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.