SoftTree
SQL Assistant™ 3.5

User's Guide

Supported database systems:

- Oracle 8i, 9i, 10g, 11g
- Microsoft SQL Server 2000, 2005, 2008
- MySQL 5, 5.1
- DB2 UDB 7, 8, 9
- Sybase ASE 12, 12.5, 15
- Sybase ASA 7, 8, 9, 10
- Microsoft Access 2003, 2007

Contents

About This Guide

This manual describes the features of the SoftTree SQL Assistant product, including how to use the graphical user interface, register editors for SQL Assistant, use code completion features, use SQL Reference functions, and database integration options. The described features and how-to instructions apply to all supported database management systems running on any platform, unless otherwise noted.

Intended Audience

This document is for Database Developers and Database Administrators.

Conventions Used in This Document

This section describes the style conventions used in this document.

Italic

An *italic* font is used for filenames, URLs, emphasized text, and the first usage of technical terms.

`Monospace`

A `monospaced` font is used for code fragments and data elements.

Bold

A **bold** font is used for important messages, names of options, names of controls and menu items, and keys.

User Input

Keys are rendered in **bold** to stand out from other text. Key combinations that are meant to be typed simultaneously are rendered with "+" sign between the keys, such as:

Ctrl+F

Keys that are meant to be typed in sequence will be separated with commas, for example:

Alt+S, H

This would mean that the user is expected to type the Alt and S keys simultaneously and then to type the H key.

Graphical symbols

 **-** This symbol is used to indicate DBMS specific options and issues and to mark useful auditing tips.

 **-** This symbol is used to indicate important notes.

Abbreviations and Product Reference Terms

DBMS – Database Management System

Oracle – This refers to all supported Oracle® database servers

SQL Server – This refers to all versions of Microsoft® SQL Server™ database servers.

Trademarks

CHAPTER 1, Overview of SoftTree SQL Assistant

Introduction

SoftTree SQL Assistant enables database developers and DBAs to realize amazing improvements in code quality and accuracy. It integrates with many widely used database editors, database management and development environments, as well as DBMS native utilities transforming them to RAD database development tools. SQL Assistant delivers unparalleled support for code typing, automatic code completion, syntax references and validations, and database object and attributes browsing. It is quite simply the ultimate SQL code assistance solution available.

SQL Assistant is fast and can be used with both small and very large database systems. It is also very flexible and can be easily customized by users to match their coding habits.

Key Benefits

- Increases database coding performance, providing advanced IntelliSense and type-ahead functionality.
- Improves code quality and accuracy.
- Provides fast code navigation and convenient syntax checking options.
- Easily integrates with most SQL and non-SQL editors.
- Provides interactive SQL help and code assistance system.
- Provides full SQL editing, database code execution, and data viewing, exporting facilities for all registered SQL and non-SQL editors.
- Fast and has small disk and memory footprint.
- Supports 7 most popular database systems.
- Can be easily installed without interrupting any existing processes or settings and used instantly.

CHAPTER 2, Connecting to Your Database

Overview

SoftTree SQL Assistant can connect to a database using either an ODBC interface, an ADO.NET interface or a native database interface. When used with pre-configured editors maintaining persistent connections to the database, SQL Assistant automatically intercepts and shares the existing connection. If a connection cannot be shared, SQL Assistant displays **SQL Assistant - Connect** dialog, which you can use to enter the required database connection properties. The entered connection properties can be saved for future use so that the next time you need that database connection you can quickly pull it by name from the saved connections list and SQL Assistant will not bother you with additional connection prompts.

Important Notes:

- The appearance of the **SQL Assistant - Connect** dialog differs for different types of connections with different prompts and options displayed on the screen.
- For ODBC based connections SQL Assistant can use both DSN and DSN-less connections. Note that DSN is a common term for "data source name" ODBC connection name.

  To use a DSN-less connection, in the **DSN / Driver** property specify the driver name, for example, "Driver={SQL Server}" without double-quotes. The driver name must be entered exactly as it appears on the Drivers tab in the Windows ODBC Administrator program.

  To use a preconfigured DSN connection, in the **DSN / Driver** property specify the ODBC DSN name as it appears in the ODBC Administrator on either File DSN or System DSN tabs, for example, "DSN={MyDSN}" without double-quotes.

  When you use a DSN-less connection to connect to your database, you must specify all required connection parameters, including the server name, port, user, and other parameters that may be required for the chosen database server type and connection method.

  When you use a DSN connection, you do not need to enter any additional parameters with the exception of user name and password in case if database side user authentication is required.

The following prompts may appear on the **SQL Assistant - Connect** dialog:

DB Type – Type of the database management system (DBMS). This parameter is defined for each target in SQL Assistant options and cannot be changed directly on the connection dialog.

Connection Type - Type of the client connection that SQL Assistant uses to communicate with the database server. The available choices depend on the chosen **DB Type** parameter andare different for different database types.

TNS Name – This is an Oracle specific parameter used with OCI and ADO.NET based connections. Enter Oracle server connection name as specified in your **TNSNAMES.ORA** file on the local computer.

Server Name - For SQL Server connections using ODBC DSN-less connections or ADO.NET, this is the name or IP address of the database server computer, or in case of connecting to a named instance name, instance name in standard server\instance format.

For MySQL connections using ODBC DSN-less connections, ADO.NET or MySQL Native, this is the name or IP address of the database server computer.

For Sybase ASE connections using ODBC DSN-less connections or ADO.NET, this is the name of the Sybase server profile defined in the Sybase client settings **SQL.INI** on the local computer.

For DB2 ADO.NET-based connections, this is the DB2 database alias defined in the DB2 client settings on the local computer.

This parameter is ignored for all other databases and connection types.

Connect As – This is an Oracle specific parameter used with OCI and ADO.NET based connections. Enter one of the following Normal, SYSDBA, SYSOPER.

Server Port – Enter this parameter only if establishing a TCP/IP based connection and if your database server is using a non-default port number to listen for new network connections. This parameter is not used with ODBC DSN connections. It is also not used with Microsoft Access connections.

Database – For SQL Server, Sybase ASE, Sybase ASA, MySQL and DB2, enter default database used for the initial connection.

For Microsoft Access connections enter full name of the Microsoft Access file to connect to.

OCI DLL – This is an Oracle specific parameter used with OCI-based connections. Enter full path for the **OCI.DLL** file.

MySQL DLL – This is a MySQL specific parameter used with MySQL Native-based connections. Enter full path for the **LIBMYSQL.DLL** file.

Data Provider – This is the name of the database connection provider used with ADO.NET-based connections. Enter provider name, for example, "System.Data.SqlClient" without double-quotes. Note that some data providers are pre-installed with the .NET framework, while other are installed with your database client software. Typically, multiple types of database providers can be used to establish a connection to the database server, yet they all require that database client softwarebe installed on the local computer. For example, both Micorsoft's data provider for Oracle System.Data.OracleClient and Oracle's data provider Oracle.DataAccess.Client can be used to establish an ADO.NET-based connection to an Oracle database. Yet, both software products internally use the Oracle client software to connect to an Oracle database. The very same Oracle client software can be used by SQL Assistant directly with via the OCI interface.

Authentication – This parameter can be used with ODBC DSN-less and ADO.NET connections only if supported by your database system.

For SQL Server, enter "Windows Authentication" or "SQL Server Authentication" without double quotes. If you chose "Windows Authentication" you do not need to specify **User Name** and **User Password** parameters.

For DB2 connections, Sybase ASE, Sybase ASA connections enter "DB Authentication" or "OS Authentication" without double quotes. Note that "OS Authentication" can only be used if your database server is configured to rely on client-side user authentication.

This parameter is not used with Microsoft Access connections.

User Name – Enter user name for the database connection. This parameter is required only for database-side user authentications.

User Password – Enter password for the database connection. This parameter is required only for database-side user authentications.

Connection Timeout – This parameter controls how many seconds to wait before canceling connect to the database server and reporting that it is unreachable. This value is supported only if it is supported by the chosen database connection method and the database driver. If the driver or the database does not support this parameter, the parameter value is ignored.

Command Timeout – This parameter controls how many seconds to wait before canceling an internal SQL command execute method call and generating an error. This value is supported only if it is supported by the chosen database connection method and the database driver. If the driver or the database does not support this parameter, the parameter value is ignored.

Encrypt Connection – This parameter controls whether to use or not to use an encrypted communication protocol for SQL Server database connection. This value is supported only with SQL Server connections if it is supported by the chosen database connection method.

Use Compression Connection – This parameter controls whether to use or not to use an encrypted and compressed communication protocol for MySQL database connection. This value is supported only with MySQL connections if it is supported by the chosen database connection method.

Save Connection Data – If checked, this option makes SQL Assistant save the specified connection parameters in its configuration file. The connection can be reused later by selecting its name from the **Server Name** or **TNS Name** drop-down lists.

Show/Hide Raw Connection String – Click on this hyperlink to show or hide connection string. The connection string contains applicable parameters specified in other options. The format of the string differs for different connection types and methods.

Important Notes: If necessary, you can specify additional non-default connection parameters in the raw connection string. If **Save Connection Data** option is checked, these additional parameters will be saved along with the default parameters and reused for future connections. When entering additional parameters using raw connection string, you must ensure that the entered parameters are supported by the database driver and client software. Consult your database driver and client software documentation for a list of supported parameters and their formats.

Oracle Database Connections And Settings

In order to connect to an Oracle database server you must have Oracle client software installed on the computer running SQL Assistant. The client software must be properly installed and configured. Its network configuration files must include connection settings for Oracle servers that you need SQL Assistant to connect to. These files must be located in the so-called ORACLE_HOME directory whose name is referenced by ORACLE_HOME variable in the system registry under HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE key.

If you have several versions of Oracle clients installed on your computer, you could choose which one you want

to use with SQL Assistant. On the **SQL Assistant - Connect** dialog click the **Options >>** button and then select the required ORACLE_HOME from the Oracle homes drop-down list. Make sure that the configuration files for the client you select contain all required Oracle server connection settings. You can verify connection settings in TNSNAMES.ORA usually located in [ORACLE_HOME]/NETWORK/ADMIN directory.

SQL Assistant supports both regular user connections and connections for users with administrative privileges. You must use second type of connection when connection as a SYS user. The **Connect As** option on the **SQL Assistant - Connect** dialog allows you to choose database connection type. Please note that for SYSDBA connection the assigned schema is SYS; for SYSOPER the assigned schema is PUBLIC -- regardless of the actual Oracle username supplied.

 **Important Note:** When working with pre-configured targets for Oracle, such as SQL*Plus, SQL Assistant silently intercepts the existing connection and uses that connection to access Oracle catalog tables and views. When working with user-defined targets, SQL Assistant opens new connection to the database server, which is not shared with the target application. This secondary connection can be opened using the same or different user credentials. The secondary connection is closed automatically in the event the target editor or SQL Assistant is closed.

SQL Server Database Connections And Settings

In order to connect to a SQL Server database server SQL Assistant uses either Microsoft SQL Server ODBC driver pre-installed by default on all Windows systems, SQL Server Management Objects (SSMO), which are installed with SQL Server Management Tools such as SQL Server Management Studio or ADO.NET SQL Server provider. The connectivity method depends on the method used by the target editor.

SQL Assistant supports 2 types of user authentication with SQL Server:

- Windows Authentication - in this case, SQL Assistant does not prompt a user for credentials, but instead it uses an access token assigned at the time the user logged on using a Windows account. SQL Assistant only prompts for the name of the target SQL Server instance at the time when connection is attempted.

- Database Authentication -- in this case, SQL Assistant requires a user to specify login name, password and the name of the target SQL Server instance at the time when connection is attempted.

 **Important Notes:**

When working with certain pre-configured targets for SQL Server that use ODBC connections, such as SQL Query Analyzer, SQL Assistant silently intercepts the existing connection and uses that connection to access SQL Server catalog tables and views.

When working with SQL Server Management Studio or SQL Server Management Studio Express that use SMO connections, SQL Assistant silently intercepts the existing connection and uses that connection to access SQL Server catalog tables and views.

When working with user-defined targets and targets that don't use known connection types, SQL Assistant opens a new connection to the database server, which is not shared with the target application. This secondary connection can be opened using the same or different user credentials. The secondary connection is closed automatically in the event the target editor or SQL Assistant is closed.

MySQL Database Connections And Settings

SQL Assistant presently supports MySQL database servers versions 5.0 and up. In order to connect to a MySQL database server, SQL Assistant uses either MySQL native client interface, if available, or MySQL

ODBC driver, which are not part of SQL Assistant. The connector and/or driver must be already preinstalled before you can use SQL Assistant with MySQL targets. MySQL client and ODBC driver can be obtained from MySQL AB http://www.mysql.com.

Important Notes:

For ODBC based connections, SQL Assistant requires MySQL ODBC driver version 3.5. Please note that MySQL ODBC driver v5.0, which is at the time of this manual was available as a beta version download, is not fully ODBC compliant and doesn't support certain standard ODBC features which are required by SQL Assistant.

When working with pre-configured targets for MySQL, such as MySQL Query Browser, SQL Assistant silently intercepts the existing connection and uses that connection to access MySQL catalog tables and views. When working with user-defined targets, SQL Assistant opens new connection to the database server, using an ODBC driver, and this connection is not shared with the target application. This secondary connection can be opened using the same or different user credentials. The secondary connection is closed automatically in the event the target editor or SQL Assistant is closed.

DB2 Database Connections And Settings

SQL Assistant presently supports DB2 UDB for Linux, Unix and Windows versions 7 and up. In order to connect to a DB2 database server, SQL Assistant uses DB2 ODBC driver, which is not part of SQL Assistant. The driver must be already preinstalled before you can use SQL Assistant with DB2 targets. DB2 ODBC driver is typically installed with DB2 client software and is part of your DB2 license. The driver can be obtained from IBM Corporation http://www.ibm.com.

Important Notes:

When working with certain pre-configured targets for MySQL that use ODBC connections, such as MySQL Query Browser, SQL Assistant silently intercepts the existing connection and uses that connection to access MySQL database catalog tables and views.

When working with user-defined targets and targets that don't use ODBC connections, SQL Assistant opens new connection to the database server, which is not shared with the target application. This secondary connection can be opened using the same or different user credentials. The secondary connection is closed automatically in the event the target editor or SQL Assistant is closed.

Sybase Database Connections And Settings

SQL Assistant presently provides support for Sybase Adaptive Server Enterprise versions 12.5 and up and also for Sybase Adaptive Server Anywhere v8.0 and up. In order to connect to a Sybase database server, SQL Assistant uses Sybase ODBC driver, which is not part of SQL Assistant. The driver must be already preinstalled before you can use SQL Assistant with Sybase targets. Sybase ODBC drivers are typically installed with Sybase client software or separately and it is part of your Sybase license. The drivers for ASE and ASA can be obtained from Sybase Corporation http://www.sybase.com.

Important Notes:

When working with Sybase targets, SQL Assistant opens new connection to the database server, which is not shared with the target application. This secondary connection can be opened using the same or different user

credentials. The secondary connection is closed automatically in the event the target editor or SQL Assistant is closed.

Microsoft Access Database Connections And Settings

SQL Assistant presently provides direct integration and support for Microsoft Access databases versions 2003 and 2007. In addition, it can be used with earlier versions of Microsoft Access database via ODBC connections. In order to connect to a Microsoft Access database, SQL Assistant uses Microsoft Access ODBC driver, which is not part of SQL Assistant. The driver must be already preinstalled before you can use SQL Assistant with Microsoft Access targets. Microsoft Access ODBC drivers are typically pre-installed on all Windows systems. Latest ODBC driver versions can be obtained from Microsoft Corporation http://www.microsoft.com.

 **Important Notes:**

When working with Microsoft Access targets, SQL Assistant opens new connection to the database file, which is not shared with the target application. This secondary connection can be opened using the same or different user credentials. The secondary connection is closed automatically in the event the Microsoft Access IDE, or non-Access target editor or SQL Assistant is closed.

Database Connection Settings and Security

By default, SQL Assistant remembers used connection settings and user credentials and stores them in the configuration file SqlAssist.sas file. Your connection settings, user names and passwords are stored encrypted so that nobody can read or change them without your authorization.

If for whatever reason you do not want to save connection settings between work sessions you can uncheck the **Save Connection Data** option on the **SQL Assistant - Connect** dialog. This option is not visible by default. To display it, click the **Options >>** button.

If your password or user name changed since you saved the connection settings causing the database connection to fail, SQL Assistant automatically prompts you to re-enter your connection settings and updates them in the configuration file.

You can use SQL Assistant's Options dialog to change your connection password proactively and to modify other connection settings. See Managing Database Connections topic in CHAPTER 13 for more information.

Automatic Database Connection Recovery

Because of SQL Assistants unique ability to share connections with certain types of SQL editor targets, its working is heavily dependent on the availability and speed of the database connection. In certain situations, if a database connection breaks, SQL Assistant may attempt to automatically repair the broken connection in order to maintain uninterrupted code entry. If an attempt to repair broken connection fails, SQL Assistant displays its own **Connect to Database** dialog independent of the target editor. You can use the dialog to provide the

necessary connection details as described in other topics of this chapter. For more information, see topics specific to your database type:

Oracle Database Connections And Settings

SQL Server Database Connections And Settings

MySQL Database Connections And Settings

DB2 Database Connections And Settings

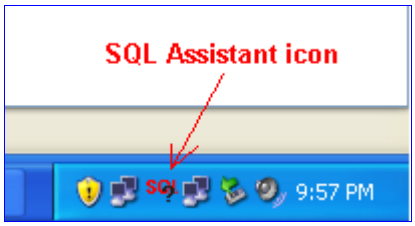
Sybase Database Connections And Settings

CHAPTER 3, Using SQL Coding Assistant

Starting and Stopping SQL Assistant

During the installation, the setup program places SQL Assistant shortcut to the Windows Startup folder so that when you logon to the system Windows loads SQL Assistant automatically. If you need to start SQL Assistant manually, simple click SQL Assistant shortcut in the SQL Assistant Program Folder or run SqlAssist.exe file.

When SQL Assistant is running, its icon appears in the Window system tray.



To exit SQL Assistant:

1. Right-click the SQL Assistant icon in the Window system tray.
2. Select **Exit** command from the popup menu.

Tip: If SQL Assistant has been registered as an Eclipse, or as a Microsoft SQL Server Management Studio add-on or as a Visual Studio 2005/NET add-on, the host program loads the add-on automatically. The host program automatically unloads the add-on when the host program is closed. If you have SQL Assistant's add-on registered for any of these programs, you do not need to run SQL Assistant as an icon in the system tray. You can then disable "Load on Windows Startup" option" in the SQL Assistant options. See Managing SQL Assistant Load Methods topic for more information.

Temporarily Pausing SQL Assistant

To pause SQL Assistant services without exiting the program

1. Right-click the SQL Assistant icon in the Window system tray.
2. Select **Suspend** command from the popup menu. SQL Assistant suspends its activities and does not display any SQL Assistance popups in this mode. SQL Assistant icon in the Windows system tray changes its color from red to gray.

To resume SQL Assistant services:

1. Right-click the SQL Assistant icon in the Window system tray.

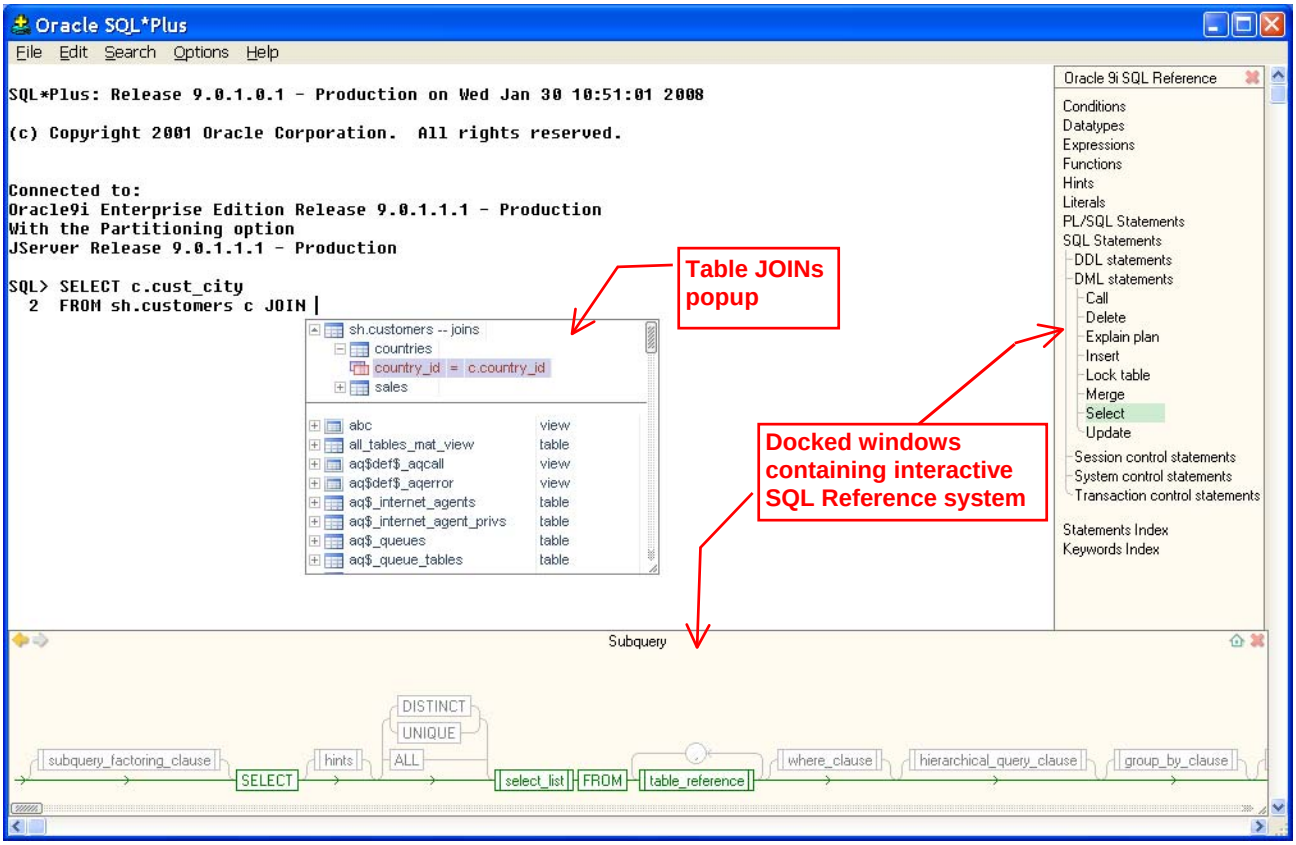
2. Select **Resume** command from the popup menu. SQL Assistant resumes its normal activities. SQL Assistant icon in the Windows system tray changes its color from gray back to red.

Tip: If you have SQL Assistant and target editor menu integration enabled, you can use SQL Assistant's Suspend and Resume command available in editor's right-click context menu and/or top level menu.

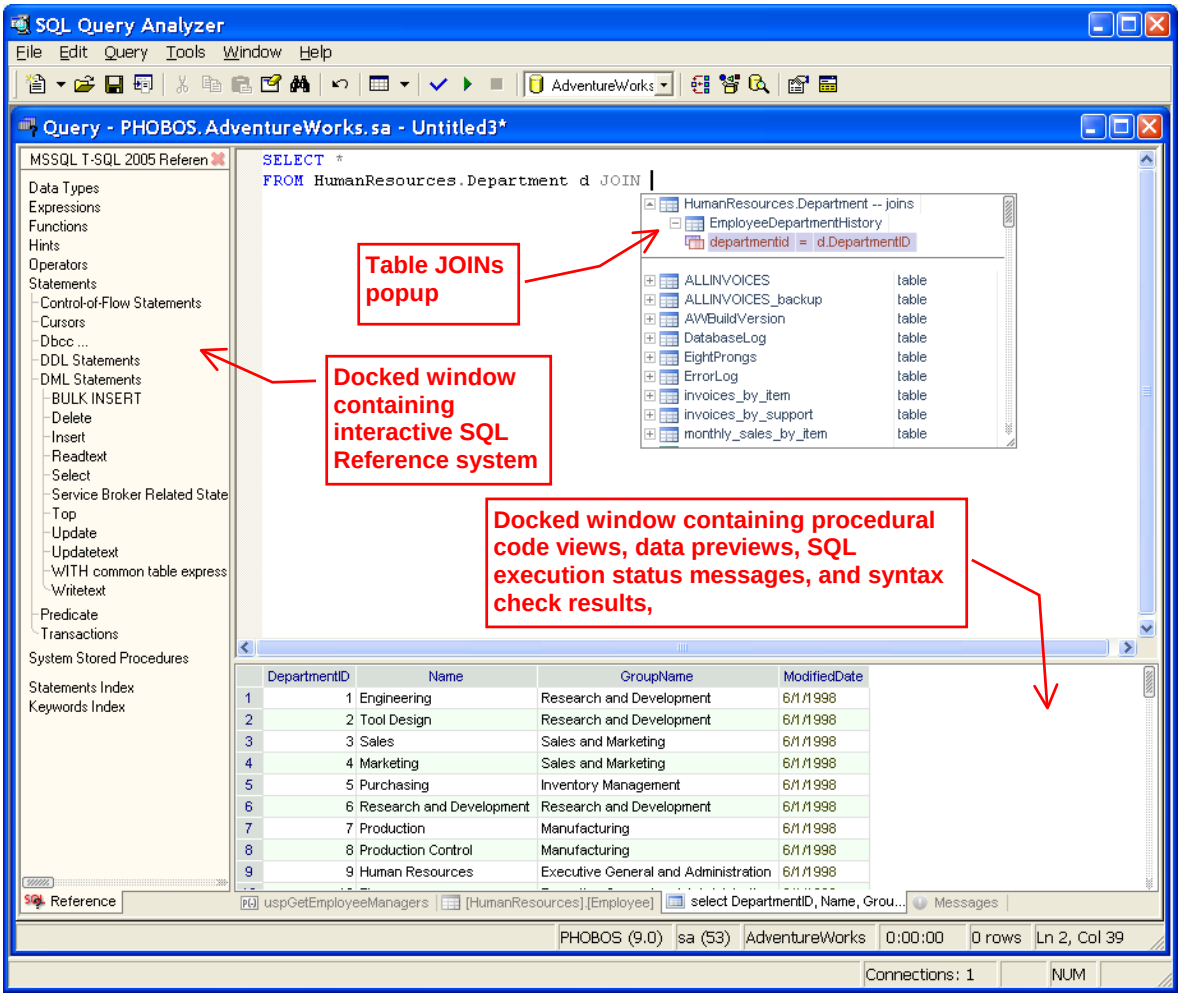
SQL Assistant Windows and Appearance

SQL Assistant windows and popups have consistent look a in all target editors. However their availability, position and management functions depend on the target editor type and chosen SQL Assistant options. For example, when working with in SQL*Plus, which comes as one of the pre-registered targets, the SQL Reference window appears on the right hand side of the screen while in all other pre-registered targets it is by default displayed as a docket window on the left hand side of the target editor screen.

Below is an example SQL*Plus screenshot demonstrating several active SQL Assistant windows.



In comparison, a similar set of SQL Assistant windows in Microsoft SQL Server Query Analyzer may look like the following example screenshot.



There are two types of SQL Assistant windows that can appear in the target editor workspace:

- **Docked windows** are windows docked to the sides of the editor workspace area. SQL Assistant currently supports both vertical and horizontal docket Windows. The SQL Reference System and Code Structure View are always displayed in the vertically docked window. They share the same window and appear as two different tab pages. All other SQL Assistant windows appear as tabs in the horizontally docked window in the bottom part of the editor workspace area.

  Docked windows can be resized separately or together with the target editor.

  In Integrated Development Environments (IDE) supporting multiple-editor window interface, each target editor window can have different set of SQL Assistant docked windows attached to it.

  Docket windows utilize tab interface to display multiple pages with different contents. This tabbed interface reduces the number of unused windows displayed at one time and maximizes the editor's workspace for development and management.

- **Popups** are floating windows displayed next to the editing position. They may show and hide automatically providing context based assistance with the SQL code typing.

  Popup windows can be moved and resized

Manually Invoking SQL Assistant Windows and Popups

By default, SQL Assistant displays help popups automatically. It monitors text entered in the SQL editor and as you type SQL commands it display context-based popups containing items that you may want to insert in the text.

Using the Keyboard Hot Keys

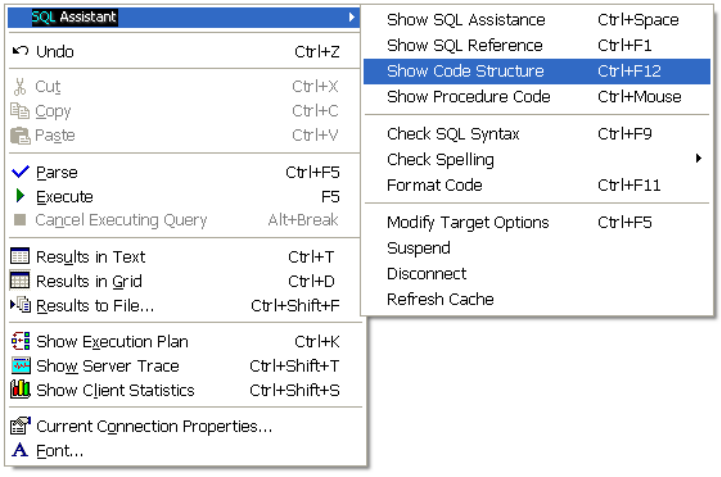
If for whatever reason SQL Assistant popup is not displayed and you want to display it at the current cursor position, you can use the global Ctrl+Space hot key or a custom hot key if you changed the default key. The type of the popup displayed is context driven. Read the following topics for more details. The hot key used to invoke SQL Assistant popup can be changed on the Options dialog. For more information, see Customizing Hot Keys topic in CHAPTER 13.

Press the Esc key at any time to immediately close SQL Assistant's popup.

Using Context and Top-level Menus

In order to improve your SQL Assistant experience and relieve you from the need to remember various hot keys, SQL Assistant supports direct integration with top-level right-click popup menus available in your development environments. To use SQL Assistant functions right-click on the text in the editor where you want to invoke SQL Assistant then select the top-most item in the context menu. The item text should read **SQL Assistant** and lead to the next menu level containing specific SQL Assistant functions.

Note: In editors that do not support right-click context menus, for example, in SQL*Plus, SQL Assistant creates it own right-click menu. The following example, demonstrates SQL Assistant commands displayed in the Microsoft SQL Query Analyzer context menu.

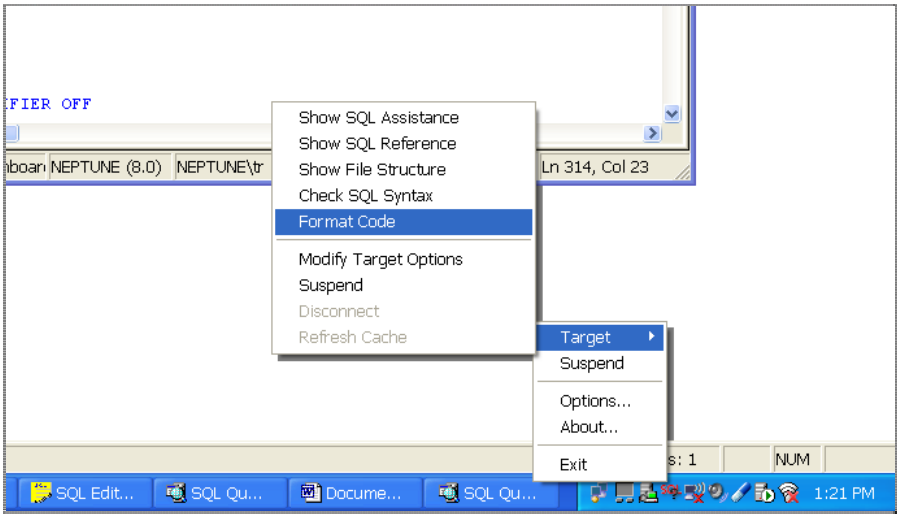


The right-click context menu integration is enabled by default in all pre-configured and new targets, while top-level menu integration is disabled by default. See Customizing Target Editor Menu Integration topic in

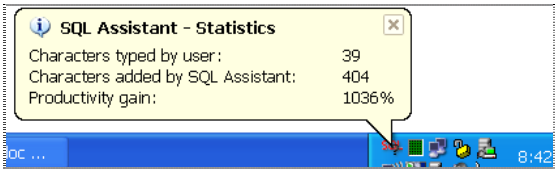
CHAPTER 7 for more information on how to customize SQL Assistant integration with target editor menus.

Using System Tray Icon Menu

All commands available in the SQL Assistant menus integrated with the target editor are also available in the menu associated with the SQL Assistant icon displayed in the Windows system tray. This menu is always available. Right-click on the system tray icon to display the menu then choose the **Target** menu branch as on the following screenshot.



In addition to invoking various SQL Assistant commands and features, you can use the system tray icon to get a glimpse at SQL Assistant usage statistics. Rest the mouse pointer over the SQL Assistant's icon in the system tray area and after a couple of seconds you should see usage statistics balloon looking similar to is demonstrated on below example screenshot.



Read the following topic for information about usage statistics and how they are calculated.

Understanding and Using SQL Assistant's Usage Statistics

SQL Assistant gathers keyboard usage statistics in all target editors where it is being active. The usage statistics provide you with a tool for analyzing which coding methods and SQL Assistant usage technique provides most gains. For example, you can use it to compare whether entering SQL queries with SQL Assistant's help starting from the FROM and JOIN clauses and then adding SELECT clause to the built query skeleton is more efficient then starting with the SELECT clause first and then adding JOIN.

SQL Assistant provides 3 types of usage statistics:

Global-level statistics for all target editors and all editing sessions – this statistics are displayed on the **About** tab page in the SQL Assistant's Options dialog and also displayed as a balloon for the SQL Assistant's system tray icon. The numbers in these places show keyboard usage in all target editors ever used with SQL Assistant in all editing sessions.

Editor-level statistics for all editing sessions – this statistics are available in the SQL Assistant's menus – both in the right-click context menu and top-level menu, in case target editor top-level menu integration is enabled. The numbers show keyboard usage for all instances of current target editor in all editing sessions.

Instance-level statistics for the current sessions – this statistics are available in the SQL Assistant's menus – both in the right-click context menu and top-level menu, in case target editor top-level menu integration is enabled. The numbers show keyboard usage for the current editor instance in the current editing session only.

Meaning of Statistics.

All statistics are real-time and consist of the following numbers.

Typed by User – this is the number of characters typed by user in the target editor. This number includes all characters and digits, carriage returns, tabs and other characters that part of the code.

Added by SQL Assistant – this is the number of characters that SQL Assistant added to the code. This number includes all characters and digits, carriage returns, tabs and other characters that part of the code. It includes text inserted using SQL Assistant's popups, text generated by code snippets, tabs inserted by auto-indent feature and other SQL Assistant's dynamic code auto-formatting features working in the background as you enter the code.

Typing Speed – this is the number measuring your average typing speed as an average number of characters added to the code per minute. This number includes all characters including these characters typed and characters added for you by SQL Assistant. Note that this statistic calculates typist speed during active periods of code entry. Any inactive periods 30 seconds and longer are ignored.

Productivity Gain - this is the ratio of how much code you were able to enter with the help of SQL Assistant expressed as a percentage. This number is only available for the global statistics level. For example, if you typed 50 characters, and 100 more characters were added for you by SQL Assistant's popups and/ code snippets, your productivity gain was 200%.

 **Tip**: An average SQL coder well familiar with the database schema and remembering most table and column names can type in average anywhere from 20 to 50 characters per minute. The same average coder working with a non-familiar database schema can type in average only from 1 to 5 characters per minute because most of the time is spent researching names and typing/correcting names of schema, tables and table columns. SQL Assistant can help in both cases improving average typing speed by a factor of 2 or more.

Disabling and enabling statistics collection

In case you do not want SQL Assistant to gather keyboard usage statistics, you can disable this feature.

1. Use any available method to display SQL Assistant's menu.

2. Click the **Statistics** submenu and in that submenu click the **Active** command. The checkmark next to the Active command will disappear and the SQL Assistant will stop collecting usage statistics.

In case you want to re-activate this feature, repeat steps described above.

Resetting Statistics

To reset statistics and begin anew:

1. Use any available method to display SQL Assistant's menu.
2. Click the **Statistics** submenu and in that submenu click the **Reset** command.

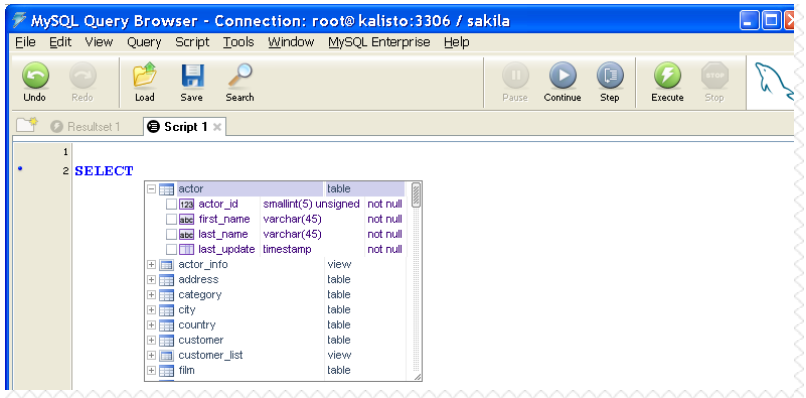
How to Build Advanced SQL Commands With Only a Few Keystrokes

The following examples demonstrate several important techniques for fastest SQL code generation complete with multiple tables joins, column selection list, joins and other bells and whistles.

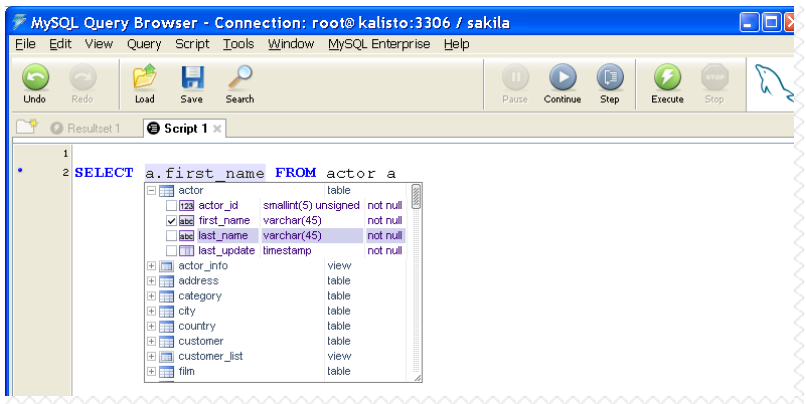
Example 1: Building complete SELECT starting with column names

1. Type **SELECT** and press spacebar. The SQL Assistant's objects popup will appear.

2. Press **Arrow Right** navigation key to expand columns of the **actor** table. The result should like the below image

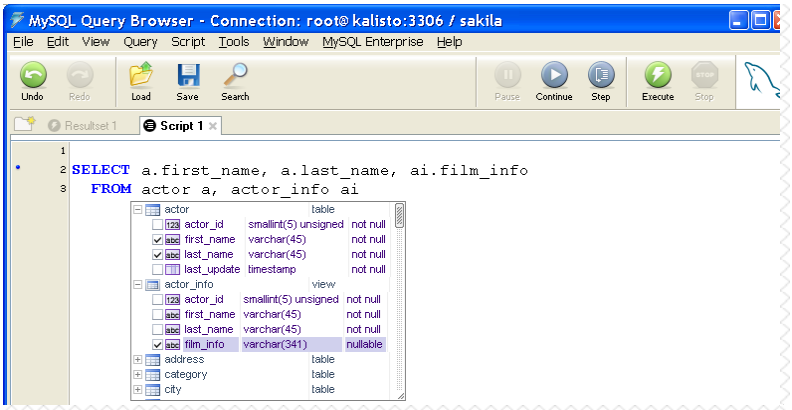


3. Press **Arrow Down** twice to move the selection to **first_name** column. Press **Arrow Right** key to pick this column. Note that the SELECT in the editor will change to the following
`SELECT a.first_name FROM actor a`

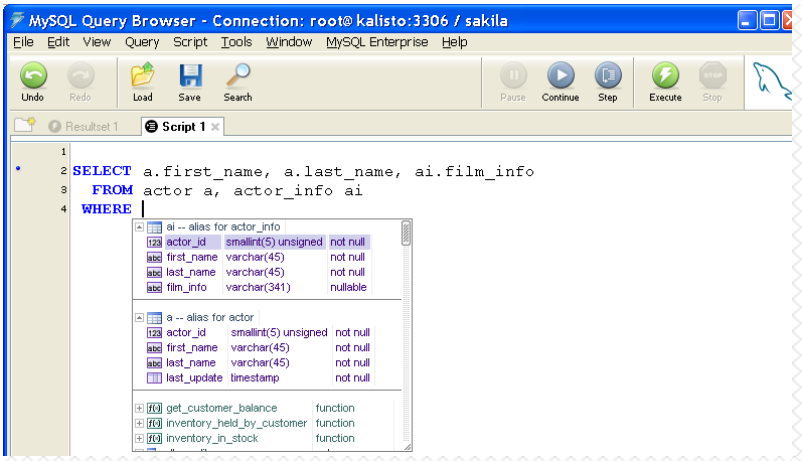


Press **Arrow Down** to select the next column **last_name** and **Arrow Right** again to add this column to the SELECT list.

4. Using **Arrow Down** and **Arrow Right** keys, scroll to the next table **actor_info** and expand that table too, then using the same the technique pick **film_info** column from **actor_info** table.

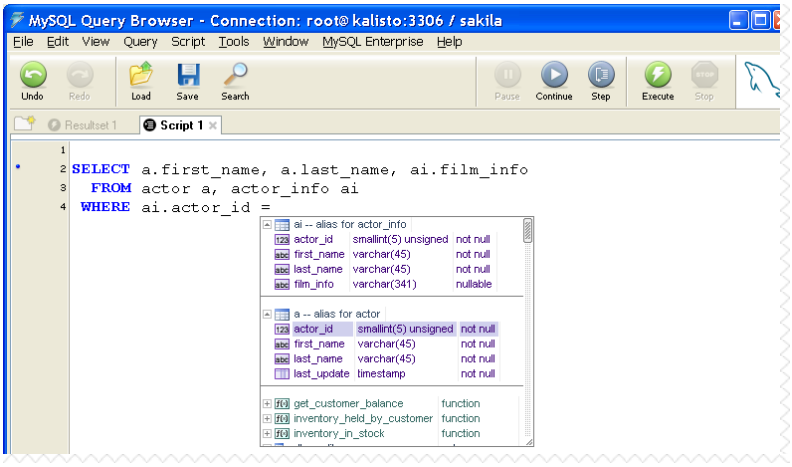


5. Click at the end of the generated query and add WHERE clause, in other words, click after "ai" and press Enter key. Type WHERE and press space. The SQL Assistant's columns popup will appear for the previously inserted query



6. Using the **Arrow Down** scroll to the second line and press the Enter key to paste **ai.actor_id.** The SQL Assistant column popup will disappear.

7. Type the equal sign. The SQL Assistant's columns popup will appear again. Scroll down to the **a.actor_id** column and press enter or simply double click that column



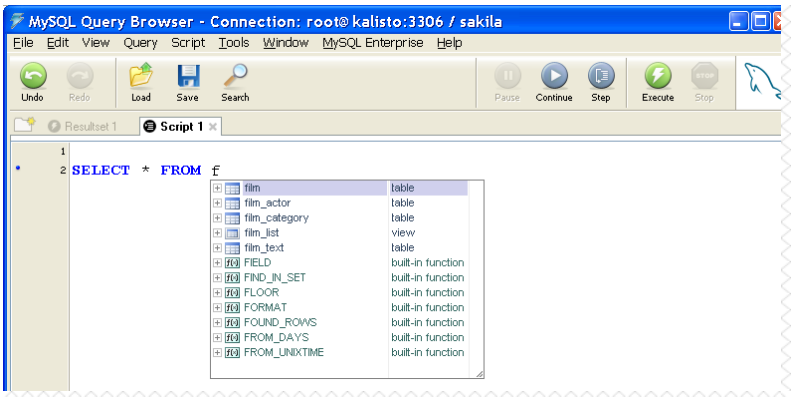
You now have the complete SELECT statement with multiple tables columns and a WHERE clause.

Note: This result required only 22 quick key-presses using mostly 2 adjacent navigation keys and 2 mouse clicks. Compare this to the length of the text, that is 106 characters long, and you come up with 79% saving. Moreover, the coding of such query with 3 joined tables took only a few seconds.

Example 2: Building complete SELECT starting with joins

1. Type **SELECT * FROM** and press spacebar. The SQL Assistant's objects popup will appear.

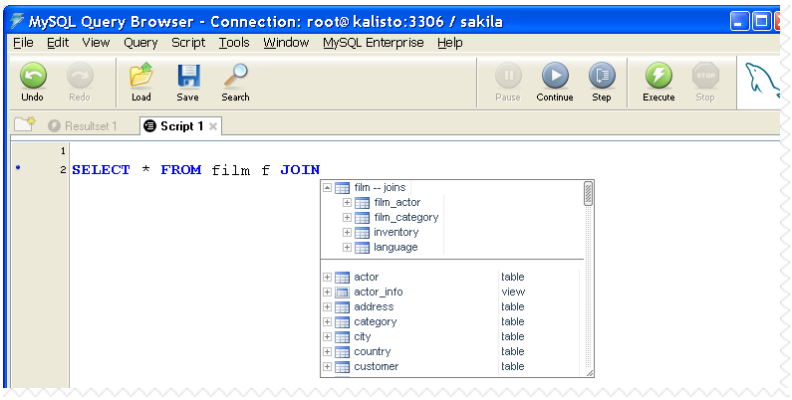
2. Type letter F. The object list will be automatically filter to show objects beginning with the letter F. The result should like the below image:



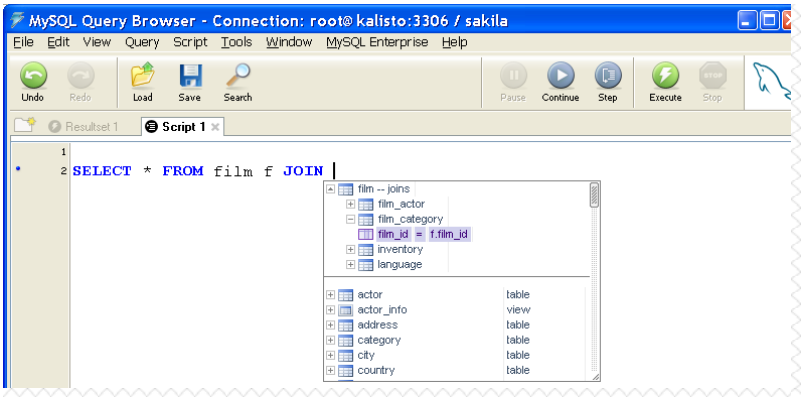
Press Enter key to pick the **film** table. SQL Assistant will modify the SQL query in the editor to the following:

```
SELECT * FROM film f
```

3. Now type **JOIN** and press spacebar. The SQL Assistant's object -joins popup will appear.



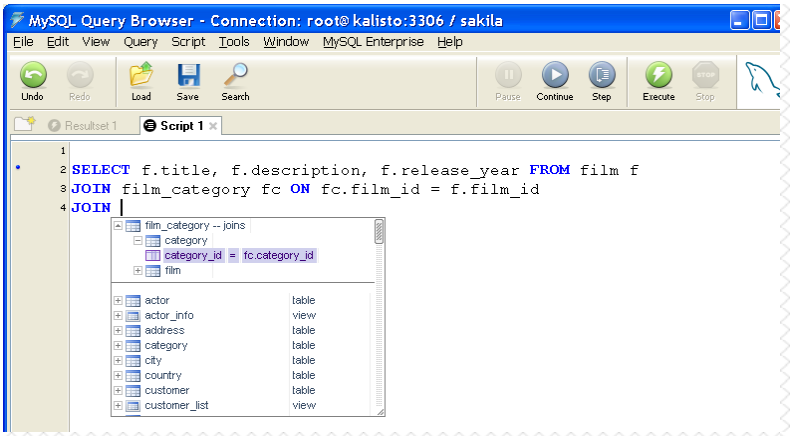
Using the **Arrow Down** and **Arrow Right** keys, expand the **film_category** table join The result should like the below image:



Now press the Enter key to pick it for the query and you are almost done. In the editor you now have the following query:

```
SELECT *
FROM film f JOIN film_category fc ON fc.film_id = f.film_id
```

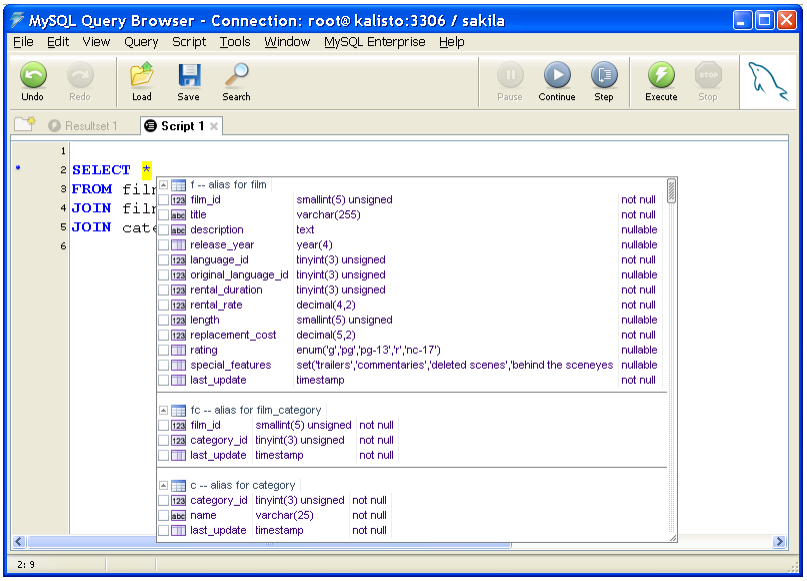
4. Type **JOIN** again at the end of the query to add another join. Follow same steps as described above, but this time pick the **category** table join:



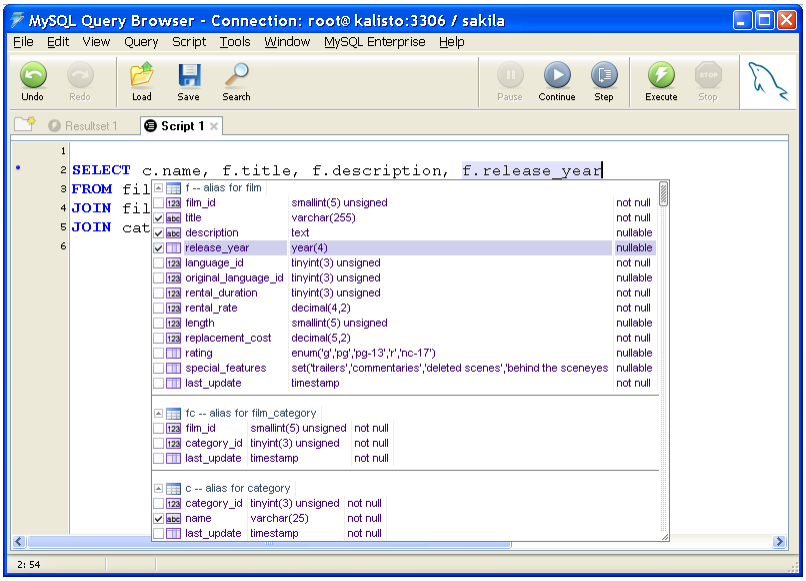
You now have:

```
SELECT *
FROM film f
JOIN film_category fc ON fc.film_id = f.film_id
JOIN category c ON c.category_id = fc.category_id
```

5. Click near to the asterisk and press Ctrl+Space to invoke SQL Assistant's core refactoring and column expansion popup (for more info see Advanced Code Expansion for * and Object Columns and Arguments topic)
You should get the following:



6. Using mouse or navigation keys select the required columns from the referenced tables clicking the check boxes.



The resulting query may look like below

```
SELECT c.name, f.title, f.description, f.release_year
FROM film f
    JOIN film_category fc ON fc.film_id = f.film_id
    JOIN category c ON c.category_id = fc.category_id
```

Note: This result required only 35 quick key-presses, including spacebar presses. Compare this to the length of the text, that is 146 characters long, and you come up with 76% saving. Moreover, the coding of such query with 3 joined tables took only a few seconds.

Example 3: Creating multi-line comments with 4 keys

1. Type **/**** and then press Ctrl+Enter hot key. You are going to get

```
/*****************************************
* |
*****************************************/
```

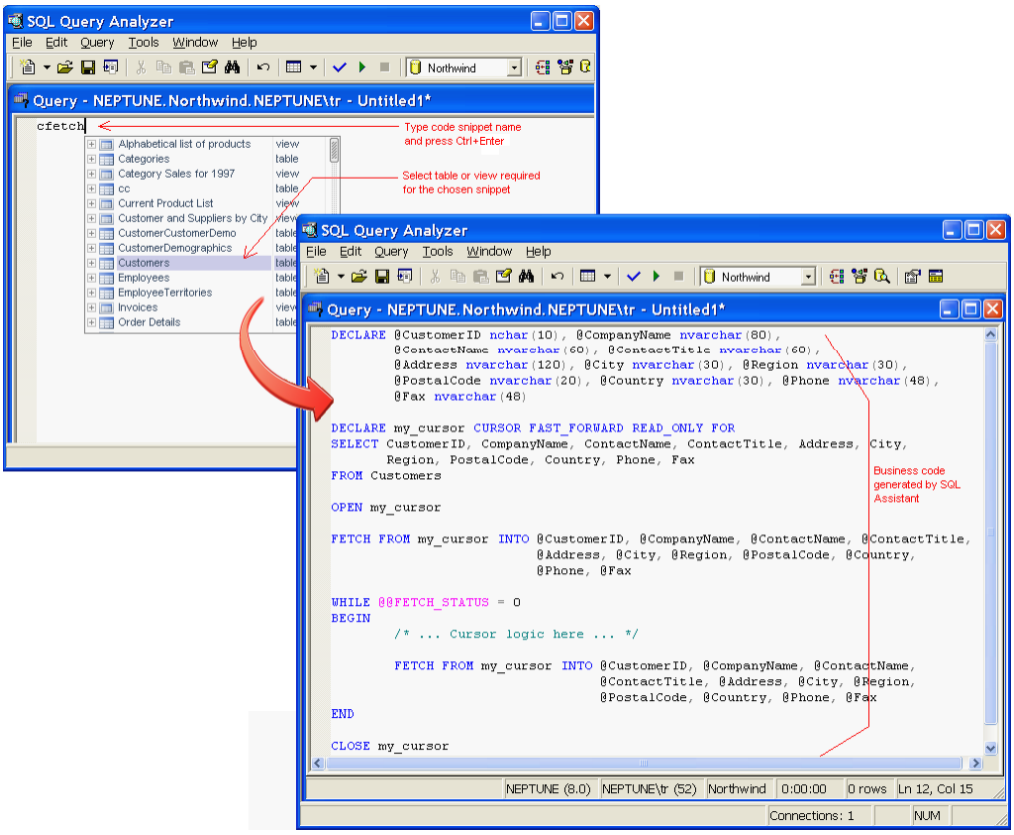
and the cursor will be positioned on the second line with the exact position indicated by the pipe | character.

Tip: The above simple example involves one of the pre-defined code snippets that are part of standard SQL Assistance configuration. SQL Assistant is packed with a large number of pre-defined snippets. Some snippets are very simple like automatic adding of the END keywords to every BEGIN and indenting the code, other are more complicated and can be used to generate a body of a stored procedure or a complete cursor logic along with table references, variables and loops. Code snippets provide very efficient methods for quick code entry.

Example 4: Generating complete-cursor logic with 7 keys and 1 click

1. Type **cfetch** and then press Ctrl+Enter hot key. The object popup will appear.

2. Click any of the tables or views available in the popup. The result is displayed on the following picture:



Tips:

- The above simple example involves one of the pre-defined code snippets that are part of standard SQL Assistance configuration.
- You can create your own snippets to quickly generate common code used in your queries and procedures.
- For more information for how to use advanced code snippets, see CHAPTER 6, Using and Creating Code Snippets for Fast Code Entry

Using Object Name Code Completion Features

The contents of the SQL Assistant popup is context and database-driven. For example, when you type FROM keyword in a SELECT statement or an UPDATE keyword the popup list is populated with items that may want to insert into the text. Such items may include table and view names, table function names, schema names, and so on. The following object types are supported:

Oracle:

- Schemas
- Tables
- Views
- Packages
- Types
- Procedures
- Functions
- Aliases

Microsoft SQL Server:

- Databases
- Schemas
- Tables
- Views
- Table Functions
- Procedures
- Functions
- Aliases

DB2:

- Schemas
- Tables
- Views
- Procedures
- Functions
- Table Functions
- Aliases

MySQL:

- Schemas
- Tables
- Views
- Procedures
- Functions

Sybase Adaptive Server Enterprise:

- Databases
- Schemas
- Tables
- Views

- Procedures
- Functions

 **Sybase Adaptive Server Anywhere:**

- Databases
- Schemas
- Tables
- Views
- Procedures
- Functions
- Table Functions

 **Microsoft Access:**

- Tables
- Views (Queries)

For your convenience, items of different types are displayed in different colors and indicated by different icons displayed on the left side of the popup list.

If the item you want is below the visible area of the popup, scroll through the list to locate the item and then double-click or press Enter key to insert it into the text. Alternatively, you can start typing the first characters of the item to display only those items that begin with the typed characters.

If the item you want is not in the popup list, continue typing normally and the popup will disappear automatically.

The popup containing object, schema and database names can appear when you type space character after SELECT, FROM, JOIN, TRUNCATE TABLE, DECLARE, EXEC, EXECUTE, CALL or USE keywords. It can also appear after a database or schema name followed by a dot character. Set of keywords that trigger the automatic popup is controlled by the SQL Assistance type parameter that you select in SQL Assistant options for the current target type, such as T-SQL or PL/SQL.

Using Object Name Auto-Completion

You can use the auto-completion feature when modifying the existing code. Using Ctrl+Space shortcut you can make SQL Assistant to auto-complete partially entered object name, for example, if in the Microsoft SQL Server Query Analyzer connected to the master database you type

*SELECT * FROM sysda*

and then press Ctrl+Space hot key, SQL Assistant automatically completes the text as

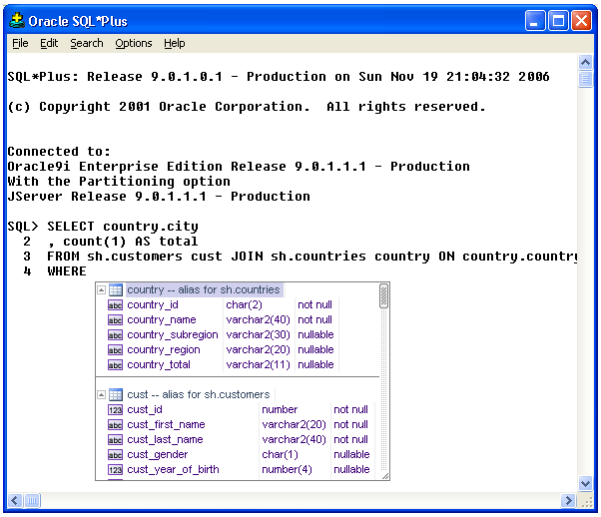
*SELECT * FROM sysdabases*

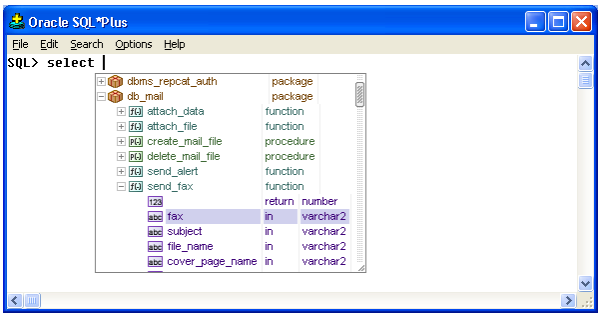
because there is only one object in the database catalog whose name begins with "sysda." In case multiple names match the entered text, a regular SQL Assistant object popup appears on the screen with the list of objects whose names begin with "sysda" prefix.

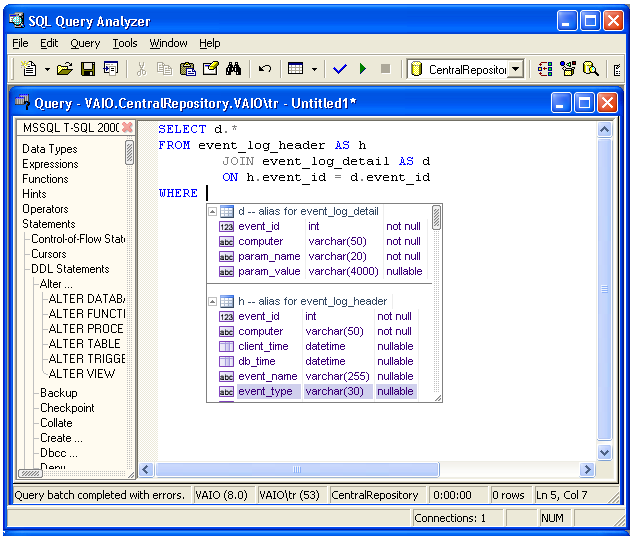
Using Column and Parameter Names Completion Features

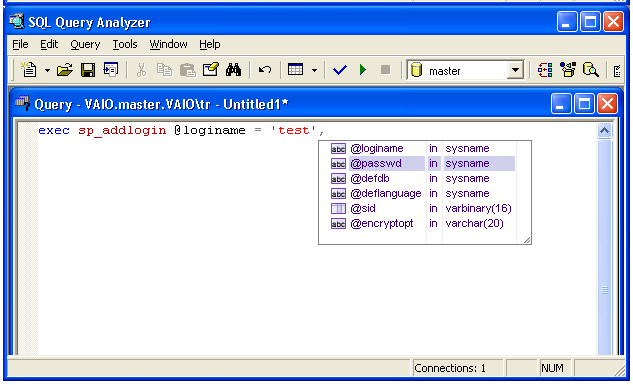
The table/view column list popup appears expanded automatically when SQL Assistant is invoked after a dot character, comma, equal sign or end of procedure name. The popup item list is normally limited to column names of the referenced table or parameters of the referenced procedure or function.

Below are several examples demonstrating column and parameter names completion feature:









SQL Assistant popup window contents depend on the popup invocation context and the target environment.

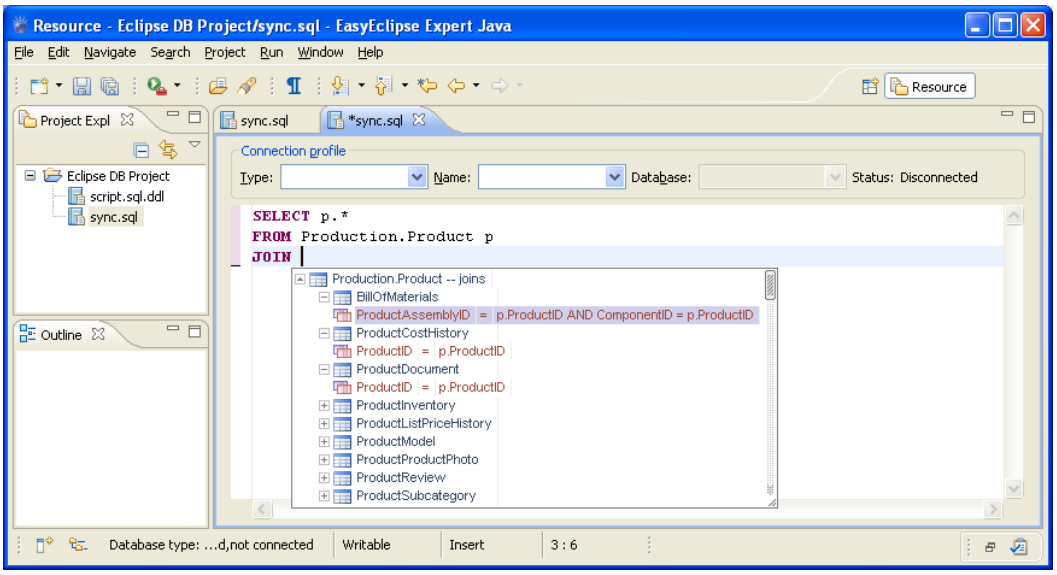
Using JOIN Clause Completion Features

The JOIN clause completion popup appears automatically immediately after typing JOIN keyword or after typing ON keyword or equal sign which are part of the JOIN clause. The JOIN popup may also appears after WHERE clause correlated sub-queries.

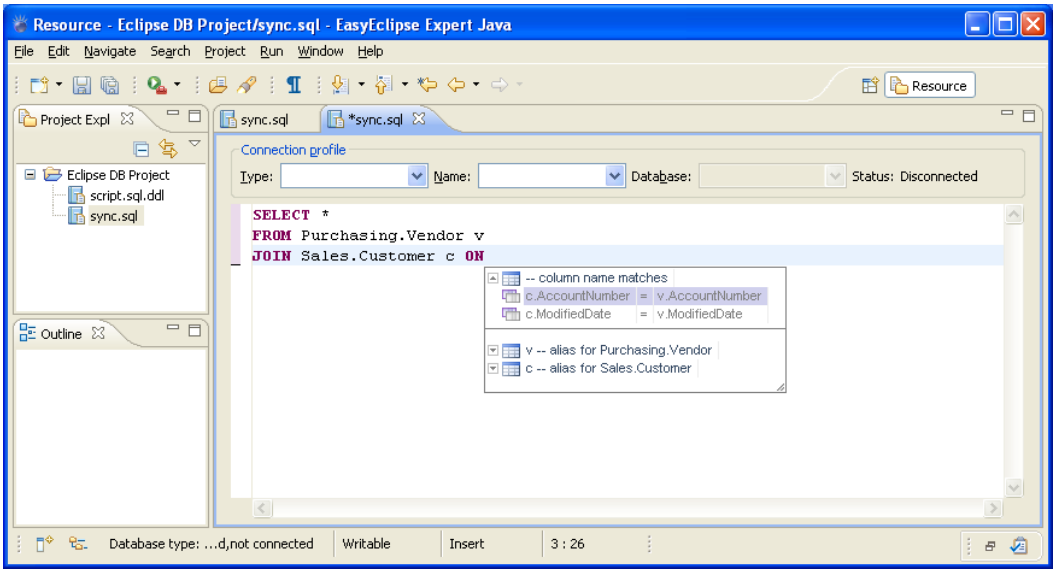
In the popup appearing after the JOIN keyword, you are presented with a list of suggestions based on the analysis of the relationship constraints defined for the tables referenced in the JOIN clause as well as the list of all other tables in the database so that you can pick other tables without referenced constraints defined. The list of suggestions appears on top of the popup and a horizontal line separates it from the rest of the popup contents.

You can select an object name to add it to the current SQL statement in the editor and then type ON keyword and select a condition for the join. However, it is must more efficient to expand the table required for the join to display a list of suggested join conditions for that table and the select one of them. SQL Assistant will automatically generate the entire JOIN clause including table names, aliases and columns.

The following sample screenshot demonstrates how to use JOIN clause completion feature when using referential integrity based suggestions



In addition to using referential integrity definitions for JOIN clause suggestions, SQL Assistant can use column-name matches. It uses different color styles for display to help you recognize and distinguish between different suggestion types. The following sample screenshot demonstrates column name match based JOIN suggestions:



Note that both types of JOIN suggestions can appear in the same popup list. Purple color suggestions indicate referential constraints; gray color suggestions indicate column name matches.

Tips:

- If a referential constraint consists of multiple columns, all columns referenced in constraint appears on the same line. Note that is the line is long, some of the columns may appear cut and no visible, to display these columns you can increase width of JOINs popup dragging its right edge. See Working with SQL Assistant Popups topic for details on how to resize, and manipulate SQL Assistant's popups.

- Use can use the keyboard navigation keys to quickly expand and collapse tables suggested for a JOIN and show/hide their columns. Use Arrow Right key to expand the selected table level, and Arrow Right to collapse the level.

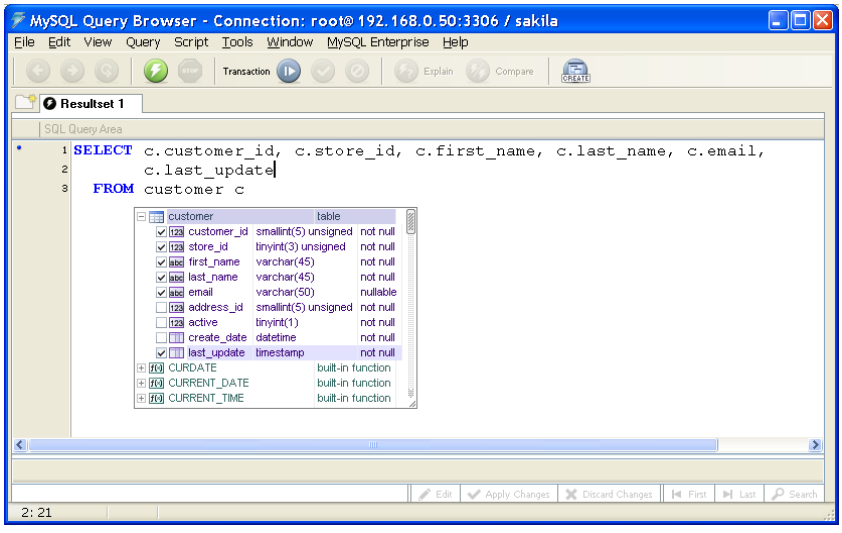
Using Multiple Columns Selection in DML Statements

When using SQL Assistant to generate DML statements such as SELECT, INSERT and UPDATE, you can choose which columns to include with the statement as soon as you type the first keyword. Here is how to use that feature. For example, you want to code a SELECT statement to retrieve data from several columns of the *customer* table. Use the following technique:

1. Type SELECT and press space bar. SQL Assistant objects popup will appear.

2. Select the *customers* table if you can see it right away or continue typing table name to narrow down the list until you see the *customers* table.

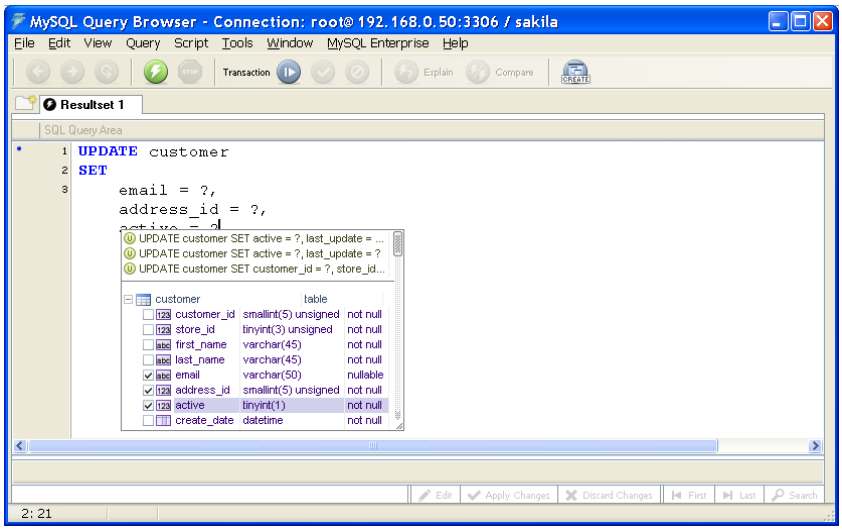
3. Click ⊞ plus sign in front of the *customers* table name to expand the table and display its columns. Alternatively, you can highlight the table with a keyboard and press Arrow Right navigation key to expand the *customers* table.



4. Click on checkboxes on front of columns you want to include with the SELECT statement. Alternatively, use Arrow Up and Arrow Down keyboard navigation keys to scroll the list and then use Arrow Right and Arrow Left keys to check or uncheck the required columns. As you select or unselect columns, you will see the SELECT statement for the current column selection automatically generated for you behind the SQL Assistant's popup.

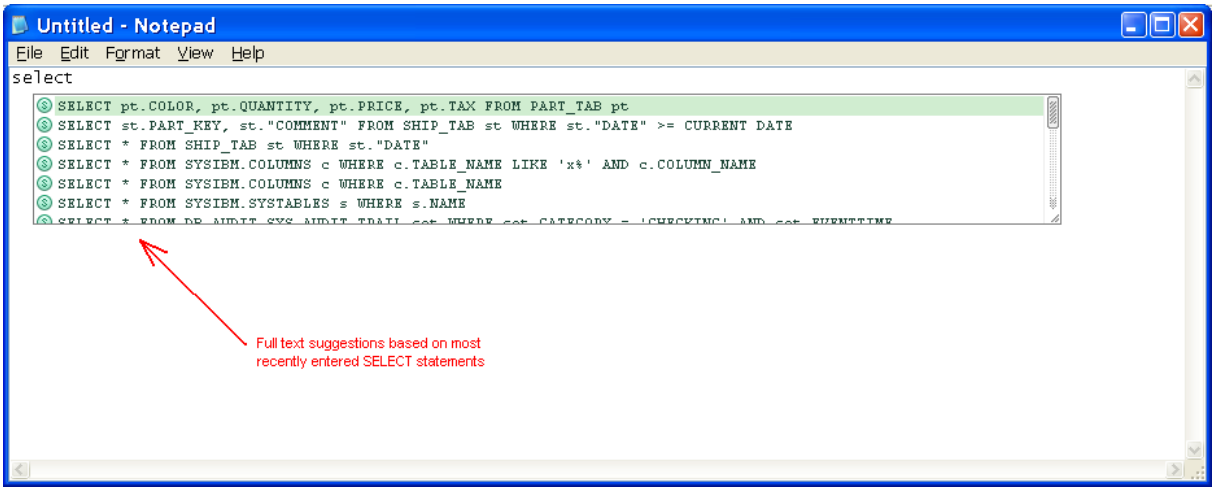
5. Press Esc or Enter keys to close the popup.

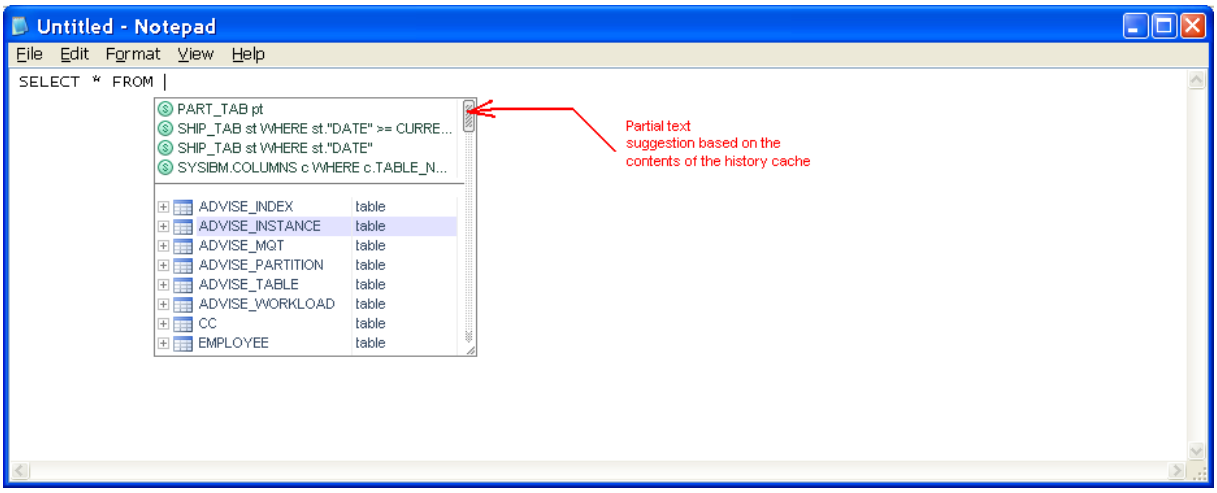
Same technique can be used for INSERT and UPDATE statements. You can see this on the sample screenshot below.



Using Context-based Suggestions Based on Historical Coding Patterns

SQL Assistant saves SQL DML statements code you have previously typed manually or built with the help of SQL Assistant's various code-suggestion and code-generation features. It saves this SQL code in the cache file and ranks frequency of each entered statement. When you type the code, it retrieves that same code for you if you begin to type the same text again later. It is also smart to identify most frequently typed parts of the code and suggest them during the typing. In essence, this feature is similar to the well-known AutoComplete feature, which is available in web browsers. Here in further in the documentation this feature will be called as **History Cache.**





Depending on the context, the code from the **History Cache** can be suggested exactly as it was entered or partially, based on the matching text patterns. On the sample screenshots above you can see both full text and partial text suggestions based on the text of previously entered SQL SELECT queries that are stored in the **History Cache**.

The size of the cache can be controlled in SQL Assistant settings. See CHAPTER 13, Customizing SQL Assistant Behavior topic for more details. The default cache size is 32 Kbytes. SQL Assistant uses mix of Most Frequently Used (MFU),, Most Recently Used (MRU) and First In First Out (FIFO) rules to maintain cache data. When choosing the right cache size, evaluate your system performance and check how it affects SQL Assistant response times when you are typing new code. Large cache allows SQL Assistant to save more history data, and more history data allows it to improve SQL statement ranking and deliver better context-based

suggestions. Small cache allows SQL Assistant to process the data in the cache faster and this can increase its response times at the cost of less accuracy in suggestions and less choices.

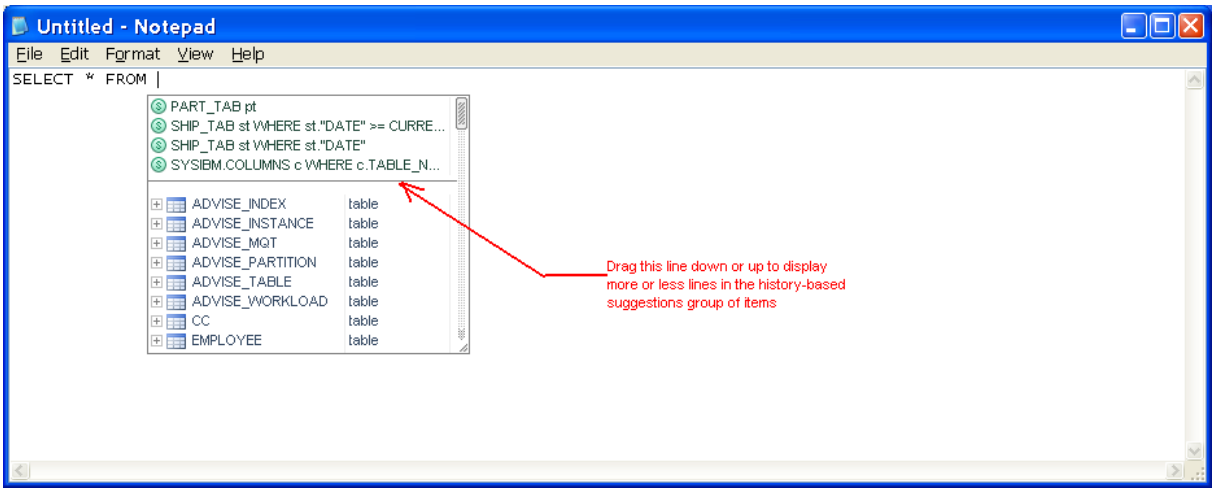
Another factor to consider when using code suggestions based on your historical coding patterns is which statements to show in the suggestion part of the SQL Assistant popups. The following choices are available:

- **Most Recently Used** – This is the default option. If selected, this option makes SQL Assistant to show suggestions matching the already entered text against most recently used SQL statements.
- **Most Frequently Used** – If selected, this option makes SQL Assistant to show suggestions matching the already entered text against most frequently used SQL statements.
- **No Sort** – If selected, this option makes SQL Assistant to show suggestions matching the already entered text against most records in the cache in the order they were saved or updated. The exact physical order is unknown and depends on your coding patterns.

The number of **History Cache** based suggestions appearing in SQL Assistant's popups is controlled by **History Items Shown in Lists** option that can be configure in SQL Assistant system options for type of assistance. The default value is 3 for all types.

If you would like to hide **History Cache** based suggestions in popups, set the value to 0 (zero).

When the popup is already displayed on the screen, you can dynamically adjust the number of displayed items by dragging the horizontal line separating history-based suggestions appearing in the popup.



The adjusted size will be effective for the lifetime of the target editor instance. To set the size permanently, use the available system options as described in the previous paragraphs.

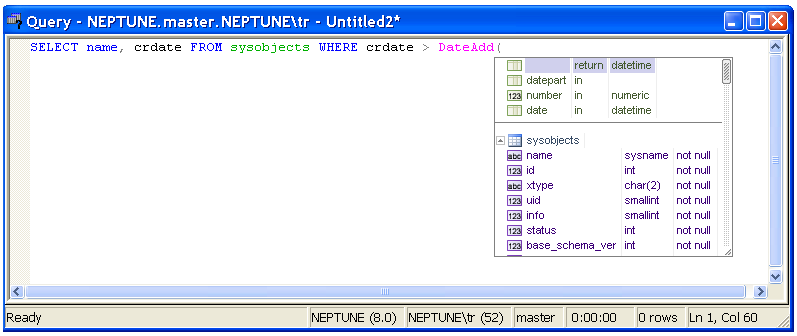
The position of the historical items within the popup is controlled by the **List Items** option group in the SQL Assistant system options, which you can use to organize which types of items appear in popups and in which order. Yet, the only usable position for the historical items is to appear at the top of the list or at the bottom of the list. The default is to appear at the top of the list

Using Function Argument Hints Features

The function argument list popup appears expanded automatically when SQL Assistant is invoked after an

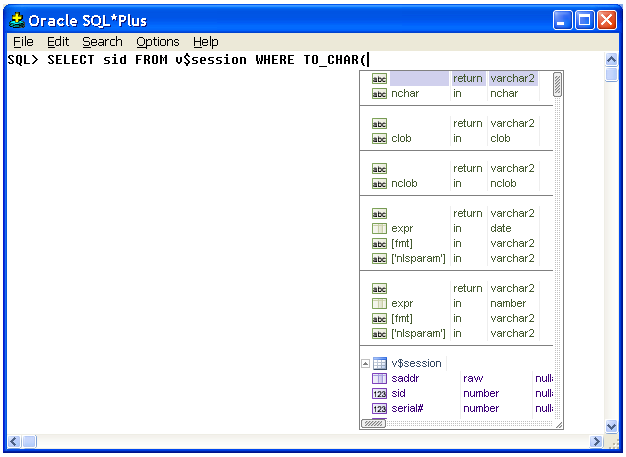
open-parenthesis character "(" or after comma within function arguments. The popup item list is normally limited to hints describing function arguments and the function return code, and also in a separate section, list of column names of tables and views referenced in the same SQL statement.

Below is an example popup demonstrating function argument completion feature.



Tips:

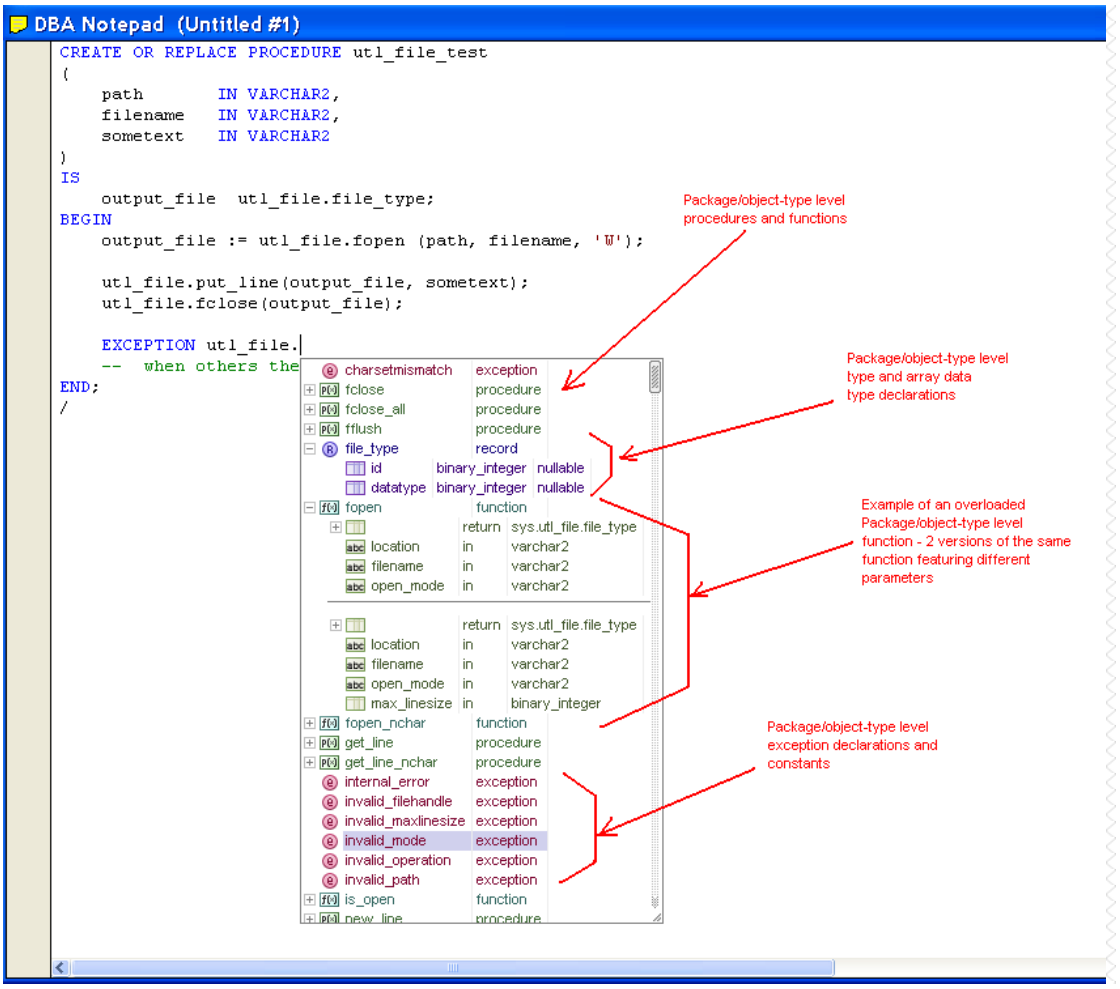
- Function argument names and the return code are displayed in a different color and provided as hints to help you enter values having correct data types and in the correct order. After inserting the text from the argument popup you should replace it with the actual values or valid script variables
- Optional arguments are displayed in [ ] brackets.
- In case multiple overloaded versions of the same function are available having different argument types or different number of arguments, each function version appears in a separate section as on the following sample screenshot:



Using Advanced Oracle Package and Object Type Attribute Completion Features

The advanced popup for Oracle packages and object types appears after package and type names. The popup contains lists of various attributes and functions available in the referenced objects, in other words provides graphical visibility for what is available in the referenced packages and types. Red text notes on the following

sample screenshot describe how to read the popup contents.



Tips:

- The popup list may contain multiple levels of information. Items prefixed with a ⊞ plus sign contain various sub-items such as object attributes, functions, function parameters and so on. To expand such sub-items click on the plus sign. Similarly to expand sub-items of sub-items, click on the plus sign displayed next to the sub-items
- If for whatever reason you don't want to paste or auto-generate any code when the popup appears after the initial SQL statement keyword, just continue typing the code normally. The popup will disappear automatically.
- In case multiple overloaded versions of the same function are available having different argument types or different number of arguments, each function version appears in a separate section with sections separated by a horizontal line

Using Local and Global Variable Names Completion Features

This feature is currently supported for T-SQL targets only. The variable name list popup appears expanded automatically when SQL Assistant is invoked after single @ character or after double @@ characters.

For local variables prefixed with a single @ characters, the popup item list includes all variables declared above the current line in the same unit of code such as a stored procedure, trigger, function, or other procedural object, in other words between the predecessor CREATE or ALTER command and the current position or between the last "go" batch delimiter and the current position.

For global variables prefixed with a double @@ characters, the popup item list includes all standard SQL Server global variables.

Just like any other SQL Assistant list, as you type the code, the context of the variable names list is automatically filtered to the matching items only.

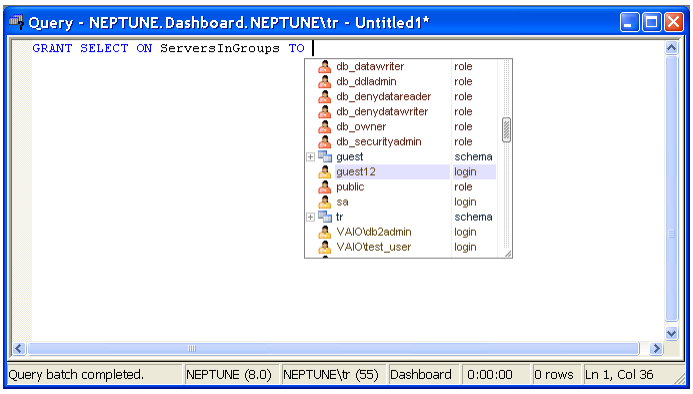
Using User/Role Names Completion Features

The user/role name list popup appears expanded automatically when SQL Assistant is invoked after GRANT, REVOKE and DENY commands. Type for example,

GRANT SELECT ON MyTable TO

SQL Assistant will display a list of users and roles to which you can grant the SELECT privilege. Note that SQL Assistant also automatically displays object list popup after ON keyword.

Below is an example popup demonstrating user name completion feature.



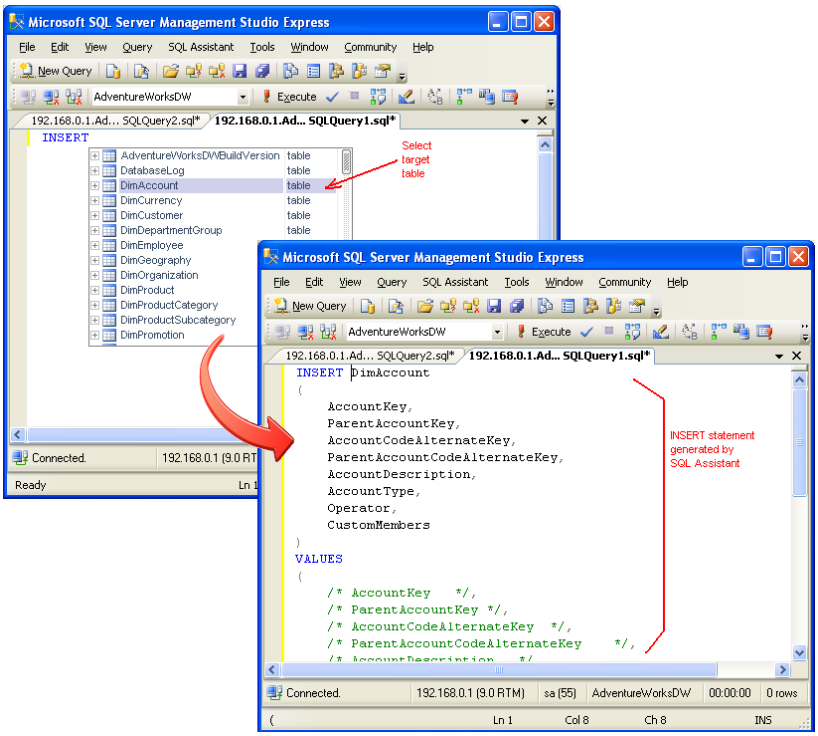
Using Code Auto-Expansion and Auto-Generation Features

Automatic Generation of DML Statements

SQL Assistant supports several handy code auto-generation features. Code auto-generation is context-based and triggered after you activate SQL Assistant is certain places of the SQL code. Several examples are provided in the following topics.

If you use SQL Assistant's popup displayed immediately after the SELECT keyword, and then pick a table or view or table function name in the popup list, SQL Assistant will automatically insert into the code the complete SELECT statement for the chosen object, adding object columns immediately after the SELECT keyword, and following by the FROM clause containing the selected object table or view. Similarly, if you choose an object in SQL Assistant's popup after the INSERT or UPDATE keyword, SQL Assistant will generate and automatically insert into the code the complete text of the INSERT or UPDATE statement.

The following example for SQL Server demonstrates the working of the automatic SQL statement code generation feature.



Tips:

- Hold down the SHIFT key while choosing an item in the SQL Assistant popup appearing after the initial SQL statement keyword to paste just the selected object name without generating additional code.
- If for whatever reason you don't want to paste or auto-generate any code when the popup appears after the initial SQL statement keyword, just continue typing the code normally. The popup will disappear automatically.
- You can also press the Esc key to dismiss the popup.

Automatic Generation of Variable Declarations

SQL Assistant can automatically generate variables for table, view or table function columns. This feature can be useful when coding cursors, batch selects and updates and similar SQL operations requiring declaration of a value holder variable for every column in a given object.

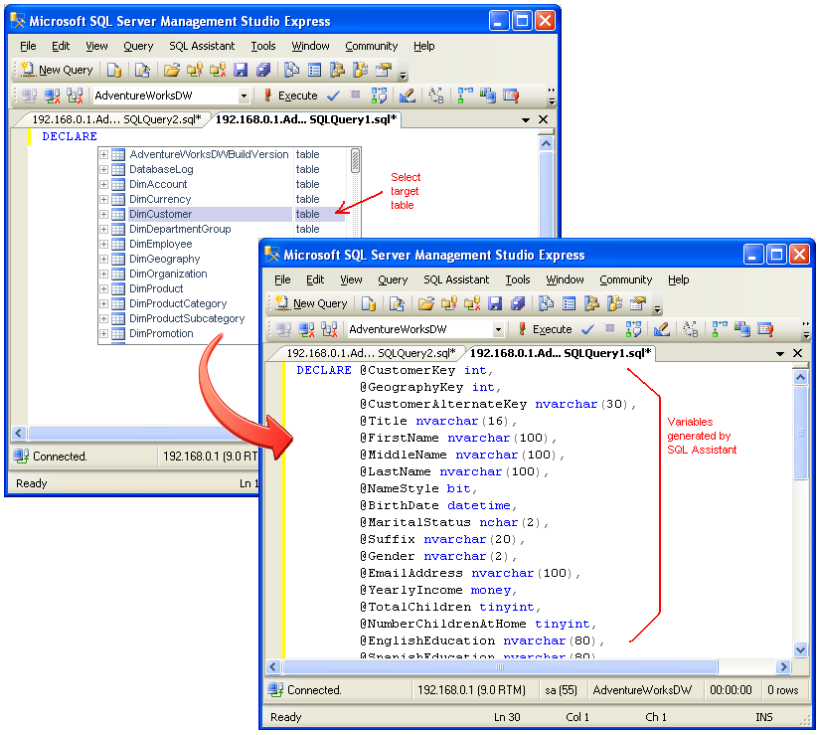
To use this feature:

1. Type DECLARE keyword then type space. SQL Assistant will display a list of objects in the database.

2. In the popup list choose the object that you want to target. SQL Assistant will automatically insert as many variable declarations into the code as there are columns in the chosen object.

Note: The names and data types of the declared variables match exactly names and data types of corresponding columns in the chosen database object.

The following example for SQL Server demonstrates the working of the automatic code generation feature

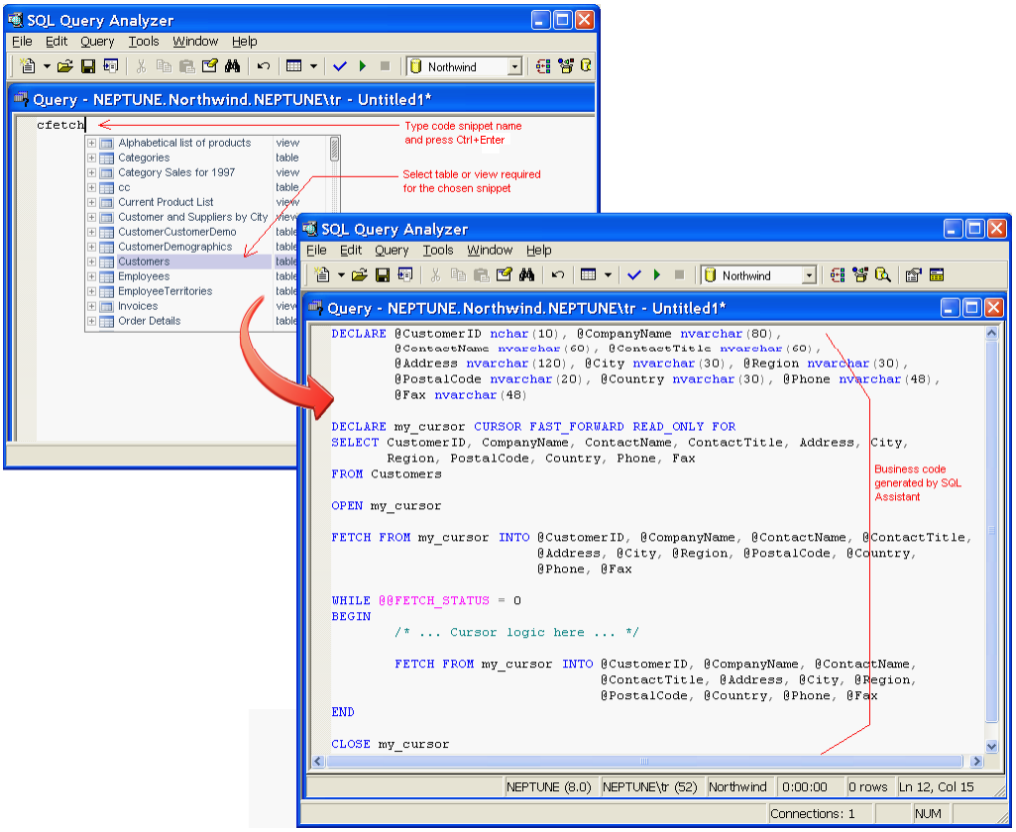


Tips:

- If for whatever reason you don't want to paste or auto-generate any code when the popup appears after the DECLARE keyword, just continue typing the code normally. The popup will disappear automatically.
- Hold down the SHIFT key while choosing an item in the SQL Assistant popup appearing after the DECLARE keyword to paste just the selected object name without generating additional code.
- You can also press the Esc key to dismiss the popup.

Advanced Interactive Code Snippets

Using SQL Assistant's advanced code snippets features, you can automatically generate entire code blocks with complete procedural and business logic based on the popup selections. Interactive code snippets can take your object selection and for the selected objects obtain from the database its attributes, parameters, columns and so on and then plug this information into snippet placeholders, For example, you can use the predefined *cfetch* snippet to generate complete block of code with table variable declarations, cursor declaration and other cursor loop and handling commands as on the following sample picture.



For more information for how to use advanced code snippets, see CHAPTER 6, Using and Creating Code Snippets for Fast Code Entry

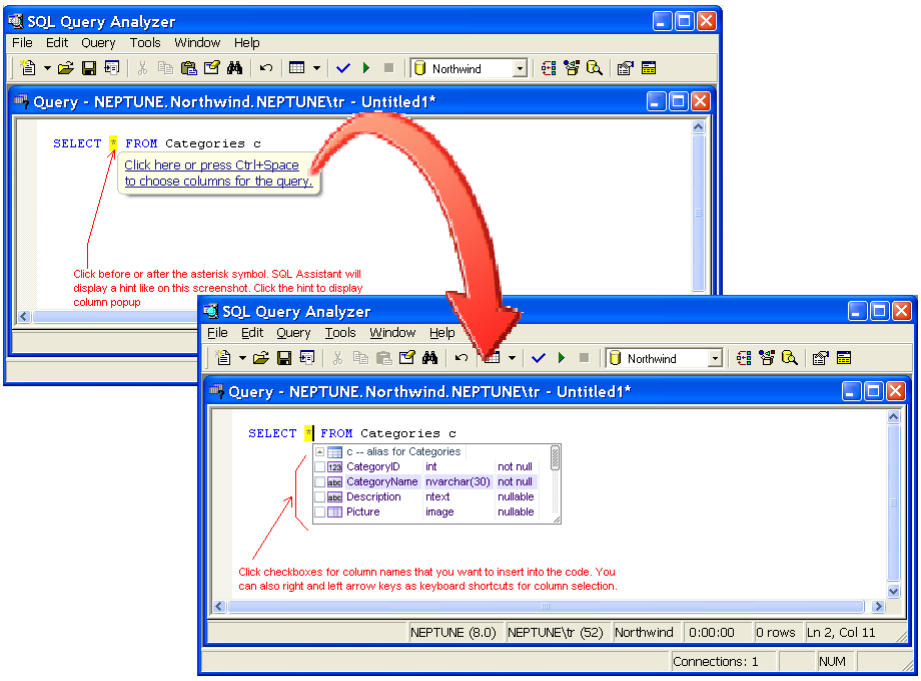
Advanced Code Expansion for * and Object Columns and Arguments

SQL Assistant constantly monitors SQL queries entered into the editor and senses SQL syntax elements that you might need to expand. For example, if you enter a code like

```
SELECT * FROM Categories
```

After you enter the above text, if you place the cursor near the asterisk symbol, SQL Assistant will highlight that symbol and display a hint. If you click on the hint or simply use the default Ctrl+Space hot key, you can get SQL Assistant to display a list of columns for the table referenced in the SELECT statement. You can then use the

column popup to choose columns that you want to use in place of the asterisk. The following sample image demonstrates how it works in the editor.



Tips:

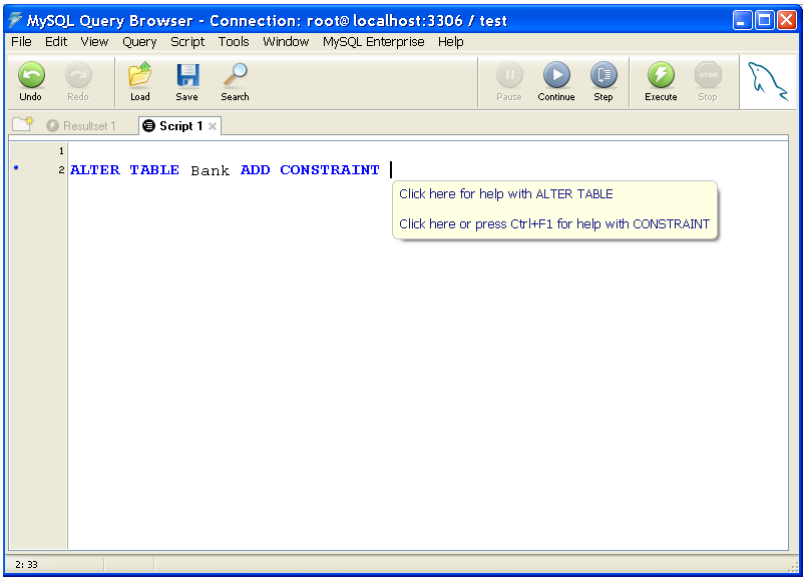
- Use the Arrow Right or Arrow Left navigation keys to select columns using the keyboard only. Use Arrow Up and Arrow Down keys to scroll the popup list.
- If you select an item in the popup and then unselect it, SQL Assistant will automatically remove it from the code.
- Use the Esc key to dismiss the popup at any time.
- Code Auto-Expansion is available for various SQL code elements, for example to expand the query and add additional columns to the SELECT clause, click after the table name in the FROM part for which you want to add additional columns and press Ctrl+Space, or wait for a second or two from the hint and then click on the hint.
- To get assistance with DDL commands like CREATE TABLE, DROP STATISTICS, ALTER TRIGGER and many others, click near the keywords of the statement for which you need help and then press Ctrl+F1, which is the default hot key for SQL Reference, or press whatever other hot key you have configured for SQL Reference in SQL Assistant options.

Advanced Code Expansion and Reference for DDL Commands

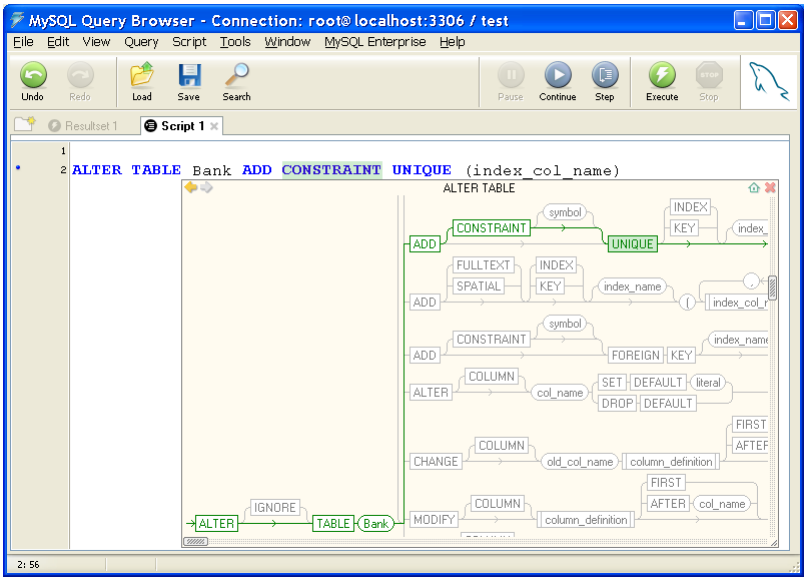
SQL Assistant constantly monitors SQL queries entered into the editor and senses DDL commands that you might need to get help with. For example, if you type a code like the following:

```
ALTER TABLE Categories ADD CONSTRAINT
```

After you enter the above text, if you pause for a couple of seconds, SQL Assistant will display a hint as demonstrated on the following screenshot



If you click the first hyperlink offering help with the ALTER TABLE syntax, SQL Assistant will popup context based SQL Reference window for ALTER TABLE command, which you can use to interactively build the required ALTER TABLE command.

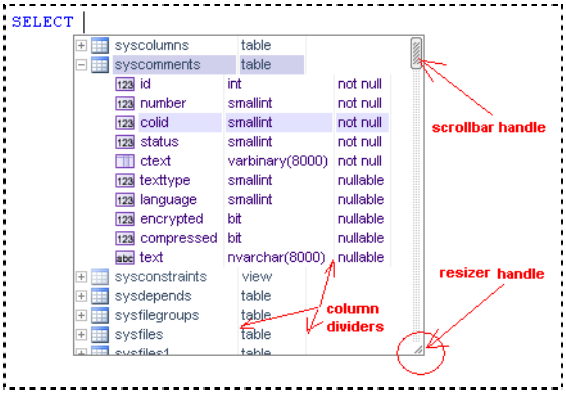


Similarly if you pick the second hyperlink offering help with the CONSTRAINT syntax, SQL Assistant will popup SQL Reference topic with description of CONSTRAINT related types and available options..

For details on how to use SQL Reference, see CHAPTER 7, Using Interactive SQL Reference System

Working with SQL Assistant Popups

A typical SQL Assistant popup looks like on the following screenshot.



The following topics describe how you can use the keyboard and mouse actions to work with the popup.

Navigation Keys

The following navigation keys are supported in the SQL Assistant popups:

Arrow Down – moves the logical selection to the next item. If pressed immediately after the popup appears on the screen, it selects the top item. If pressed while the last visible item is selected and there are more items available in the list, then it scrolls the contents down to the next item and selects it. Can be repeated until the last available item is reached.

Arrow Up – works just like the Arrow Down but in an opposite direction.

Page Down – scrolls the contents by one logical page and moves the logical selection to the top visible item. The logical page size is controlled by the popup list size and can be adjusted as described in the following Resizing Content topic.

Page Up – works just like the Page Down but in an opposite direction.

Arrow Left and **Arrow Right** – select an item in the current line and works with column and argument popups with multiple-choice options.

Tips:

- The most efficient way to use SQL Assistant popups is to start typing the item you want so that only items beginning with that text remain in the list and then using The Arrow Down key move the selection to required item and then hit the Enter key to paste the selected item text into the editor.
- You can also use the computer mouse to scroll the SQL Assistant popup contents and then click the item you want to paste into the editor.

Selection Keys

The SQL Assistant supports 2 item selection keys:

- Enter key is the default key, which is the standard key used in all Windows controls to select an item in a list and other multiple-choice selection control. This key also allows you to tab through the text while SQL Assistant popup remains displayed on the screen.
- Tab key is an alternative key, which is supported for compatibility with Microsoft development environments featuring so called Intellisense® technology. This key requires 2-hand typing – right hand for scrolling and selecting items in a list and left key for pressing the Tab key.

 **Tip:** Using SQL Assistant options you can customize which selection keys to use with SQL popups and other features. You can choose to use a single key such as Enter or Tab or you can choose to use both keys. See Customizing SQL Assistance Types topic for more information.

Scrolling Content

The contents of the SQL Assistant popup can be scrolled using either the keyboard navigation keys Arrow Up/Down and Page Up/Down or using the mouse. When using the mouse you can scroll the popup contents by dragging its scrollbar handles or clicking little arrows available at the extremes of scrollbars to scroll in small increments. See the sample screenshot with comments in the beginning of Working with SQL Assistant Popups topic for information on where to locate scrollbar handles.

Resizing Content

To resize the SQL Assistant popup, drag the resizer handle in the bottom-right corner of the popup window. See marked screenshot in the beginning of Working with SQL Assistant Popups topic for information on where to locate the resizer handle.

Resizing Individual Columns

Depending on the popup type several columns of text can be displayed in the item list. Moreover different parts of the list can have different number of columns and column width and positions. If the column width is insufficient and some parts of text appear cut, you can resize these columns so you can see the complete text. Note that thin gray vertical lines indicate column boundaries. To change size of a particular column, rest the mouse pointer over the vertical line indicating right boundary of that column. The mouse pointer should change to  . Press the left mouse button and while holding it pressed drag the line to the desired position. See marked screenshot in the beginning of Working with SQL Assistant Popups topic for information on where to locate lines indicating column boundaries.

Moving Content

In case SQL Assistant popup covers part of the editor screen that you want to see, click on an empty area

within the popup window, and while holding the left mouse button pressed, drag the popup window to the part of the screen where it is convenient for you.

Refreshing Content

For performance reasons SQL Assistance uses in-memory cache for catalog data retrieved from the database so that it does not need to query the database each time it needs to display the SQL Assistant popup. This internal cache is not updated automatically when changes occur in the database catalog data during active SQL Assistant sessions. For example, when new stored procedures or tables are created in the database or table columns are altered, the SQL Assistant is not aware of these changes and so they are not reflected in the contents of popups. Use either of the following methods to refresh the internal SQL Assistant cache:

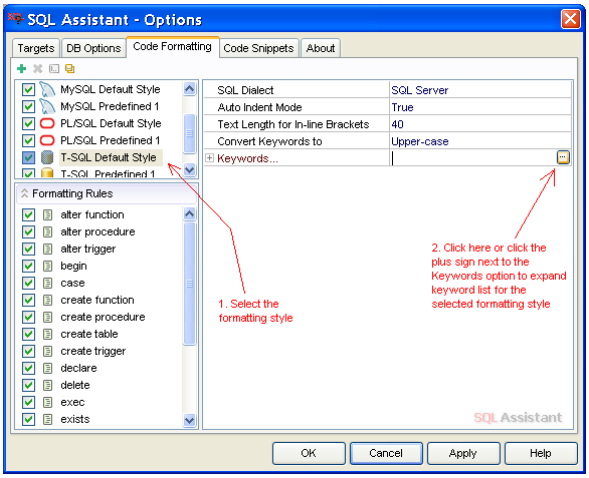
Method 1: Use **Refresh Cache** item in one of SQL Assistant menus. Example screenshots available in topic Manually Invoking SQL Assistant Popups in CHAPTER 3 demonstrate some of SQL Assistant menus.

Method 2: While the popup is displayed on the editor screen, press the **F5** hot key. Note that this Refresh method cannot be used if F5 key is also used as a hot key in the editor.

Method 3: While the popup is displayed on the editor screen, right click on the popup and then click the **Refresh** menu that appears on top of the popup window.

Using Keyword Capitalization and Formatting Feature

SQL Assistant can be configured to automatically format keywords as you type them. The set of keywords recognized by the program is controlled by the Code Formatting settings that you select in SQL Assistant options.



If the Keyword Capitalization feature is enabled, SQL Assistant automatically changes keywords that you type the case specified in the **Convert Keywords** rule. For example, if you type `alter table mytable` and the **Convert Keywords** rule is set to **Uppercase,** SQL Assistant automatically updates the entered text to `ALTER TABLE mytable`.

You can customize how to convert keywords and which keyword to convert in SQL Assistant Options. You can

also disable automatic keyword formatting feature. For more information, see Customizing Keywords Used With the Keyword Capitalization Feature topic in CHAPTER 13.

Using Smart Auto-Indent Feature

If the Auto-Indent option is enabled, pressing Enter key in the target editor will add a new line with the caret positioned on it, along with the indent which SQL Assistant assumes to be reasonable in the current code point. Note that the indentation is based on the code formatting patterns configured for the current SQL dialect. The indent, in other words, the amount of spacing in the new line, is calculated from the left side of the editor screen in accordance with the SQL code format patterns for the current SQL statement and relative to the indent of the previous line. For example, if you code an Oracle PL/SQL procedure and within BEGIN…END construct you type some text like in the following paragraphs:

```
BEGIN
    IF my_variable > 0 THEN
    END IF;
END;
```

and press Enter key after the THEN keyword, SQL Assistant will automatically insert a new line and will enter as many spaces (or tabs) as configured in the format patterns. The resulting code will look like the following:

```
BEGIN
    IF my_variable > 0 THEN
        |
    END IF;
END;
```

Here, the pipe sign represents new position of the editing caret. In comparison, if you press Enter key after a text line like "*my_variable := 55;"* the indent of the inserted line will be the same as the indent of the previous line.

```
BEGIN
    my_variable := 55;
    |
END;
```

 **Tip:** SQL Assistant also automatically un-indents closing brackets, END, END IF, END LOOP, LEAVE and some other compound SQL statement closing keywords so that the indent of these SQL code closing keywords matches the indent of the opening keywords;

See Customizing Code Formatting Patterns topic in CHAPTER 13 for details on where and how to change code formatting patterns.

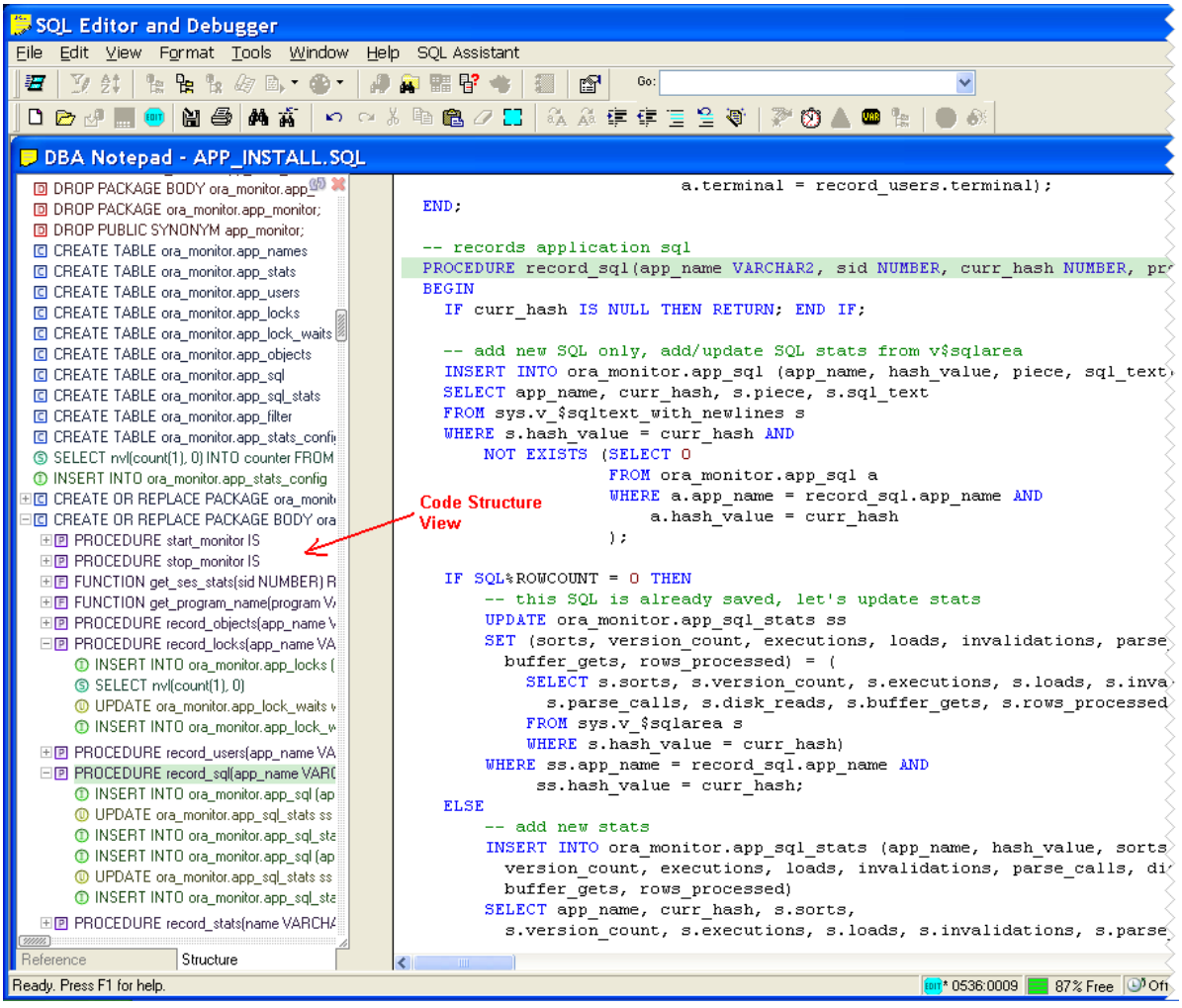
CHAPTER 4, Using Code Structure View and Code Bird's View

Overview of Code Strunсture View

The **Code Structure View** tool displays a structure of the code in a tree-like form adjacent to the left side of the target editor window.

The **Code Structure View** shows references to most important SQL commands in the code entered in target editor. The references are grouped together by procedures, loops and conditional statements and intended to represent their level in the code structure. The references make it easy to move quickly through code.

You can invoke the **Code Structure View** by clicking using Ctrl+F12 hot key. **Code Structure View** can be also invoked using SQL Assistant's menu available in the system tray or menus available in the target editor. To use target editor menus, the menu integration option must be enabled. For details, see Manually Invoking SQL Assistant Popups topic in CHAPTER 3.



The view can show references to many different types of SQL commands including but not limited to the following commands: CREATE, DROP, EXECUTE, SELECT, INSERT, DELETE, UPDATE, TRUNCATE, PRINT, USE, GRANT, REVOKE, and DENY.

 **Tips:**

1. For CREATE PROCEDURE, CREATE FUNCTION and CREATE PACKAGE commands, a plus sign is displayed next to the command. You can use that plus sign to expand the CREATE command and display the nested SQL commands which are part of the expanded procedure, function or package.

2. A plus sign is also displayed for various loop structures such as FOR loops, WHILE, UNTIL and generic LOOP loops and for conditional IF statements. Just like for CREATE commands you can click the plus sign to expand the nested SQL commands.

3. Different icons and colors are used for different types of SQL statements in order to improve visual appearance of the code structure and to allow users to locate the required SQL commands faster.

Working with Code Structure View Interface

Code Navigation

You can instantly navigate to any command in the target editor by clicking the appropriate hyperlink in the **Code Structure View** or by pressing the Enter key while the command's hyperlink is selected in the view. The target editor window will be scrolled as needed to make the selected SQL command visible on the screen and the caret will be set to the beginning of that command line.

Another helpful feature of the **Code Structure View** is dynamic code highlighting. When you move mouse pointer over command hyperlinks displayed in the Code Structure View, SQL Assistant automatically highlights referenced commands in the target editor if these commands are available in the visible frame of the target editor window.

Expanding / Collapsing Multiple Levels

The **Code Structure View** supports right-click context menu, which can be used to quickly expand or collapse all levels or certain levels only. Commands are available expand/collapse up to 4 levels or all levels at once.

Scrolling Content

To scroll the **Code Structure View** window contents, using the mouse drag window scrollbar handles as needed, or click little arrows available at the extremes of scrollbars to scroll in small increments. See marked screenshot in the beginning of Working with SQL Assistant Popups topic for information on where to locate scrollbar handles.

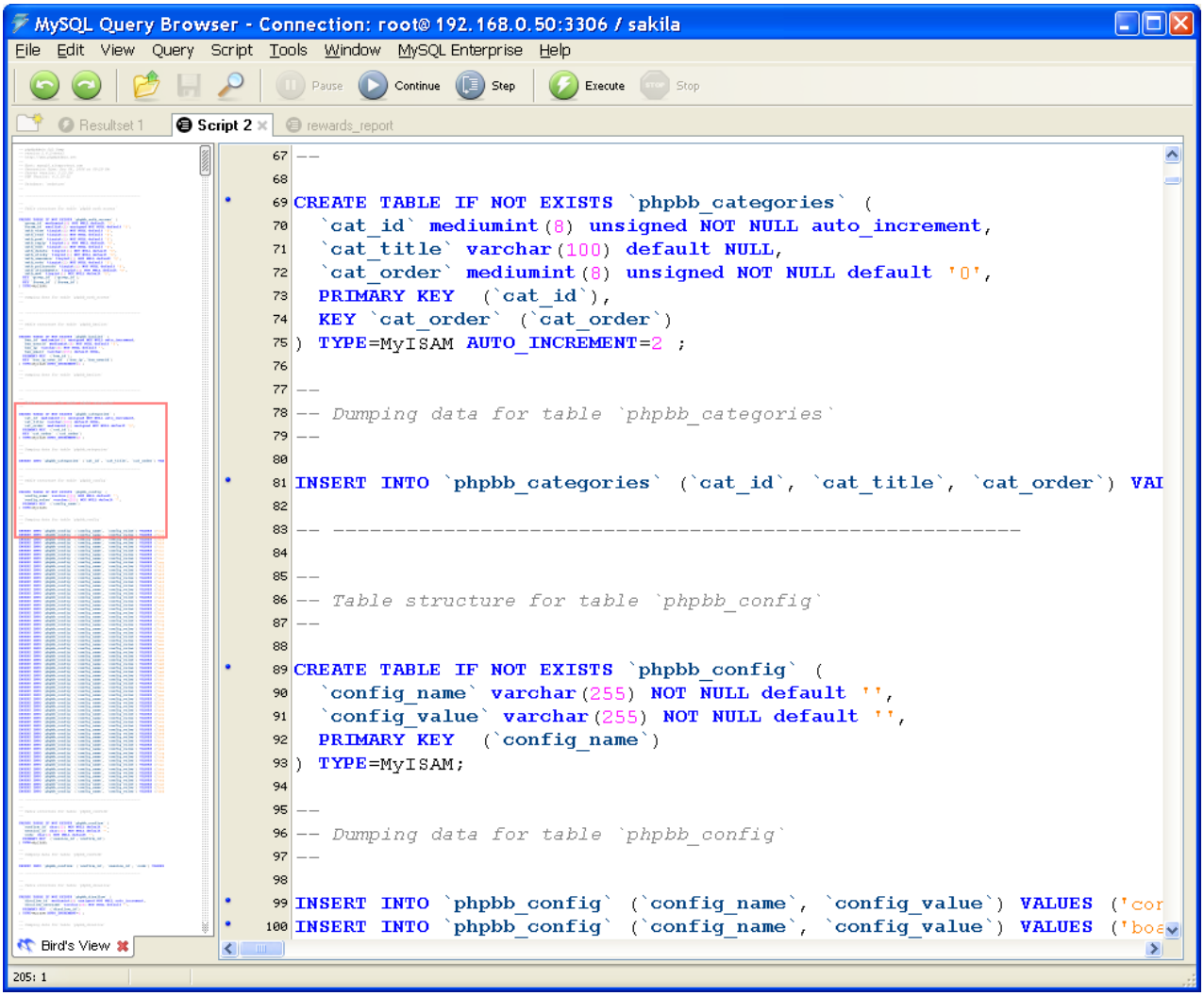
Resizing Content

To resize the **Code Structure View** window, drag the resizer handle in the bottom-right corner of the window. See marked screenshot in the beginning of Working with SQL Assistant Popups topic for information on where to locate the resizer handle.

Overview of Code Bird's View

The **Code Bird's View** tool displays a miniature view of the code loaded in the editor. The view appears adjacent to the left side of the target editor window. The sole purpose of the **Code Bird's View** is to easy the code navigation while working with large scripts.

You can invoke the **Code Bird's View** by clicking using Shift+F12 hot key. **Code Bird's View** can be also invoked using SQL Assistant's menu available in the system tray or menus available in the target editor. To use target editor menus, the menu integration option must be enabled. For details, see Manually Invoking SQL Assistant Popups topic in CHAPTER 3.



Using Partial Code Display vs. Full Code Display

Fully loading very large scripts into the **Code Bird's View** and creating scaled down page views may take a while especially on old slow computers. Note that each page view requires taking a separate screenshot. SQL Assistant uses several performance optimization techniques to load the pages faster. On opening of the view, SQL Assistant scans first 1000 lines of the script and automatically builds page views for the initially scanned lines only. Typically, in most editors opened full screen, first 1000 lines represent about 15 to 20 pages of code. If the entire script file is longer than 1000 lines, additional lines are scanned and page views are added to the **Code Bird's View** when you scroll through the file or enter new code only. This Partial Code Display method allows SQL Assistant to work efficiently and quickly update the **Code Bird's View** in the background**.**

If you would like to force scanning of the entire file right away, right-click the **Code Bird's View** area and from the context menu select the **Refresh** command. SQL Assistant will scroll the entire file and load all page snapshots into the **Code Bird's View**. This will provide you with a Full Code Display and allow fast navigation within any part of the loaded file.

Working with Code Bird's View Interface

Code Navigation

The viewed code area in the target editor is identified by a red rectangle displayed in the **Code Bird's View**.

The red rectangle can be dragged up and down, with an immediate change of the displayed part of code in the target editor. Dragging the box below or above the visible area of the view makes **Code Bird's View** automatically scroll the content in the same direction and automatically scroll code in the editor to match the position of the box.

You can instantly navigate to any page in the target editor by clicking that page in the **Code Bird's View.** The currently visible

In addition, you can use the scrollbars available in the **Code Bird's View** to scroll the content.

Refreshing Content

The **Code Bird's View** content is refreshed automatically as you make changes in the code and/or scroll code pages. However, in certain cases SQL Assistant might be unaware of code changes in the target and as a result will not refresh the view automatically. For example, this can happen if you open an existing file in the same editor window replacing the previous text. This can also happen if you do a global search and replace in the same editor window or multiple windows at once. These kinds of changes are performed mostly in the editor's memory without screen changes outside of the visible area. In such cases, you may want to invoke the **Code Bird's View** refresh manually not waiting for the screen updates so that the content you see in the **Code Bird's View** matches what you got in the file.

In case you need to force the refresh, right-click anywhere in the **Code Bird's View** to display the context menu. In the context menu, click the **Refresh** command.

 **Important Notes:** The forced Refresh will make SQL Assistant to rescan the entire file. If the file is very large, or you are running SQL Assistant on a slow computer, it may take more than a few seconds to rescan the file and create a snapshot of every page. That's why if the file is longer than 4000 lines, SQL Assistant displays a prompt asking to confirm the loading of the entire file. For more information, see Using Partial Code Display

vs. Full Code Display topic in this chapter.

Scaling the View

The **Code Bird's View** automatically scales the view based on the content type and length. In case you want to change the default text scaling, right-click anywhere in the **Code Bird's View** to display the context menu. In the context menu, click the **Scale** command to display the next menu level and then choose the desired scaling factor.

Note that the scaling factor is represented as a fraction of the original text size. For example, **x4** scaling factor means that the text in the **Code Bird's View** will appear 4 times smaller then the text displayed in the target editor. The actual text size is dependent on the font type and font size used in the editor and may vary for different environments and target types.

Scrolling Content

To scroll the **Code Bird's View** window contents, using the mouse drag window scrollbar handles as needed, or click little arrows available at the extremes of scrollbars to scroll in small increments. See marked screenshot in the beginning of Working with SQL Assistant Popups topic for information on where to locate scrollbar handles.

Note that by default SQL Assistant scrolls the contents the **Code Bird's View** automatically and synchronously with the scrolling of the target editor window. The red rectangle indicating the current page is also moved automatically to reflect the current position in the file. If you would like to freeze the automatic contents scrolling, right-click anywhere in the **Code Bird's View** to display the context menu. In the context menu, click the **Auto Scroll** command. To restore automatic scrolling, repeat the same operation.

 **Tips:**

- You may want to disable Auto Scroll in case you need to make a quick change in some part of the file and then return to the previous editing position. After making the change, click on the red rectangle in the **Code Bird's View** and SQL Assistant will return the editor to the previous position.
- The **Code Bird's View** freezing can be also used along with a small scaling factor as way to visually compare different parts of the code in the same file, having one part of the file displayed in the **Code Bird's View** area and another part in a different place of the same file displayed in the editor area.

Resizing Content

To resize the **Bird's View** window, drag the resizer handle in the bottom-right corner of the window. See marked screenshot in the beginning of Working with SQL Assistant Popups topic for information on where to locate the resizer handle.

CHAPTER 5, Using Code Formatter and Beautifier

Overview

In addition to the automatic keyword format-as-you-type feature described in Using Keyword Capitalization and Formatting Features topic, SQL Assistant provides the capabilities for formatting selected blocks of code, or all the code in the current file. If you highlight a portion of the code in the editor, for example, a single SQL statement or a group of SQL statements, and then invoke the SQL Assistant 's Code Formatter function, SQL Assistant formats the highlighted code only. If you invoke this function when nothing is highlighted, SQL Assistant formats the entire code available in the editor.

The code formatting is pattern-driven. The Code Formatter formats the code in accordance with the current preferences in SQL Assistant Options. See the following Setting Code Formatter Patterns topic for more details on the code formatting patterns and how to customize them.

There are several ways to invoke the feature:

- Use the default Ctrl+F11 hot key or a custom hot key, if you changed the default key.
- Use SQL Assistant's system tray icon menu (see Using System Tray Icon Menu topic for details)
- Use target editor's context or top-level menus, if menu integration option is enabled (see Using Context and Top-level Menus topic for details)

Important Notes:

Only the text in the editor is changed. If that text is associated with a file, the file is not updated until you use your editor's **File | Save** feature.

If for whatever reason you want to undo the formatting, use your editor's **Edit | Undo** features to undo the changes. If the entire text is reformatted, you may need to use the Undo feature twice, first to undo the insertion of the formatted code and again to retrospectively undo the editor reset command. This will restore text in the editor as it was before you invoked SQL Assistant's Code Formatter feature.

Code Format patterns are configured per SQL dialect and style. For example, different formatting patterns are used for Oracle PL/SQL and for MySQL. You may have virtually unlimited number of styles defined for the same dialect. For example, you can configure different styles for formatting SQL trace files and formatting stored procedures codes. See the following topic for more information.

Setting Text-wrapping Options

SQL Assistant's 's Code Formatter and Beautifier function supports the following two options for controlling explicit text wrapping:

- **Text length for in-line brackets** – this option controls treatment of sub-queries and expressions within brackets and allows keeping short expressions on a single line. In other words If the text within brackets is shorter than the specified maximum value, the entire contents of the text within brackets is kept on the same line. If the text is longer than the specified value, brackets are move

to new lines and the text within brackets is reformatted using regular formatting rules.
The following example demonstrates how this parameter affects code wrapping. Note that wrapping could be also affected by the formatting patterns. This example uses the default settings.

```
SELECT id as customer_id, store_id, first_name, last_name,
        (SELECT max(id) FROM orders o WHERE o.cust_id = c.id) AS last_order_id,
        email, address_id, active,  create_date, last_update
FROM cust c;
```

If the described parameter is set to 60 characters, after formatting you should get the following result:

```
SELECT id as customer_id,
       store_id,
       first_name,
       last_name,
       (SELECT max(id) FROM orders o WHERE o.cust_id = c.id) AS last_order_id,
       email,
       address_id,
       active,
       create_date,
       last_update
FROM cust c;
```

Note that the entire sub-query appears on a single line. However, if the option is set to a shorter value, for example, 30 characters, the result would look different::

```
SELECT  id as customer_id,
       store_id,
       first_name,
       last_name,
       (
          SELECT max(id)
          FROM orders o
          WHERE o.cust_id = c.id
       ) AS last_order_id,
       email,
       address_id,
       active,
       create_date,
       last_update
FROM cust c;
```

- **Line length for text wrapping** – this option controls overall line wrapping and is used when formatting long lines of text, which are normally not wrapped. For example, you may have a WHERE condition in your query using a long object or column names line and/or complex expression, like in the below example:

```
...
WHERE long_table_name1.long_column_name1 = long_table_name2.long_column_name2
...
```

In case the above text length exceeds the specified option value, as a result of the formatting, the text will be wrapped to 2 lines. If it does not exceed the option value, it will be kept on a single line as in the above example.

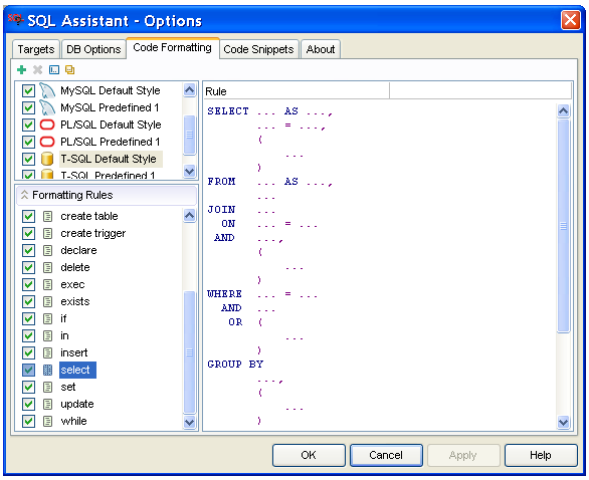
Setting Code Formatter Patterns

You can customize to a certain extent how the code is formatted. Use SQL Assistant Options to edit code-formatting patterns.

1. On the Options screen activate **Code Formatting** tab.

2. Choose SQL dialect and formatting style that you want to modify.

3. Click the type of SQL statement whose formatting pattern you want to modify. Below is a sample screen demonstrating formatting pattern for T-SQL SELECT statement



4. Edit the pattern as required. For more information on how to use and change SQL Assistant's options, see topic Customizing Code Formatting Patterns in CHAPTER 13.

Tips:

- Use formatting style management icons available in the left-top corner of the Options dialog to create new, rename, duplicate or delete formatting styles.

  Note that the icon functions are sensitive to the location of the focus in the Code Formatting tab. For example, if the focus is set to the left-top box containing formatting style names, and you click the X button, it will delete the selected formatting style entirely, including all associated formatting patterns. However, if the focus is set to left-bottom box containing pattern names and a SQL pattern name is highlighted, clicking the same button will delete the selected SQL pattern.

- The contents of the right side of the Code Formatting tab is also code sensitive. If a SQL pattern is selected, it will display the pattern definition. If the formatting style is selected, it will display properties of the formatting style.

- The default style for each supported SQL dialect and its formatting patterns are protected and may not be deleted. Yet, you can still change their definitions as you see the fit.

Commenting and Uncommenting Code Blocks

SQL Assistant code formatter provides two functions for commenting and uncommenting blocks of code easily and for nicely formatting the comment block.

To quickly comment a block of code:

1. Using mouse or keyboard highlight block of code that you want to comment out.
2. Right-click anywhere in the editor workspace and in the popup menu select **SQL Assistant** then **Comment Code** submenu. In that menu, click either **Comment with – –** command or **Comment with /* */** command.

To uncomment previous commented block of code:

1. Using mouse or keyboard highlight block of code that you want to comment out.
2. Right-click in the editor workspace and in the popup menu select **SQL Assistant** then **Comment Code** submenu. In that menu, click **Uncomment** command.

CHAPTER 6, Using and Creating Code Snippets for Fast Code Entry

Overview

SQL Assistant's Code Snippets feature provides a way for you to insert ready-made snippets of code into your SQL scripts. To insert a code snippet, type the shortcut name of the snippet and then press Ctrl+Enter hot key. If the snipped uses non-default hot key, press that hot key instead of Ctrl+Enter.

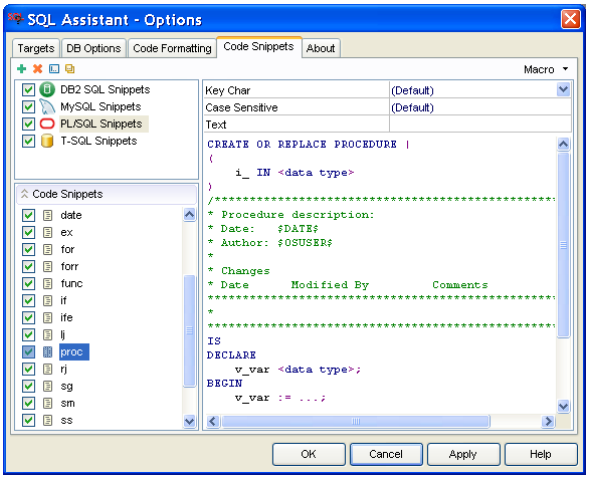
SQL Assistant comes with a number of ready to use code snippets. For example, if in a PL/SQL editor you type "for" without quotes and then press Ctrl+Enter default hotkey (or whatever you have selected as a hotkey for this particular code snippet), the following text will be inserted at the current caret position:

```
FOR i IN 0..| LOOP

END LOOP;
```

Use SQL Assistant Options dialog to customize pre-configured snippets and create new snippets.

1. On the Options screen activate **Code Snippets** tab.
2. Choose SQL dialect for which you want to configure code snippets
3. Select the required snippet. Below is a sample screenshot demonstrating code snippet options.



4. Edit the snippet code as required.

To disable a snippet, you can uncheck the box next to the snippet name. If a snippet is disable, its definition remains in the SQL Assistant options but the snippet is not active and cannot be used.

To delete a snippet complete, click on the snippet name and then press the Del key on the keyboard.

For more information on how to use and change SQL Assistant's options see CHAPTER 13.

 **Tip:** The vertical bar in the snippet code indicates where SQL Assistant will place the editing caret after the snippet text is inserted into the editor. Note that this predefined caret position does not work in buffered text entry editors like SQL*Plus where the caret is always placed at the end of the last line.

To create new and manage existing code snippets use SQL Assistant's Options dialog. It also possible to assign different hot keys to different code snippets. For more information see Customizing Existing and Creating New Code Snippets topic in CHAPTER 13.

Macro-variables and Dynamic Code Generation

Code snippets can contain certain macro-variables whose values are dynamically replaced when the snipped is inserted. There are two types of macro-variables: passive and active. Passive variables have easy to guess results. The following passive macro-variables are supported in all snippets:

Variable	Meaning
$DATE$	Current system date
$TIME$	Current system time
$LOGIN$	Login name for the current database connection
$USER$	Database user name for the current database connection
DB	Name of the current database (in the context of the current database connection)
$SERVER$	Name of the current database server (in the context of the current database connection)
$OSUSER$	Network name of the current user
$MACHINE$	Name of the computer where macro-variable is processed
SA_TARGET	SQL Assistant current target caption, for example, "SQL Server Enterprise Manager"
$SA_VERSION$	SQL Assistant version

For example, if you have a code snippet named "fun" having text like below:

```
CREATE OR REPLACE FUNCTION |
(
   v_in IN <data type>
)
RETURN <data type>

/*************************************************************
* Function description:
* Date:  $DATE$
* Author: $OSUSER$ connected as $LOGIN$
```

```
*
* Changes
* Date    Modified By         Comments
*************************************************************
*
*************************************************************/
IS
DECLARE
    v_ret <data type>;
BEGIN
   v_ret := ...;

   RETURN v_ret;
END;
```

If you type "fun" without quotes and then press Ctrl+Enter (or whatever you have selected as a hotkey for this code snippet), the text of the code snippet will be inserted at the current caret position and the $DATE$ macro-variable will be automatically substituted with the current system date, for example, *1/1/2007*; similarly the $OSUSER$ macro-variable will be substituted with your Windows login name, for example, *MyDomain\MyName and $LOGIN$* macro-variable will be substituted with your database login name, for example, *SomeLoginName.* The caret will be then placed at the point marked with the pipe | symbol so that you can type your function name immediately after the snipped code insertion.

 **Tip:** In Oracle, MySQL and DB2 targets, $LOGIN$ and $USER$ variables always render the same value.

In comparison, active macro-variables if referenced in a snippet code, will cause SQL Assistant to display an object selection popup and dynamically generate the contents based on the attributes of the selected object. The following active macro-variables are supported in all snippets:

Variable	Meaning
$OBJECT$	This will be replaced with the name of the selected object.
$COLUMNS$	This will be replaced with the comma-separated list of columns of the selected object. If there is more columns that fit on a single line the code will wrap and additional lines with columns will be added as needed This macro-variable can be used only with tables, views, and table functions.
$COLUMNS_V$	This is virtually the same as $COLUMNS$ macro, except that each column will be inserted on a separate line. The entire insertion will be a vertical comma-separated list of columns. Positions of commas and other text before and after each inserted column name is controlled by pre and post macro-text elements. See Tips section below for more details.
$COLUMNS+TYPES$	This will be replaced with the comma-separated list of columns of the selected object and their data types. This macro-variable can be used only with tables, views, and table functions.
$COLUMNS+TYPES_V$	This is virtually the same as $COLUMNS+TYPES$ macro, except that each column/type pair will be inserted on a separate line. The entire insertion will be a vertical comma-separated list of column/type pairs . Positions of commas and other text before and after each inserted column name is controlled by pre and post macro-text elements. See Tips section below for more details.

$ARGS$	This will be replaced with the comma-separated list of arguments of the selected object. This macro-variable can be used only stored procedures and user-defined functions.
$ARGS_V$	This is virtually the same as $ARGS$ macro, except that each argument will be inserted on a separate line. The entire insertion will be a vertical comma-separated list of arguments. Positions of commas and other text before and after each inserted argument name is controlled by pre and post macro-text elements. See Tips section below for more details.
$ARGS+TYPES$	This will be replaced with the comma-separated list of arguments of the selected object and their data types. This macro-variable can be used only stored procedures and user-defined functions.
$ARGS+TYPES_V$	This is virtually the same as $ARGS+TYPES$ macro, except that each argument/type pair will be inserted on a separate line. The entire insertion will be a vertical comma-separated list of arguments and their types. Positions of commas and other text before and after each inserted argument/type pair is controlled by pre and post macro-text elements. See Tips section below for more details.

For example, if you have a code snippet named "cfetch" having text like below

```
DECLARE v_$COLUMNS+TYPES$;

DECLARE my_cursor CURSOR FOR
SELECT $COLUMNS$
FROM $OBJECT$;

OPEN my_cursor;
fetch_loop:

LOOP
   FETCH FROM my_cursor INTO v_$COLUMNS$
   IF at_end <> 0 THEN
      LEAVE fetch_loop;
   END IF;

   /* ... Cursor logic here ... */

END LOOP fetch_loop;

CLOSE my_cursor;
```

If you type "cfetch" without quotes and then press Ctrl+Enter (or whatever you have selected as a hotkey for this code snippet), you will be presented with a prompt for an object name. If, for example, you select "Customers" table, the text of the code snippet will be inserted at the current caret position and the $OBJECT$ macro-variable will be automatically substituted with the selected table name. The columns of the selected "Customers" table and their data types, will be inserted as variable declarations along with the defined **v**_.prefix for variable names and the following DECLARE CURSOR statement will be generated using Customer's table columns and the results output to the generated SQL variables:

```
DECLARE v_CustomerID nchar(10), v_CompanyName nvarchar(80),
        v_ContactName nvarchar(60), v_ContactTitle nvarchar(60),
        v_Address nvarchar(120), v_City nvarchar(30), v_Region nvarchar(30),
        v_PostalCode nvarchar(20), v_Country nvarchar(30), v_Phone nvarchar(48),
```

```
        v_Fax nvarchar(48);

DECLARE my_cursor CURSOR FAST_FORWARD READ_ONLY FOR
SELECT CustomerID, CompanyName, ContactName, ContactTitle, Address, City,
       Region, PostalCode, Country, Phone, Fax
FROM Customers;

OPEN my_cursor;
fetch_loop:

LOOP
   FETCH FROM my_cursor INTO INTO v_CustomerID, v_CompanyName, v_ContactName,
                              v_ContactTitle, v_Address, v_City, v_Region,
                              v_PostalCode, v_Country, v_Phone, v_Fax;
   IF at_end <> 0 THEN
      LEAVE fetch_loop;
   END IF;

   /* ... Cursor logic here ... */

END LOOP fetch_loop;

CLOSE my_cursor;
```

Tips:

- The prefix text entered before or after active macro-variable is honored in all expanded elements. For example, if in SQL Server editor you use a snippet with the code like @$COLUMNS$, when expanded, every column will be prefixed with the @ sign effectively inserting variable names into the code.
- To insert macro-variable into the snippet code, you can type its name including $ sign delimiters or you can simply use the **Macro** button available in the top-right corner of the tab to paste the required macro-variable name.

Special Cases for Column/Variable and Argument/Value Pairs

To simplify programming of dynamic code generation, SQL Assistant supports special use cases for code snippets containing active macro-variables referenced on the same line and separated by equal sign. For example, consider predefined "si" snippet for T-SQL having the following definition:

```
DECLARE @$COLUMNS+TYPES$

SELECT
   @$COLUMNS$ = $COLUMNS$
FROM $OBJECT$
WHERE
```

If this snipped is invoked, the SELECT clause will be generated as a list of pairs of "variable name = column name" syntax tokens. If you type "si" without quotes and then press Ctrl+Enter (or whatever you have selected as a hotkey for this code snippet), you will be presented with a prompt for an object name. If, for example, you select "Customers" table, the text of the code snippet will be inserted at the current caret position and the $OBJECT$ macro-variable will be automatically substituted with the selected table name. The columns of the selected "Customers" table and their data types, will be inserted as variable declarations along with the specified @.prefix for variable names and the following SELECT statement will be generated using Customer's table columns and the results output to the generated SQL variables:

```
DECLARE @CustomerID nchar(10), @CompanyName nvarchar(80),
        @ContactName nvarchar(60), @ContactTitle nvarchar(60),
        @Address nvarchar(120), @City nvarchar(30), @Region nvarchar(30),
        @PostalCode nvarchar(20), @Country nvarchar(30), @Phone nvarchar(48),
        @Fax nvarchar(48)

SELECT
   @CustomerID = CustomerID, @CompanyName = CompanyName,
   @ContactName = ContactName, @ContactTitle = ContactTitle,
   @Address = Address, @City = City, @Region = Region,
   @PostalCode = PostalCode, @Country = Country, @Phone = Phone, @Fax = Fax
FROM Customers
WHERE
```

After this code is inserted into the target editor, the editing cursor will be positioned immediately after the WHERE keyword so that if you need a WHERE clause for this generated SELECT-INTO-VARIABLES query and you hit the spacebar key, SQL Assistant's column names popup will appear right away allowing you to quickly pick column or columns from the Customer table that you want to filter on.

Similar snippets can be used for other operations. SQL Assistant comes with several sample snippets for each SQL dialect. You can customize them as well as define your own active code snippets.

 **Note:**

Any text entered between active macro variable and the equal sign, will repeat in every generated value pair. It is not important if the text is adjacent to the variable name or it is space or tab separated. The following example shows what would happen if you define a test snippet like below:

```
EXEC $OBJECT$
   $ARG$ /* test suffix 1 */ = /* test suffix 2 */ $ARG$_var
```

When invoked, this test snippet will make SQL Assistant prompt you for the procedure name. Perhaps you pick "SalesByCategory" procedure in the sample "Northwind" database. The text of the resulting code may look like below:

```
EXEC SalesByCategory
   @CategoryName /* test suffix 1 */ = /* test suffix 2 */ @CategoryName_var,
   @OrdYear /* test suffix 1 */ = /* test suffix 2 */ @OrdYear_var
```

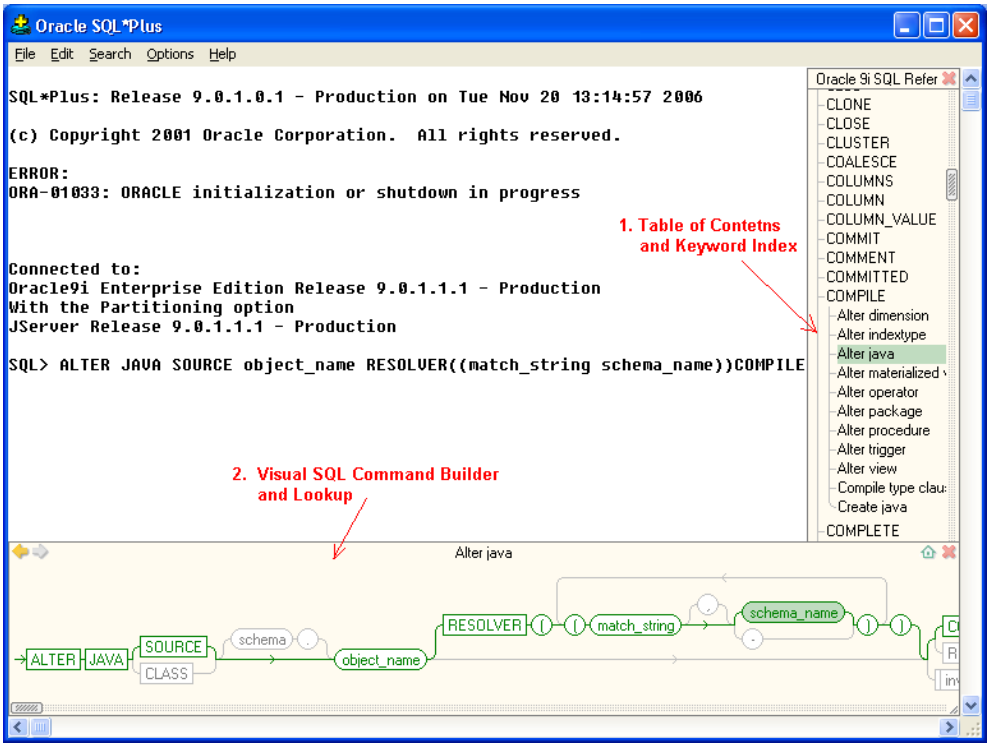
CHAPTER 7, Using Interactive SQL Reference System

Overview

SQL Assistant features a handy interactive SQL reference system. Different SQL Reference versions are provided for different versions of Oracle and SQL Server databases so that you can lookup and interactively build SQL commands fully compatible with your database version.

The SQL Reference consists of 2 main parts:

1. Table Of Contents, Statements and Keywords Indexes
2. Visual SQL Command Builder and Lookup



The following topics in this chapter describe how to invoke and use different parts of the SQL Reference system.

Invoking SQL Reference System

SQL Reference can be invoked at any time using the keyboard hot key assigned to it. The default hot key is

Ctrl+F1. If this key is already reserved for some function in your SQL editor or development environment, you can assign a different hot key in SQL Assistant options. For more information see Customizing Hot Keys topic in CHAPTER 13.

SQL Reference can be also invoked using SQL Assistant's menu available in the system tray or menus available in the target editor. To use target editor menus, the menu integration option must be enabled. For details, see Manually Invoking SQL Assistant Popups topic in CHAPTER 3.

Using SQL Reference Index and Table of Contents

SQL Reference Table of Contents may appear on the left or right hand side of the target editor window and the visual SQL Command Builder may appear at the bottom of the editor window or as a popup below the editing caret. The actual position depends on editor capabilities.

To select a specific SQL command, function or topic:

1. In SQL Reference Table Of Contents click hyperlink for the appropriate topic. If that topic contains subtopics, a list of hyperlink subtopics will appear under the selected topic name

2. In the Subtopics list select the required command or function. Visual SQL Command Builder window will appear at the bottom of the editor screen. You can use this window to review the syntax or you can use it to interactively build the required command and paste it into the editor

 **Tip:** If you are not sure in which topic or subtopic to find the required command or function you can use either Statements or Keywords Indexes to quickly locate the required text. These 2 Indexes are available as the last 2 items below the Table Of Contents Click on either of these items to expand the list and then in the expanded list locate the required statement or function.

Using SQL Commands Syntax and Functions Lookup

To quickly check syntax of a known SQL command:

Method 1

1. Pres Ctrl+F1 to invoke SQL Reference window.

2. Click **Statement Index** hyperlink. The Index will expand.

3. In the expanded Index locate and click the command whose syntax you want to lookup. Visual SQL Command Builder and Lookup windows will appear at the bottom of the editor screen.

Method 2

1. Pres Ctrl+F1 to invoke SQL Reference window.

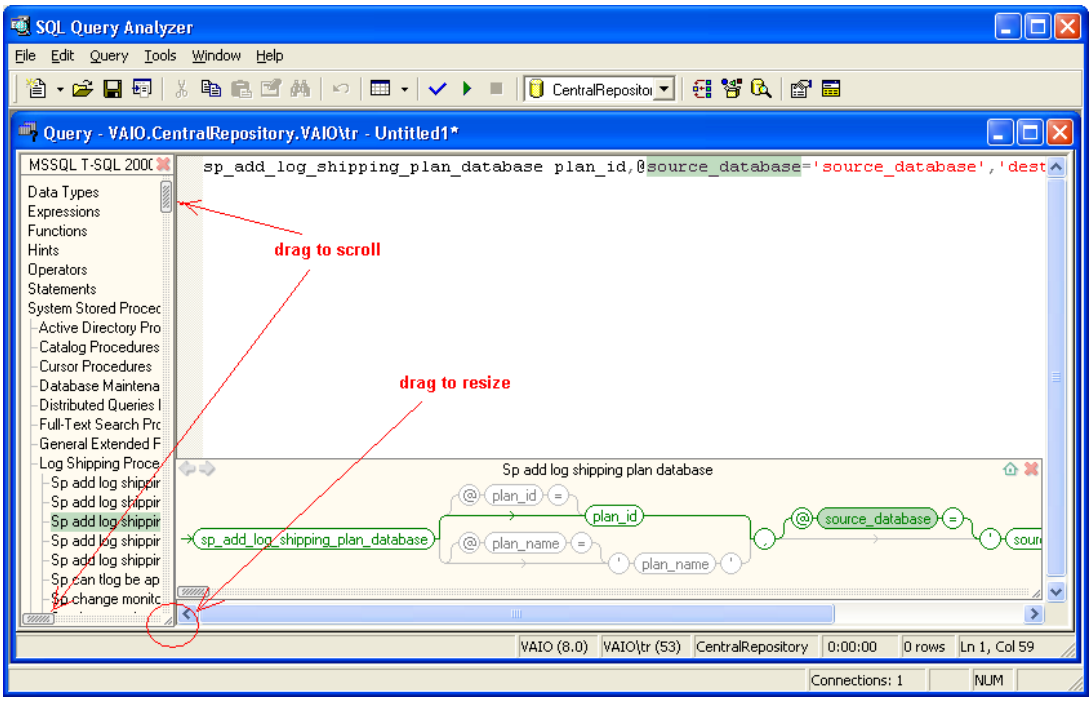
2. In that window type the command whose syntax you want to check. The SQL Reference will automatically locate that command in the Table of Contents and scroll it so the required commands appears on top of the visible list of items.

3. Click the command whose syntax you want to lookup. Visual SQL Command Builder and Lookup windows will appear at the bottom of the editor screen.

If you do not know the exact command or function name you can use hyperlinks in the Table of Contents to browse topic and subtopics. For example, if you are looking for Log Shipping Procedures in Microsoft SQL Server but not sure about their names and syntax:

1. Pres Ctrl+F1 to invoke SQL Reference window.

2. Click **System Stored Procedures** hyperlink. The Topic will expand



3. Locate the required procedure, for example, *sp_add_log_shipping_plan_database* and click on the hyperlink. The procedure syntax and parameters will appear I na new window on the bottom of the editor screen.

To scroll the **Table of Contents** window contents drag and its scrollbar handles as pointed on the picture above.

To resize the **Table of Contents** window drag its bottom-right corner as needed..

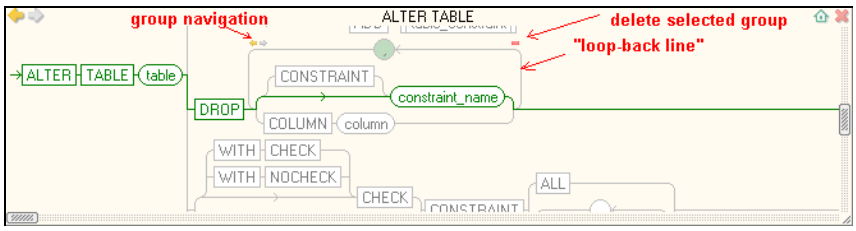
To pick a different version of the SQL Reference click the drop-down list displayed on top of the **Table of Contents** window and then select the required version.

Working with Visual SQL Command Builder Interface

The previous topic describes how to locate SQL commands and functions in the SQL Reference and how to open the **Visual SQL Command Builder** window. Once you have the **Visual SQL Command Builder** window open you can use it to graphically build the required SQL command or simply paste its syntax into the editor window.

Most SQL commands supports multiple option groups and alternative syntaxes. To paste syntax for any specific option group double-click the last syntax element in the group.

Optional syntax groups are displayed as parallel horizontal chains. If these syntax groups are non-exclusive and can be used together, a "loop-back line" is displayed around them.



To completely replace previously pasted syntax group simply click the last element of the required group.

To add another syntax group to already pasted text without replacing the previous group, click the circle with comma symbol displayed in a middle of the "loop-back line,"

If you want to add new syntax group in front of the previously pasted group use the little yellow arrows navigation links to navigate syntax groups and highlight their text in the editor.

To delete unneeded option group from the text in the editor, single click on that option name displayed in **Visual SQL Command Builder** window and then click the little red minus sign.

If a syntax option contains multiple sub-options, its name is displayed as an underlined hyperlink. If you click on such name the **Visual SQL Command Builder** window will refresh and show available sub-options.

Use yellow arrows in the top-left corner of the **Visual SQL Command Builder** window to navigate displayed screens.

To synchronize the current topic with the Table of Contents click the little home icon in the top-right corner of the **Visual SQL Command Builder** window.

Once you have the correct command syntax pasted edit non-syntax elements such as specific object names, column names, parameters and so on.

Scrolling Content

To scroll the **Visual SQL Command Builder** window contents, using the mouse drag window scrollbar handles as needed, or click little arrows available at the extremes of scrollbars to scroll in small increments. See marked screenshot in the beginning of Working with SQL Assistant Popups topic for information on where to locate scrollbar handles.

Resizing Content

To resize the **Visual SQL Command Builder** window, drag the resizer handle in the bottom-right corner of the window. See marked screenshot in the beginning of Working with SQL Assistant Popups topic for information on where to locate the resizer handle.

Moving Content

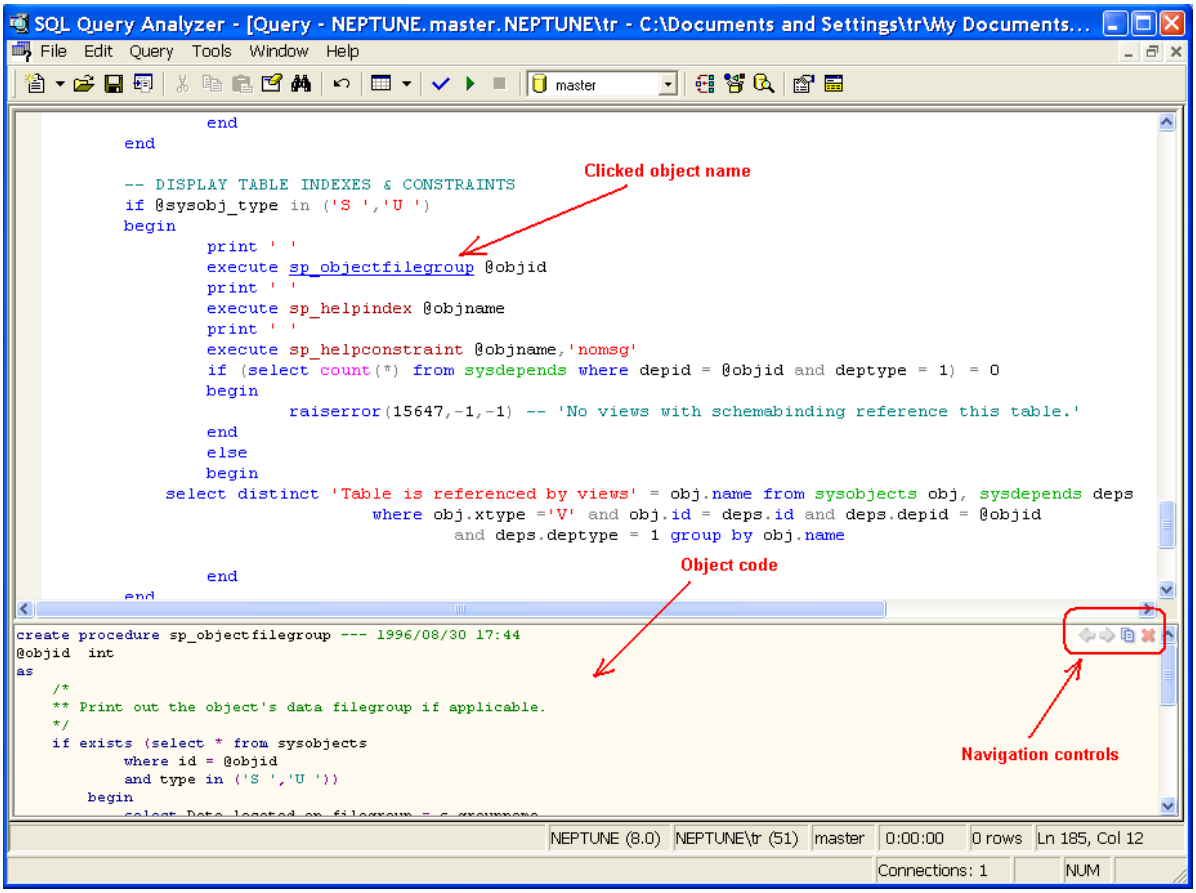
In case the **Visual SQL Command Builder** window covers part of the editor window that you want to see, click on an empty area within the **Visual SQL Command Builder** window, and while holding the left mouse button pressed, drag the window to the part of the screen where it is convenient for you.

CHAPTER 8, Using Procedural Code View

Overview

Any SQL developer who programs stored procedures or functions or simply anyone who writes SQL code using SELECT from a database view frequently needs to pick at the source code of the referenced object to learn the business logic encapsulated in that object. Typical methods to peek at the code involve using some kind of the database browser utility external or integrated with the editor. Such utilities require many mouse clicks or keystrokes to find and reach the needed object and then open the DDL of that object in a separate editor or save it to a file. This is a very time consuming process, especially when working with large systems containing many thousands of objects.

SQL Assistant supports an elegant single click method to lookup the required code without a need to go anywhere outside of the target editor window. This method is based on the so-called "hot mouse tracking" feature. Hot tracking is the visual effect whereby the text under the mouse pointer reacts to pointer movement and turns into a hyperlink. To activate that feature, hold down the Ctrl key and then move the mouse pointer over the name of the object whose source code you want to preview. The object name under the pointer will turn to a hyperlink. Click the hyperlink to display the Procedural Code View window with the source code of the clicked object.



SQL Assistant supports 2 additional methods for invoking Procedural Code View:

- Highlight the word or simply click on the words referring to a procedural object or view in the database then use **Target / Show Procedure Code** command in SQL Assistant's system tray icon menu (see Using System Tray Icon Menu topic for details)
- Highlight the word or simply click on the words referring to a procedural object or view in the database then use **SQL Assistant / Show Procedure Code** command in the target editor's context or top-level menus. This method is available only if the menu integration option is enabled (see Using Context and Top-level Menus topic for details)

 **Important Note:**

The **Show Procedure Code** menu can be used with procedural objects of different types including:

- Views – in all supported database systems. *See notes for Oracle below*.
- Stored procedures –in all supported database systems
- User defined functions –in all supported database systems
- Oracle packages – applicable to Oracle database systems only
- Oracle types – applicable to Oracle database systems only
- SQL Server triggers – applicable to SQL Server database systems only

 **Tip:**

In Oracle systems, view definition is stored as a LONG data type and because of that it cannot be used in regular catalog queries. The following technique can be used to add support for Oracle views to the **Procedural Code View**:

1. First create a new user-defined function in your Oracle database as in the example below:

```
CREATE OR REPLACE FUNCTION view_text(v_owner VARCHAR2, v_name VARCHAR2)
RETURN VARCHAR2
AS
    v_long LONG;
    v_len NUMBER;
    v_text VARCHAR(4000) := 'CREATE OR REPLACE VIEW ' || v_owner || '.'
                            || v_name || ' AS' || CHR(10);
BEGIN
    SELECT text, text_length INTO v_long, v_len
    FROM all_views
    WHERE owner = v_owner AND v_name = view_name;

    v_text := v_text || SUBSTR(v_long, 1, 4000 - LENGTH(v_text));
    IF v_len > 4000 THEN
        v_text := SUBSTR(v_text, 1, 3991) || CHR(10) || CHR(10) || 'more...';
    END IF;

    RETURN v_text;
END;
```

2. Double-click on SQL Assistant icon in the system tray to open the Options dialog and then in the DB Queries option group modify **DDL Code (Oracle)** query text adding at the end the following text:

```
UNION ALL
SELECT view_text(owner, view_name)
FROM all_views
WHERE owner = :OWNER AND view_name = :OBJECT
```

3. Close the Options dialog and wait several seconds for SQL Assistant to reload options in all open editors

and for the new settings to take effect. For more information on how to customize SQL Assistant database queries see Customizing Database Catalog Queries topic in CHAPTER 13.

Working with Code View Interface

Navigating Code Views

The previous topic describes how to invoke the **Procedural Code View** window when this view is not open. You can use the same methods to view code of a different procedural object while the Code View window is already open. Every time you invoke Code View, the view content is automatically refreshed with the source code of the newly requested object. The Code View window keeps track of requested objects and allows you to navigate their codes much like you navigate pages in a web browser. Use yellow arrows in the top-right corner of the **Procedural Code View** window to navigate displayed screens.

The navigation history is only available while the **Procedural Code View** window is open. If you close it and then invoke the **Procedural Code View** again, the previous navigation history gets lost.

Tip:

You can use the same invocation methods with the **Procedural Code View** window methods to lookup source code of the objects whose names are mentioned in the displayed code. This way you can drill-down from source code of a top level procedure to source other of objects called within that object code. The navigation buttons in the top-right corner can be used to go back to the source code of the previously viewed object.

Scrolling Content

Use standard scroll bars available in the **Code View** window to scroll the contents. In addition you can use regular keyboard navigation keys and the mouse wheel control if such is available.

Resizing Content

To resize the **Code View** window, drag the top edge of the window up or down. Note that when you place mouse pointer over the top edge of the **Code View** window the cursor shape changes to resize shape as on the following screenshot.

Make sure the cursor takes the right shape before dragging the window edge.

Copying Content

The **Code View** window supports 'copy to the Clipboard' function that can be used to copy the entire content or

highlighted portions of the content only. Use the  icon to copy the text. Note that if no text is highlighted in the **Procedural Code View,** the entire content is copied, otherwise only the highlighted text is copied to the Clipboard. The copied text can be then pasted into the target editor or any other program using standard **Edit / Paste** menu or Ctrl+V hot key.

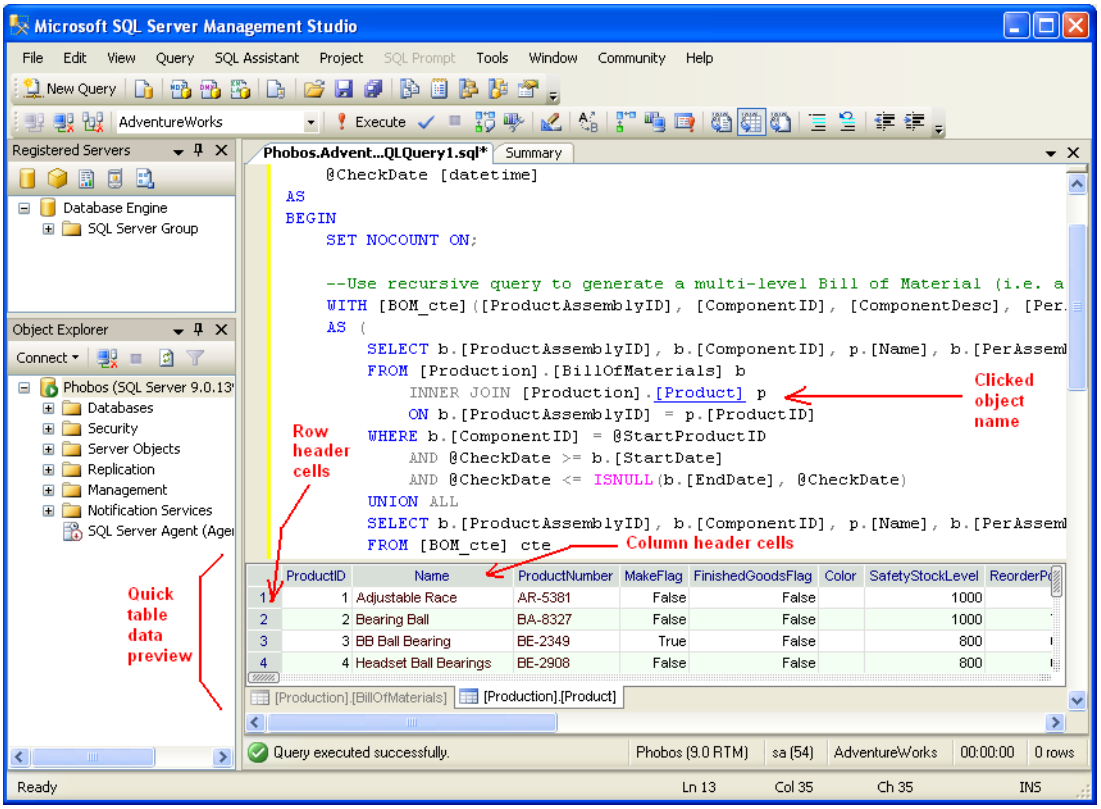
CHAPTER 9, Using Table Data Preview

Overview

Table Data Preview facility allows picking at the data of a table or view referenced in the code. Typical methods to peek at the data involve using some kind of an external application the database browser utility external or integrated with the editor. Such utilities require many mouse clicks or keystrokes to find and reach the needed object and then open the data of that object in a separate editor or save it to a file. This is a very time consuming process, especially when working with large systems containing many thousands of objects.

SQL Assistant supports an elegant single click method to lookup the required data without a need to go anywhere outside of the target editor window or to type and execute the complete SELECT statement. This method is based on the so-called "hot mouse tracking" feature. Hot tracking is the visual effect whereby the text under the mouse pointer reacts to pointer movement and turns into a hyperlink. To activate that feature, hold down the Ctrl key and then move the mouse pointer over the name of the object whose source code you want to preview. The object name under the pointer will turn to a hyperlink. Click the hyperlink to display the Table Data window with the source code of the clicked object.

Tip: For performance reasons, only **first 100 records** are displayed in the Table Data Preview. If you need the complete contents, right-click on the preview grid and choose **Retrieve All** command from the popup menu. In case you would like to get filtered data or data displayed ina certain order, use SQL Assistant facility to execute SQL queries.



SQL Assistant supports 2 additional methods for invoking Table Data Preview:

- Highlight the word or simply click on the words referring to a table or view object then use **Target / Show Table Data** command in SQL Assistant's system tray icon menu (see Using System Tray Icon Menu topic for details)
- Highlight the word or simply click on the words referring to a table or view objects then use **SQL Assistant / Show Table Data** command in the target editor's context or top-level menus. This method is available only if the menu integration option is enabled (see Using Context and Top-level Menus topic for details)

 **Important Note:**

The **Show Table Data** menu can be used with objects of different types including:

- Tables - in all supported database systems
- Views – in all supported database systems.
- Aliases – supported in Oracle and SQL Server for aliases referring to table and view objects
- Materialized Views – applicable to database systems supporting materialized views

Working with Table Data Preview Interface

Scrolling Content

Use standard scroll bars available in the **Table Data Preview** window to scroll the contents. In addition, you can use regular keyboard navigation keys and the mouse wheel control if such is available.

Resizing Content

To resize the **Table Data Preview** window, drag the top edge of the window up or down. Note that when you place mouse pointer over the top edge of the **Table Data Preview** window the cursor shape changes to resize shape as on the following screenshot.

Make sure the cursor takes the right shape before dragging the window edge.

To resize individual columns in the data grid, drag the right-edge of the column header left or right. Note that when you place mouse pointer over the right edge of a column header the cursor shape changes to resize shape as on the following screenshot.

Make sure the cursor takes the right shape before dragging the column edge.

Tip: When column width is too narrow to fit the contents, 3 dots (also called ellipses) are drawn in each cell with non fitting data to indicate the data overflow effect. To quickly resize a column so it fits the entire content in all cells, double-click click on the right-edge of the column header. SQL Assistant will calculate the required width and resize this column as needed.

Copying Content

The **Table Data Preview** window supports 'copy to the Clipboard' function that can be used to copy the entire content or highlighted portions of the content only. Use the right-click menu over the data grid to activate the Copy function or use the Ctrl+C keyboard shortcut. The copied text can be then pasted into the target editor or any other program using standard **Edit / Paste** menu or Ctrl+V hot key.

Tips: The data grid supports both regular and irregular data selection. To select various areas individually or in groups including:

- To select an entire row of data, click on the row header cell containing the row number.
- To select an entire column, click on the column header.
- To select the entire grid contents, click the top-left cell in the row headers column, or alternatively, right-click on the grid and then click **Select All** command from the right-click menu.
- To select an individual cell, just click on that cell.
- To quickly select several adjacent cells, use "mouse lasso" effect over these cells. Click on the starting cell and then while holding the mouse left button pressed drag it around cells that you want to select.
- To select any combinations of the above, for example, to select several columns, hold down the **Ctrl** key on the keyboard and then select the required areas using methods described above.

Exporting Content

The **Table Data Preview** window supports data export function that can be used to save the contents to an operation system file in either comma separated format (CSV) or tab-separated format (TXT).

1. Right-click the data grid to activate the context menu.
2. In the context menu choose **Save All As** command
3. Use standard system file browse dialog to choose destination directory and the name of the output file. Click the **Save** button on that dialog to save the data in the chosen file. If the file already exists, SQL Assistant will prompt your permission to overwrite the file.

Loading All Records

The **Table Data Preview** window by defaults shows only first 100 records from the invoked table. To make display the complete table contents, right click anywhere in the **Table Data Preview** window and choose

Retrieve All command from the right-click context menu./

CHAPTER 10, Executing SQL Queries

Overview

SQL Assistant's code execution facility provides several important features that might be missing in the target editor, including but not limited.

- Display of results sets returned by the database engine during code execution.
- Support for batch SQL scripts and multiple result-sets returned from a single batch.
- Automatic stripping of comments for database servers not supporting comments in individual SQL commands.
- Display of code execution timing.
- Separate displays of code execution statistics and result-sets.
- Display of each result-set on a separate tab page and retain it for later use, in other words do not recycle result windows after each code execution.
- Ability to export returned data to external files, including Excel.
- Retain previously returned results on screen
- Database code execution in target editors that do not support database connectivity or code execution functions, for example, Notepad, UltraEdit and other.

SQL Assistant supports batch code execution. It automatically parses the SQL script submitted for execution executed and breaks it into individual SQL batches or SQL statements depending on the type of the database it is connected to.

 **Important Note:** For T-SQL compatible databases, SQL Assistant uses "go" as a batch separator. If "go" is not found, it executes the entire script as a single batch. For all other databases it uses semicolon as a statement separator and automatically parses and breaks the script into compound and regular SQL statements executed individually one after another.

Working with SQL Code Execution Interface

Invoking SQL Code Execution Function

As with all other SQL Assistant functions, the **SQL Code Execute** function can be applied either to the entire file in the editor or just to the highlighted text. If no text is highlighted, SQL Assistant executes the entire file.

You can use either of the following methods to execute SQL code.

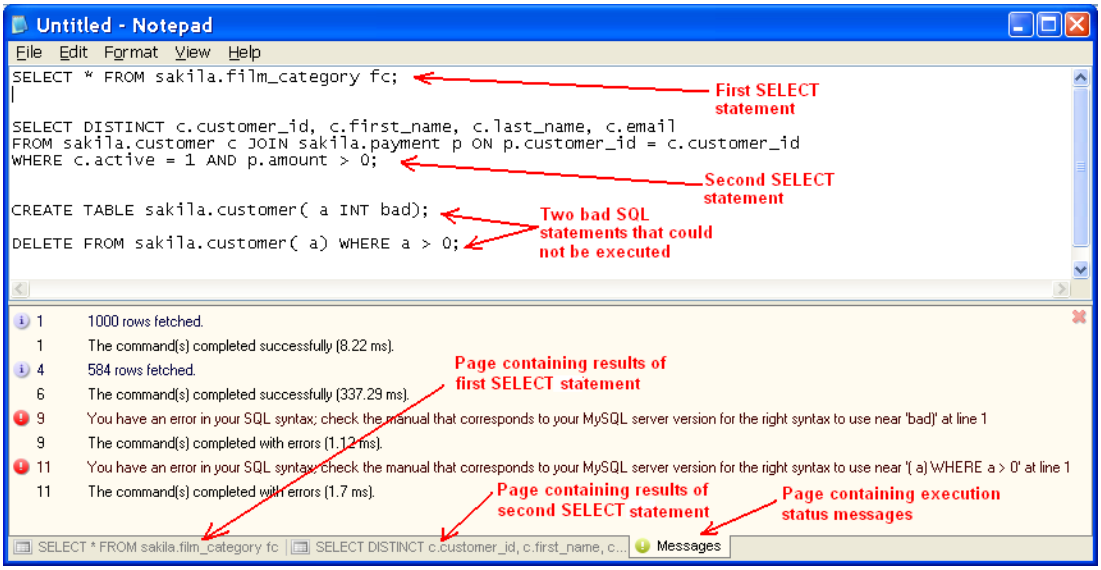
- Use the default Ctrl+ Shift + F9 hot key or a custom hot key if you changed the default key.
- Use SQL Assistant's system tray icon menu (see Using System Tray Icon Menu topic for details) and choose **Target / Execute SQL Code** command.
- Use target editor's context or top-level menus, if menu integration option is enabled (see Using

Context and Top-level Menus topic for details). In that menu select **SQL Assistant / Execute SQL Code** command.

The results of the code execution are displayed using one or multiple tab pages in the horizontally docked SQL Assistant window.

Reading and Understanding Code Execution Output

Errors and other messages returned by the database server during code execution are written to the **Messages** tab page along with the SQL Assistant's status messages. Normal database messages are displayed as dark blue text, error messages as dark red text and SQL Assistant's status messages as black text. Line Different icons



Line numbers and message type icons on the **Messages** tab page indicate type of each database message and the line in the original text where the error occurs. Note that line numbers are absolute and counted from the first line in target editor. That is it. If you highlight and execute a block of code in a middle of the script, the report lines will show line offset from the beginning of the script, not from the beginning of the highlighted text

In case the executed code returns result-sets, each retuned result-set is displayed on a separate tab page. The tab page name begins with the definition of the result-set query. The result-set is displayed in a grid control similar to the grid control used for the Table Data Preview feature

Scrolling Content

Use standard scroll bars available in the **Messages** and **Result-set** windows to scroll the contents. In addition, you can use regular keyboard navigation keys and the mouse wheel control if such is available.

Locating Errors

Another helpful feature of the **Messages** window is dynamic code highlighting. When you move mouse pointer

over messages displayed in the **Messages** window, SQL Assistant automatically highlights the referenced SQL statement in the target editor, if this statement is available in the visible frame of the target editor window. In case the referenced SQL statement is not visible on the screen, click on the error message and SQL Assistant will automatically scroll text in the target editor to make the referenced SQL statement appear on the screen. The cursor is automatically set in the beginning of the SQL statement.

Resizing Content

To resize the docked window containing output messages and returned result-sets window, drag the top edge of the window up or down. Note that when you place mouse pointer over the top edge of the window the cursor shape changes to resize shape as on the following screenshot.

Make sure the cursor takes the right shape before dragging the window edge.

For instructions on how to resize columns in grid controls see Resizing Content topic in CHAPTER 9, Using Table Data Preview.

CHAPTER 11, Using SQL Syntax Checker

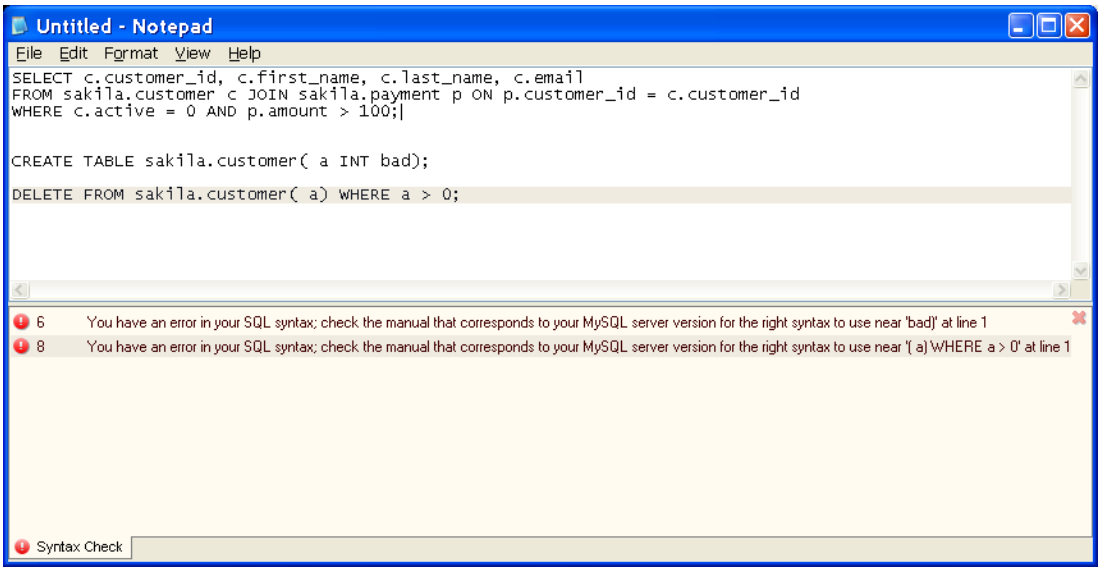
Overview

SQL Assistant's **SQL Syntax Checker** allows you to check for SQL syntax correctness without actually having to execute the code. Using this tool you can ensure your code will work when executed and if there are any syntax errors, correct them before the code execution.

Important Notes:

- The **SQL Syntax Checker** uses available syntax checking method provided by your database server software. Its your database software, not SQL Assistant, that validates the code and reports whether the code is syntactically correct or not.
- Because the database behavior and available methods differ for different database servers and event different database server versions, the result of the syntax check may vary in different environments.
- All error messages returned by the database are displayed in the **Syntax Check** tab. In case no syntax errors have been found, a simple **Syntax Check Completed** message is displayed.

The syntax checker outputs results of the syntax check to a separate window named **Syntax Check**. Each message in that window begins with a line number in the editor indicating where a bad SQL has been found and following by an error message text. In certain cases several different messages might be reported for the same bad SQL statement. As a result several lines with the same line number will appear in the Syntax check window.



Important Notes for Oracle Targets: Oracle's methods for parsing and checking syntax of DDL statements cause these statements to be executed immediately. You are advised to use SQL Syntax check selectively and apply it to DML statements only.

Working with SQL Syntax Checker Interface

Invoking SQL Syntax Checker

You can use either of the following methods to invoke the SQL syntax checker.

- Use the default Ctrl+Shift+F9 hot key or a custom hot key if you changed the default key.
- Use SQL Assistant's system tray icon menu (see Using System Tray Icon Menu topic for details) and choose **Target / Check SQL Syntax** command.
- Use target editor's context or top-level menus, if menu integration option is enabled (see Using Context and Top-level Menus topic for details). In that menu select **SQL Assistant / Check SQL Syntax** command.

 **Tip:** As with all other SQL Assistant functions, the **SQL Syntax Check** function can be applied either to the entire file in the editor or just to the highlighted text. If no text is highlighted, SQL Assistant validates the entire file.

The results of the syntax checking are displayed in the **Syntax Check** window

Scrolling Syntax Check Content

Use standard scroll bars available in the **Syntax Check** window to scroll the contents. In addition, you can use regular keyboard navigation keys and the mouse wheel control if such is available.

Locating Syntax Errors

Another helpful feature of the **Syntax Check** window is dynamic code highlighting. When you move mouse pointer over error messages displayed in the **Syntax Check** window, SQL Assistant automatically highlights the referenced SQL statement in the target editor, if this statement is available in the visible frame of the target editor window. In case the referenced SQL statement is not visible on the screen, click on the error message and SQL Assistant will automatically scroll text in the target editor to make the referenced SQL statement appear on the screen. The cursor is automatically set in the beginning of the SQL statement.

Resizing Content

To resize the **Syntax Check** window, drag the top edge of the window up or down. Note that when you place mouse pointer over the top edge of the **Syntax Check** window the cursor shape changes to resize shape as on the following screenshot.

Make sure the cursor takes the right shape before dragging the window edge.

CHAPTER 12, Using Spell Checker

Overview

SQL Assistant's **Spell Checker** can seamlessly integrate Microsoft Word's multilingual in-line proofreading functions into your SQL editor. The Spell Checker is an intelligent tool that selectively checks for spelling errors in relevant parts of the SQL code. Because SQL scripts are special documents containing special SQL instructions, object references and object attribute references, as well as other programming constructs, checking the entire content of the script is really meaningless. That is why SQL Assistant parses the content first and applies spell check functions only to comments and to hard-coded string values.

SQL Assistant provides 2 different ways to check spelling:

- **On-demand Spell Checker** – the interactive spell checker is activated manually using hot keys or context menus. The spell checker scans the highlighted text and provides spelling suggestions as required.
- **Real-time Spell Checker** – this non-interactive spell checker runs in the background all the time. As you type code, the spell checker checks the text, and then marks possible errors with red wavy underlines. You can then use the context menus to display suggested changed for marked words.

 **Important Notes:** The Spell Checker requires that you have Microsoft Word 97 or later installed on the system. The Spell Checker can only use installed proofreading tools and languages. To check spelling in other languages, use the Microsoft Office installation CD to install additional language support.

 **Tip:** If you wish to check the correctness of your SQL code syntax rather then the spelling, use the **SQL Syntax Checker** function. This function is described in Using SQL Syntax Checker topic in CHAPTER 3.

Using On-demand Spell Checker

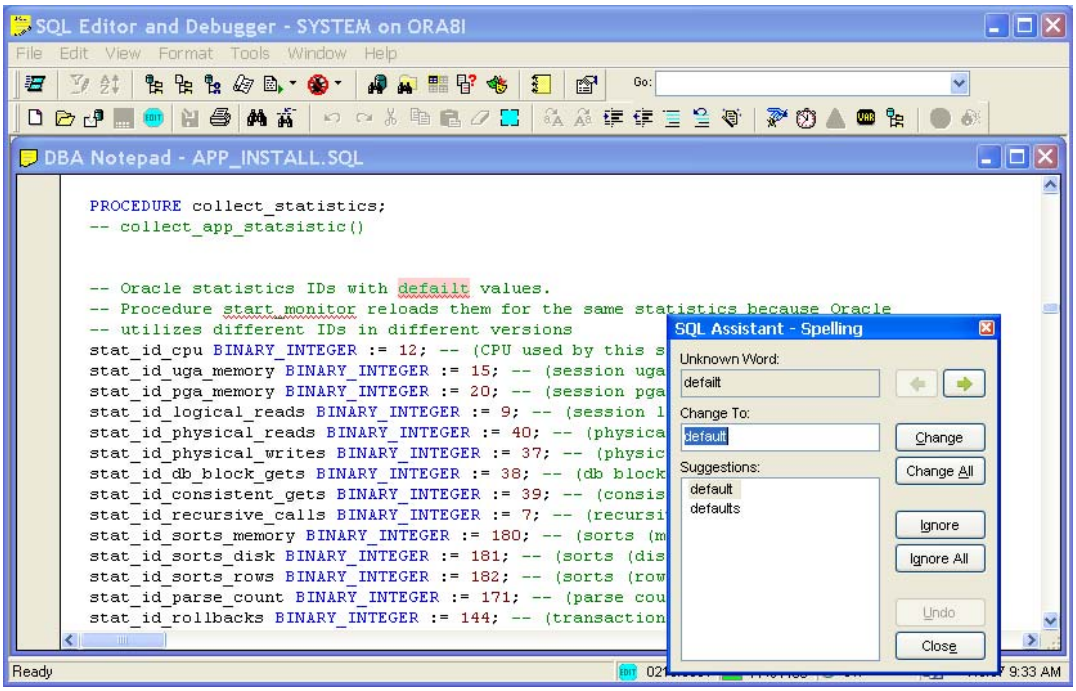
The interactive on-demand spell checker provides straightforward interface that mimics spell check functions available in many office and email programs. When it finds a possible spelling error, it displays the **SQL Assistant – Spelling** dialog window containing list of suggested corrections and also controls you can use to navigate, modify and ignore errors.

As most other SQL Assistant functions, the spell check can be run for the highlighted portion of the text only or if nothing is highlighted, for the entire content of the target editor all at once. Typically, to run a spell check in SQL code, you should highlight the relevant portion of the text and then activate the spell checker. Then use the **SQL Assistant – Spelling** dialog to make the necessary corrections. Note that the dialog disappears automatically when you reach the last error in the text or you can close it at any time using the Esc key or the Close button.

You can use either of the following methods to invoke the interactive spell checker.

- Use the default Ctrl+Shift+F7 hot key or a custom hot key if you changed the default key.
- Use SQL Assistant's system tray icon menu (see Using System Tray Icon Menu topic for details) and choose **Target / Check Spelling / Check** command.
- Use target editor's context or top-level menus, if menu integration option is enabled (see Using Context and Top-level Menus topic for details). In that menu select **SQL Assistant / Check Spelling / Check** command.

The following sample screenshot demonstrates the interactive on-demand spell checker used with "DB Tools for Oracle" target editor.



The following functions and controls are available in the spell checker dialog:

Unknown Word – this is the possible misspelled word that the spell checker suggests to modify.

Change To – in this editor box you can type new text that you want to use as a replacement for the incorrectly spelled word.

Arrow Left and Right – this controls can be used navigate between misspelled words.

Change – this button updates the misspelled word that SQL Assistant has highlighted in the target editor with the text in the **Change To** edit box.

Change All – this button updates all occurrences of the misspelled word in the editor with the text in the **Change To** edit box. You can use this function if the same spelling error repeats multiple-time.

Ignore – this button makes the spell checker to skip the highlighted word and move on to the next error.

Ignore All – this buttons makes the spell checker to skip all occurrences of the highlighted word and don't check their spelling. Note that this selection is not persistent and applies only to the current spell check session.

Undo – this button undoes the last change. The button can be used repeatable to recursively undo previous changes. note that only changes made in the current spell check session can be undone. To undo older changes use your editors **Edit / Undo** features.

Close – this button closes the spell checker dialog and ends the spell check session. It can be used at any time. Alternatively you can use the Esc hot key to dismiss the spell checker dialog

Important Notes:

Only the text in the editor is changed. If that text is associated with a file, the file is not updated until you use your editor's **File Save** feature.

If for whatever reason you want to undo the changes, use your editor's Edit | Undo features to undo the changes. In case multiple spelling changes have been made, you may need to use the Undo feature multiple times, to undo each change.

Using on Real-time Spell Checker

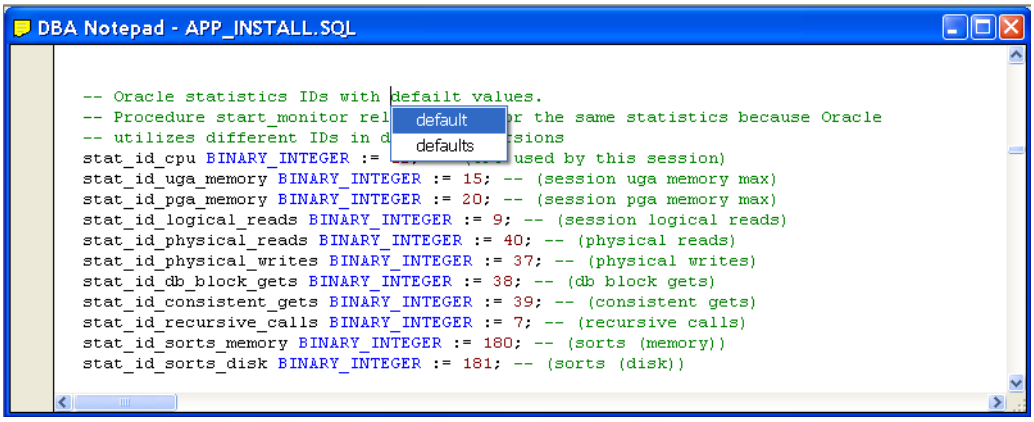
The **Real-time Spell Checker** is non-interactive. If activated, the spell checker quietly runs in the background and as you type new code or modify already entered code it marks possible errors with red wavy underlines. The spell checker only checks text in comments and in hard-coded string values, in other words, values enclosed in single quotes.

If you wish to correct errors, you can simply edit the misspelled words directly in the editor. If the new text is correct, the wavy underlines disappear automatically.

If you would like to see a list of spelling suggestions for a certain word, highlight the word for which you need help and then use either of the following methods to invoke the spelling suggestions popup list.

- Use SQL Assistant's system tray icon menu (see Using System Tray Icon Menu topic for details) and choose **Target / Check Spelling / Suggestions** command.
- Use target editor's context or top-level menus, if menu integration option is enabled (see Using Context and Top-level Menus topic for details). In that menu select **SQL Assistant / Check Spelling / Suggestions** command.

The popup list with suggestions will appear below the highlighted word. You can choose the correct word or if you don't find any press Esc key to dismiss the suggestions list.



Similarly, if you would like to see a list of synonyms for the highlighted word, you can use the **Synonyms** command in the same menu to display the popup list containing possible synonyms for the highlighted word.

To deactivate the spell checker use the **Deactivate** command available in the SQL Assistant menus, which are described in the previous paragraphs.

Choosing Spell Check Language

By default the Spell Checker uses the language selected as default language in Microsoft Word options. If you need to check spelling in a different language make sure that language support is installed on your system. If the required language is not available, use the Microsoft Office installation CD to install additional language support.

To pick a non-default language for the spell check use either of the following options:

- Use SQL Assistant's system tray icon menu (see Using System Tray Icon Menu topic for details) and choose **Target / Check Spelling / Language** command and then from the next level menu select the required language.
- Use target editor's context or top-level menus, if menu integration option is enabled (see Using Context and Top-level Menus topic for details). In that menu select **SQL Assistant / Check Spelling / Language** command and then from the next level menu select the required language.

 **Tip:** The language selection only applies the current target editor session. If you close the target editor and then reopen it or simply switch to a different editing window within the same target editor you would need to repeat the language selection again.

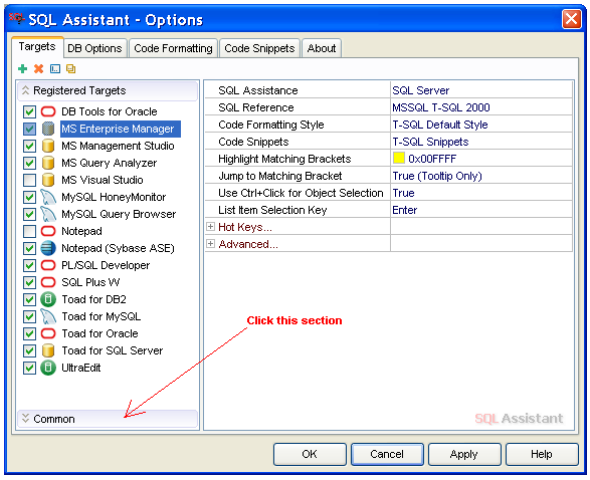
CHAPTER 13, Customizing SQL Assistant Behavior

Customizing Hot Keys

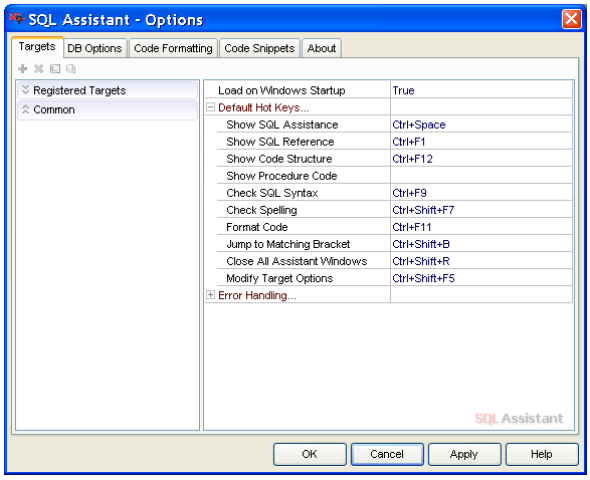
To customize default global hot keys:

1. Double-click SQL Assistant icon in the Windows System Tray. The **SQL Assistant - Options** dialog will appear

2. Click the **Common** section on the left side of the Options dialog screen.



The **Common** section will be displayed.



3. If you want to change hot key invoking SQL Assistant popup window in a target SQL editor, click **SQL Assistance** option and then press the keyboard buttons that you want to use as a new hot key. For example, if you want to use Shift+Tab as the new hot key, press the Shift key and while holding it down press the Tab key.

If you want to change hot key invoking SQL Reference, click **SQL Reference** option and then press the keyboard buttons that you want to use as a new hot key. For example, if you want to use Ctrl+F1 as the new hot key, press the Ctrl key and while holding it down press the F1 key.

If you want to change hot key invoking SQL editor target registration function, click **Add / Modify Target** option and then press the keyboard buttons that you want to use as a new hot key. For example, if you want to use Ctrl+F5 as the new hot key, press the Ctrl key and while holding it down press the F5 key.

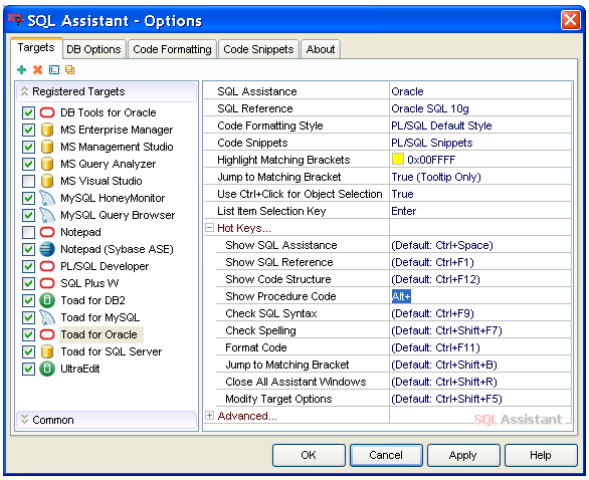
Important Note: The chosen hot keys must be unique and not used by your SQL editor, which you want to register with SQL Assistant. Failure to pick a unique hotkey may lead to incorrect SQL Assistance behavior. If you use several editors you can pick different hot keys for different editors as demonstrated in the following section.

To customize target specific hot keys:

1. Double-click SQL Assistant icon in the Windows System Tray. The **SQL Assistant - Options** dialog will appear

2. Click the **Targets** tab page.

3. In that page, on the left hand side of the dialog select the target type in the targets list



4. To choose editor specific hotkey, on the right side of the Options screen select the hot key option you want to change and then press the new hot key. Hot key can be a single key or a combination of Ctrl, Alt, Shift and other key and a group of keys together.

   Tip: The **(default)** text indicates that the default global hot key is used for the selected target for a particular SQL Assistant feature. This is a special option value, which cannot be typed in the hot key field. If you want to undo the custom hot key and switch back to the default, select the feature to undo for which you want to switch back to the default, and then press the Esc key.

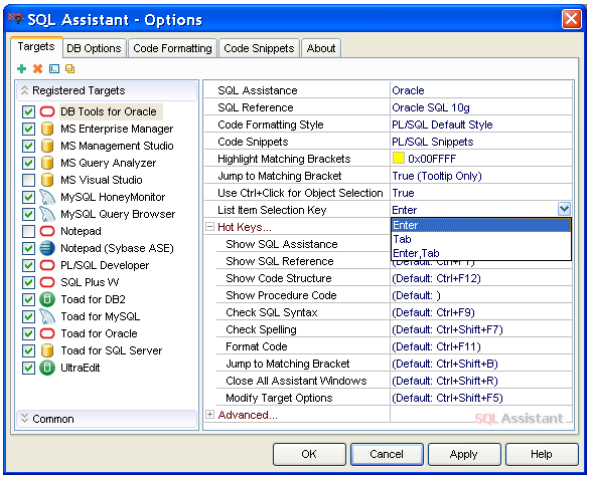
To customize selection keys used in SQL Assistance popup lists:

1. Double-click SQL Assistant icon in the Windows System Tray. The **SQL Assistant - Options** dialog will appear.

2. Click the **DB Options** tab page.

3. In that page, on the left hand side of the dialog select the **SQL Assistance** section and then select SQL Assistance type in the types list.

4. Click the **Options** option on the right hand side of the screen. The little button will appear on right side of the field. Click this button to show additional options.



5. On the next line, choose the desired Selection key in drop-down list. Available options are:

   Enter – this will allow you using the Enter key as an item selection key

   Tab – this will allow you use the Tab key as a selection key.

   Enter, Tab – this will allow you using both Enter and Tab keys as selection keys.

To customize code snippet activation keys:

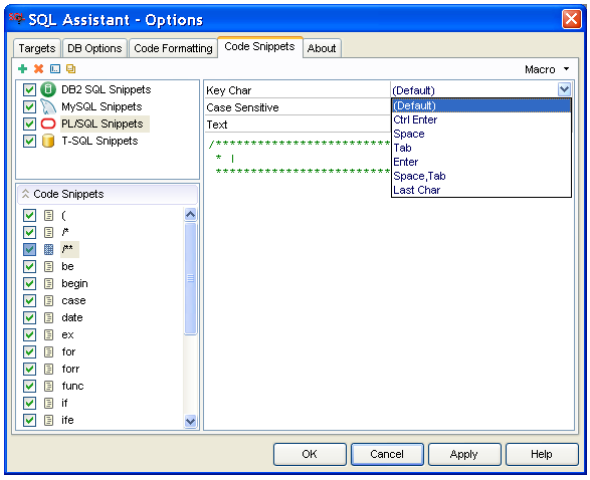
1. Double-click SQL Assistant icon in the Windows System Tray. The **SQL Assistant - Options** dialog will appear.

2. Click the **Code Formatting** tab page.

3. In that page, on the left hand side of the dialog select the SQL dialect whose code snippets you want to customize

4. Below that select the snippet by its short name displayed in the snippet list

5. On the right side of the screen expand the Key Char drop-down list and then select the required snipped activation key.



Available options are:

Ctrl+Enter – this will allow you using the Ctrl+Enter key combinations as a snippet activation key. If this key combination is pressed, SQL Assistant expands the snipped short name with the snippet code.

Space – this will allow you using Space key as a snippet activation key. The snippet name must be typed and then Space key pressed to active it.

Enter – this will allow you using the Enter key as a snippet activation key. The snippet name must be typed and then Enter key pressed to active it.

Tab – this will allow you use the Tab key as a snippet activation key. The snippet name must be typed and then Tab key pressed to active it.

Enter, Tab – this will allow you using both Enter and Tab keys as a snippet activation key. Pressing either Enter or Tab can be used to activate the snippet the result the same as above.

Last char – this is a special option that allows SQL Assistant automatically insert a code snippet whenever its short name is typed. Beware of snippets with similar names. If you define 2 snippets with a similar name like *snip* and *snip2* and set both to activate automatically using **Last char** option, you will not be able to use *snip2* because the snippet *snip* will always be activate first.

Managing SQL Assistant Load Methods

SQL Assistant supports 2 different methods for loading SQL Assistant code into the target editor. Different methods are available and can be used with different types of editors:

- **Target editor monitoring and hooking** – this generic method can be used with virtually all registered editor targets. SQL Assistant must be running as a system tray application (see Starting and Stopping SQL Assistant topic for details) in order to be able to watch for new editor instances and hook their keyboard and mouse event queues.

- **Native add-ons** – this method is currently only available for several applications for which an SQL Assistant add-on is available:
  ► Eclipse Integrated Development Environment and Eclipse-based derivative products
  ► Visual Studio 2005

- Visual Studio NET
- Visual Studio for Team Database Professionals (formerly Data Dude)
- Microsoft SQL Server Management Studio
- Microsoft SQL Server Management Studio Express.

The add-on for Eclipse must be copied to *plugins* directory of your Eclipse installation. The *plugins* directory location differs for different Eclipse versions and for derivative Eclipse products. Consult your Eclipse documentation for details for where to find this directory. If copied to the correct directory, the host application automatically loads the add-on on the startup.

The add-on for all Visual Studio and SQL Server Management Studio based products is registered in the system registry in the add-ons registration area so the host application automatically loads into each new instance of the Query Window.

SQL Assistant's system tray application may be used concurrently with these add-ons but theoretically is not required.

See CHAPTER 14, Registering and Unregistering Targets for SQL Assistance for more details on how to register targets, and modify SQL Assistant add-on loading methods.

Customizing Target Editor Menu Integration

For user convenience, SQL Assistant can integrate its menus wit the target editor menus. By default, it adds SQL Assistance menu items to the editor's right-click Context menus only. Also, by default it registers SQL Assistant add-in for SQL Server Management Studio and adds SQL Assistance menu items to the target editor top-level menu.

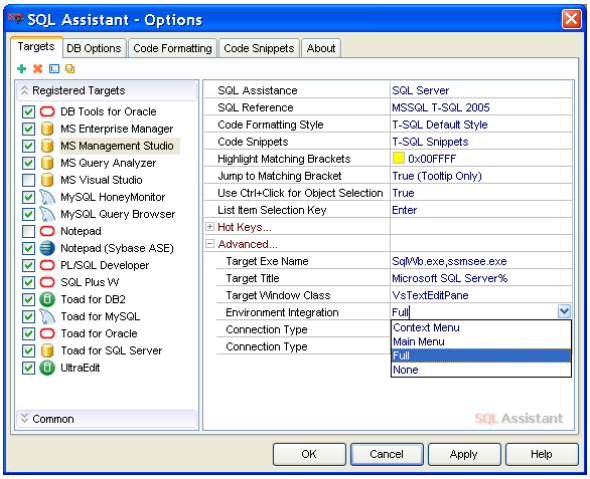
To change which menus get updated with SQL Assistant commands, use the following method:

1. Double-click SQL Assistant icon in the Windows System Tray. The **SQL Assistant - Options** dialog will appear.

2. Click the **Targets** tab page.

3. In the targets list select the target whose menus you want to modify.

4. Click the **Advanced…** option on the right hand side of the screen. The little button will appear on right side of the field. Click this button to show advanced options.



5. Choose the **Environment Integration** option and then in the drop-down list choose the required menu integration option. Note that the **Full** option means that both the top-level and context editor menus can be modified. The **None** option means that neither menu can be modified.

   Tip: Menu integration is not required. It is only provided for your convenience so that you don't need to remember all supported hot keys. You can also use SQL Assistant's system tray menu as described in Using System Tray Icon Menu topic in CHAPTER 3.

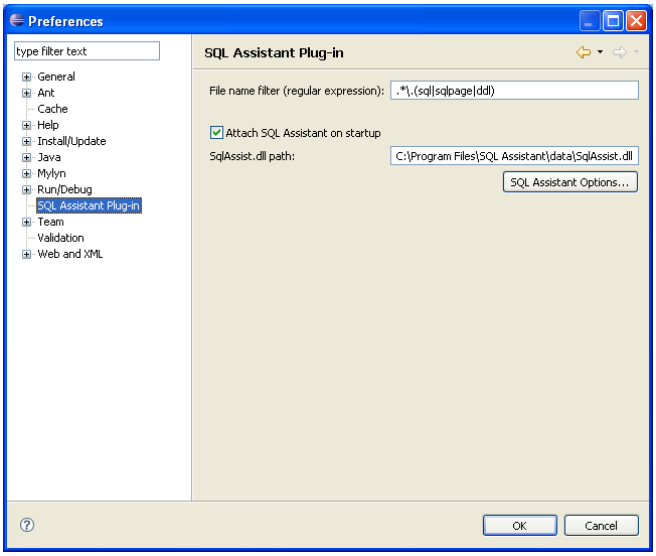
Important Note: For target editors such as Eclipse, SQL Server Management Studio and Visual Studio in addition to changing the **Environment Integration** option you may also enable or disable the SQL Assistant Add-in option. The menu integration only works if the **Register SQL Assistant Add-in** option is set to True.

Customizing Settings for Eclipse-based Target Editors

Several Eclipse specific settings can be customized directly in the Eclipse IDE:

1. In Eclipse, click **Windows -> Preferences** menu The **Preferences** dialog will appear.

2. Click **SQL Assistant Plug-in** item in the Preferences tree.



3. On the right hand side of the dialog, customize SQL Assistant settings as required.

4. Click the **OK** button on the bottom of the dialog to save Changes.

Tip: In case you use specialized SQL editors within the Eclipse environment, it is also recommended that you disable their **Code Assistant auto-activation** feature in the Eclipse Preferences. See Configuring Eclipse-based Target Editors topic for more information.

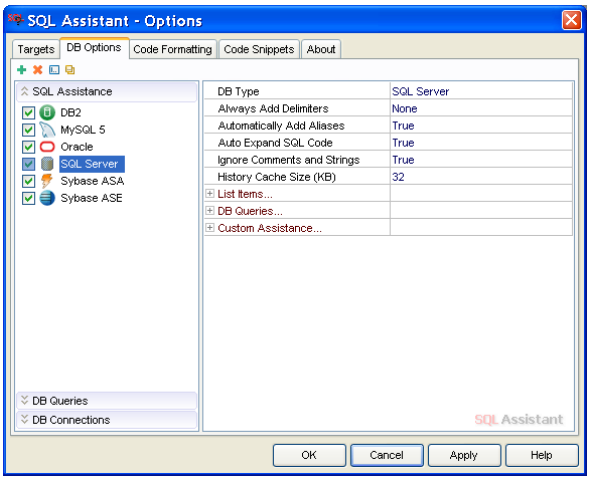
Customizing SQL Assistance Types

SQL Assistant comes with several pre-configured SQL assistance types called for clarity "Oracle", "DB2", "SQL Server", "My SQL 5", "MS Access", "Sybase ASA" and "Sybase ASE." All default pre-configured types are wired to the complete set of default database catalog queries. SQL Assistance interface allows you to customize the default assistance types or create your own.

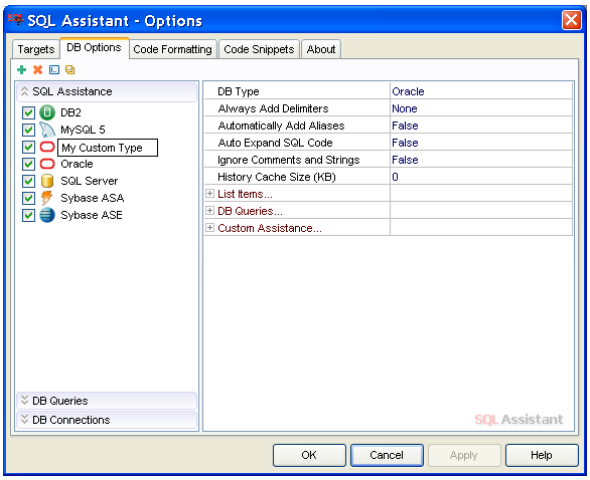
The following example demonstrates how to create your own type for SQL Server using a limited subset of catalog queries:

1. Double-click SQL Assistant icon in the Windows System Tray. The **SQL Assistant - Options** dialog will appear.

2. Click the **DB Options** tab page.

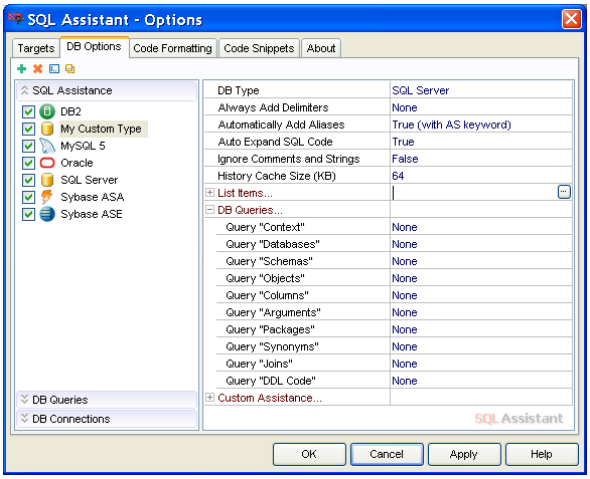


3. Click the Plus Sign icon  in the left top corner of the **DB Options** tab page to add a new SQL Assistance type and in the new row type *My Custom Type* or any name you want to use.

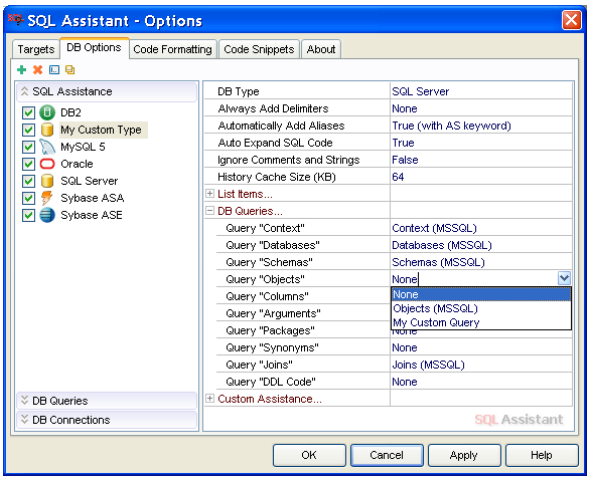


4. In the **DB Type** drop-down select type of the database for which the new SQL Assistance type will be used.

5. Click the field next to the **DB Queries** option. The little button will appear on right side of the field. Click this button to show DB Query associations and their properties.



6. Choose query associations as needed.

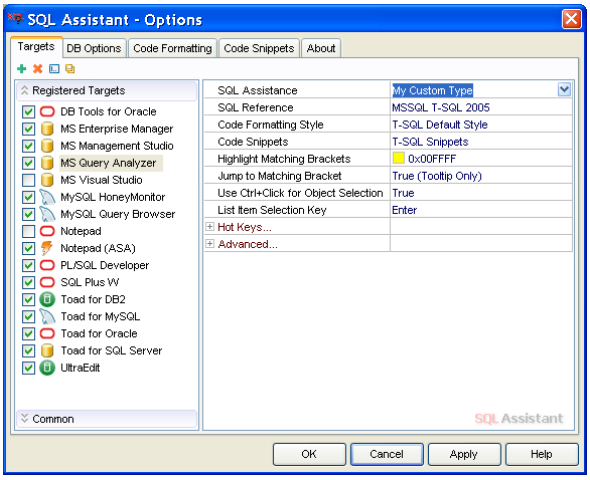


For example, in the DB Type option select "SQL Server", for "Objects", "Columns" and "Joins" select items as on the following sample screenshot. Note that if you leave "Databases" option associated with the "None" value, SQL Assistant will not be looking for database names in the database catalog and so no database names will appear in SQL Assistant popups.

Tip: You can define your own database catalog queries in the DB Queries page and then associate them with SQL Assistance types. See Customizing Database Catalog Queries topic for more information.

7. The created SQL Assistance Type can be then associated with specific development environment or a standalone editor.

   Here is how: Click the **Targets** tab page on **SQL Assistant - Options** dialog.



8. On the right hand side of the screen select the target editor for which you want to use the new custom configuration and then on the right side of the screen, select My Custom Type item in the **SQL Assistance** option drop-down list.

9. Choose type and version of **SQL Reference** that you want to associate with this target and also the associated type of **SQL Snippets**. Choose "None" as a value for items that you do not want to use.

10. Click the OK button to save all changes and close the Options dialog.

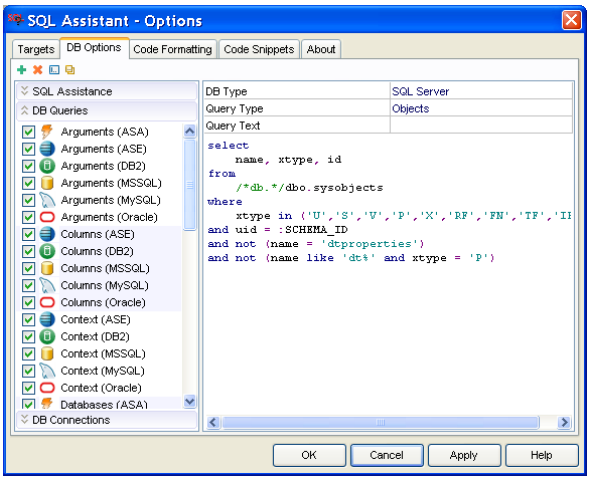
Customizing Database Catalog Queries

SQL Assistant uses several types of pre-defined database queries that are tuned for working with multiple versions of Oracle and SQL Server databases. If needed you can modify these queries and tune them for the specific version of the database you are working with. You can also enter new queries and assign them to your own custom SQL Assistance configuration.

To modify existing queries for reading database catalog information:

1. Double-click SQL Assistant icon in the Windows System Tray. The **SQL Assistant - Options** dialog will appear.

2. Click the **DB Options** tab page.



3. Select **DB Queries** section on the left-side of the Options screen

4. Select name of the catalog query you want to modify. The query text will appear on the right side of the screen.

5. Change query type, database target type, and edit the query text as needed.

6. If you want to modify other queries, repeat steps 4 and 5 for each query that you want to change. When done, click the OK button to save your changes and close the Options dialog.

To create a new query for reading database catalog information:

Use steps 1 to 6 as described above. On step 4 instead of selecting an existing query, click the Plus Sign icon in the left top corner of the **DB Options** tab page to add a new query.

To quickly create a new query from an existing query:

Use steps 1 to 6 as described above. On step 4 select an existing query, which you want to copy, click the Copy Sign icon in the left top corner of the **DB Options** tab page to add a new query. Name the new query as required.

To delete an existing query which you don't want to use anymore:

Use steps 1 to 6 as described above. On step 4 instead of editing query text, click the Delete Sign icon in the left top corner of the **DB Options** tab page to delete the selected query.

Tip: You can associate your custom catalog queries with both predefined and custom SQL Assistance types. This feature provides you with nearly complete control over SQL Assistant behavior and appearance. See Customizing SQL Assistance Types topic for more information.

Using Advanced Filtering For Fast Database Catalog Data Access

The previous topic describes how you can manage and customize database catalog queries. You can use the same technique for tuning database catalog query performance and filtering of the returned results. This is a very handy feature when you work with very large databases running various ERP applications with tens of thousands of objects, columns, procedures and so on. To improve the performance and limit query results only to specific object types or schemas you can use the following techniques:

1. Edit the slow performing catalog query and add the required performance hints. For example, if your Oracle database server is tuned for cost-based optimization and you want to use rule-based optimization for catalog queries you can add /*+ RULE */ hint after each SELECT keyword in text of each catalog query.

2. Filter out unused databases and schemas. For example, if in SQL Server you never use databases msdb and model, you can filter them out and add WHERE clause to the "Databases" catalog query with the text like WHERE name NOT IN ('msdb', 'model')

3. Filter in used schemas only. For example, if in your Oracle database you only work with schemas SHIPPING and RECEIVING you can add a WHERE clause to the "Schemas" catalog query with the text like WHERE username IN (' SHIPPING', 'RECEIVING')

Using techniques described above you can tune SQL Assistant to fit your requirements and maintain top performance.

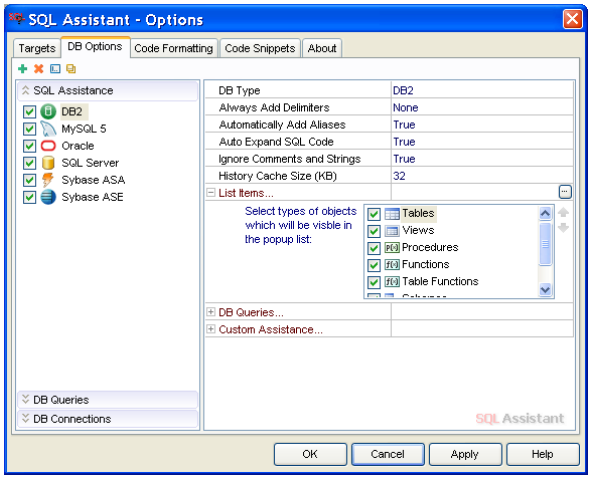
Using Object Type Filtering

You have several tools for controlling what objects are displayed in SQL Assistant popups. For example, you can use the **List Items** filter to enable only certain types of objects.

To use this filter type:

1. Activate **DB Options** tab.

2. Select **SQL Assistant type** whose behavior you want to modify.



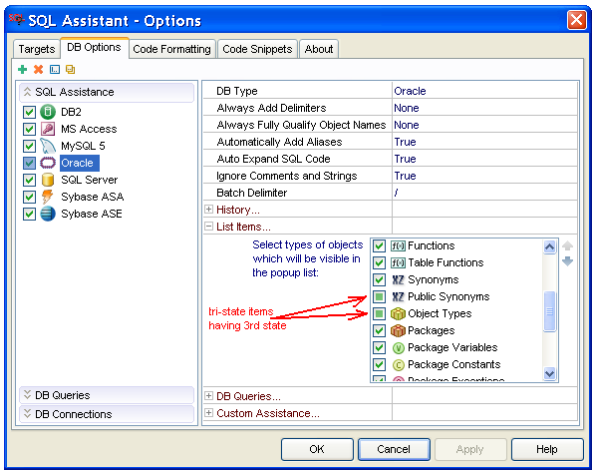
3. Expand **List Items** option group as shown on the sample image.

4. Select the required list item types and press the **Ok** button to apply changes and close the Options dialog.

Note: Different types of items can be configured for different types of supported database systems. If you are working with multiple database system types, you may need to customize each type individually.

Note: Some types of items support 3 states:

- **Enabled** (checkbox is checked) – items of this type always shown in the popups.
- **Disabled** (checkbox is unchecked) – items of this type are hidden in the popups
- **Enabled after first letter match** (checkbox is in a 3rd state with the box painted) – items of this type appear in the popups only after you type some text after popup appearance and their names match the text.



This special state allows you to hide items making the popup list to busy and making it difficult to locate what is needed. For example, Oracle aliases by default are marked with the 3rd state because if checked, thousands of them appear in the beginning of each popup list.

Changing Order of Objects in Object Name Popups

Certain object types can appear in different positions in the Object Name Popup list. To modify the default order:

1. Activate object type filter as described in the previous topic.
2. Click item type whose display position you want to change in the object name popups. Be careful not to remove the checkmark when selecting the required type.
3. Click the little Arrow Up and Arrow Down icons displayed next to the item type list to move the selected item type, up or down.

Tip: Note that not every type can be repositioned. This dependency is a result of internal constraints that cannot be modified. The Arrow Up and Arrow Down icons become enabled only for types whose order in the popup list can be modified.

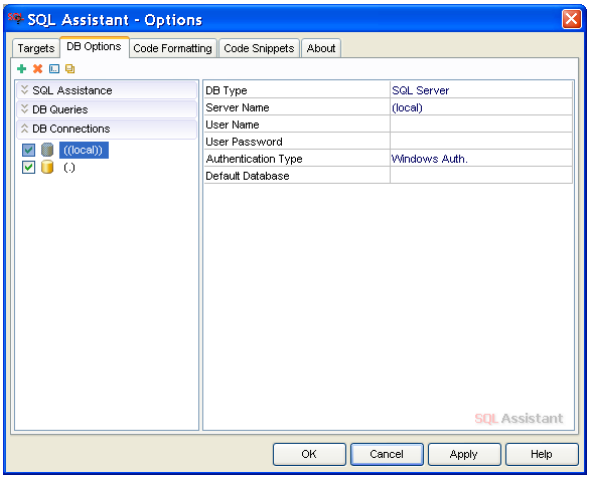
Managing Database Connections

Use the following instructions to manage database connections saved in SQL Assistant configuration files

To modify an existing connection:

1. Double-click SQL Assistant icon in the Windows System Tray. The **SQL Assistant - Options** dialog will appear.

2. Click the **DB Options** tab page.



3. Select **DB Connections** section on the left-side of the dialog screen

4. Select name of the connection you want to modify. The connection properties will appear on right side of the screen.

5. Edit properties as required.
   Note: Server, user, password, connection type and database name are optional properties. If not specified, you can enter them later on the Connection dialog, which is displayed if SQL Assistant needs to establish a new connection to the database not shared with the target editor.

   Also, note that different set of properties is available for different connection types. For example for Oracle connections you may need to enter Oracle TNS name, connection type (Normal, SYSDBA or SYOPER) and/or path to Oracle's OCI.DLL

   Additional information of supported connection types and their properties is available in CHAPTER 2, Connecting to Your Database.

6. If you want to modify other connections, repeat steps 4 and 5 for connection you want to change. When done, click the OK button to save your changes and close the Options dialog.

To create a new connection:

Use steps 1 to 6 as described above. On step 4 instead of selecting an existing connection, click the Plus Sign icon  in the left top corner of the **DB Options** tab page to add a new connection.

To quickly create a new connection from an existing connection:

Use steps 1 to 6 as described above. On step 4 select an existing connection, which you want to copy, click the Copy Sign icon  in the left top corner of the **DB Options** tab page to add a new connection. If adding SQL Server connection, use your SQL Server instance name as the connection name. If adding Oracle connection, use your Oracle TNS connection name as you would enter it in SQL*Plus and other Oracle programs.

To delete an existing connection which you don't want to use anymore:

Use steps 1 to 6 as described above. On step 4 instead of editing connection properties, click the Delete Sign icon  in the left top corner of the **DB Options** tab page to delete the selected connection.

Customizing Brackets and SQL Code Matching and Navigation

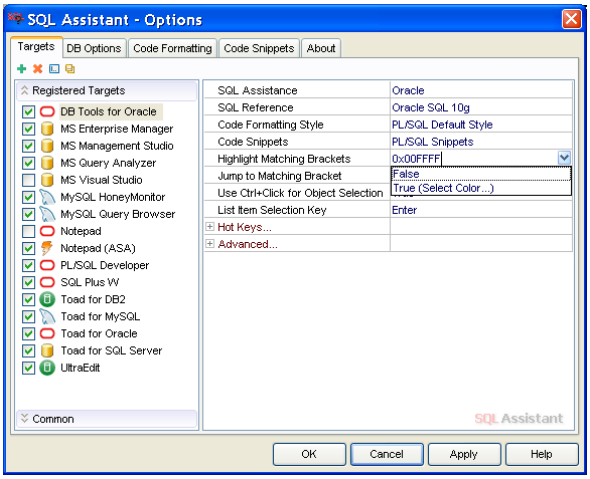
SQL Assistant supports automatic brace matching which gives you immediate feedback on misplaced brackets or open-ended SQL code blocks such as BEGIN without END or IF without END IF, and other SQL structures requiring beginning and ending keywords. It also provides quick code navigation methods using matching bracket jumping so you can quickly navigate from the start of a SQL block to the end or visa versa.

You can change the behavior of this feature by modifying the following options:

- **Highlight Matching Bracket** – Using this option you can control the color used to highlight matching brackets and block delimiters or turn the highlighting completely off.
- **Jump to Matching Bracket** – Using this option you can control how to jump between matching brackets or disable this option completely.

Both options are available in the Targets options group and can be configured differently for different targets. To modify these options:

1. Double-click SQL Assistant icon in the Windows System Tray. The **SQL Assistant - Options** dialog will appear.
2. Click the **Targets** tab page.



3. On the left hand side of the screen, select the target whose options you want to modify.

4. On the right hand side of the screen, change bracket related options as required.

5. If you want to modify options for other targets, repeat steps 3 and 4 for each target. When done, click the OK button to save your changes and close the Options dialog.

Customizing Existing and Creating New Code Snippets

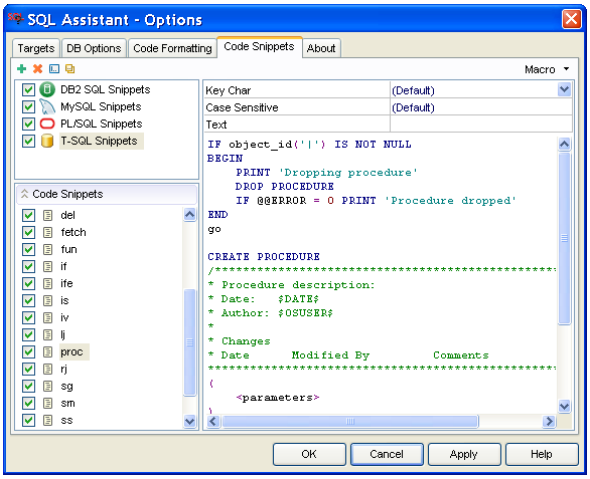
Any changes you make in code snippets on the **Code Snippets** tab page do apply only to the selected SQL dialect appearing in the SQL dialects list in the top-left corner of the tab page. Before you change any code snippets, make sure to select the SQL dialect you want to modify.

To modify existing code snippets:

1. Double-click SQL Assistant icon in the Windows System Tray. The **SQL Assistant - Options** dialog will appear.

2. Click the **Code Snippets** tab page.

3. On the left hand side of the screen, select the SQL dialect whose keyword list you want to modify.



4. Select name of the snippet whose code you want to modify. The snipped code will appear on right side of the screen.

5. Change snippet hot key, case sensitivity, and edit the snippet text as needed.

6. If you want to modify other code snippets, repeat steps 4 and 5 for each snippet. When done, click the OK button to save your changes and close the Options dialog.

To create a new snippet for reading database catalog information:

Use steps 1 to 6 as described above. On step 4 instead of selecting an existing snippet, click the Plus Sign icon in the left top corner of the **Code Formatting** tab page to add a new snippet.

To quickly create a new snippet from an existing snippet:

Use steps 1 to 6 as described above. On step 4 select an existing snippet, which you want to copy, click the Copy Sign icon in the left top corner of the **Code Formatting** tab page to add a new snippet. Name the new snippet as required.

To delete an existing snippet which you don't want to use anymore:

Use steps 1 to 6 as described above. On step 4 instead of editing snippet text, click the Delete Sign icon in the left top corner of the **Code Formatting** tab page to delete the selected snippet.

Tip: When you activate the **Code Snippets** tab, the first available SQL dialect is selected by default in the list the left-top corner of the page. If you make changes in code snippets often, it is a good idea to set the used SQL dialect to appear first in the list. Use drag-and-drop method to rearrange order in which SQL dialect names appear in the list and place the required SQL dialect first. You changes will be remembered and next time you open the Options dialog, the SQL dialect items on the **Code Snippets** tab will appear in the chosen order.

Customizing Keywords Used With the Keyword Capitalization Feature

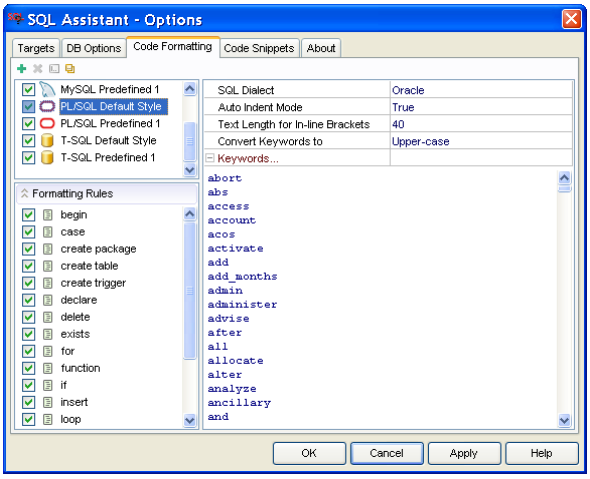
Any changes you make in keywords and code formatting rules on the **Code Formatting** tab page do apply only to the selected SQL dialect appearing in the SQL dialects list in the top-left corner of the tab page. Before you make any changes in formatting rules, make sure to select the SQL dialect you want to modify.

To enable automatic keyword reformatting and/or customize keywords list:

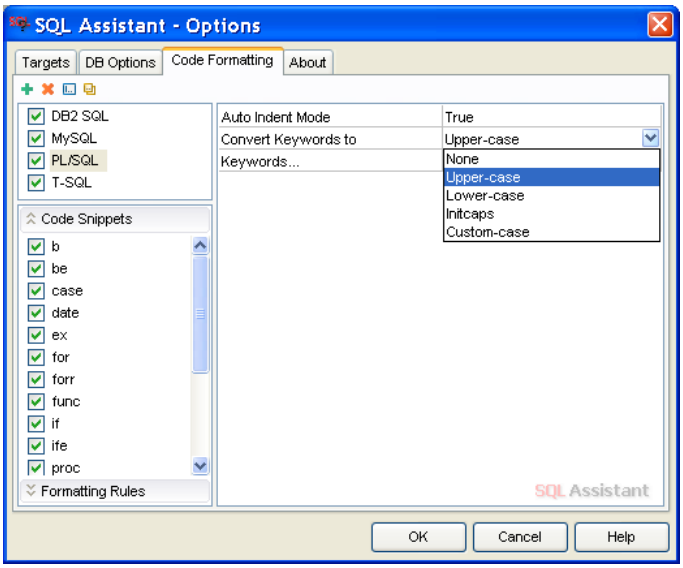
1. Double-click SQL Assistant icon in the Windows System Tray. The **SQL Assistant - Options** dialog will appear.

2. Click the **Code Formatting** tab page.

3. On the left hand side of the screen, select code formatting style whose keyword list you want to modify.



4. Click the field next to the **Keywords** option. The little button will appear on right side of the field. Click this button to expand the list of keywords and the formatting function applied to these keywords.



In the **Convert Keywords to** drop-down list choose **Uppercase** function to automatically convert keywords to upper case. Choose **Lowercase** function to automatically convert keywords to upper case. Choose **Initcaps** function to automatically convert keywords to lower case with first letters in upper case. Choose **Custom-case** option to have the keywords formatted exactly as they are entered in the keywords list.

Note that choosing the **None** option effectively disables the automatic keyword formatting feature.

5. Edit keyword list as needed.

6. Click the OK button to save all changes and close the Options dialog.

To disable automatic keyword reformatting:

1. Double-click SQL Assistant icon in the Windows System Tray. The **SQL Assistant - Options** dialog will appear.

2. Click the **Code Formatting** tab page.

3. On the left hand side of the screen select the SQL dialect whose keyword list you want to modify.

4. In the **Convert Keywords to** drop-down list choose **None** to disable automatic keyword formatting.

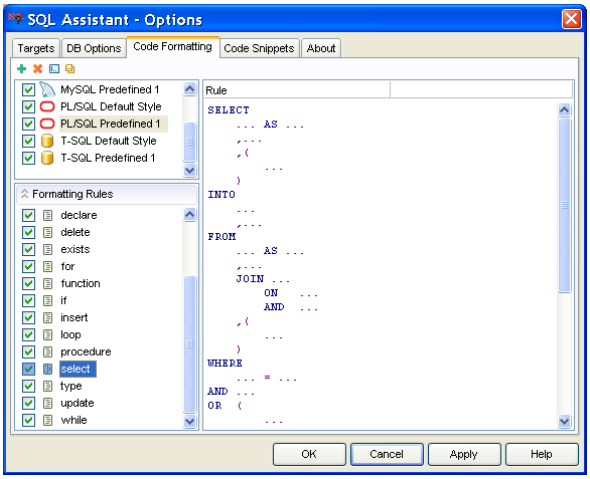
5. Click the OK button to save all changes and close the Options dialog.

Tip: When you activate the **Code Formatting** tab, the first available SQL dialect is selected by default in the list the left-top corner of the page. If you make changes in code formatting often, it is a good idea to set the used SQL dialect to appear first in the list. Use drag-and-drop method to rearrange order in which SQL dialect names appear in the list and place the required SQL dialect first. You changes will be remembered and next time you open the Options dialog, the SQL dialect items on the **Code Formatting** tab will appear in the chosen

order.

Customizing Code Formatting Patterns

Code formatting patterns consist of keywords, triple dots indicating text between keywords, parenthesis and commas indicating code flow, and white spaces consisting of space and tab characters



It is important to know that the code-formatting pattern may include more keywords, then the actual SQL statement which is formatted using the pattern. The keywords which are in the pattern but not in the SQL statement are ignored during the processing and do not affect the results.

Spacing and positions of commas, parenthesis and logical operations are very important. To get an idea how this affects the code formatting, let's consider the following T-SQL DECLARE statement with multiple variables

DECLARE @var1 int, @var2 datetime, @var3 char(5), @var4 int

If you specify formatting pattern for DECLARE as

DECLARE ...,
...

The code formatter will produce the following code

DECLARE @var1 int,
@var2 datetime,
@var3 char(5),
@var4 int

However, if you specify formatting pattern for DECLARE as below (note the position of the comma character)

DECLARE ...
,...

The code formatter will produce the following code

```
DECLARE    @var1 int
           ,@var2 datetime
           ,@var3 char(5)
           ,@var4 int
```

Similarly, if a formatting pattern for WHERE clause in a SELECT statement is defined as

```
WHERE …
    AND …
```

The code formatter will produce the following code

```
WHERE col_a = @col_a
    AND col_b = @col_b
```

However, if you specify formatting pattern for WHERE as below (note the position of the AND logical operator)

```
WHERE … AND
    …
```

The code formatter will produce the following code

```
WHERE col_a = @col_a AND
    col_b = @col_b
```

 **Tip:** The best way to learn how the code formatting works is to modify the default options and to apply code formatting to your SQL file so you can see the results of your changes.

Customizing Error Handling Options

SQL Assistant supports 4 error-handling modes. To choose which mode is right for you, open SQL Assistant options and activate the **Targets** page. On the Targets page on the left hand side of the screen expand the **Common** section and then on the right hand side of the screen expand the **Error Handling** option group.

The following error handling modes and options are supported:

- **Show system tray notifications –** In this mode, if an error occurs, SQL Assistant displays a small notification message in the system tray area above the SQL Assistant icon. The message contains first 60 characters of the error message text. If you click on the notification message the full Error Message dialog will appear. The Error Dialog describes the error and allows you to report the error to SoftTree Technologies technical support and obtain a tracking number for the reported error. The rest of the error handling is the same as in the **Display error messages** mode described in the next paragraph. Note that this error handling method is only available on Windows XP, Vista, 2003 and later Windows versions.

- **Display error messages –** In this mode, if an error occurs, SQL Assistant displays its Error Message

dialog describing the error and also asks if you would like to send an error report to SoftTree Technologies technical support system. If you do not want to send an error report, you can simply click the **Close** button. If you click the **Send** button, the error message, SQL Assistant' version, name and version of the target editor, and your Windows system type and version will be sent to the support system. The error dialog contains 2 optional edit fields, which you can use to describe the steps taken before this error occurred and optionally your contact email. Please note that the error data contains no confidential or personal information unless you explicitly enter such data yourself. Each submitted error report is processed by the online support system and a unique number is assigned to each reported error. This number can be used later to track the status of the issue. If you specify a valid contact email, you allow SoftTree Technologies to use it for asking additional questions about the error if there is such need, for example asking for additional information on how to reproduce the error. The provided email can be also used to notify you about status changes. For example, if a bug in SQL Assistant program code is causing an error, SoftTree Technologies may use the provided email to notify you when this bug is fixed. Note that the contact email will be used for communications regarding the reported error. It will not be used for any other purposes.

 **Tip:** The error dialog is resizable. This is it. If you cannot see the entire error text displayed in the top white area of the dialog, drag the edge of the dialog to make it bigger.

- **Ignore all errors –** In this mode, if an error occurs, SQL Assistant silently ignores it.

- **Silently report errors to support –** This is similar to the first option, except that errors are logged automatically to the support system and SQL Assistant does not ask you to describe the steps. On one hand, this option may appear less interrupting while still helpful to improve future SQL Assistant versions. On another hand, you do not get a chance to enter any related error description or comments, which undoubtedly diminishes the value of the reported errors and makes it more difficult to analyze and troubleshoot them.

SQL Assistant also allows you to specify default contact email, which the error reporting service may use in the **Display error messages** and the **Silently report errors to support** error-handling modes. Use **Your email address (optional)** option to specify your default contact email. Note that this is an optional value and it can be left blank.

CHAPTER 14, Registering and Configuring Targets for SQL Assistance

Overwiew of Target Registration Modes

SQL Assistant supports 2 different methods for hooking SQL Assistant code into the target editor: **add-ons** and **dynamic monitoring and hooking**. For several pre-configured targets, SQL Assistant provides specially created add-ons that allow full and easy SQL Assistant integration with the target environment. The add-ons must be properly installed and configured with the target environment

In comparison, the dynamic application monitoring and hooking method is generic and it can be used with many Windows-based programs, including new and unknown programs. You can register new programs with SQL Assistant and configure SQL Assistant settings for use with the newly registered programs.

See Managing SQL Assistant Load Methods topic in CHAPTER 13 for more information about SQL Assistant loading into the target editor environment

The following topics describe how to register and unregister SQL Assistant add-ons and new target editors.

Installing and Enabling SQL Assistant Add-ons for Preconfigured Targets

SQL Assistant add-on for Visual Studio 2005, Visual Studio NET, Visual Studio for Team Database Professionals (formerly Data Dude), Microsoft SQL Server Management Studio, Microsoft SQL Server Management Studio Express is preinstalled by default, and located in the SQL Assistant installation directory, but the add-on is not enabled by default. Do the following to enable the add-on:

1. Double-click SQL Assistant icon in the Windows System Tray. The **SQL Assistant - Options** dialog will appear.

2. On **Targets** tab page, on the left hand side of the screen, select the target for which you want to enable the add-on.

3. On the right hand side of the screen, expand the **Advanced…** option group.

4. Use the drop-down list or simply double-click the value in the **Register SQL Assistant Add-on** option to change the value from False to True.

5. Click the OK button to save all changes and close the Options dialog.

6. To verify the registration, close SQL Assistant systray icon and restart the target editor. SQL Assistant menu should appear in the target and SQL Assistant should be active in the Query Editor windows.

SQL Assistant add-on for Eclipse IDE and derivative products such as Easy-Eclipse and a number of others software products is not installed by default. These products can be installed anywhere on the disk and they typically do not register itself with the system. SQL Assistant does not know where to find them and that is a why a simple manual installation step is required to install the add-on for Eclipse. Do the following to install and enable this add-on:

1. Open Windows Explorer and locate SQL Assistant installation directory. The default path is `C:\Program Files\SQL Assistant`

2. Expand **Data** subfolder and right-click `com.softtree.sa.eclipse.jar` file. Windows Explorer popup menu will appear over the file.

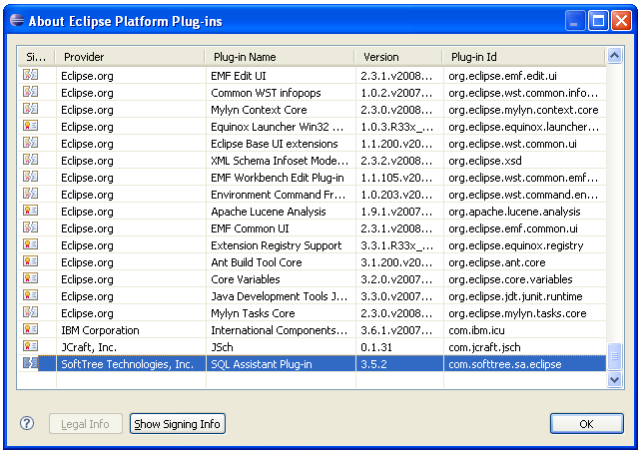
3. Click **Copy** command in the Explorer's popup menu.

4. Locate **plugins** subfolder of your Eclipse installation. For example, if you have Eclipse installed in the root folder of the drive C, the subfolder for plugins typically would be `C:\eclipse\plugins` Some Eclipse based products use `C:\Documents and Settings\{user name}\workspace\.metadata\.plugins` as the startup folder for their plugins. For exact plugins subfolder name and location consult your Eclipse documentation

5. Right-click the plugins subfolder. Windows Explorer popup menu will appear.

6. Click **Paste** command in the Explorer's popup menu. That is it.

7. To verify the registration, close SQL Assistant systray icon and restart the Eclipse IDE. Click **Help -> About Eclipse Platform** menu to display the Eclipse's About dialog box. On this dialog click **Plug-in Details...** button. The About Eclipse Platform Plug-ins dialog box will appear. Scroll the list to the bottom. SQL Assistant add-on should appear at the end of the list as on the following sample screenshot.

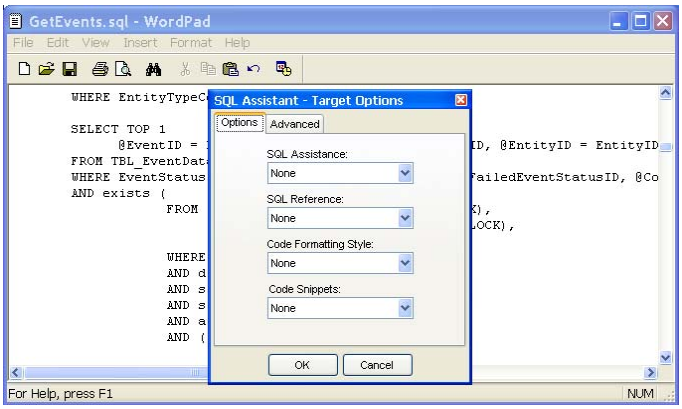


Registering New Targets for SQL Assistance

The simplest way to register a new editing target is to use the Ctrl+F5 hot key. The following example demonstrates how Windows WordPad application can be registered with SQL Assistant:

1. Start WordPad

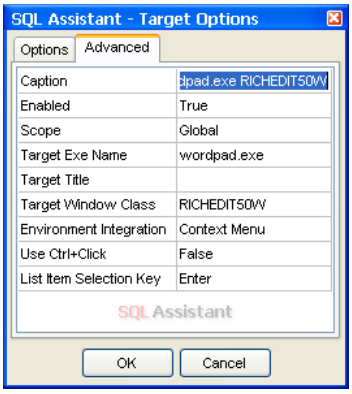
2. Make sure the focus in is in the WordPad window, click any text in the editor area. Press Ctrl+F5 hot key. The SQL Assistant dialog will appear on the screen as on the sample screenshot below.



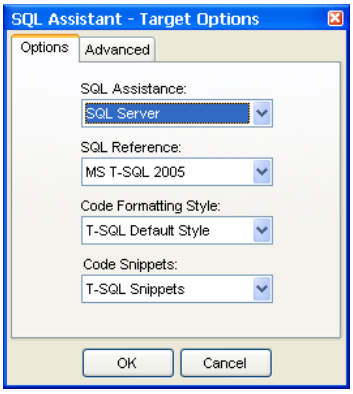
3. Customize target options.

For example, if you would like to use WordPad as an editor for SQL Server files, choose "SQL Server" in the **SQL Assistance** drop-down. If you want to use code snippet shortcuts with this editor, choose T-SQL in the **Code Snippets** drop-down.

Before:



After:



Change **SQL Assistance** option to "SQL Server."

Choose version of **SQL Reference** that you want to use with the new target. If you are working with Microsoft SQL Server 2000, choose "MS T-SQL 2000." If you are working with Microsoft SQL Server 2000, choose "MS T-SQL 2005".

If you would like to use code snipped shortcuts, change **Code snippets** option to "T-SQL."

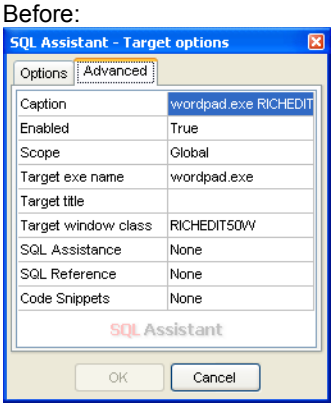
Customize advanced target options.

For example, you can change target name in the **Caption** option to "WordPad" or other descriptive name that will be used to reference this editing target in SQL Assistant Options on **Targets** page.

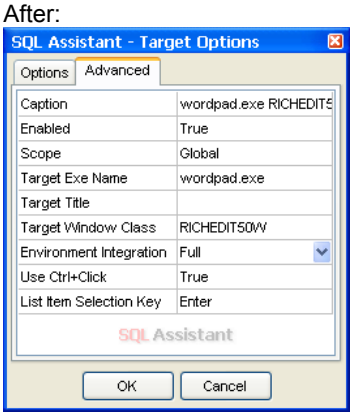
Leave the default **Enabled** option as "True."

Use "Global" for the **Scope** option If you want to register WordPad permanently and make SQL Assistant always assist with editing files in WordPad. If you want to apply options only to the currently running instance of the editor, choose either "Current process" or "Current window." For WordPad both options have the same effect because WordPad editor does not feature multiple document interface and can edit only one file at a time.

Before:



After:



Don't change **Target exe name** and **Target window class** options unless you are instructed to do so by technical support.

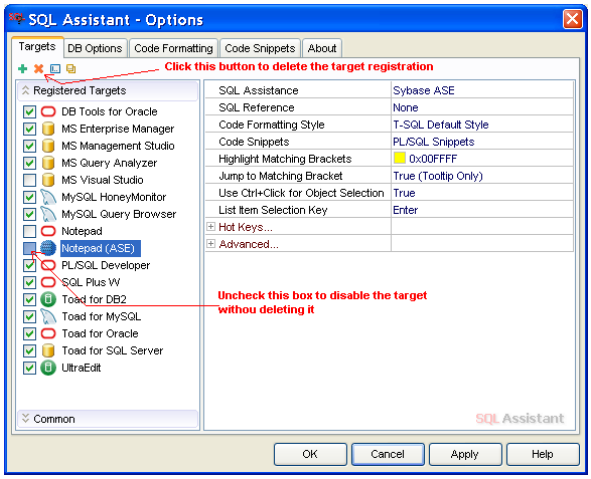
If the same executable file (**Target exe**) is used to start different programs and don't want to use SQL Assistant with all of these programs, specify an optional **Target Title** filter, which causes SQL Assistant to compare program title in addition to program name. For example, Microsoft SQL Server Manager for SQL Server 2000 is started using MMC.EXE, which is a general purpose Microsoft tool for launching various server management applications. With the "Microsoft SQL Server Manager %" **Target Title** you can limit SQL Assistant to the Microsoft SQL Server Manager application only.

Tip: The percent sign can be used in **Target Title** filter property as a wildcard symbol much like a LIKE operator in SQL queries. Using this wildcard you can target applications whose title changes depending on the database connection type or name of the opened object or the file being edited.

Unregistering Previously Registered and Preconfigured Targets

To un-register already registered targets including these that come pre-configured with the default installation you can use the following simple method:

1. Double-click the SQL Assistant icon in the Windows system tray. The Options dialog will appear.
2. Click **Targets** tab page.
3. Select the editing target you want to unregistered and delete permanently from the settings and then click the Delete button in the left top corner of the screen.



Disabling Targets Without Deleting Their Registrations

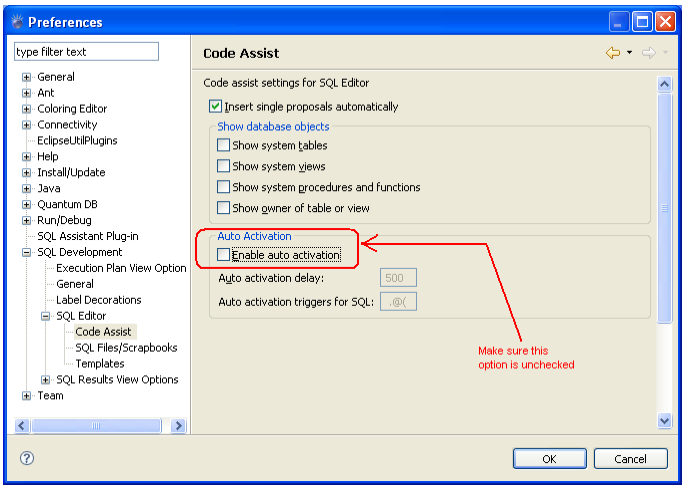
If you do not want to delete an existing target permanently but simple want to disable SQL Assistance until later time, you can uncheck **Target Enabled** checkbox as demonstrated on the sample screenshot in the previous topic.

Configuring Eclipse-based Target Editors

In case you use specialized SQL editors within the Eclipse environment, it is recommended that you disable the **Code Assistant auto-activation** features in the Eclipse Preferences. SQL IntelliSense ++ features available in SQL Assistant supercede Code Assistant features of these editors. If left with default settings, **Code Assistant auto-activation** will interfere with SQL IntelliSense causing major inconveniences.

Different types of Eclipse compatible SQL editors can use different names for this feature and can have the settings located in different places. The following screenshot demonstrate where and how to disable this feature in the default SQL Editor that comes with Eclipse Data Tools Platform.

The following screenshot demonstrates options that should be modified.



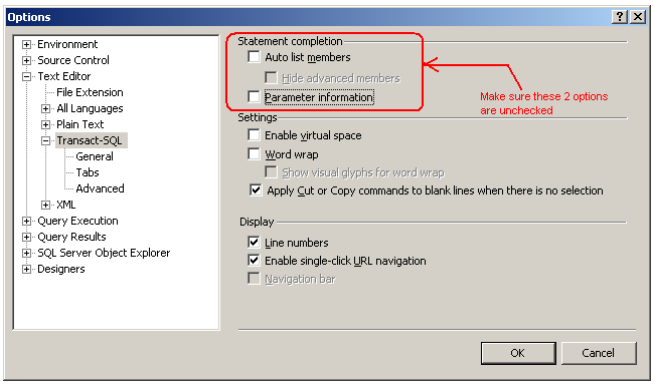
The Preferences dialog can be accessed using **Window / Preferences** menu available in the in the Eclipse IDE.

Configuring Native Tools Provided with SQL Server 2000 and 2005

No changes or customization are required in settings of SQL Query Analyzer, SQL Server Management Studio 2005 and Management Studio Express 2005. SQL Assistant can be used with these tools using their default settings.

Configuring Native Tools Provided with SQL Server 2008

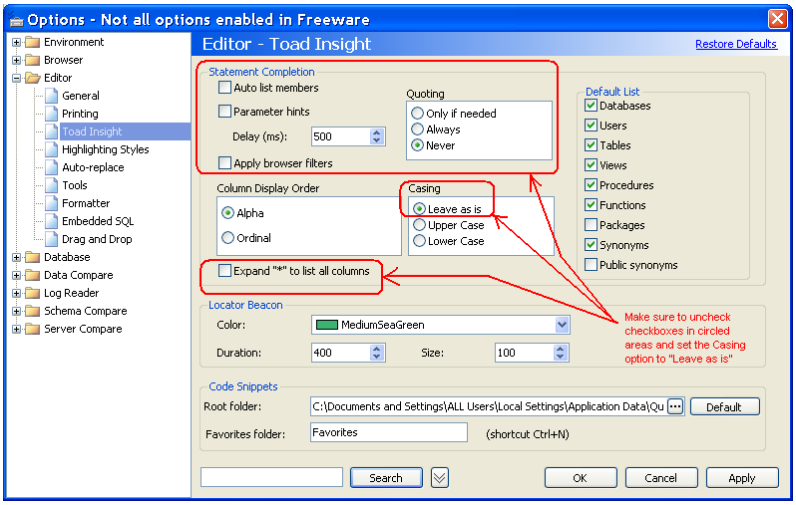
It is recommended that you disable the **Statement completion** features in SQL Server Management Studio 2008 Options. SQL IntelliSense ++ features available in SQL Assistant supercede **Statement completion** features of SQL Server Management Studio. If left with default settings, **Statement completion** will interfere with SQL IntelliSense causing significant inconveniences. The following screenshot demonstrates options that should be modified.



The Options dialog can be accessed using **Tools / Options** menu available in SQL Server Management Studio 2008.

Configuring Toad Editors

If you use SQL Assistant with recent versions of Quest Software Toad tools, it is recommended that you disable the **Statement Completion** and **Casing** features in Toad Options. SQL IntelliSense ++ features available in SQL Assistant supercede **Statement Completion** features of Toad Editors. If left with default settings, **Statement Completion** will interfere with SQL IntelliSense causing significant inconveniences. The following screenshot demonstrates options that should be modified.



The Options dialog can be accessed using **Tools / Options** menu available in Toad.

Resolving Keyboard Hotkey Conflicts.

It is possible that certain default keyboard hotkeys used by SQL Assistant are also used by you target editor. yet is very easy to resolve such conflicts. Every default hotkey in SQL Assistant can be changed to any hotkey of user choice. This customization can be done both globally across all registered targets in the Common section of SQL Assistant Options, and individually for each target. In the latest case you can choose different hotkeys for use with different targets. For more information about customizing hotkeys, see Customizing Hot Keys topic in CHAPTER 13.

CHAPTER 15, Installation and Uninstallation

Installation

The SQL Assistant's setup program provides intuitive and simple installation method. Simply run the setup program and follow prompts displayed on the screen.

Uninstallation

The SQL Assistant supports standard uninstallation mechanism for removing program files from the computer.

To uninstall the SQL Assistant:

1 Click Windows **Start** button, from the Start Menu select **Settings**, then **Control Panel**.

2 Double-click **Add/Remove Programs** icon in the Control Panel.

3 Select the SQL Assistant item in the programs list, click the **Add/Remove** button.

APPENDIX A, Hardware and Software Requirements

Minimum Requirements

1. Intel or AMD-based workstation or server running one of the following operating systems:
   - Windows Vista
   - Windows 2003
   - Windows XP
   - Windows 2000
2. 128 MB RAM
3. 5.0 MB free disk space for full installation
4. VGA or better monitor
5. Required database client software (consult your database system documentation for details)
6. If ODBC database interface is used, ODBC 3.0 and compatible database connectivity driver
7. If spell check feature is used, Microsoft Word 97 or later must be installed on the system.

Database Server Software

Any of the supported database servers:

- Oracle 8i, 9i, 10g, 11g
- Microsoft SQL Server 2000, 2005
- DB2 UDB 7, 8, 9
- MySQL 5, 5.1
- Sybase ASA 8, 9, 10
- Sybase ASE 12, 12.5, 15

Database Client Software

Depending on the database type and selected target editor you may need to have the following database client software installed on the computer running the target editor:

- **Oracle OCI** version 8.0 or later, which is part of the **Oracle client** software provided by Oracle Corporation.
- **Oracle ODBC Driver** provided by Oracle Corporation.
- Microsoft **SQL Server** ODBC driver for SQL Server 2000 and later provided by Microsoft Corporation
- **Sybase ASE ODBC Driver** for ASE 12 and 12.5 provided by Sybase Corporation
- **Adaptive Server Enterprise** ODBC driver for ASE 15 provided by Sybase Corporation
- **Adaptive Server Anywhere** ODBC drivers for ASA 7 or later provided by Sybase Corporation
- **IBM DB2 ODBC DRIVER** for DB2 7 and later, which is part of IBM DB2 ODBC drivers software provided by IBM Corporation.
- **MySQL ODBC 3.51 Driver** provided by MySQL AB Corporation.
- **MySQL Connector** v5.0 or later which is part MySQL client software provided by MySQL AB Corporation.
- **ADO.NET** software and database connectivity drivers, which are part of Microsoft NET

Framework v2.0 or later provided by Microsoft Corporation

 **Tip:** Note that the text in bold indicates actual driver names available directly from the database vendors. SQL Assistant currently does not support connectivity drivers provided by other third party vendors.

APPENDIX B, License Agreement

SOFTWARE PRODUCT USER LICENSE

Copyright laws and international copyright treaties, as well as other intellectual property laws and treaties protect this SOFTWARE PRODUCT. The SOFTWARE PRODUCT is licensed, not sold.

CAUTION: Loading this software onto a computer indicates your acceptance of the following terms. Please read them carefully.

GRANT OF LICENSE: SoftTree Technologies, Inc. ("SoftTree Technologies") grants you a license to use the software ("Software"). One licensed copy of the Software may either be used by a single person who uses the software personally on one or more computers, or installed on a single workstation used non-simultaneously by multiple people, but not both. One licensed copy of the Software can be used with any number of database servers.

You may make other copies of the Software for backup and archival purposes only. You may permanently transfer all of your rights under this Software LICENSE only in conjunction with a permanent transfer of your validly licensed copy of the product(s).

The Software and associated add-in components are licensed on a RUN-TIME basis, which means, that for each computer on which the Software is installed, a valid run-time license must exist.

RESTRICTIONS: Unregistered versions (shareware licensed copies) of the Software may be used for a period of not more than 30 days. After 30 days, you must either stop using the Software, or obtain a validly licensed copy.

You must maintain all copyright notices on all copies of the Software. You may not sell copies of the Software to third parties without express written consent of SoftTree Technologies and under SoftTree Technologies' instruction.

EVALUATION copies may be distributed freely without charge so long as the Software remains whole including but not limited to existing copyright notices, installation and setup utilities, help files, licensing agreement, In executing such an act as distributing without the similar copyright or license violation, to the maximum extent permitted by applicable law you may be held liable for loss of revenue to SoftTree Technologies or SoftTree Technologies' representatives due to loss of sales or devaluation of the Software or both.

You must comply with all applicable laws regarding the use of the Software.

COPYRIGHT: The Software is the proprietary product of SoftTree Technologies and is protected by copyright law. You acquire only the right to use the Software and do not acquire any rights of ownership.
For your convenience, SoftTree Technologies provides certain Software components in the source code format. You may customize this code for your environment, but you agree not to publish, transfer, or redistribute in any other form both the original code and the modified code.
You agree not to remove any product identification, copyright notices, or other notices or proprietary restrictions from the Software.
You agree not to cause or permit the reverse engineering, disassembly, or decompilation of the Software. You shall not disclose the results of any benchmark tests of the Software to any third party without SoftTree Technologies' prior written approval.

DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS: You may not rent, lease or transfer the Software except as outlined under GRANT OF LICENSE - use and copy.

Without prejudice to any other rights, SoftTree Technologies may terminate this Software LICENSE if you fail to comply with the terms and conditions of this Software LICENSE. In such event, you must destroy all copies of the Software and all of its component parts.

WARRANTY DISCLAIMER: SoftTree Technologies provides this license on an "as is" basis without warranty of any kind; SoftTree Technologies disclaims all express and implied warranties, including the implied warranties of merchantability or fitness for a particular purpose.

LIMITATION OF LIABILITY: SoftTree Technologies shall not be liable for any damages, including direct, indirect, incidental, special or consequential damages, or damages for loss of profits, revenue, data or data use, incurred by you or any third party, whether in an action in contract or tort, even if you or any other person has been advised of the possibility of such damages.

SoftTree Technologies, Inc.
Ilyce Ct 62,
Staten Island NY, 10306
USA