

# 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.

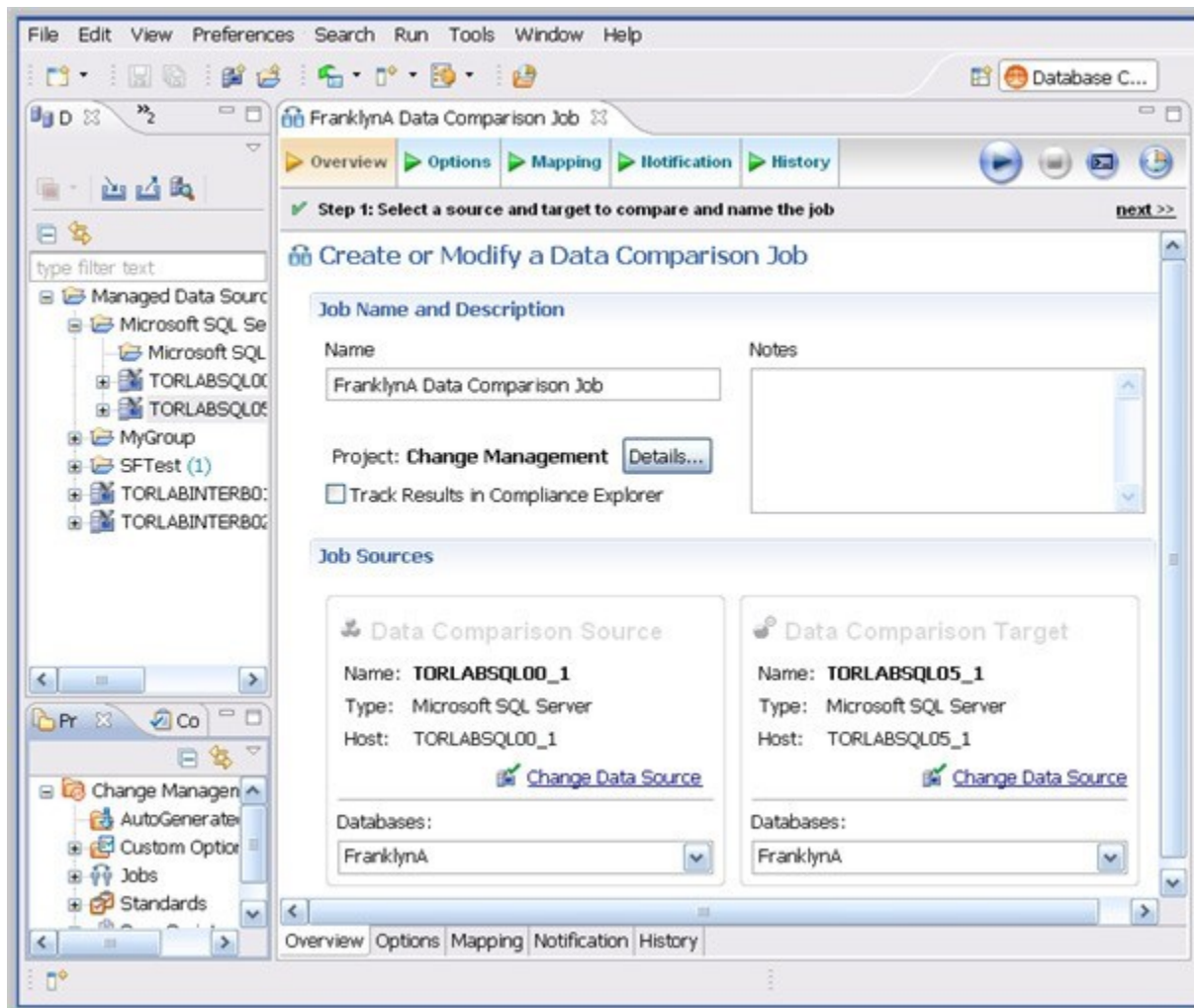
## To build a tabbed frame

- 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

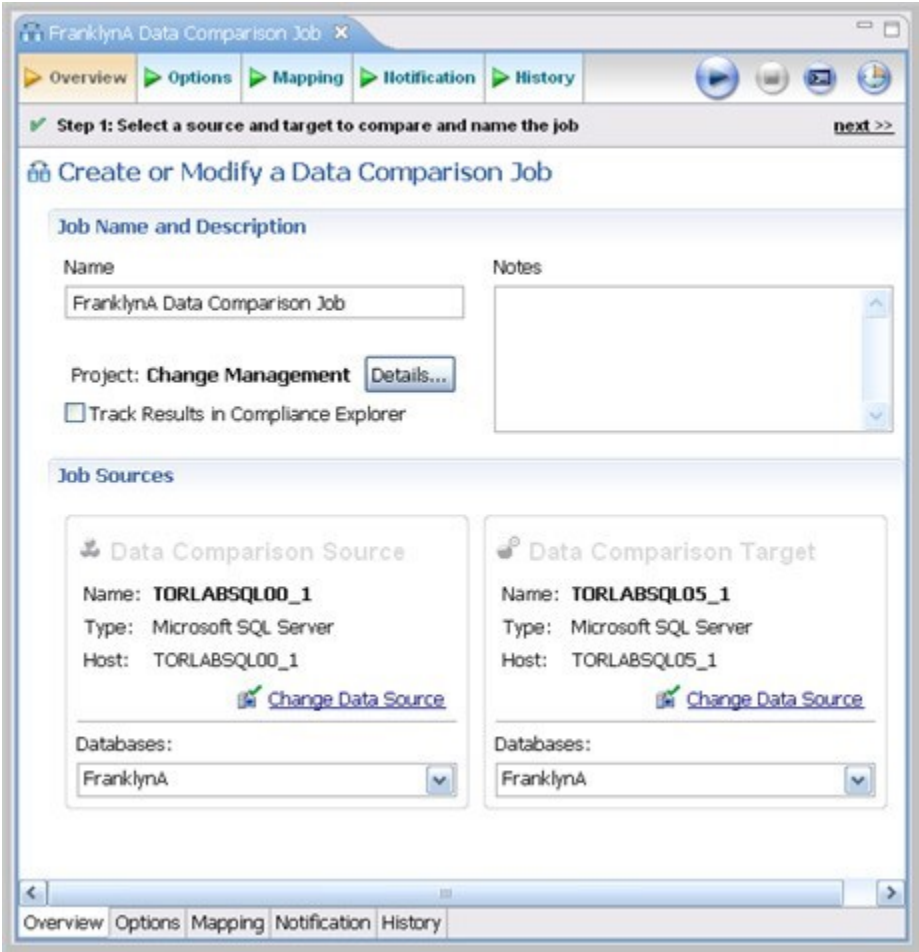
The Data Comparison Job editor compares the data repositories of two specified data sources at the table and row levels. In addition, the editor lets you specify table mappings and data matching functions that synchronize it at the global, table, and row level. It also provides high-level mapping refinement 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. Additionally, the editor provides the ability to define standard settings templates and archive individual data source properties to be used in comparison jobs.

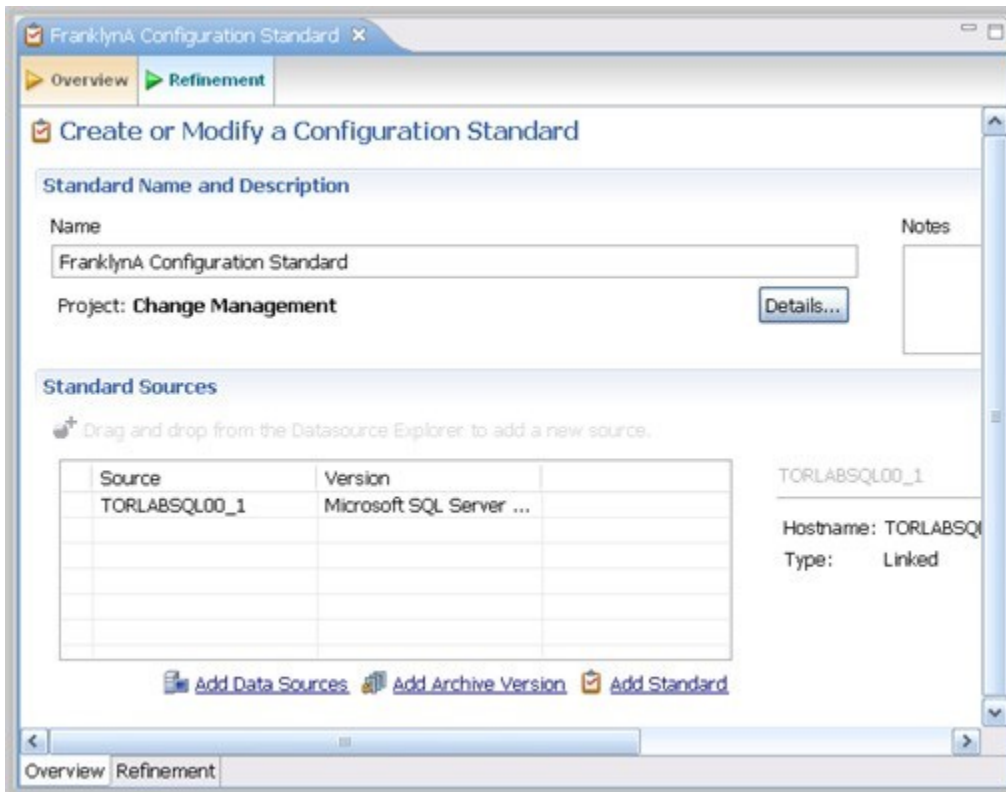


For more information on comparing the configuration settings of two specified data sources, see [Comparing database configuration settings](#).

## Configuration Standard Editor

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

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



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

## Configuration Archive Job Editor

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

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

FranklynA Configuration Archive

Overview Refinements Options Notification History

Step 1: Select a data source and name the job [next >>](#)

### Create or Modify a Configuration Archive Job

**Archive Name and Description**

Name: FranklynA Configuration Archive

Notes

Project: Change Management [Details...](#)

**Archive Source**

Configuration Archive Source

Name: TORLABSQL00\_1

Type: Microsoft SQL Server

Host: TORLABSQL00\_1

Overview Refinements Options Notification History

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:

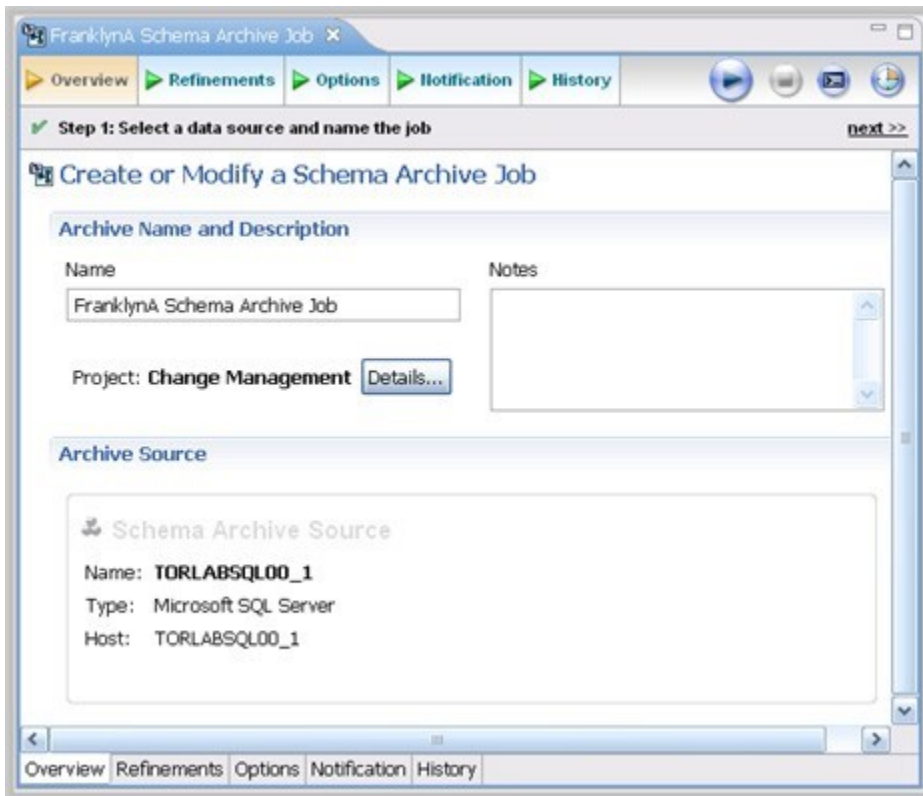
DBMS Source/Target	Objects Compared
Sybase / SQL Server	Tables, Unique Keys, Check Constraints, Foreign Keys, Indexes, Primary Keys, Views
Oracle / SQL Server	Tables, Unique Keys, Check Constraints, Foreign Keys, Indexes, Primary Keys, Views

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.

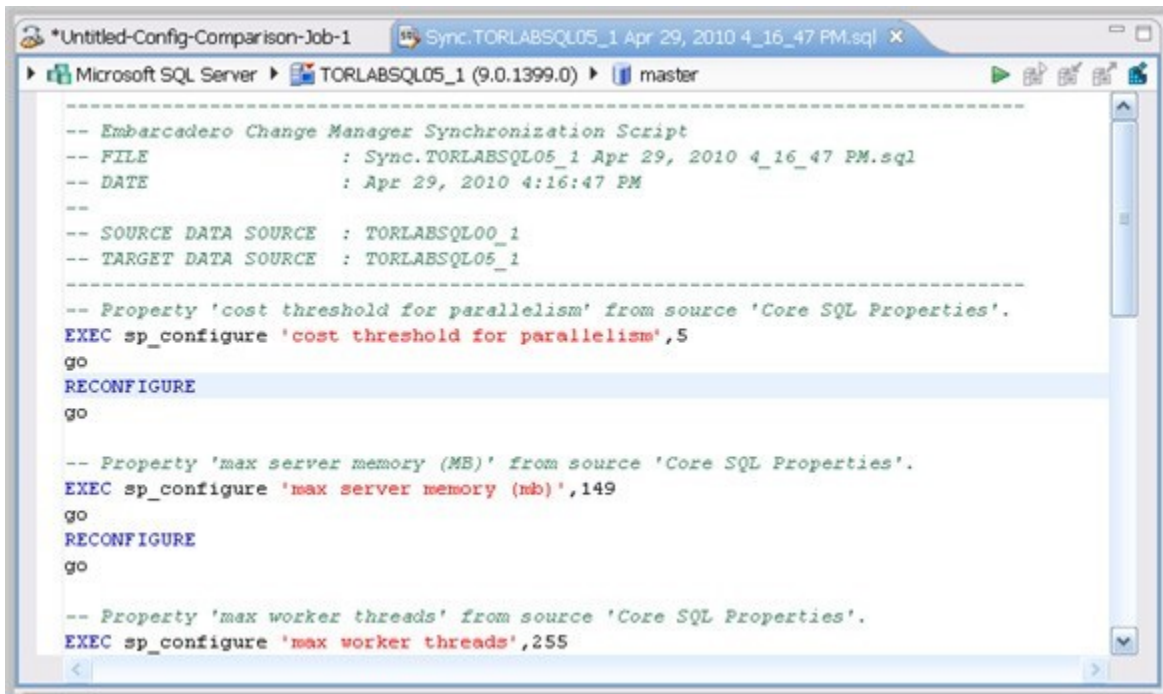
Schema archives are snapshots of existing schema values. Each data source in your environment can contain multiple archive versions. These versions can be used as baselines in schema comparison jobs against other data sources, archives, or standards.



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

## SQL Editor

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



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.



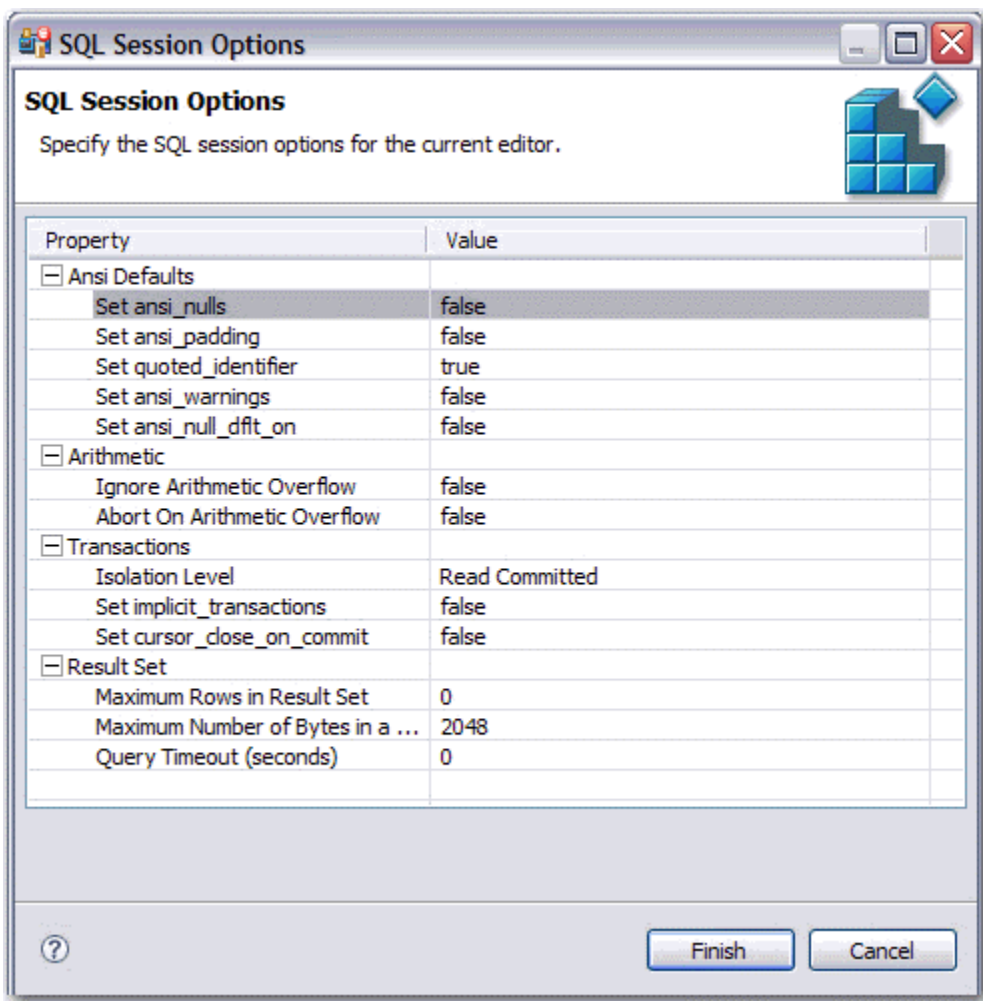
The synchronization script generation process does not support all object parameters. The following object parameters should be manually added after a synchronization script is generated, as required:

- `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

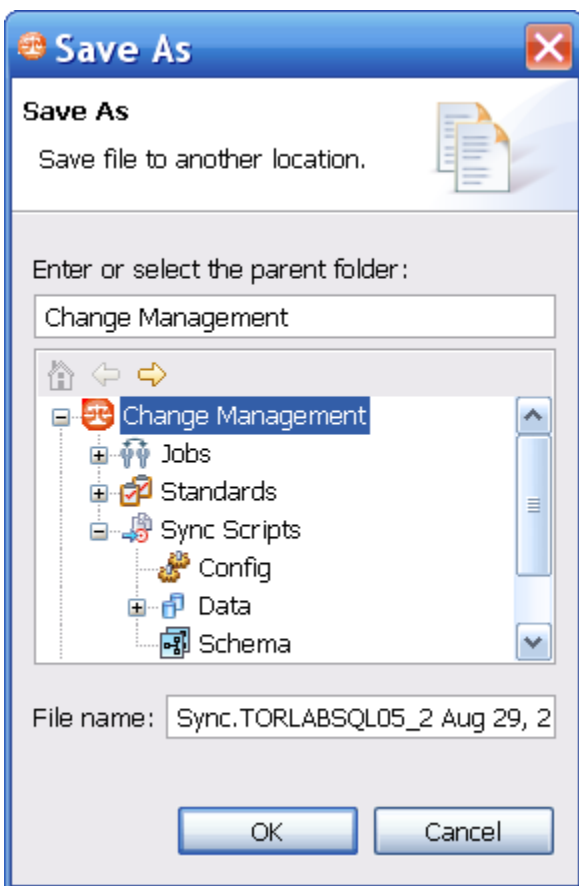
1. On the Toolbar, click the **SQL Session Options** icon.
2. In the SQL Session Options dialog, click on individual parameters in the **Value** column to change the configuration of each property, as specified.
3. Click **Finish**.

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.