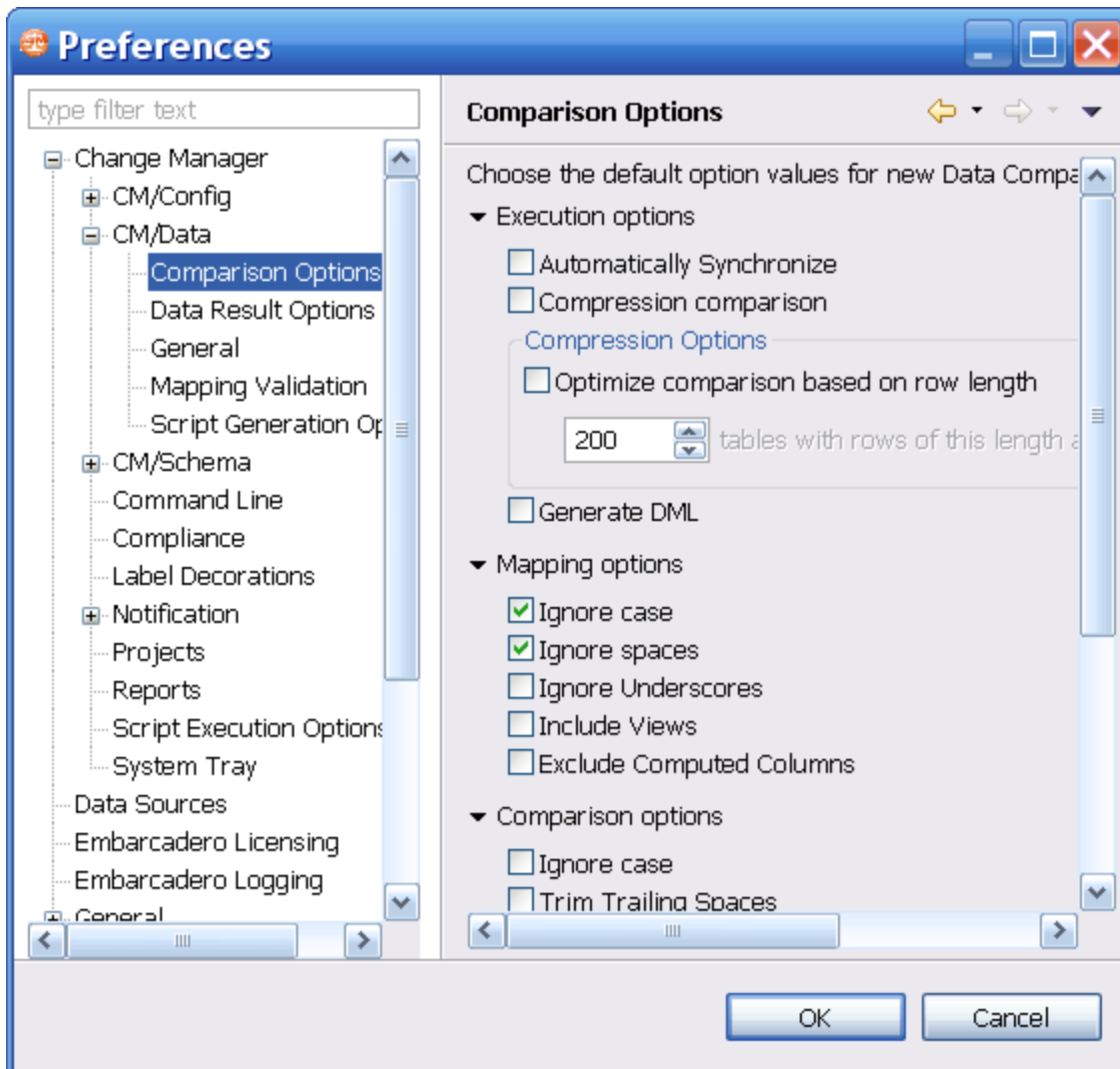
- The Server Information group displays the results of executing `sp_server_info`, where the attribute name is the configuration parameter and the value is its value.
- The Version Information group, which contains the following parameters:
 - Build Number
 - Build Type
 - Compilation Date & Time
 - Compilation Kernel
 - Compilation OS Bit Size
 - Compilation OS Platform
 - EBF Version
 - Post Build Type
 - Product
 - Product Code
 - Version

Preferences reference

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

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

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



DB Change Manager Preferences

The following topics describe preferences specific to DB Change Manager:

- [CM/Config > Archive Options](#)

- [CM/Config > Comparison Options](#)
- [CM/Data > Comparison Options](#)
- [CM/Data > Data Result Options](#)
- [CM/Data > General](#)
- [CM/Data > Mapping Validation](#)
- [CM/Data > Script Generation](#)
- [CM/Schema > Archive Options](#)
- [CM/Schema > Comparison Options](#)
- [Notification > Email Notification](#)
- [Notification > Email Notification > Advance Settings](#)
- [Notification > Email Notification > Contacts](#)
- [Notification > Email Notification > Templates](#)
- [Notification > File System Notifier](#)
- [Notification > Popup Notification](#)
- [Notification > Popup Notification > Advanced Settings](#)
- [Projects](#)
- [Reports](#)
- [Script Execution Options](#)
- [System Tray](#)
- [Task Scheduler](#)
- [Data Sources](#)

CM/Config > Archive Options

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

- [Extract DDL](#)

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

CM/Config > Comparison Options

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

- [Extract DDL](#)

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

CM/Data > Comparison Options

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

Data comparison options are grouped into these categories:

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

Execution Options

Execution Options control aspects of processing the job:

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

 **Compression Comparison** is not available for Sybase.

- **Optimize comparison based on row length.** The value determines when compression is applied on a table during the data comparison process, based on the total size of all of the rows. If the size of the rows is greater than this value, then compression is automatically enabled.
- **Generate DML** indicates that the synchronization DML used in the job is automatically placed in a separate file. If this option is selected, you can specify on the Notifications tab to send the resulting output with any notification.

Mapping Options

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

- **Ignore Case** indicates that objects are analyzed without regard to case. For example, the table **Product_Prices** is compared to the table **PRODUCT_PRICES**.
- **Ignore Spaces** indicates that spaces are ignored when analyzing objects. For example, the table **Product Prices** is compared to the table **ProductPrices**.
- **Ignore Underscores** disregards underscores in object names for comparison purposes. For example, the table **Product_Prices** is compared to the table **ProductPrices**.
- **Include Views** indicates that the comparison process 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.

For more information, see [Customize Database Mapping](#).

Comparison Options

The Comparison Options specify how the job handles naming conventions:

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

Resolution Options

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

- **Drop and Recreate Foreign Keys** indicates that all foreign keys are dropped prior to applying changes during the resolution process. This option should be used with **Wrap Statements in Transaction** to ensure that foreign keys do not remain dropped in the event that the synchronization script used to resolve it fails. If this option is not selected, a data comparison job generates ordered table insert and delete statements, and you must do the following tasks:
 - In the Mapping page, select all dependent tables. If a referenced table is missing from the comparison, DB Change Manager cannot ensure the referential integrity of the data.
 - If you also change the comparison key, you must set the same column as a comparison key for all dependent tables. If you do not do so, DB Change Manager cannot insure the referential integrity of the data.

 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.


 This option is not supported for DB2 for LUW scripts.

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

CM/Data > Data Result Options

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

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

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

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

CM/Data > General

The General options relate to data mapping:

- **Remembered dialog decisions.** When you select a default masking rule on a database or table, this option affects the masking rule used on the corresponding tables or columns. **Yes** means the rule is used on all. **No** means the tables or columns use the rule set explicitly for them. The default Prompt allows you to make the decision at the time you select the masking rule on the database or table.
- **Don't show again decisions.** When the **Automatically Synchronize** option is toggled, some mappings become invalid and are removed. If you want to see this information, select this box.

CM/Data > Mapping Validation

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

Mapping Validation

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

▼ **Comparison Mapping Validation**

Comparison Key required for comparison: Error ▼

Column data type mapping compatibility: Error ▼

▼ **Auto Synchronization Mapping Validation**

Relational table cycles: Error ▼

Parent tables not mapped: Error ▼

Child tables not mapped: Error ▼

Source and Target Table Mapping Validation: Error ▼

Check Constraints on target table: Warning ▼

Data truncation with length differences: Error ▼

Data truncation with scale/precision differences: Error ▼

Nullable column to Non-Nullable column: Error ▼

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

Non-Unique column to Unique column: Error ▼

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

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

Unmapped source columns: Warning ▼

Unmapped target columns: Warning ▼

JDBC Driver support for data types: Error ▼

Data Type conversion support on Data Movement: Error ▼

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

- **Error** means that an error message indicating that there is a problem with the mapping validation is issued.

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

CM/Data > Script Generation

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

- **Warn about duplicate rows.** By default, DB Change Manager issues a warning whenever it encounters duplicate rows while generating a synchronization script.
- **Exclude duplicate rows.** If you want to automatically ignore duplicate rows while generating a synchronization script, select this option.
- **Set quoted identifier ON/OFF.** For SQL Server and Sybase only, if the quoted identifier database option is off, this option prefixes the data synchronization script with `SET QUOTED IDENTIFIER ON` and suffixes it with `SET QUOTED IDENTIFIER OFF`.

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

CM/Schema > Archive Options

The **Include Table Dependencies** option extracts all table dependencies. By default, it is not selected.

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

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

The **Automatic Purge Options** delete archive files after a number of days, number of versions, and/or when the size of the files reaches a certain limit. The purge conditions are checked and enforced when an archive job is saved or when a new version is created.

- The number of **Days** after which archives are deleted.
- The number of **Versions** stored, after which the oldest version is deleted.
- The number of **MB** total size of the version files, after which the oldest version is deleted.

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

CM/Schema > Comparison Options

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

The list of options differs depending on the DMBS of the source and target data sources. The parameters in the panel correspond to the job options listed on the Options tab of the [Schema Comparison Job Editor](#). The option categories include:

- [CM/Schema > Comparison Options > Include Objects Options](#)
- [CM/Schema > Comparison Options > Compare Options](#)
- [CM/Schema > Comparison Options > Decorator Options](#)
- [CM/Schema > Comparison Options > Identifier Options](#)
- [CM/Schema > Comparison Options > DDL Extract Options](#)
- [CM/Schema > Comparison Options > Additional Options](#)
- [CM/Schema > General](#)
- [CM/Schema > Mapping Options](#)
- [CM/Schema > Refinements Options](#)
- [Command Line](#)
- [Compliance](#)
- [Label Decorations](#)

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

CM/Schema > Comparison Options > Include Objects Options


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

These options include:

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

CM/Schema > Comparison Options > Compare Options

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

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

These options include:

- **Ignore Storage** ignores storage comparison when running the job.
- **Ignore Partition Differences** (*Oracle only*) ignores partition differences when running the job.
- **Ignore Column Order** ignores table column order differences when running the job.
- **Ignore Check Constraints Quotes** ignores quotes in check constraints when running the job.
- **Ignore Table Comment** ignores table comments when running the job.
- **Ignore Text Case** ignores syntax case in text object comparisons when running the job. (views, procedures, trigger, etc.)
- **Ignore Text Comments** ignores comments in text objects when running the job.
- **Ignore Text Whitespace** ignores white space when running the job.

- **Ignore Tablespace File Name** ignores tablespace file names when running the job.
- **Ignore Password Differences** ignores differences in object passwords when running the job. (group, login, role, user, etc.)
- **Ignore Tablespace** (*IBM only*) ignores tablespaces for table, index, cluster, etc.
- **Ignore Object Permissions** ignores the permissions on objects when running a job.
- **Ignore Logging/No Logging Differences** (*Oracle only*) ignores object differences where one is set to **Logging** and the other is set to **No Logging**.
- **Ignore Sequence Start Value** (*Oracle only*) ignores the **Start Value** property of an **Oracle Sequence**.
- **Ignore Data Capture** (*IBM only*) ignores the **Data Capture** property for tables.
- **Ignore File Group** (*SQL Server only*) ignores the File Group property for tables and indexes.
- **Ignore Segment** (*Sybase only*) ignores the **Segment** property for tables, indexes, and partition definitions.

CM/Schema > Comparison Options > Decorator Options

Decorator Options affect particular database objects.

These options include:

- **Preserve Target File Group** (on extended alter) (*SQL Server only*) When an extended alter is generated for an object, the target objects file group remains the same as the original object if this option is enabled. If this option is disabled, the targets objects file group are set to the one used by the source object.
- **Preserve Target Storage** (*Oracle only*) The functionality is similar to **Preserve Target File Group** but this option applies to the storage properties of Oracle objects.
- **Preserve Target Tablespace** (*Oracle only*) The functionality is similar to **Preserve Target File Group** but this option applies to the tablespace properties of Oracle objects.
- **Preserve Target Segment** (*Sybase only*) Preserves the **Segment** property for tables, indexes, and partition definitions.
- **Recompile** This option triggers a recompilation for dependent procedures, when the target object needs to be dropped and recreated (extended altered).
- **Comment Out Extended Alters for Tablespaces** An extended alter is essentially a drop and then a recreate. For tablespaces, an extended alter is a dangerous operation that may cause data loss. This option is by default enabled and generates commented out DDL for tablespace extended alter.

CM/Schema > Comparison Options > Identifier Options

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

These options include:

- The **Ignore Name Case** option specifies that the syntax case is ignored when doing comparisons. This option is deselected by default.
- The **Ignore Constraint Names** option specifies that differences in table constraint names are ignored when doing comparisons. This option is deselected by default.
- The **Ignore Specific Name** (*IBM DB2 for LUW only*) option ignores SQL specific names in procedures, functions, and methods. This option is deselected by default.

CM/Schema > Comparison Options > DDL Extract Options

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

CM/Schema > Comparison Options > Additional Options

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

These options include:

- **Enable Dependency Linking.** When this option is activated, the comparison contains information about dependencies between objects. This information is critical for the ordering of the generated DDL. If the option is not enabled, the schema comparison is faster but when DDL is generated for more than one object, the ordering is incorrect. By default, **Enable Dependency Linking** is deselected for Oracle, so all GRANT statements for Oracle objects are not extracted and compared.

CM/Schema > General

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

- **Show uncomment alter tablespace script confirm dialog.** There is an alter tablespace script confirm dialog that displays by default. Deselect this option if you do not want to see this dialog.
- **Do not ask about running the created jobs.** Select this option if you want to ignore the system prompt when running schema comparison jobs. If you select this option, you can also select whether to automatically run a schema comparison job when an object is created.
- **Automatically launch jobs created from selected objects.** Select this option if you want to automatically run a schema comparison job when an object is created.

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

CM/Schema > Mapping Options

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

- **Ignore Case.** By default, schema comparison jobs ignore the case of the texts. Deselect this option if case is important to you.
- **Ignore Spaces.** By default, schema comparison jobs ignore spaces in the texts. Deselect this option if spaces are important to you.
- **Ignore Underscores.** By default, schema comparison jobs compare underscores in the texts. Select this option if you do want to ignore underscores.
- **Show Users.** (*Oracle only*) By default, schema comparison jobs show users from the source and target that have at least one object. Select this option if you want to show all users from the target, regardless of whether they have any objects in their schema, and map them with users from the source.

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

CM/Schema > Refinements Options

The **Refinements Options** determine whether new objects are included. By default, they are included when you run a schema comparison job. You can configure the job to ignore new objects by unchecking this option.

Command Line

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

- **Windows (*.bat).** By default, the command line interface produces batch files.

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

Once you configure new defaults, click **Apply**. For more information, see [Generating Command Line Scripts](#).

Compliance

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

These options include:

- **Compliance Met**. By default, this value is 100%, which allows a job to pass.
- **Compliance Critical**. By default, this value is 75, meaning that 75-99% compliance results in a warning.
- **Compliance Not Met**. By default, this value is 0, meaning that 0-74% compliance results in a failure.

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

Label Decorations

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

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

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

Notification > Email Notification

You can specify email server parameters that enable email notifications for jobs.

The parameters include:

- **Host**. The name of the email host server is required.
- **Port**. The default port number is 25.
- **Sender Address**. The email address to use for sending email from DB Change Manager to others.
- **User Name**. The user name of the sender on the mail server.
- **Password**. The password of the sender on the mail server.
- **Web Server Report Directory**. The root directory of the web server where DB Change Manager saves reports; for example, c:\apache2\htdocs.
- **Web URL Prefix**. The URL used by DB Change Manager to specify links in the report and in the notification email; for example, <http://www.websserver.com/>.

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

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

Notification > Email Notification > Advance Settings

Advanced Settings lets you map SMTP properties to values. Refer to <https://javaee.github.io/javamail/FAQ> for more information.

SMTP properties are listed at: <http://https://javaee.github.io/javamail/docs/api/com/sun/mail/smtp/package-summary.html>.

Notification > Email Notification > Contacts

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

To add a contact

1. In **Preferences > Notification > Email Notification > Contacts**, click **Add**.
2. Enter an email address, and then click **OK**.


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

Notification > Email Notification > Templates

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

Available actions include:

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

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

- **Add.** Opens the Add Template dialog.
- **Remove.** Removes the selected template.

To create a new email notification template

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

Notification > File System Notifier

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

- **Enable file system notification service.** By default, file system notifications are not enabled. Select this option to enable them.
- **Report, Log, and Script Directories.** By default, file system notifications are created in the Documents and Settings/username/ Change Manager Results directory. Click **Browse** to change the default directories.

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

Notification > Popup Notification

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

- **Enable system tray notification service.** By default, system tray notifications are not enabled. Select this option to enable them.
- **Display notification time.** By default, the notification lasts for 3000 milliseconds.
- **Show only when minimized to tray.** By default, the popup displays when the **System Tray** is minimized.
- **Play sound.** By default, if notifications are enabled, a sound accompanies a notification.
- **Custom sound.** You can change the default sound and select a .wav sound file.

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

Notification > Popup Notification > Advanced Settings

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

- **Generate notification when job is run from.** By default, if notifications are enabled, a notification is generated for each job run from the Workbench or from the command line. You can limit notifications to one or the other.
- **Generate notification when job finishes.** By default, a notification is generated for each job that succeeds or fails. You can limit notifications to one or the other.
- **Generate notification for threshold.** This option is currently unavailable.

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

Projects

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

- **Suppress Project Required Warning Dialog.** This option disables the warning dialog when using the Project Explorer. By default, it is disabled.

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

Reports

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

- The Report Style tab
 - **Logo.** You can browse to upload a logo. The logo should be 80 pixels wide by up to 420 pixels tall.
 - **Accent color.** You can select an accent color from the color palette.

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

- The Paper Setup tab
 - **Paper size.** You can use the dropdown list to select from a list of standard paper sizes, and choose between **Portrait** or **Landscape** orientations.
 - **Margins.** You can set the **Top**, **Right**, **Left**, and **Bottom** margins for the paper. If you prefer not to use inches, set the **Unit** option to either centimeters or millimeters.

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

- The Contents tab
 - **CM/Config: Include matching configuration parameters.** By default, matching configuration parameters are included in a report.
 - **CM/Data: Include matching data fields (database level).** By default, matching data fields in databases are included in a report.

- **CM/Data: Include matching data fields (table level).** By default, matching data fields in tables are included in a report. You cannot change this option.
- **CM/Data: Include detail reports in CM/Data notifiers.** By default, details are included in a data job notification.
- **CM/Schema: Include matching schema objects.** By default, matching schema objects are included in a report.
- **CM/Schema: Include DDL in Schema Reports.** By default, DDL is included in a report. If you do not need the DDL in the report, and to improve performance, deselect this option.
- **Split reports when exceeding _ number of objects.** By default, reports including over 1000 objects are split into multiple sections, to improve performance. You can change the default number of objects in a section.

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

- The Font tab
 - **Path.** By default, DB Change Manager includes the **FreeSerif.ttf** font for use in reports. You can change this to another font on your system.
 - **PDF Encoding.** If you change the font, make sure that the appropriate PDF encoding is set. Leave the field blank to use the default encoding.

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

Script Execution Options

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

- **Connection Auto Commit.** By default, auto commit applies to synchronization scripts for data comparison jobs.

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

System Tray

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

- **Show System Tray Icon.** By default, the **System Tray** icon is shown.
- **Minimize to System Tray.** You can minimize the system tray by selecting this option.

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

Task Scheduler

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

- **Always schedule auto generated batch files.** By default, automatically generated batch files do not run on schedule. Select this option to require scheduling.
- **Batch files directory.** By default, batch files are saved in **Documents and Settings\username\change_manager_x\workspace\Change Management\AutoGeneratedBatchFiles**. You can change this to another directory.

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

Data Sources

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

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

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

Eclipse SQL Development Preferences

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

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

SQL Development > SQL Editor

The SQL Editor options include:

- **Enable the SQL Parser for Files Smaller than 700 KB**
By default, for better performance files smaller than 700 KB are enabled, and larger files are disabled. You can configure the file size.
Deselecting **Enable the SQL Parser** disables many of the “smart” SQL Editor features including code formatting, auto completion, semantic validation, and hyperlinks.
- **Severity Level for Semantic Validation Problems**
This option determines how semantic code errors are flagged in the SQL Editor and the Problems view.

SQL Development > SQL Editor > Code Assist

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

- **Enable Auto Activation**
By default, Code Assist is enabled. The auto-activation delay determines how long before the Code Assist feature activates.
- **Insert Single Proposals Automatically**
By default, if only a single code completion suggestion is returned, it is automatically inserted. To disable automatic inserts, deselect this option.
- **Fully Qualify Completions Automatically**
By default, code completion results are the minimum required to identify the object. Select this option to return specific, fully-qualified code completions.
- **Code Assist Color Options**
By default, the background color of code completion proposals is white and the foreground is black. You can modify these options using the color picker.

SQL Development > SQL Editor > Code Formatter

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

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

SQL Development > SQL Editor > Results Viewer

The Results Viewer options specify how the Results view displays results, and whether the results are shown in the SQL Editor or sent to a file.

Results sets options include:

- **Maximum number of result sets**
By default, up to 100 results sets are included.
- **Suppress execution messages**
Select this option to disable system messages in the Results view.

By default, results are shown in SQL Editor. The Show results in editor options include:

- **Grid refresh interval (ms)**
This option defines the speed in milliseconds that the Results view refreshes.
- **Maximum result rows to sort**
By default, the results will sort up to 25,000 rows.
- **Results Format**
By default, Results are in a grid format. You can change that to HTML, XML, or text format.
- **Stripe the rows of the results table**
This option adds intermittent highlighted bars in the Results view.
- **Display results in separate tab in the SQL Editor**
This option opens the Results view in a separate tab on the Workbench. You can save results to a file. When selected, you specify:
 - **Prompt to save file**
Whether DB Change Manager will prompt you to specify a location for the file. If so, you must specify a default location for the file.
 - **Include SQL text**
To include SQL text, you must specify a file format and text qualifier.

SQL Development > SQL Editor > Syntax Coloring

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

SQL Development > SQL Filters

The SQL Filters options specify whether to ignore system objects when performing jobs. By default, system objects are ignored. For more information, see [Filtering Data Source Objects](#).

User interface reference

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

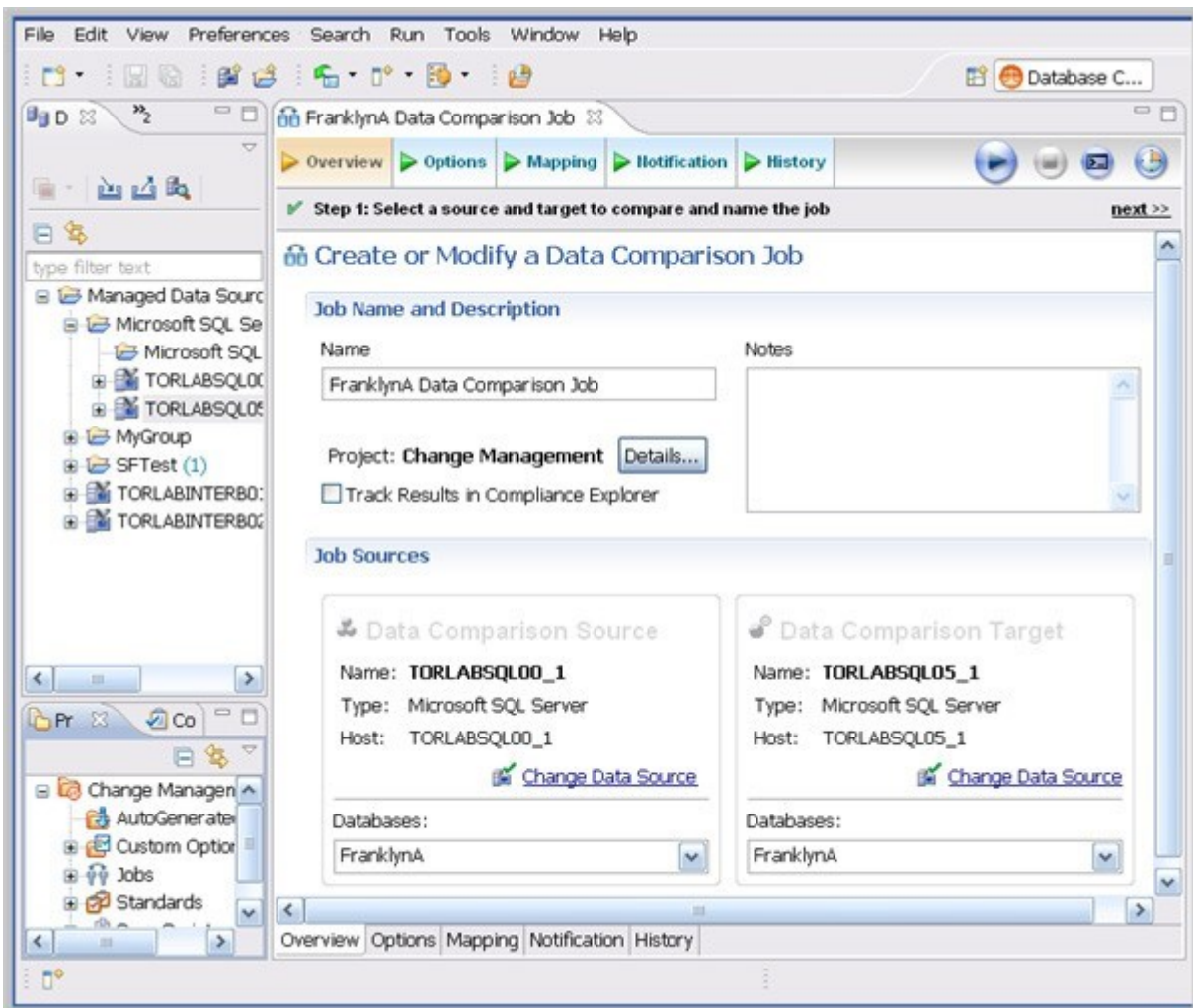
The following interface components compose the DB Change Manager work environment:

- [DB Change Manager Workbench](#)
- [Welcome Page](#)
- [Workspace](#)
- [Views](#)
- [Editors](#)

DB Change Manager Workbench

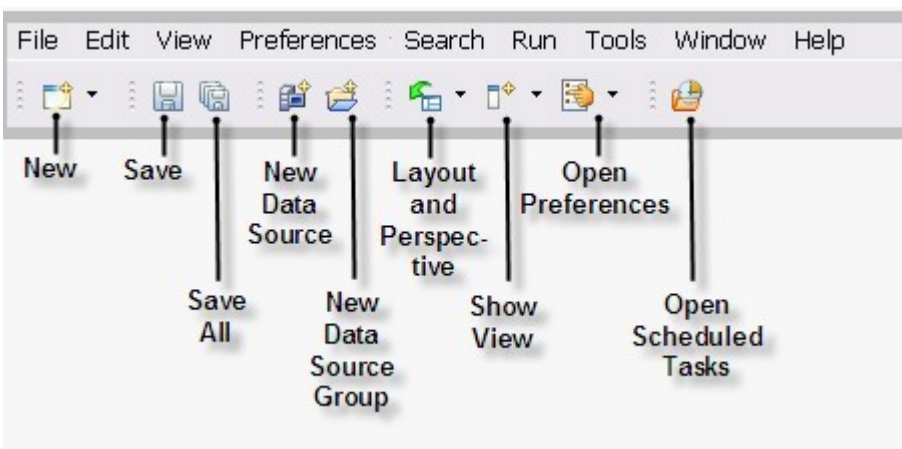
The Workbench is the DB Change Manager development environment. It provides the interface to create, manage, and navigate DB Change Manager processes through interaction with views, editors, and menus.

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



The Workbench space is composed of Views and Editors, as well as the Menu Bar and Command Toolbar.

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



License Selection Dialog

The license Selection dialog opens if, at startup, the product detects more than one license available. You can choose the license you want to use from a list that includes details about each license.

Welcome Page

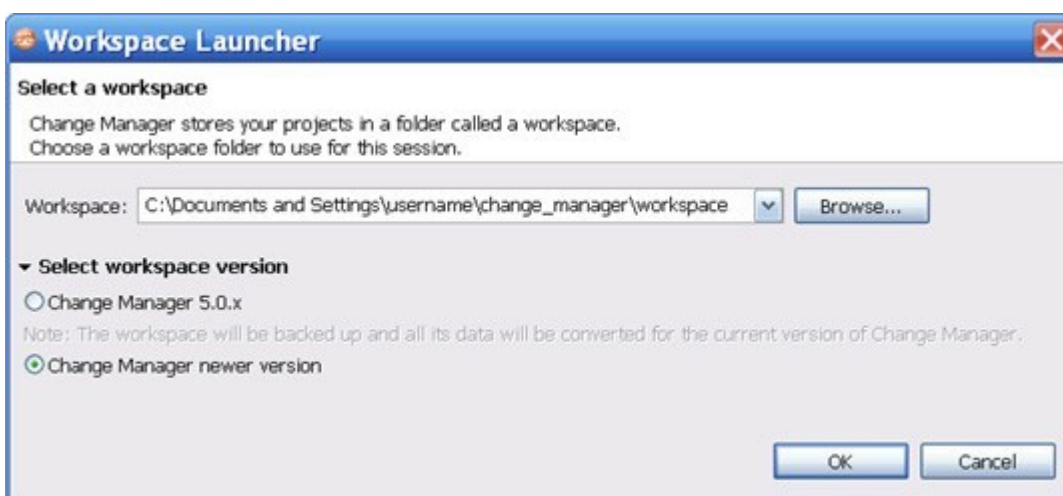
The Welcome page opens when you launch DB Change Manager. It provides easy access to information for new users, and helps you get started with the application.

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

Workspace

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

The Workspace launcher allows you to switch between multiple DB Change Manager workspaces, or convert workspaces to new versions of DB Change Manager.



To switch DB Change Manager workspaces

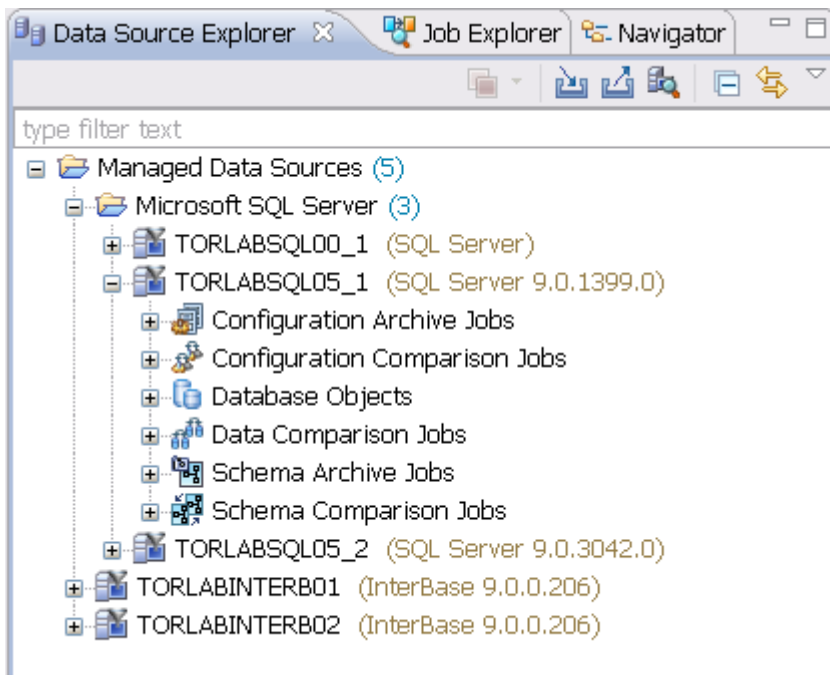
1. From the Main menu, click **File > Switch Workspace**.
2. In the Workspace Launcher dialog, browse to the directory and select it.
3. Select the version of the older workspace.
4. Click **OK**.

Views

- Job explorer
- Navigator view
- Compliance Explorer
- SQL Errors view
- Data Diff view
- Extract DDL Scripts
- SQL Log
- SQL Errors Log
- Search view

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

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



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

All views have two ways to access command menus:

- Right-click the view tab

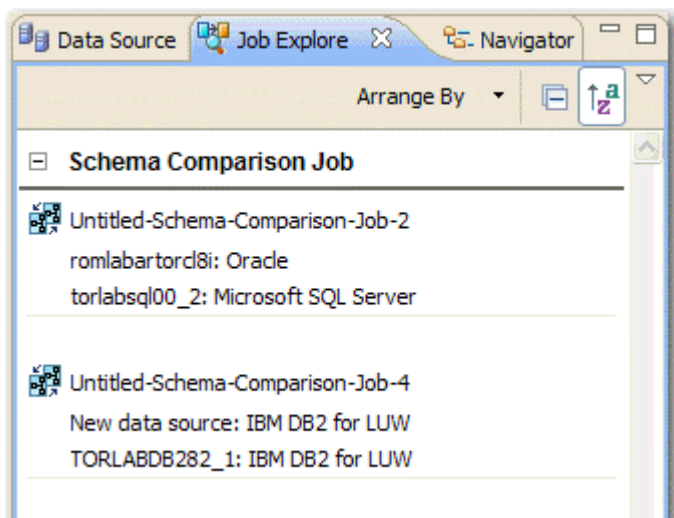
- Click the icon in the view toolbar

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

- **To move a view.** Click the tab of the view and drag your mouse. A frame indicates where the view will fit in the Workbench. Release the button to move the view to its new location.
- **To toggle a floating view.** Right-click the tab of the view and select Detached. The view switches to a floating window format that can be used to position it ovetop of embedded views in the Workbench.
- **To build a tabbed view.** Drag a view over another view and release the mouse button when the frames meet. Both views overlap and create tabs within the same frame. You can combine multiple views in this way.

Job explorer

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



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

To view jobs in Job Explorer

1. Navigate to Job Explorer, and then 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

- In Job Explorer, double-click a job.

Viewing change history

Each time a job runs, its history is recorded and listed in the Job Explorer. You can see the history of a specific job in the 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.

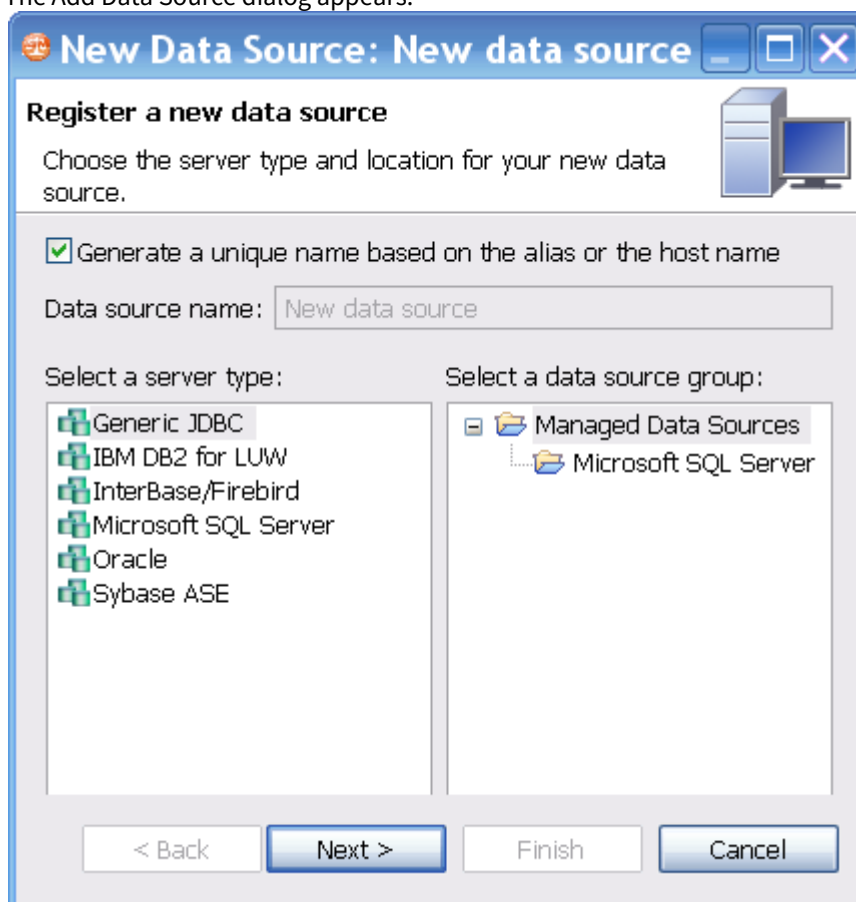
To create a working set

1. Click the menu icon in the Data Source Explorer toolbar.
2. Click **Select Working Set**.
3. In the Select Working Set dialog, click **New** to define a new set.

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 then choose **Deselect Working Set**.

To add a data source

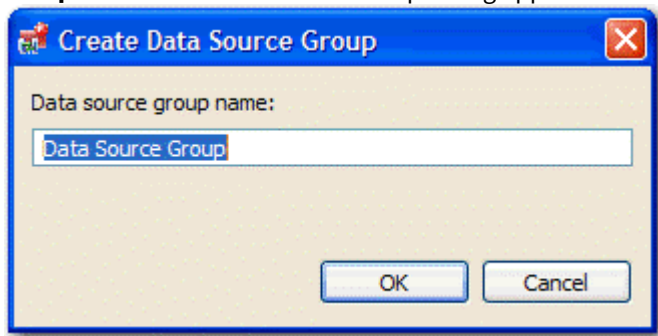
1. Right-click on the Managed Data Sources node in Data Source Explorer, and then select **New > Data Source**. The Add Data Source dialog appears.



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.
3. 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 the **Managed Data Sources** node in Data Source Explorer, and then 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 then 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.

Data Source Properties

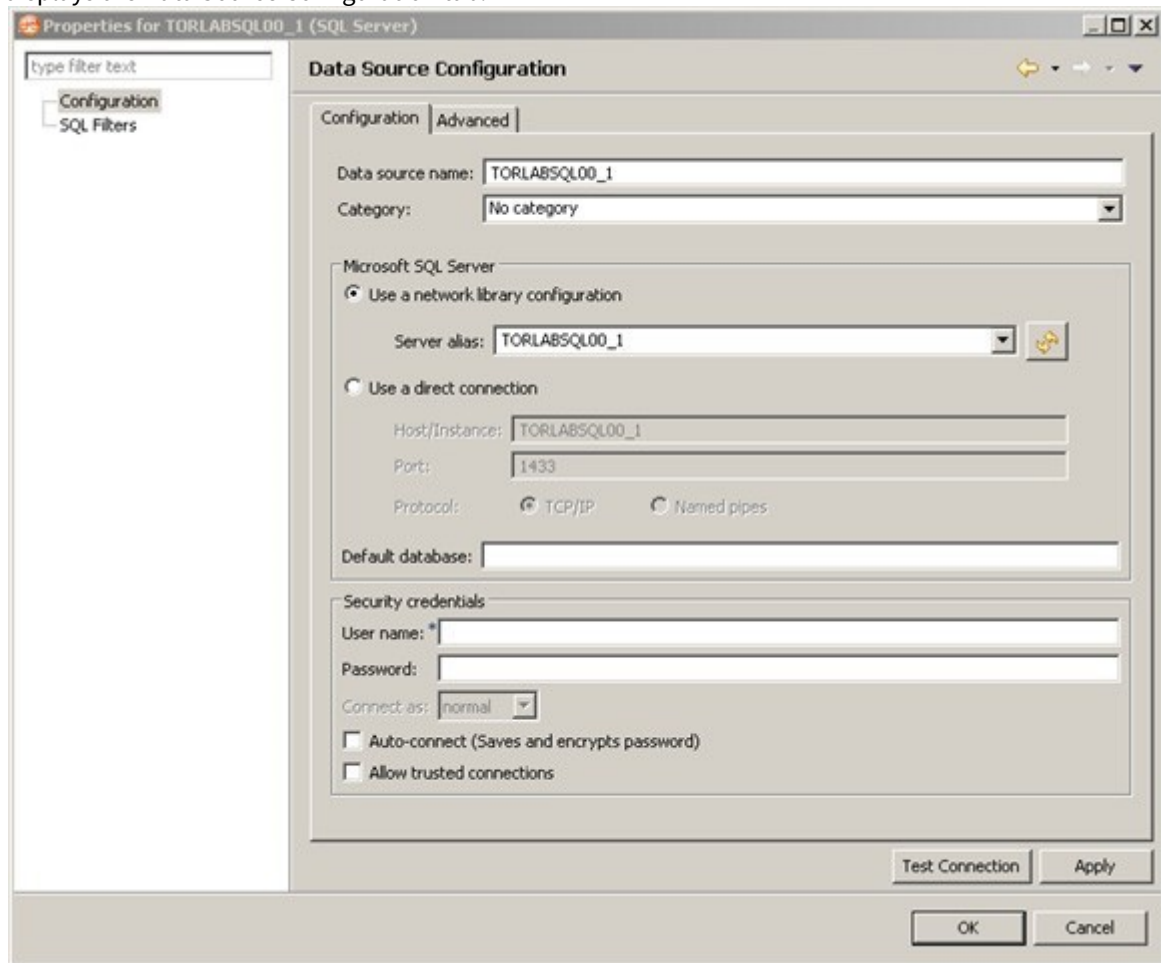
After you import a data source, you can view and change its properties.

Basic Data Source Configuration

The name and connection properties you specified when you set up the data source are available in the Configuration pane.

To change data source properties

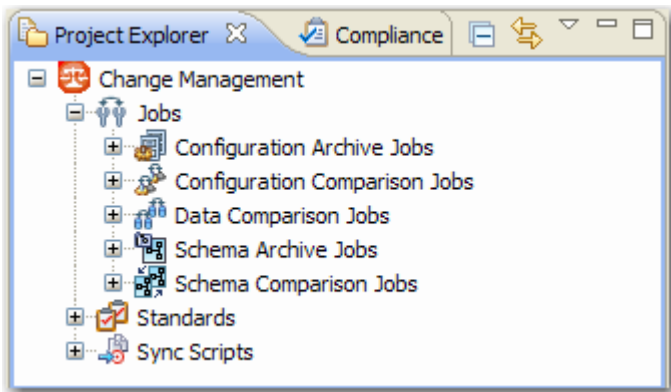
1. In the Data Source Explorer, right-click on the data source, and then select **Properties**. The properties pane displays the Data Source Configuration tab.



2. In the Data Source Configuration pane, you can review and update important details about:
 - The name
 - The type of database and its network connections
 - The default database
 - The user name and password for the connection, and whether it is encrypted and trusted
3. Click **Apply** to save the configuration.
4. Click **Test Connection** to validate the configuration.

Lifecycle Categories for Organizing Data Sources

You can categorize data sources into the phases of the database lifecycle: Development, Test, QA, and Production. Use the Category dropdown menu to assign a category to a data source. The color coding of the category, if set, is displayed in the Data Source Explorer so you can easily differentiate the usage of multiple databases.



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.

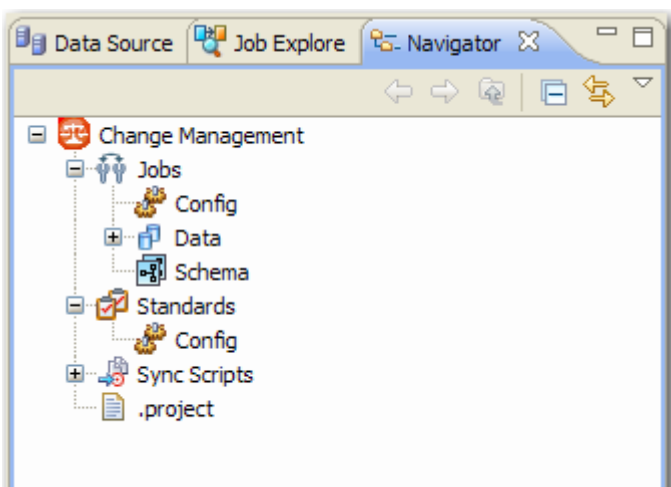
You can use the **File > New > Project** command to create project folders to organize and manage your jobs. When you create a new project, a folder is added to the Project Explorer with the name that you specified. You can add or delete files under this folder by dragging and dropping them from the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

To create a new project

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


Navigator view

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



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

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

 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:

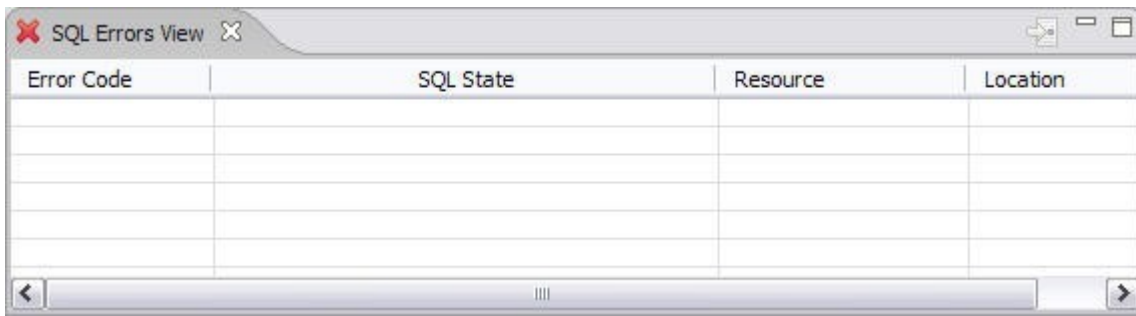
- The day the job was last run
- The day the job last failed, if applicable

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

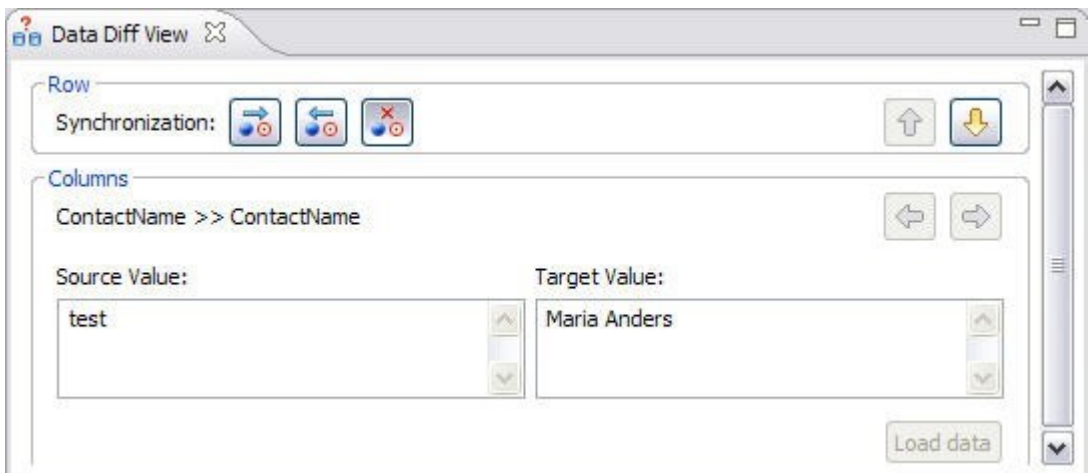
The SQL Errors view logs any errors that may occur when executing SQL scripts.



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

Data Diff view

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



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

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

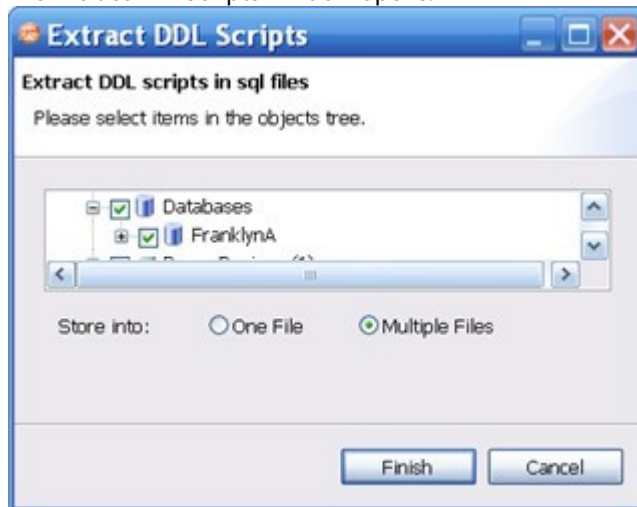
This view lets you specify changes without resolving the repositories of both data sources with synchronization scripts.

Extract DDL Scripts

To extract the DDL of the archive contents to a file

1. Open the Version tab of the archive.

- In the **Archive Contents** list, right-click the object, and then select **Extract DDL Scripts**. The Extract DDL Scripts window opens.



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

SQL Log

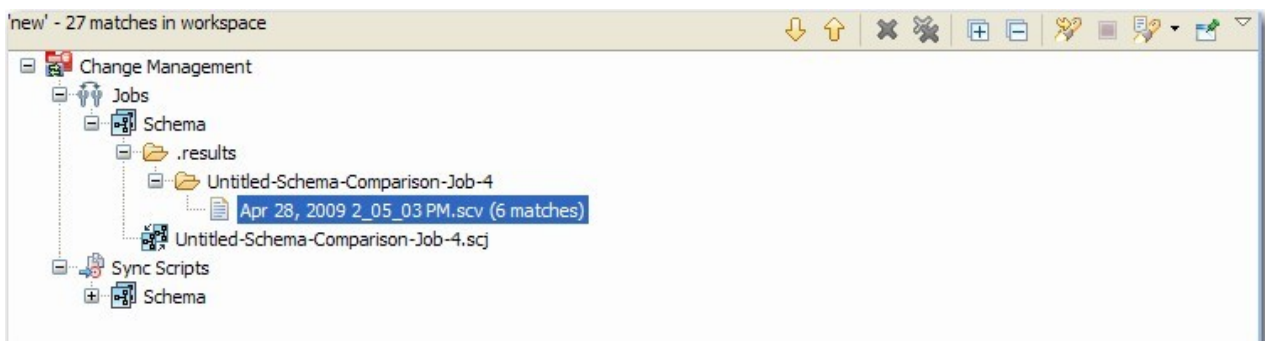
The SQL Log captures all SQL commands executed by SQL Editor and the system. SQL Log entries are listed by SQL **Statement** name, **Date** issued, **Host/Server**, **Service**, **User**, **Source**, and the **Time (in milliseconds)** it took to execute the command.

SQL Errors Log

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

Search view

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



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

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

For more information about 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 generated by DB Change Manager.

To open the Search dialog

- Click the **Search** icon in the Toolbar, or select **Search > Search** from the Main Menu.

The Search dialog contains two tabs:

- Database Search
- File Search

To search for databases

1. Navigate to the Database Search tab.
2. Use the parameter fields provided to search and locate a database:
 - a. 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.
 - b. 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 /).
 - c. Select **Case Sensitive** to take case into account when searching for string matches.
 - d. Select **Search Indexed Data** to include any information that is retained by DB Change Manager's indexing function.
 - e. Select **Apply SQL Filters** to apply any filters you may have associated with the registered data sources in the system.
 - f. Specify whether the search is limited to **Declarations, References, or All Occurrences** in the **Limit Search to** section.
 - g. 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 are returned.
3. Click **Search**.
DB Change Manager returns a list of all possible script selections based on the variables you entered.

To search for synchronization scripts

1. Navigate to the File Search tab.
2. Use the parameter fields provided to search and locate a synchronization script:
 - a. 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 /).
 - b. Select **Case Sensitive** to take case into account when searching for string matches.
 - c. Select **Regular Expression** to identify the string as a regular function in terms of search criteria.
 - d. 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 searches 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** searches for configuration archive job files, exclusively.
- e. Select **Consider Derived Resources** to include derived resources in the search.
3. Click **Search**.
DB Change Manager returns a list of all possible script selections based on the variables you entered.

Editors

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

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

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 fits into the Workbench. Release the button to move the editor to its new location.

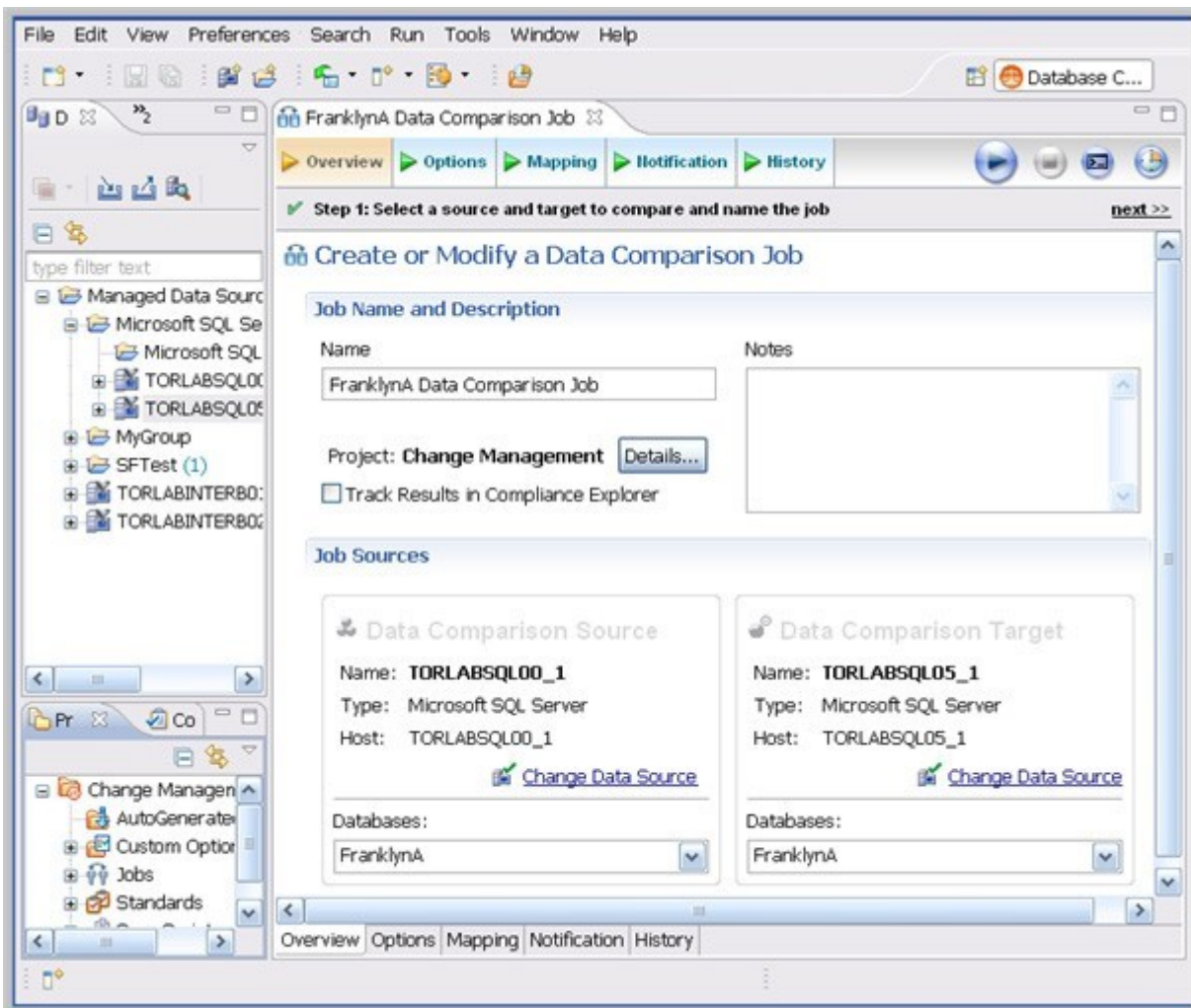
To build a tabbed frame

- Drag the editor over another editor and release the mouse button when the frames meet.

Both editors overlap and create a tabbed frame. Multiple editors can be combined in this format.

Data Comparison Job Editor

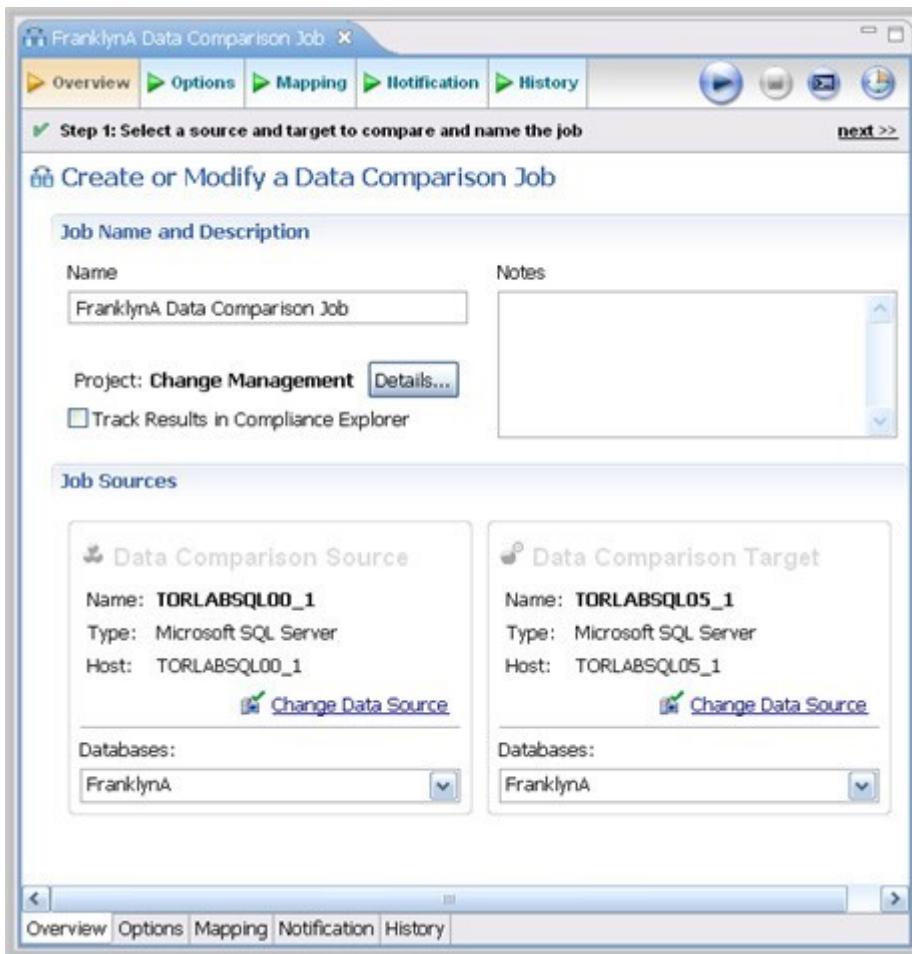
The Data Comparison Job editor compares the data repositories of two specified data sources at the table and row levels. In addition, the editor lets you specify table mappings and data matching functions that synchronize it at the global, table, and row level. It also provides high-level mapping refinement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



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

Configuration Comparison Job Editor

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

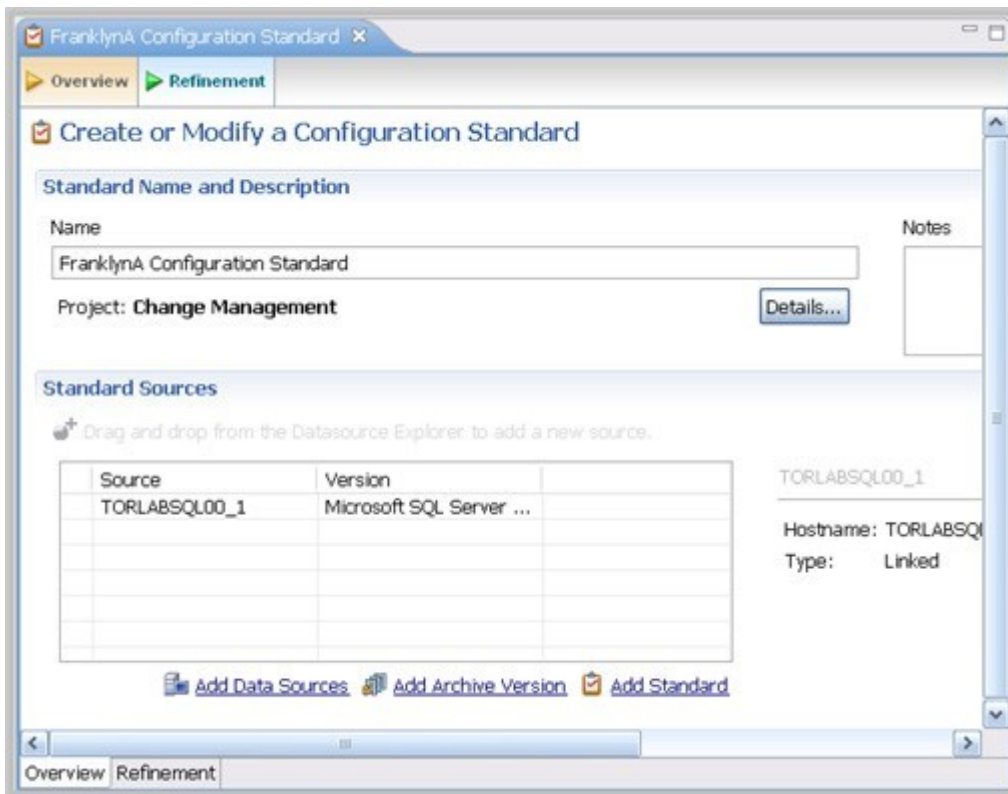


For more information on comparing the configuration settings of two specified data sources, see [Comparing database configuration settings](#).

Configuration Standard Editor

The Configuration Standard editor defines a configuration standard. A configuration standard is a set of customized data source configuration properties. Once defined, standards are used in configuration comparison jobs against data sources and archives to compare against a specific set of values and threshold operators. You use standards in configuration comparison jobs to ensure that configuration values fall within acceptable ranges.

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

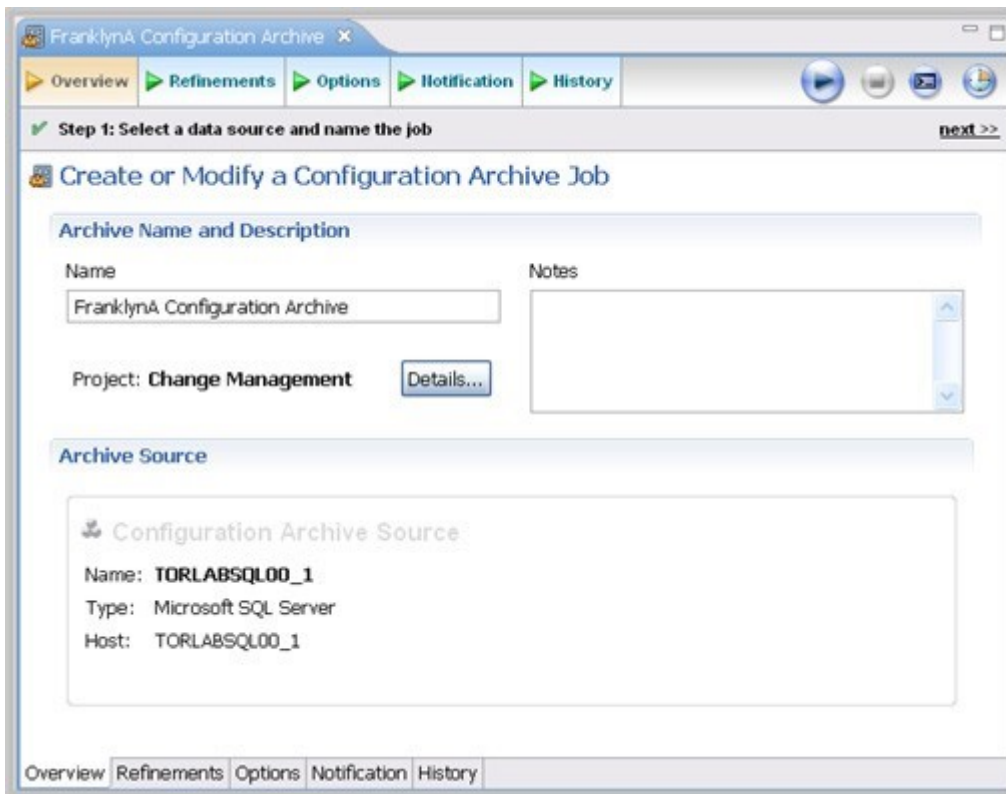


For more information on creating configuration standards, see [Creating a standard for configuration settings](#).

Configuration Archive Job Editor

The Configuration Archive Job editor provides the ability to capture a snapshot of existing data source configuration values. You can version the same archive multiple times, so you can store and use archive configuration values from different points in time.

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



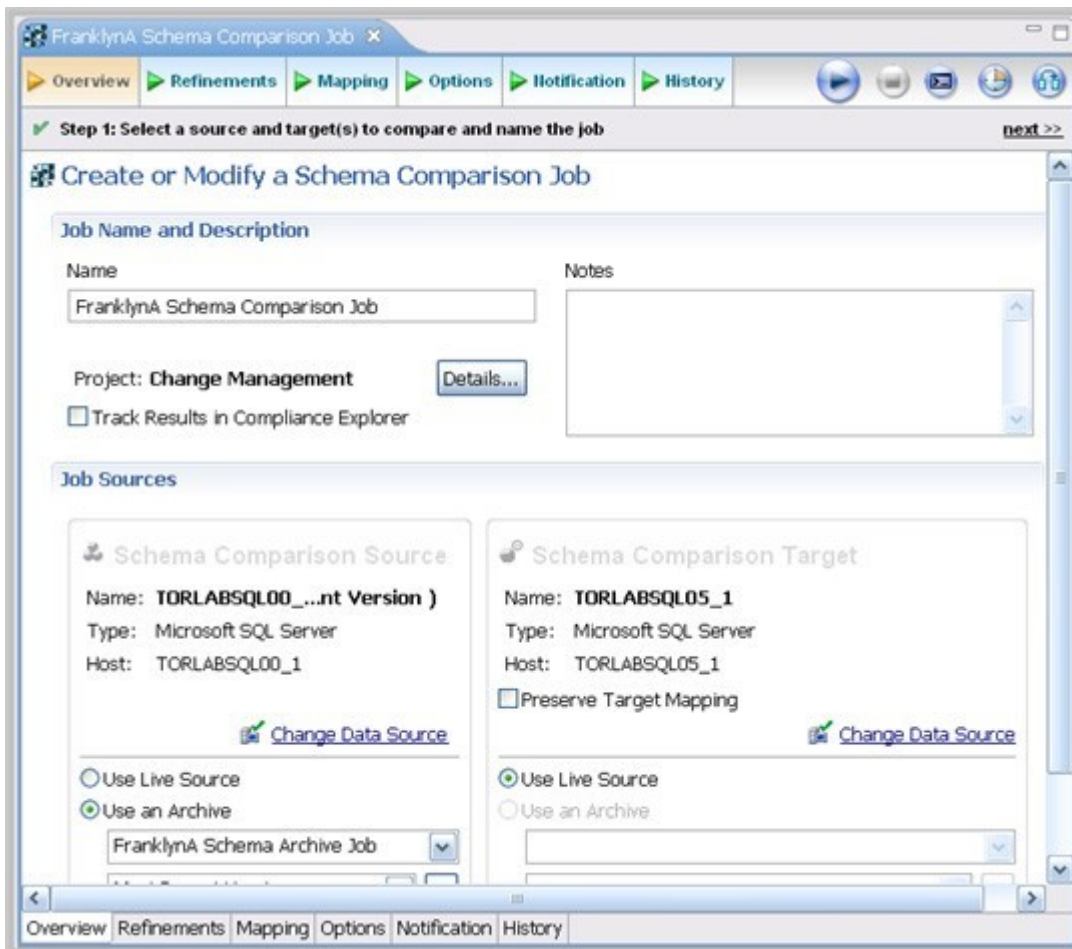
For more information on creating configuration archives for data sources, see [Creating a configuration archive](#).

Schema Comparison Job Editor

The Schema Comparison Job editor compares two schemas on registered data sources, to identify differences between them. You can compare the schemas of two live data sources, a live data source and an archive, or two archives. For example, in order to ensure that approved changes are rolled forward between testing and production environments, you could first archive a baseline schema and use it to compare with the live environment once the move is completed.

DB Change Manager can support cross-platform schema comparisons, but such jobs are limited to the following DBMS combinations and data source objects:

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

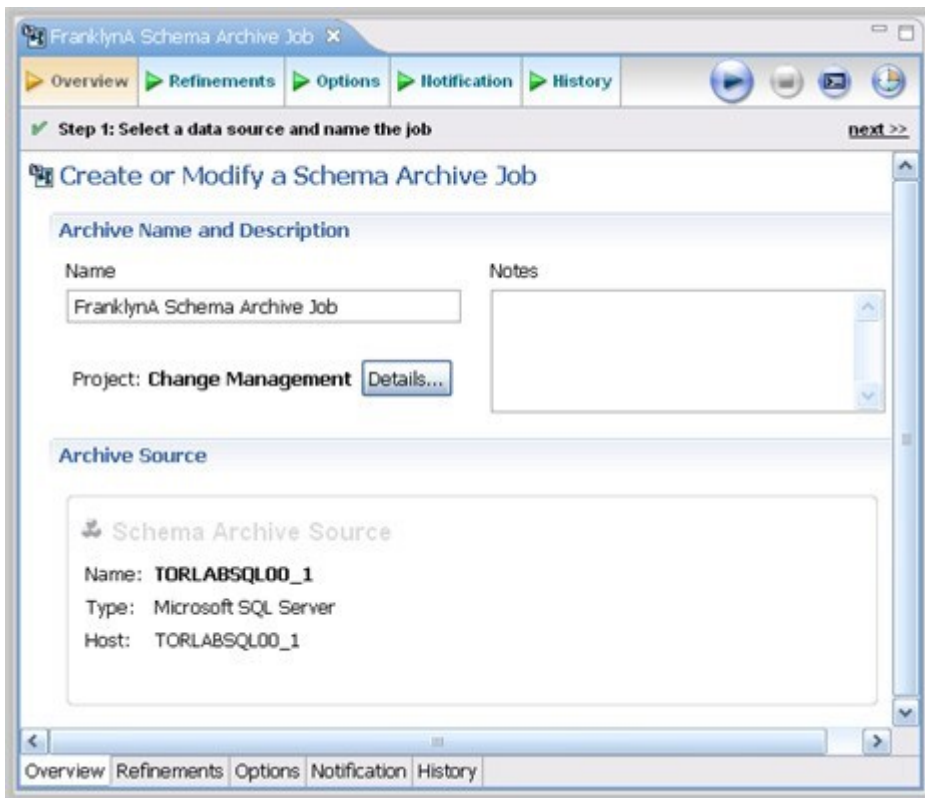


For more information on comparing the schemas of two data sources, see [Comparing schemas](#).

Schema Archive Job Editor

The Schema Archive Job editor captures a "snapshot" of an existing schema archive. The interface provides the ability to version the same archive multiple times, so you can store and use 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, see [Creating a schema archive](#).

SQL Editor

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

```

*Untitled-Config-Comparison-Job-1  Sync.TORLABSQL05_1 Apr 29, 2010 4_16_47 PM.sql x
Microsoft SQL Server  TORLABSQL05_1 (9.0.1399.0)  master
-----
-- Embarcadero Change Manager Synchronization Script
-- FILE      : Sync.TORLABSQL05_1 Apr 29, 2010 4_16_47 PM.sql
-- DATE     : Apr 29, 2010 4:16:47 PM
--
-- SOURCE DATA SOURCE : TORLABSQL00_1
-- TARGET DATA SOURCE : TORLABSQL05_1
-----
-- Property 'cost threshold for parallelism' from source 'Core SQL Properties'.
EXEC sp_configure 'cost threshold for parallelism',5
go
RECONFIGURE
go

-- Property 'max server memory (MB)' from source 'Core SQL Properties'.
EXEC sp_configure 'max server memory (mb)',149
go
RECONFIGURE
go

-- Property 'max worker threads' from source 'Core SQL Properties'.
EXEC sp_configure 'max worker threads',255

```

When DB Change Manager initially generates a script, it appears in SQL Editor. By default, new scripts are automatically saved and appear in the Navigator and Project Explorer in a default directory specified by DB Change Manager. You can edit the SQL scripts and save your changes.

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

To run SQL code

1. Open the SQL file you want to run, ensure it is associated with the correct database
2. Click the **Execute** icon.
DB Change Manager executes the code on the data source you specified. Results are displayed in the same or a new tab.

To execute a transaction

1. Open the transaction file you want to run and ensure it is associated with the correct database.
2. Click the **Start Transaction** icon.
DB Change Manager executes the transaction on the data source you specified.

To commit a transaction

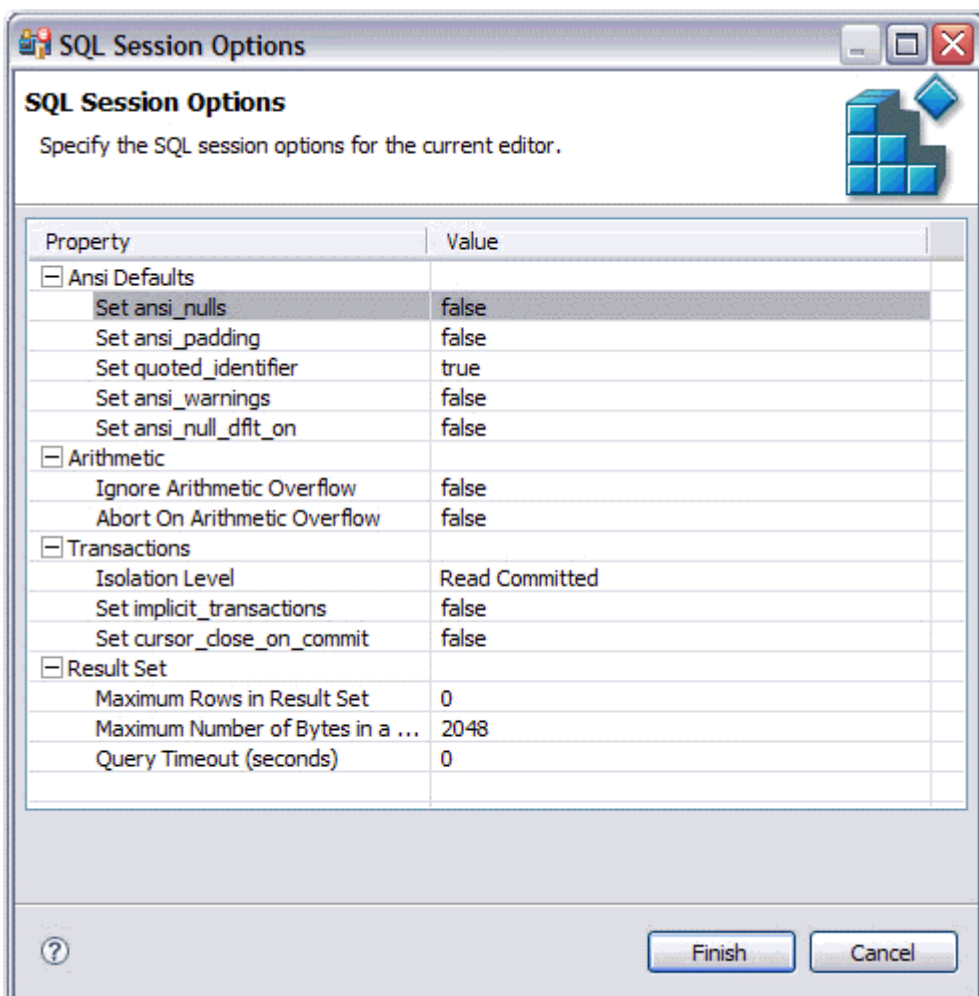
1. Open the transaction file you want to commit and ensure it is associated with the correct database.
2. Click the **Commit Transaction** icon.
DB Change Manager commits the transaction on the data source you specified.

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

- i** 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
 - statistics

SQL Session Options

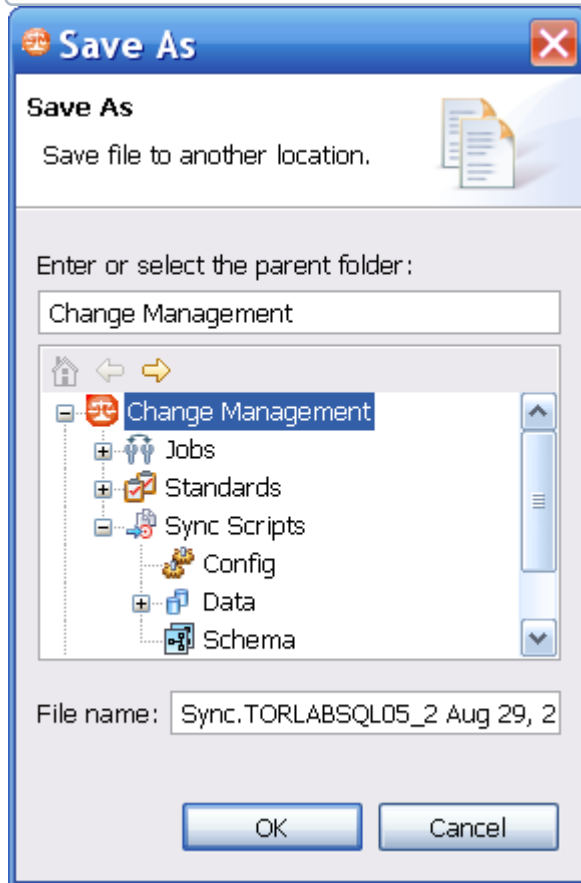
The SQL Session Options dialog provides configuration parameters for running code in the development environment.



To modify SQL session options

1. On the Toolbar, click the **SQL Session Options** icon.
2. In the SQL Session Options dialog, click on individual parameters in the **Value** column to change the configuration of each property, as specified.
3. Click **Finish**.
The session options change and DB Change Manager executes the code as specified when you launch the script.

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



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

Eclipse plugin extensions reference

DB Change Manager publishes extensions that can be implemented as plugins to customize the application.

The **Notifications extension** helps you code a plugin for DB Change Manager that implements any type of job execution notification required for your organization. DB Change Manager includes email notification by default, and you can code additional Java plugins to add to DB Change Manager.

The **Configuration Parameters extension** helps you code a plugin for DB Change Manager that adds additional configuration parameters and values for configuration standards, archive jobs, and comparisons. DB Change Manager includes plugins for standard configuration properties and version information, and you can code additional plugins that query data sources using Java JDBC for additional or environment-specific information.

To write Eclipse plugins, you need to have some experience coding in Java. For instructions, see:

- [Creating and implementing a notification plugin](#)
- [Creating and implementing a custom configuration plugin](#)

Creating and implementing a notification plugin

Notifications can be set to alert you if a job is successful, unsuccessful, or either result. Additionally, a notification can also indicate if a job was launched from the Workbench or from a command line, and can produce a report in a number of different file formats.

You can select the following options and option parameters using the job notification plugin:

Notification Option	Option Parameters
Notify When Run From	<ul style="list-style-type: none"> • Workbench • Command Line • Both
Attached Report	<ul style="list-style-type: none"> • None • CSV • XML • HTML • PDF • RTF
Send Notification	<ul style="list-style-type: none"> • Always • Only on Success • Only on Failure

Notification parameters are set independently on each job configured in DB Change Manager. Additionally, each set of notification options apply individually per job selected.

The notification plugin must be defined in Eclipse and then activated in DB Change Manager. The following tasks provide a high-level overview of defining and implementing the notification plugin:

- [Create a plugin project](#)
- [Configure the target platform](#)
- [Define dependencies](#)

- [Implement a notifier class](#)
- [Implement a viewer contribution](#)
- [Specify setup constants](#)
- [Implement the DirectoryNotifier class](#)
- [Implement the DirectoryNotifierContribution class](#)
- [Register the extensions](#)
- [Deploy the Notifier plugin](#)

Create a plugin project

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

To create a plugin project

1. Select **File > New > Other**.
The Select a Wizard dialog appears.
2. Select **Plugin Project**, and then click **Next**.
3. Type a name for the plugin in the appropriate field, and leave the remainder of the parameters as they appear.
4. Click **Next**.
5. Retain the default parameters on the next dialog, and then click **Finish**.
The new plugin project is created in Eclipse and is ready for plugin development.

Configure the target platform

The target platform for the plugin development process needs to be identified in DB Change Manager. This indicates to the new plugin for what product it extends, and grants the plugin access to the system.

To configure the target platform

1. Select **Window > Preferences**.
The Preferences dialog appears.
2. Choose the **Plugin Development > Target Platform** node, and then click **Browse**.
3. Navigate to the install directory for DB Change Manager.
4. Click **Apply**, and then click **OK**.
The Preferences dialog closes and the target platform of the plugin is now indicated.

Define dependencies

The new plugin requires a pair of dependency definitions on the DB Change Manager Notification plugin:

- `com.idera.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 then click **Add**.
3. Choose `com.idera.change.notifications`, and then click **OK**.
The dependency is added.
4. Choose `org.eclipse.ui.forms`, and then 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 then close the editor.

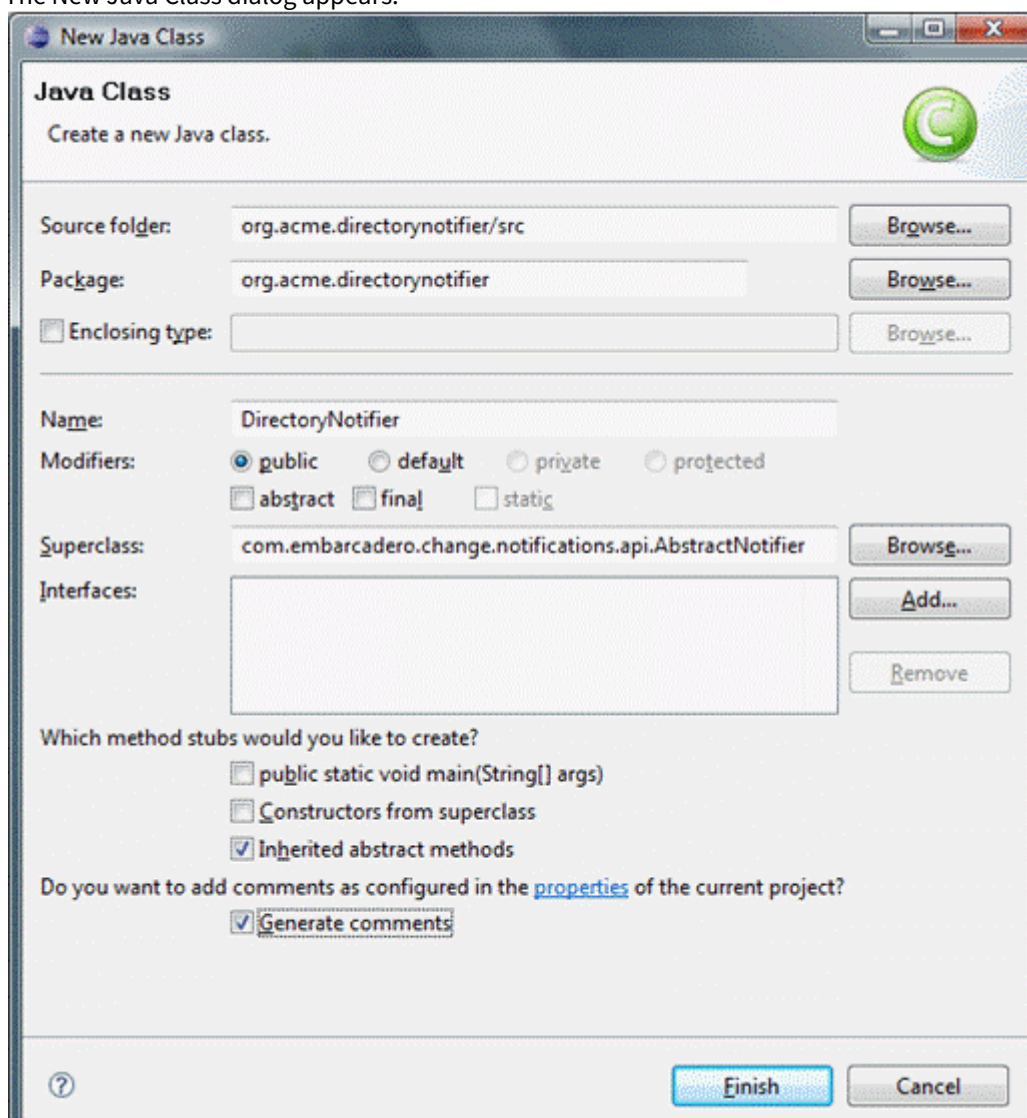
Implement a notifier class

Each notification plugin requires a `Notifier` class that must implement the interface `com.idera.change.notifications.api.INotifier`. To do this, you subclass `com.idera.change.notifications.api.AbstractNotifier` and customize it.

i If `AbstractNotifier` or `INotifier` cannot be found in Eclipse, the target platform or plugin dependencies are not configured properly. Ensure that you have configured these prerequisites and then search again for the notifier classes.

To implement a notifier class

1. Right-click on your project, and then select **New > Class**.
The New Java Class dialog appears.



2. Enter the appropriate information in the fields provided to define the new Java class:
 - a. In the **Name** field, type "DirectoryNotifier".
 - b. In the **Package** field, type "org.acme.directorynotifier".
 - c. In the **Superclass** field, type "com.idera.change.notifications.api.AbstractNotifier".

- 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 returns true, because 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 is shown for this type of notification.
- `sendNotification` is called after a job runs and is responsible for the notification. The three parameters are as follows:

Parameter	Type	Description
<code>notifierData</code>	<code>iNotifierData</code>	This instance contains the configuration for the notification of the job just run.
<code>jobMetaData</code>	<code>Map<String, String></code>	<p>This map contains information about the job execution. The keys in the map correspond to entries in the <code>NotificationPropertyEnum</code>.</p> <p>For example, to get the data and time of the execution, you would code:</p> <pre>"jobMetaData.get(NotificationPropertyEnum.DATE_TIME.getTag())"</pre> <p>NOTE: The keys are the tag of the enum entries, so ensure that <code>getTag()</code> is called when accessing <code>jobMetaData</code>.</p>
<code>notificationInfo</code>	<code>iReportGenerator</code>	Used to generate the report and access any DDL or sync script output.

Implement a viewer contribution

Each notifier can be configured for a job in DB Change Manager.

The notifier enables a user to select a target directory in which to put results. The email notifier that is supplied with the application enables users to select the email template that will be used to generate the message as well as the addresses of those who will receive it.

In order to add a directory field to the job editor, you write a class that implements the interface named `com.idera.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.

Specify 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:

```
/**
 * Constantsforthisnetworksharenotifier
 */
publicfinalclassDirectoryNotifierConstants
{
    //Overviewfilename publicstaticfinalString
OVERVIEW_FILE_NAME="outcome.txt";
    //PersistedJobSettingsKey
publicstaticfinalStringTARGET_DIRECTORY="directorynotifier.targetdir";
    /**
 * Constantutilityclasscannotbeinstantiated
 */
privateDirectoryNotifierConstants()
{
}
}
```

Implement 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.TA
RGET_DIRECTORY);
```

2. Enter the following code to define that the report and output file need to be placed in a directory whose name is composed of the job execution date and time.

 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.separatorChar +
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.seperatorChar +
        reportFile.getName()); eportFile.renameTo (destinationReportFile);
    }

```

5. 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);

```

Implement the DirectoryNotifierContribution class

Add the code below to the DirectoryNotifierViewerContribution class to provide functionality for the interface:

1. The DirectoryNotifierViewerContribution class requires three class-level private attributes. These attributes store notification data, which is the composite on which any options are placed, as well as the folderField, which is used to enter a file directory address.
2. The createFormContent attribute creates a new composite and provides a layout template for options and entry fields. The code below adds a label and a text field to the interface:

```

private INotifierData notificationData;
private Composite body;
private Text folderField;

public Composite createFormContent (Composite parent, FormToolkit toolkit)
body = toolkit.createComposite (parent);
body.setLayout (new GridLayout ());

```

```

        toolkit.createLabel (parent, "Target Directory:");

```

```

String folder = notificationData == null ? " " " " :

```

```
notificationData.getNotificationProperties().get
(DirectoryNotifierConstants.TARGET_DIRECTORY);

        folderField = toolkit.createText (parent, folder, SWT.SINGLE |
SWT.BORDER);
        folderField.setLayoutData (new GridData (SWT.FILL, SWT.CENTER, true,
false));
        folderField.addModifyListener (new ModifyListener () {
            public void modifyText (ModifyEvent e)
            {
                notificationData.setNotificationPropert
(DirectoryNotifierConstants.TARGET_DIRECTORY, folderField.getText().trim() );
            }
        });
        return body;
```

The code does the following:

- Creates a new composite body, and sets it to have a grid layout.
 - Creates a label for the target directory field.
 - Retrieves the folder from the job, if the job is new or unsaved, or defaults to an empty string if the result is null.
 - Initializes the `folderField` text element and sets it to justify.
 - Adds a listener for changes on the `folderField` element that updates the `notificationData`, thus enabling save actions.
 - Returns the body to be added to the DB Change Manager job editor to which this class applies.
3. Add the `setData` method using the following code, which is called to update the user interface with job information. This method accepts and stores the new instance of `INotifierData` containing job-specific data for the notifier. If the `folderField` has already had something entered, the notification properties are 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());
    }
}
```

Register the extensions

Once you have coded the plugin and implemented the notifier and user interface, you include definitions that explain how DB Change Manager should interact with the plugin.

You register the extensions that define the `Notifier` and `NotifierViewerContribution` to DB Change Manager. When the plugin is added to DB Change Manager, the extensions are registered and DB Change Manager understands the purpose of the plugin.

1. Open the `MANIFEST.MF` file.
2. Navigate to the `plugin.xml` tab and display the source.
3. Add the following code:

```
<extension point="com.idera.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 by clicking **File > Save**.

When `org.acme.directorynotifier` is found, the notification extension point is registered. The `notifier` and `viewContribution` attributes are the class names for two implementations and Eclipse reflectively registers these objects for use in DB Change Manager.

Deploy the Notifier plugin

Once the plugin has been defined, you need to deploy the plugin in DB Change Manager.

1. Stop DB Change Manager if it is currently running.
2. Select **File > Export**.
The Export dialog appears.
3. Choose **Deployable Plugins and Fragments** from the **Plugin Development** group.
4. Click **Next**.
5. Select `org.acme.directorynotifier` and ensure that the target directory is attributing to the DB Change Manager installation folder.
6. Click **Finish**.
7. Start DB Change Manager, and then create a new data comparison job.
8. Define a source and target data source, and then navigate to the Notifications tab.
The Directory Notifier appears as a section to the tab.
9. Choose **Enable Directory Notifier** to enable the feature.
10. Specify the **Target Directory**, such as **C:\ChangeManagerNotifications**, and then choose **PDF** in the **Attached Report** section.
11. Execute the job, and then open the target folder. A new folder is created containing two output files from the job.

Creating and implementing a custom configuration plugin

Follow the instructions in these sections to code an Eclipse plugin for DB Change Manager that adds additional configuration parameters and values for configuration standards, archive jobs, and comparisons.

- [Create a project plugin](#)
- [Configure the target platform](#)
- [Define dependencies](#)
- [Implement a property source](#)
- [Determine support](#)
- [Define properties](#)
- [Gather results](#)
- [Synchronize the job](#)
- [Register extensions](#)
- [Deploy the custom configuration plugin](#)

Create a project plugin

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

To create a plugin project

1. Select **File > New > Other**.
The Select a Wizard dialog appears.
2. Select **Plugin Project**, and then click **Next**.
3. Type a name for the plugin in the appropriate field, and leave the remainder of the parameters as they appear.
4. Click **Next**.
5. Leave the parameters on the next screen as they appear, and then click **Finish**.
The new plugin project is created in Eclipse and is ready for plugin development.

Configure the target platform

The target platform for the plugin development process needs to be identified in DB Change Manager. The target platform indicates to the new plugin for what product it extends, and grants the plugin access to the system.

1. Select **Window > Preferences**.
The Preferences dialog appears.
2. Choose the plugin **Development > Target Platform** node, and then click **Browse**.
3. Navigate to the install directory for DB Change Manager, and then click **Apply**.
4. Click **OK**.
The Preferences dialog closes and the target platform of the plugin is now indicated.

Define dependencies

The plugin requires a dependency for configuration comparison jobs in DB Change Manager. This dependency can be set by opening the MANIFEST.MF file located in the META-INF directory of your project.

1. Select the Dependencies tab, and then click **Add**.
2. Choose `com.idera.change.config.propertysource`, and then click **OK**.
The dependency is added to the plugin and appears in the **Dependencies** list.

3. Press *CTRL-S* or click **File>Save**.

Once the dependency definition is saved, you can close the editor and the system retains your selection.

Implement a property source

Each configuration property source plugin must contain a class that extends `com.idera.change.config.propertysource.api.ConfigurationPropertySource`.

i If `ConfigurationPropertySource` cannot be found in Eclipse, it means you have misconfigured the target platform or target dependencies.

1. Right-click the project, and then select **New>Class**.
The New Java Class dialog appears.
2. Name the class `JDBCVersionInformation`, set the package to `org.acme.versioninformation`, and then set the superclass to `com.idera.change.config.propertysource.api.ConfigurationPropertySource`.
3. Click **Finish**.
The new class is generated.

Determine support

Each class that subclasses `ConfigurationPropertySource` contributes properties to the configuration comparison job functionality.

The implementation uses the `isSupported` method, which determines if there are attributes available for a specific data source.

For example, if a user added the Sybase `@@Version` tag as a configuration property, it would be pointless to run the code for Oracle, SQL Server, or DB2 for LUW data sources.

The method has the following signature:

```
boolean isSupported (IDataSourceIdentifier dataSourceIdentifier)
```

This method is called by DB Change Manager for any registered content plugin when a data source is used in a configuration archive job, a standard, or a comparison job. When it encounters a data source identifier, it returns true or false and indicates whether the property source provides parameters for the data source.

Implement the following code to define this method:

```
public boolean isSupported (IDataSourceIdentifier dataSourceIdentifier)
{
    return dataSourceIdentifier.getDatabaseVersion().getDBMSType() !=DBMSType.DB2;
}
```

Define properties

The `getAvailableProperties` function returns a set of instances of `IConfigurationProperty` instances.

There are several utility methods to help instantiate these instances, named of the form, `createConfigurationProperty`.

Implement the following code to define the `getAvailableProperties` method:

```
public Set(IConfigurationProperty> getAvailableProperties (IDataSourceIdentifier
identifier)
{
```

```

Set<IConfigurationProperty> results = new HashSet<IConfigurationProperty>();
for (EVersionProperties prop : EVersionProperties.values())
{
    results.add (createCOnfigurationProperty (prop.getDisplayName(),
prop.name(),"", prop.getType(), "" ));
}
return results;
}

```

Because the EVersionProperties code cannot be resolved, you can add a Java enum to define the properties provided by the plugin.

Create a new file named EVersionProperties.java and implement the following code:

```

package org.acme.versioninformation;
import java.sql.Connection;
import java.sql.SQLException;
import com.idera.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 ther eis a JDBC error
         */
        public String getValue (Connection connection) throws SQLException
        {
            return connection.getMetaData().getDatabaseProductName();
        }
    },
    PRODUCT_VERSION (ConfigurationPropertyType.STRING, "Product Version")
    {
        /**
         * @returns the JDBC metadata's information for product version
         * @throws SQLException if there is a JDBC error
         */
        public String getVAlue (Connection connection) throws SQLException
        {
            return connection.getMetaData().getDAtabaseProductVersion();
        }
    },
    MAJOR_VERSION (ConfigurationPropertyType.NUMERIC, "Major Version")
    {
        /**
         * @returns the JDBC metadata's information for major version
         * @throws SQLException if there is a JDBC error
         */
        public String getValue (Connection connection) throws SQLException
        {
            return "" + connection.getMetaData().getDataMajorVersion();
        }
    }
}

```

```

    };
//Attributes private ConfigurationPropertyType type; private String name;
/**
 * Constructor to set entry attributes
 * @param type the type
 * @param name
 */
private EVersionProperties (ConfigurationPropertyType type, String name)
{
/**
    this.type = type;
    this.name = name;
 * Given a JDBC connection return the parameter's value
 * @param connection the database connection
 * @throws SQLException if there is a JDBC error
 */
abstract String getValue (Connection connection) throws SQLException;
/**
 * @return the display name of the parameter
 */
public String getDisplayName()
{
return name;
}
/**
 * @return the type of the parameter
 */
public ConfigurationPropertyType getType()
{
return type;

}
/**
 * Finds a property of a name and returns it
 * @param name the name of the property
 * @return the property
 */
public static EVersionProperties getProperty (String name)
{
    for (EVersionProperties p : values())
    {
        if (p.name().equals (name))
        {
            return p;
        }
    }
    throw new RuntimeException (EVersionProperties.class.getName() + "not found for
" + name + "");
}
}

```

In the code above, the four entries that capture from JDBC the product name, version, minor version, and major version are added to the enum.

Gather results

DB Change Manager gathers results when it collects all of the properties and values for display purposes or to store in an archive. The extraction method is called when a data source is indicated in a standard, or when an archive or comparison job is run.

```
public Set<IConfigurationPropertyResult>
extract(Set<IConfigurationProperty> properties, IDatasourceIdentifier
datasourceIdentifier)
```

A set of `IConfigurationPropertyResult` instances that correspond to the properties must be returned, and values need to be added for the properties returned from the data source, as follows:

```
public Set<IConfigurationPropertyResult> extract
(Set<IConfigurationProperty> properties, IDatasourceIdentifier datasourceIdentifier)
{
    try
    {
        Set<IConfigurationPropertyResult> results =
new HashSet<IConfigurationPropertyResult> ();
        Connection connection = getConnection (datasourceIdentifier);
        for (IConfigurationProperty configProp : properties)
        {
            EVersionProperties prop = EVersionProperties.getProperty
(configProp.getPropertyIdentifier());
            String value = prop.getValue (connection);
            results.add (createConfigurationPropertyResult (configProp, true,
value)
                );
        }
        return results;
    }
    catch (SQLException sql)
    {
        throw new RuntimeException (sql);
    }
}
```

In the above implementation, the code creates an empty set to hold the results and then retrieves the JDBC connection to the data source. For each retrieval, the enum constant `EVersionProperties` on which `getValue` is called to retrieve each property value from JDBC metadata. Finally, `createConfigurationPropertyResult` is called to create the result instance.

Synchronize 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>();
    }


```

Register extensions

Once you have completed coding the plugin, register it with DB Change Manager by implementing an Eclipse extension. When the plugin is added to DB Change Manager, the extensions are registered and DB Change Manager understands that the plugin adds a new type of notification to the module.

To register extensions

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.idera.change.config.propertysource.contribution`.
4. Click **Finish**, and then 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 then enter the following details:
 - **display-name:** JDBC Metadata version information
 - **source-identifier:** `org.acme.jdbc.versioninformation.source`
 - **class:** `org.acme.jdbcversioninformation.JDBCVersionInformation`

 The class name is used to enable DB Change Manager to instantiate the new class, while the display-name is the group under which the four JDBC metadata version properties appear.

6. Save the file by pressing `CTRL-S` or by choosing **File>Save** from the menu.
The class notification is complete, and DB Change Manager and Eclipse are informed about the property source. When the plugin is added to DB Change Manager, it will add your properties to configuration comparison and standard jobs.

Deploy the custom configuration plugin

To deploy the custom plugin

1. Shut down DB Change Manager if it is running.
2. Click **File>Export**.
The Export dialog appears.
3. Select **Deployable Plugins and Fragments** from the **Plugin Development** group, and click then **Next**.
4. Select `org.acme.jdbcversioninformation` and ensure the target directory is the same as the DB Change Manager installation folder.
5. Click **Finish**.
The new plugin is deployed.
6. Start DB Change Manager and create a new configuration archive job. When you select an Oracle, Sybase, or SQL Server data source, the Refinements and the Results tabs contain the four version properties you defined in the new plugin.