

1. SQL Prompt 5 documentation	3
1.1 What's new in SQL Prompt 5	4
1.2 SQL Prompt 5 and SQL Refactor	6
1.3 Requirements	7
1.4 Installing and running SQL Prompt	8
1.5 Licensing	9
1.5.1 Activating	10
1.5.2 Deactivating	16
1.5.3 Troubleshooting licensing and activation	19
1.6 Upgrading	22
1.6.1 Using Check for Updates	23
1.6.2 Troubleshooting Check for Updates errors	25
1.7 Speeding up your queries	26
1.7.1 Inserting suggestions into your code	27
1.7.2 Inserting columns with the column picker	36
1.7.3 Working with aliases	39
1.7.4 Inserting snippets	41
1.7.5 Managing snippets	43
1.7.6 Using SSMS templates in SQL Prompt snippets	45
1.7.7 Using keyboard shortcuts	47
1.8 Formatting your code	48
1.8.1 Expanding wildcards	49
1.8.2 Qualifying object names	50
1.8.3 Upper casing keywords	51
1.8.4 Using styles to format your code	52
1.8.5 Importing and exporting styles	54
1.9 Navigating your code	55
1.9.1 Using the object definition box and object tooltips	56
1.9.2 Scripting an object as ALTER	60
1.9.3 Finding invalid objects	61
1.9.4 Finding unused variables and parameters	63
1.9.5 Summarizing a script	64
1.10 Refactoring your code	66
1.10.1 Renaming an object	67
1.10.2 Encapsulating SQL as a new stored procedure	69
1.10.3 Splitting a table	71
1.10.3.1 Preparing the table in detail	76
1.10.3.2 What happens when I run the split table script?	78
1.11 Managing SQL Prompt behavior	81
1.11.1 Customizing inserted code	84
1.11.2 Options for formatting your code	86
1.11.3 Managing connections and memory	88
1.11.4 Changing the location of the settings, cache files and snippets folders	91
1.11.5 SQL Prompt Labs	92
1.12 Worked examples	98
1.13 Troubleshooting	105
1.13.1 Accessing the SQL Prompt log files	106
1.13.2 Configuring permissions for SQL Prompt	107
1.13.3 Error messages	108
1.13.3.1 Error 80070002 - The Add-in SQL Prompt failed to load or caused an exception	109
1.13.3.2 Using Layout SQL crashes SQL Server Management Studio	110
1.13.4 Common issues	111
1.13.4.1 SQL Prompt is not showing all linked servers	112
1.13.4.2 Removing SQL Prompt menu item after uninstalling	113
1.13.4.3 SQL Server Management Studio runs slowly after uninstalling SQL Prompt	114
1.13.5 Older versions of SQL Prompt	115
1.13.5.1 Unable to cast object of type 'System.DBNull' to type 'System.String'	116
1.13.5.2 SQL Prompt is showing an old version of my stored procedure	117
1.13.5.3 Add-in failing to load into SQL Server Management Studio on Vista	118
1.13.5.4 Candidate suggestions being displayed slowly	119
1.13.5.5 Suggestions to improve performance for pre-formatted SQL	120
1.14 Release notes and other versions	121
1.14.1 SQL Prompt 5.3 release notes	122
1.14.2 SQL Prompt 5.2 release notes	124
1.14.3 SQL Prompt 5.1 release notes	125
1.14.4 SQL Prompt 5.0 release notes	126
1.14.5 SQL Prompt 4.0 release notes	127
1.14.6 SQL Prompt 3.9 release notes	128
1.14.7 SQL Prompt 3.8 release notes	129
1.14.8 SQL Prompt 3.6 release notes	130
1.14.9 SQL Prompt 3.5 release notes	131
1.14.10 SQL Prompt 3.1 release notes	132

SQL Prompt 5 documentation

About SQL Prompt

SQL Prompt is an add-in for SQL Server Management Studio and Visual Studio that allows you to be more productive when you work with SQL code. You can use SQL Prompt to write code faster, format your code, navigate your code, and refactor your code.

For more information, see the [SQL Prompt product page](#).

Quick links

[What's new in SQL Prompt 5?](#)

[Release notes for SQL Prompt 5.3](#)

[SQL Prompt quick reference guide \(PDF\)](#)

SQL Refactor has been retired with the release of SQL Prompt 5. For more information, see: [SQL Prompt 5 and SQL Refactor](#).

What's new in SQL Prompt 5

This page describes the following new features:

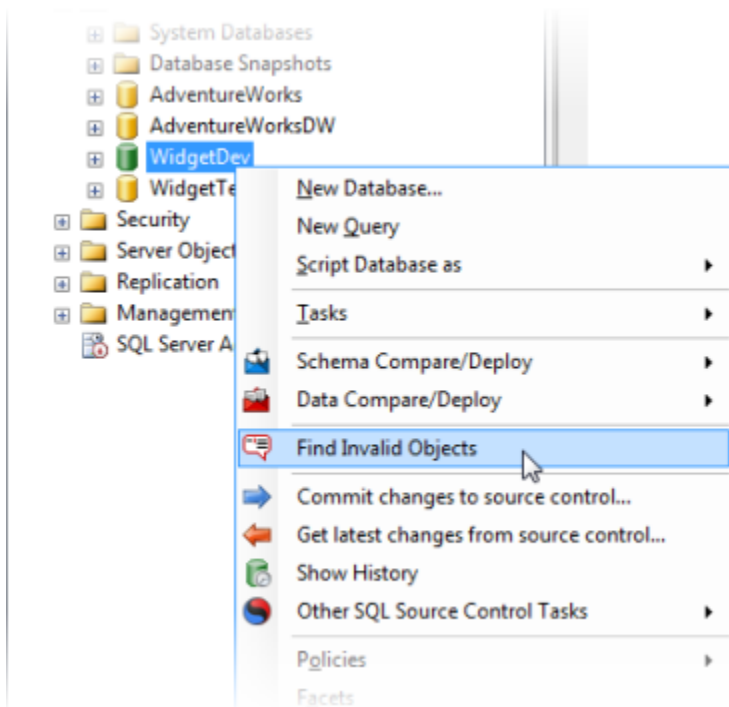
- [Find Invalid Objects](#)
- [Script Object as ALTER](#)
- [Column Dependencies](#)

Much of the functionality of the retired SQL Refactor is now available in SQL Prompt; for details, see: [SQL Prompt and SQL Refactor](#).

Find Invalid Objects

SQL Prompt 5 enables you to easily find invalid objects in your database. This can be useful, for example, if you have inherited a legacy database and you want to know if objects are referencing other objects that no longer exist in the database.

To try it out: in the SQL Server Management Studio Object Explorer, right-click the database you want to find invalid objects in, then click **Find Invalid Objects**:



For more information, see: [Finding invalid objects](#).

Script Object as ALTER

If you are navigating a query that contains a reference to an object, SQL Prompt 5 enables you to script out that object directly from the query window.

To try it out: right-click an object in a query, and click **Script Object as ALTER**.

Error rendering macro 'multimedia' : null

Keyboard shortcut: with the mouse insertion point on the name of the object you want to script out, press **F12**.

For more information, see: [Scripting an object as ALTER](#).

Column Dependencies

SQL Prompt 5 enables you to view a list of objects that reference a column.

To try it out: move the mouse pointer over a column in your query, and click the tooltip.

Error rendering macro 'multimedia' : null

For more information, see: [Viewing column dependencies](#).

SQL Prompt 5 and SQL Refactor

SQL Refactor has been retired with the release of SQL Prompt 5.

If SQL Refactor is installed on your computer, it will be uninstalled when you install SQL Prompt 5.

Much of the functionality of SQL Refactor is now available in SQL Prompt; including:

- Smart Rename (see: [Renaming an object](#))
- Qualify Object Names (see: [Qualifying object names](#))
- Uppercase Keywords in SQL Prompt 5.1 and later (see: [Uppercasing keywords](#))
- Expand Wildcards (see: [Expanding wildcards](#))
- Split Table (see: [Splitting a table](#))
- Encapsulate as New Stored Procedure (see: [Encapsulating SQL as a new stored procedure](#))
- Find Unused Variables and Parameters (see: [Finding unused variables and parameters](#))
- Summarize Script (see: [Summarizing a script](#))

For legacy support information for SQL Refactor 1.3, see [SQL Refactor 1.3 documentation](#).

Requirements

SQL Prompt is an add-in for the following development environments:

- SQL Server Management Studio 2012
- SQL Server Management Studio 2008 R2
- SQL Server Management Studio 2008
- SQL Server Management Studio 2005
- Visual Studio 2012 (SQL Prompt 5.3 and later)
- Visual Studio 2010
- Visual Studio 2008
- Visual Studio 2005

Microsoft SQL Server support

SQL Prompt supports connecting to the following versions of SQL Server:

- SQL Server 2012
- SQL Server 2008
- SQL Server 2008 R2
- SQL Server 2005
- SQL Server 2000
- SQL Server Express Edition
- SQL Server on Amazon RDS

Full support for all new features in SQL Server 2012 is in progress.

Microsoft Windows support

SQL Prompt supports the following Microsoft Windows operating systems:

- Windows Server 2008
- Windows 7
- Windows Vista
- Windows Server 2003
- Windows XP

Other requirements

- **Microsoft .NET Framework 3.5 or later**
- MDAC 2.8 or later
- 256 MB RAM
- 500 MB hard disk space available during installation

Installing and running SQL Prompt

To install SQL Prompt, you must have administrator privileges. If you are upgrading from a previous version, the installer does all the preparation for you.

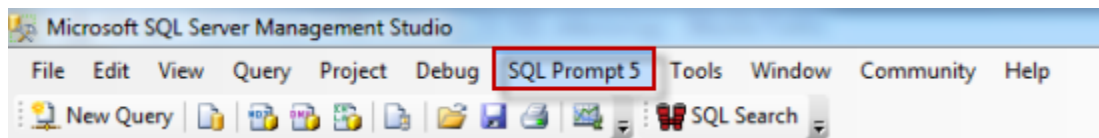
SQL Refactor has been retired with the release of SQL Prompt 5. If SQL Refactor is installed on your computer, it will be uninstalled when you install SQL Prompt 5.

Running SQL Prompt

When you install SQL Prompt, add-ins to the following editors are automatically installed:

- SQL Server Management Studio
- SQL Server Management Studio 2008 Express Edition
- Microsoft Visual Studio 2005, 2008, 2010
- Microsoft Visual Studio 2012 (SQL Prompt 5.3 only)

Start any of these editors to begin using SQL Prompt. SQL Prompt features are available from the **SQL Prompt** menu:



Customizing SQL Prompt

You can customize how SQL Prompt suggests, inserts and formats your code, to more closely match the way you prefer to write queries. To change any of the default settings in SQL Prompt:

On the **SQL Prompt** menu, select **Options**.

See: [Managing SQL Prompt behavior](#) and [Options for formatting your code.](#)

Activating SQL Prompt

When your trial period expires, you need to enter your serial number to activate the product:

On the **SQL Prompt** menu, select **Serial Number > Enter Serial Number**.

Find out more about activation

Uninstalling SQL Prompt

To uninstall SQL Prompt, use the Windows Control Panel item, **Add or Remove Programs** or **Programs and Features**.

If the SQL Prompt menu item is not removed from SQL Server Management Studio or Visual Studio, see [Removing SQL Prompt menu item after uninstalling](#).

Using SQL Prompt with Visual Studio

SQL Prompt 5 can be integrated with Visual Studio 2005, 2008, 2010 and 2012 (VS 2012 integration only available in SQL Prompt 5.3). SQL Prompt only works in Database projects and SQL Server projects.

Licensing

When you install most Redgate products (apart from free ones), you have **14 days** to evaluate them without purchase.

For a few products, you have 28 days: DLM Automation Suite, DLM Automation Suite for Oracle, SQL Prompt, SQL Source Control, Source Control for Oracle.

If you need more time to evaluate a product, email licensing@red-gate.com.

Finding your serial number

When you buy a license for a product, we'll send you an invoice that contains your serial number to activate the product. Your invoice shows how many instances of a product the serial number can be used to activate. For information about how to activate, see [Activating](#).

If you can't find your invoice, you can view your serial numbers at red-gate.com/myserialnumbers. You'll need to log in to your Redgate account with the email address and password you provided when you bought the product.

If you need to reinstall products on the same computer (eg after installing a new operating system), you can reactivate them using the same serial number. This doesn't affect the number of distinct activations for the serial number. For information about moving a serial number to a different computer, see below.

Serial numbers for bundles and suites

If you've bought a bundle or suite of products, your serial number activates all the products in the bundle or suite. For bundles containing both server and client tools (such as the SQL DBA Bundle) you will have two serial numbers.

If you deactivate a bundle or suite serial number, all products using that serial number will be deactivated.

For information on which products are included in a bundle, see [Bundle history](#).

Changing the serial number used to activate a product

To change the serial number used to activate a product, on the **Help** menu, select **Enter Serial Number**. For some products, you will need to deactivate the old serial number first.

Moving a serial number to a different computer

To move a serial number to a different computer, deactivate the serial number on the old computer, then use it to activate the product on the new computer.

To deactivate a serial number, on the **Help** menu, select **Deactivate Serial Number**. If the Deactivate Serial Number menu item isn't available, use the [deactivation tool](#).

If you can't deactivate a serial number, use the [Request Extra Activations](#) page to request more activations for your serial number. You'll need to provide your serial number and the reason for the additional activations.

Activating

This page applies to a number of Redgate products, so the screenshots below may not match your product.

When you activate a product with your serial number, the licensing and activation program sends an activation request to the Redgate activation server, using checksums of attributes from your computer. The checksums sent to the activation server do not contain any details that might pose a security risk. The activation server returns an activation response and an encrypted key to unlock the software. The licensing and activation program should activate your product within a few seconds.

If you experience problems with activating your products, you'll be directed to [activate manually](#).

- [Activating using the GUI](#)
- [Activating using the command line](#)
- [Manual activation](#)

Activating using the GUI

These instructions apply to a number of Redgate products, so the screenshots below may not match your product.

To activate your products:

1. On the **Help** menu, click **Enter Serial Number**.
The product activation dialog box is displayed, for example:



Activate SQL Compare

Enter your SQL Compare serial number

Serial number

Your serial number is on your invoice or you can [find it online](#)

☒ **Track this activation**
Sends information about this activation (including your machine name) to Red Gate.
This is useful if you contact support about your activations. [More information](#)


If you purchased SQL Compare as part of a bundle, other products may be activated by this process. The products activated are listed when activation is completed.

E-mail (optional)
Please provide the email address you would like us to send update notifications to:

☒ **I'd also like to receive the Red Gate Newsletter.** [Read our privacy policy](#)

redgate

Activate Cancel

2. Enter your serial number.
When you have entered a valid serial number,

is displayed next to the serial number box:

Activate SQL Compare

Enter your SQL Compare serial number

Serial number
000-000-123456-0000 ✓

Your serial number is on your invoice or you can [find it online](#)

☒ **Track this activation**
Sends information about this activation (including your machine name) to Red Gate.
This is useful if you contact support about your activations. [More information](#)

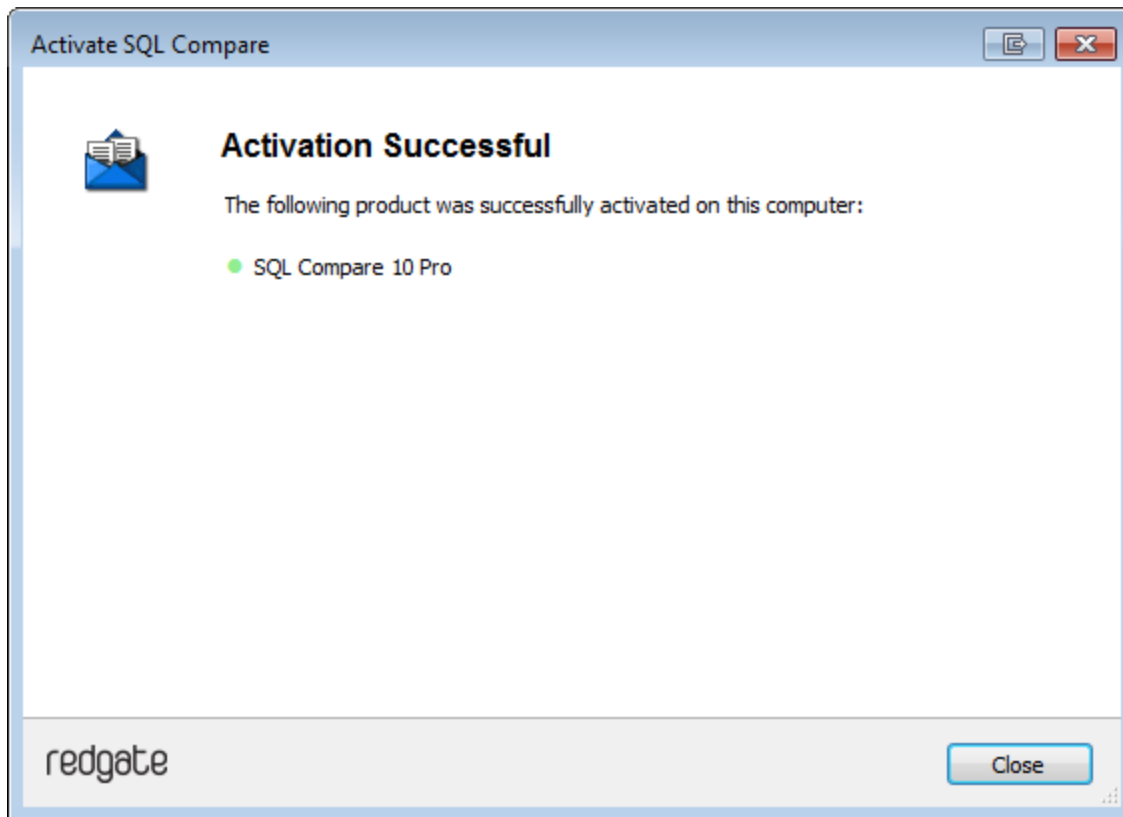
If you purchased SQL Compare as part of a bundle, other products may be activated by this process. The products activated are listed when activation is completed.

E-mail (optional)
Please provide the email address you would like us to send update notifications to:
user@example.com

☒ **I'd also like to receive the Red Gate Newsletter.** [Read our privacy policy](#)

redgate Activate Cancel

3. If you want to receive email updates from Redgate, enter your email address.
The list of identifiers and your email address may already be populated using information available to the licensing client from the Windows installation on your computer. No information is sent back to Redgate when the fields are populated.
When you activate your product, the optional information you entered is recorded by Redgate with your serial number. Your email address is not linked to the data collected should you consent to participate in the Quality Improvement Program provided with some Red Gate products.
4. Click **Activate**.
Your activation request is sent to the Red Gate activation server.
When your activation has been confirmed, the **Activation successful** page is displayed, for example:



If there is a problem with your activation request, an error dialog box is displayed. For information about activation errors and what you can do to resolve them, see [Troubleshooting licensing and activation errors](#). Depending on the error, you may want to try [manual activation](#).

5. Click **Close**.
You can now continue to use your product.

Activating using the command line

Open a command prompt, navigate to the folder where your product executable file is located and run a command with the following syntax:

```
<name of productEXE> /activateSerial:<serialNumber>
```

For example:

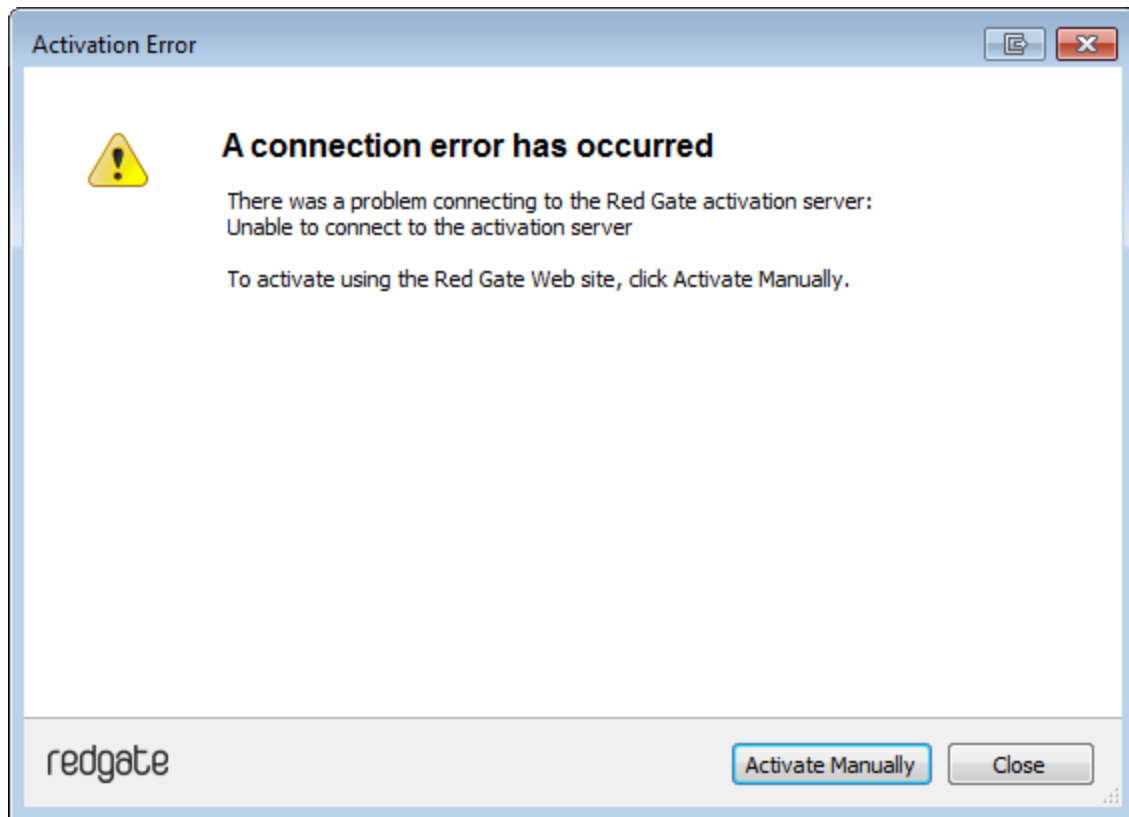
```
sqlcompare /activateSerial:123-456-789012-ABCD
```

The product activation dialog box is displayed. Follow the instructions below.

Manual activation

Manual activation enables you to activate products when your computer does not have an internet connection or your internet connection does not allow SOAP requests. You will need access to another computer that does have an internet connection.

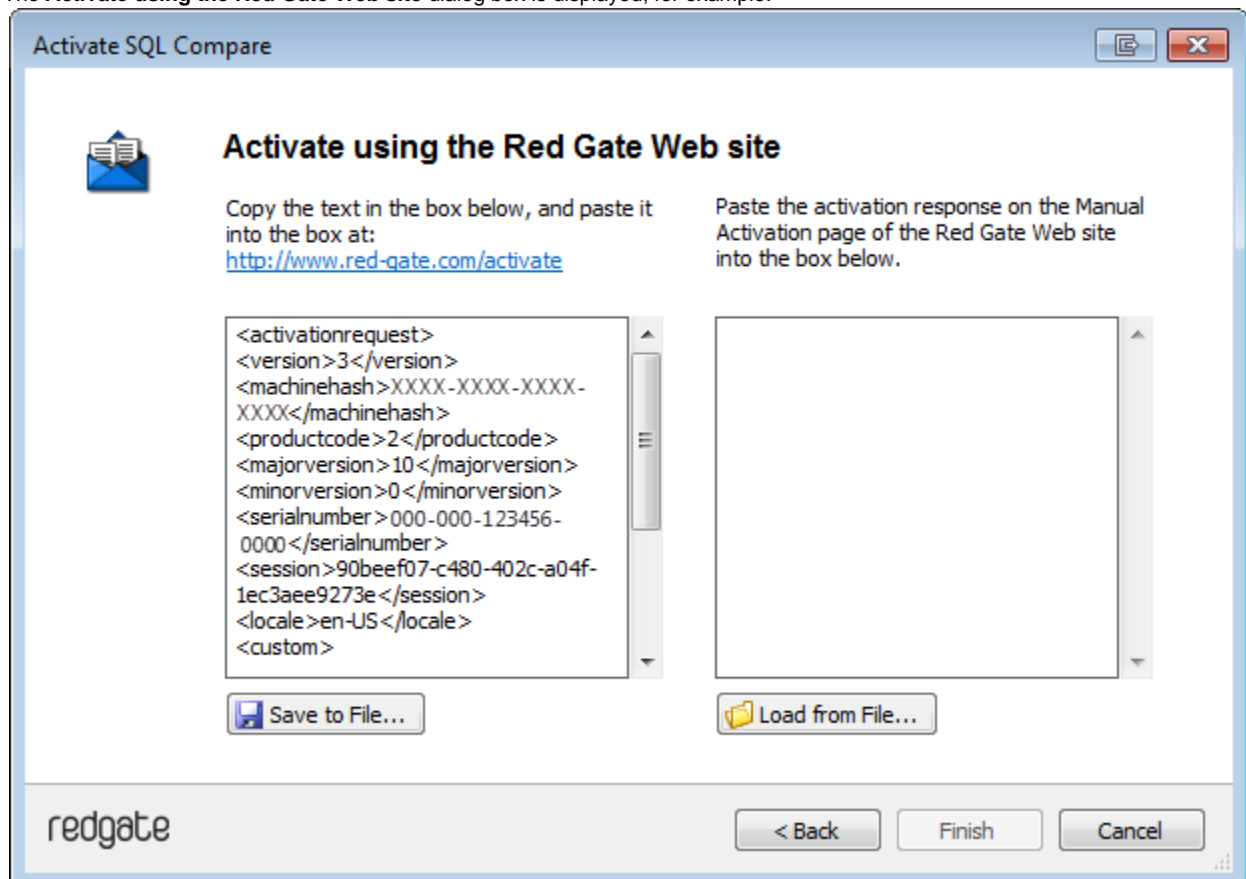
You can use manual activation whenever the **Activation Error** dialog box is displayed and the **Activate Manually** button is available, for example:



To activate manually:

1. On the error dialog box, click **Activate Manually**.

The **Activate using the Red Gate Web site** dialog box is displayed, for example:



2. Copy all of the activation request, and **leave this dialog box open** (if you close the dialog box, you may have to start again). Alternatively you can save the activation request, for example to a location on your network or to a USB device.
3. On a computer that has an Internet connection, go to the **Manual Activation** page at <http://www.red-gate.com/activate> and paste the activation request into the box under **Step 1**.

Account ▾ Quotes Shopping Cart

redgate
ingeniously simple tools

Home Products Store Community Support Our Company

I'm looking for... 🔍

Manual Activation

Use the activation request from the licensing program to generate an activation response so that you can activate products on your computer.

Step 1

Paste the activation request into the box below. Make sure you paste all of the text.

```
<activationrequest>
<version>3</version>
<machinehash>XXXX-XXXX-XXXX-XXXX</machinehash>
<productcode>2</productcode>
<majorversion>10</majorversion>
<minorversion>0</minorversion>
<serialnumber>000-000-123456-0000</serialnumber>
<session>90beef07-c480-402c-a04f-1ec3aee9273e</session>
<locale>en-US</locale>
<custom>
```

Get Activation Response

Step 2

Copy the contents of this box into your product activation dialog box.


Save to File...

Got a question?

0800 169 7433
shop@red-gate.com

4. Click **Get Activation Response**.
5. When the activation response is displayed under **Step 2**, copy all of it. Alternatively you can save the activation response to a .txt file.
6. On the computer where the licensing and activation program is running, paste the activation response or if you saved it, load it from the file.


Activate SQL Compare

 **Activate using the Red Gate Web site**


Copy the text in the box below, and paste it into the box at:
<http://www.red-gate.com/activate>

Paste the activation response on the Manual Activation page of the Red Gate Web site into the box below.

```
<activationrequest>
<version>3</version>
<machinehash>XXXX-XXXX-XXXX-XXXX</machinehash>
<productcode>2</productcode>
<majorversion>10</majorversion>
<minorversion>0</minorversion>
<serialnumber>000-000-123456-0000</serialnumber>
<session>90beef07-c480-402c-a04f-1ec3aee9273e</session>
<locale>en-US</locale>
<custom>
```

 Save to File...

```
<activationresponse>
<data>
<machinehash>XXXX-XXXX-XXXX-XXXX</machinehash>
<version>3</version>
<productcode>2</productcode>
<majorversion>10</majorversion>
<minorversion>0</minorversion>
<edition>professional</edition>
<userspurchased>1</userspurchased>
<serialnumber>000-000-123456-0000</serialnumber>
```

 Load from File...

redgate

< Back Finish Cancel

7. Click **Finish**.
The **Activation successful** page is displayed.
8. Click **Close**.
You can now continue to use your product.

Deactivating

This page applies to several Redgate products, so the screenshots below may not match your product.



Download deactivation tool

You can use the deactivation tool to deactivate a serial number so you can reuse it on another computer. You can also use it to deactivate serial numbers for products you've uninstalled.

When you deactivate a serial number for a bundle of products, all the products in the bundle are deactivated. For information about what products are in your bundle, see [Bundle history](#).

To deactivate a serial number, your computer must have an internet connection. If you can't deactivate a serial number, you can [request additional activations](#) for that serial number. You may need to do this if:

- your computer doesn't have an internet connection
- your network uses a proxy server that interrupts contact between the product and the Redgate activation server
- your serial numbers aren't displayed in the deactivation tool (eg if the product installation is corrupted)

Deactivating using the command line

Open a command prompt, navigate to the folder where your product executable file is located and run a command with the following syntax:

```
<productEXE> /deactivateSerial
```

For example:

```
sqlcompare /deactivateSerial
```

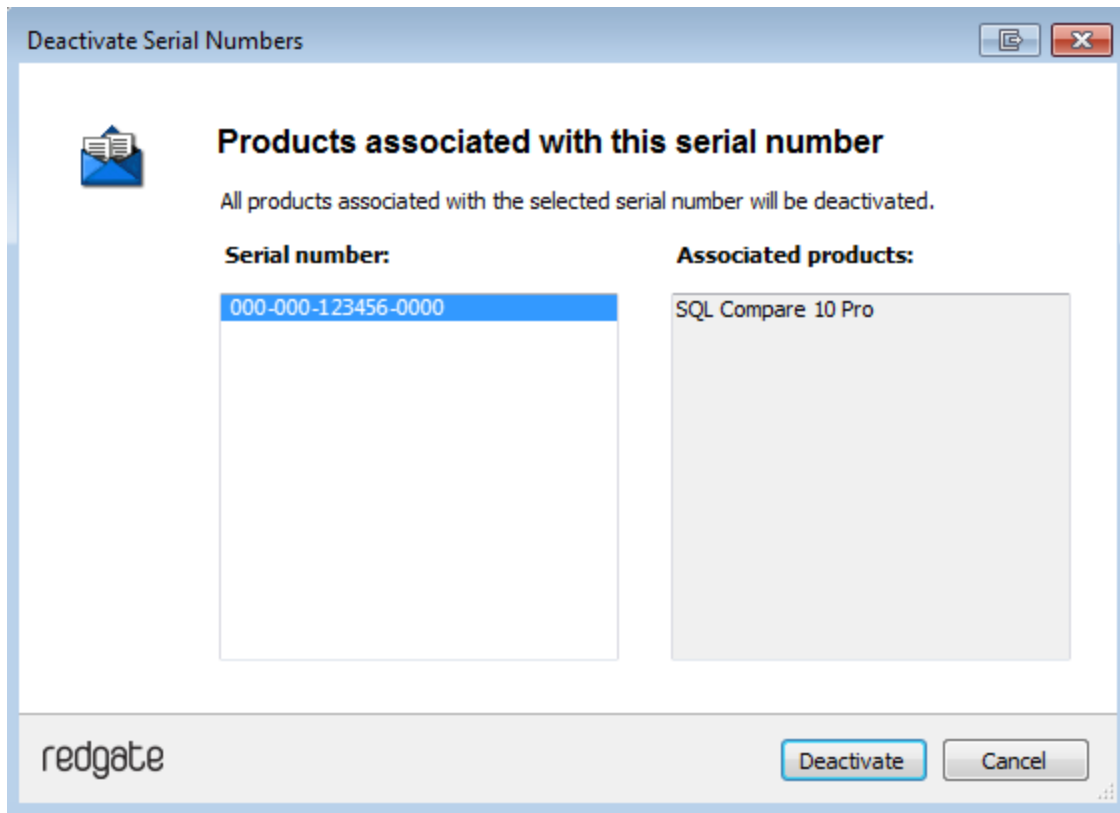
The **Deactivate Serial Numbers** dialog box is displayed. Follow the instructions below.

Deactivating using the GUI

To deactivate your products:

1. Start the deactivation tool. To do this, either [download](#) the tool and run the executable file, or on the **Help** menu of the product, click **Deactivate Serial Number**.

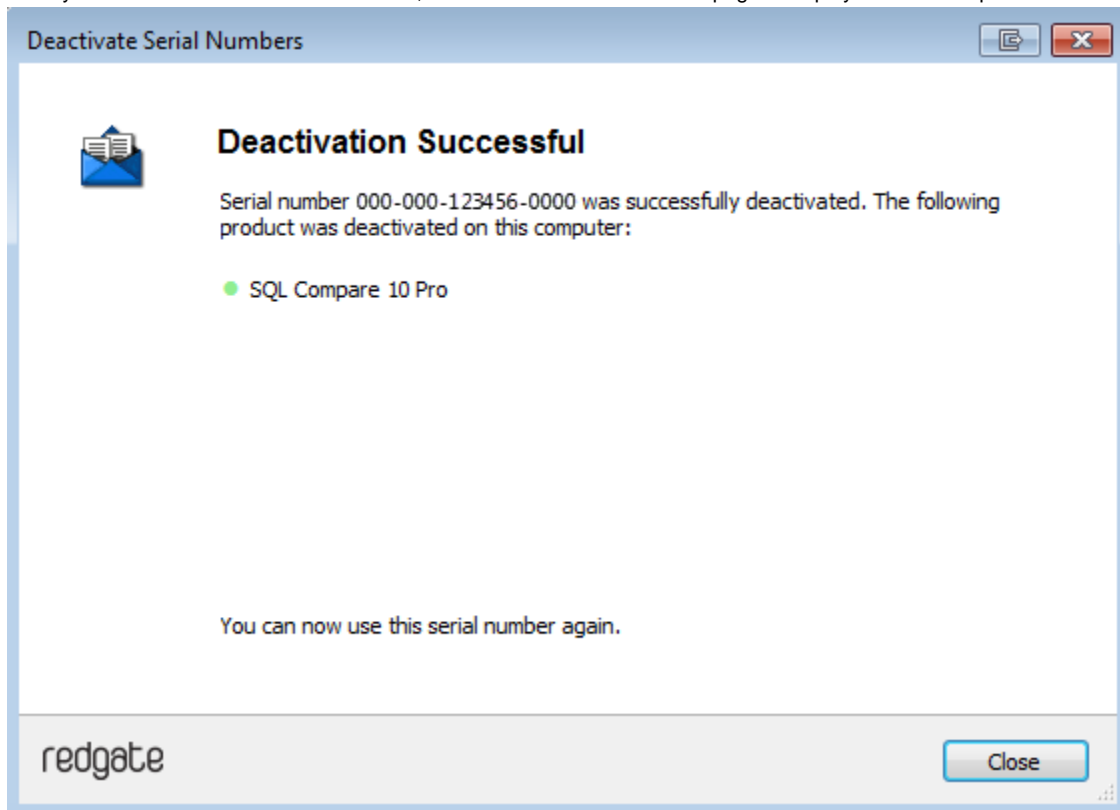
The **Deactivate Serial Numbers** dialog box is displayed. For example:



If you're running the executable file, the dialog box displays all the serial numbers for Red Gate products that have been activated on your computer.

If the serial number is for a bundle, all the products in the bundle are displayed under **Associated products**.

2. Select the serial number you want to deactivate and click **Deactivate**.
Your deactivation request is sent to the Red Gate activation server.
3. When your deactivation has been confirmed, the **Deactivation successful** page is displayed. For example:



If there's a problem with your deactivation request, an error dialog box is displayed. For information about deactivation errors and how to resolve them, see [Troubleshooting licensing and activation errors](#).

4. Click **Close**. You can now use this serial number on a different computer.

Troubleshooting licensing and activation

This page provides information about errors you may encounter when you activate Redgate products:

- [The number of activations for this serial number has been exceeded](#)
- [This serial number has been disabled](#)
- [This serial number was for a trial extension](#)
- [This serial number is not registered with the activation server](#)
- [This serial number is not for <product name>](#)
- [This serial number is not for this version](#)
- [The activation request is in the wrong format](#)
- [The activation request contains an invalid machine hash](#)
- [The activation request contains an invalid session](#)
- [The activation request contains an invalid serial number](#)
- [The activation request contains an invalid product code or version number](#)
- [There's a problem deactivating your serial number](#)
- [This serial number is not activated on this computer](#)
- [Products not activated on this computer](#)

The number of activations for this serial number has been exceeded

This error message is displayed when a serial number is activated on more computers than the number of licenses that were purchased for that serial number.

When you purchase products from Redgate, we send you an invoice that includes your serial numbers. The serial numbers enable you to activate the software a number of times, depending on how many licenses you purchased and the terms in the [license agreement](#). When this limit is reached, you will see this error message.

To fix the problem, you can:

- [deactivate](#) the product on another computer to free up a license
- [purchase](#) more licenses
- [request additional activations](#) for your serial number

This serial number has been disabled

This error message is displayed when you try to activate a product using a serial number that Redgate has disabled.

When you upgrade a product, your existing serial numbers will be disabled and we will issue new ones with your invoice. If you cannot find your new serial numbers, you can review them at <http://www.red-gate.com/myserialnumbers>

Redgate will also disable serial numbers for non-payment of invoices or breach of the terms in the [license agreement](#). If you think we have disabled your serial numbers in error, email licensing@red-gate.com

This serial number was for a trial extension

This error message is displayed when you have requested a trial extension and you try to reuse the serial number that was provided for the trial extension; trial extensions can be used one time only.

To continue using the product, you need to [purchase it](#).

This serial number is not registered with the activation server

This error message is displayed when the serial number you entered does not exist on the Redgate activation server.

To find out your serial numbers, check your invoice or go to <http://www.red-gate.com/myserialnumbers>

This serial number is not for <product name>

This error message is displayed when the serial number you entered is not for the product you are trying to activate.

To find out your serial numbers, check your invoice or go to <http://www.red-gate.com/myserialnumbers>

This serial number is not for this version

This error message is displayed when the serial number you entered is for a different version of the product you are trying to activate.

If the serial number is for an older version of the product, and you don't have that version installed on your computer, you can download it from the Release notes and other versions page.

If you want to upgrade to the latest version of the product, go to the [Upgrade center](#) to get a quote or purchase an upgrade, or email sales@red-gate.com.

The activation request is in the wrong format

This error message is displayed:

- if your internet connection does not allow SOAP requests.
Try using manual activation; on the error dialog box, click **Activate Manually**, and then follow the instructions that are displayed.
- if you are activating by email and there is a problem with the format of the activation request.
Check that you copied and pasted all of the activation request.
Alternatively, try using manual activation. Go to <http://www.red-gate.com/activate> and paste your activation request under **Step 1**.
- when you are using manual activation and there is a problem with the format of the activation request. If the format is incorrect, for example part of the request is missing, the Redgate activation server cannot process the request.
Check that you copied and pasted all of the activation request.

For more information about activating manually, see [Manual activation](#).

The activation request contains an invalid machine hash

This error message is displayed:

- if your internet connection does not allow SOAP requests.
Try using manual activation; on the error dialog box, click **Activate Manually**, and then follow the instructions that are displayed. For more information, see [Manual activation](#).
- when you are using manual activation and there is a problem with the format of the *machinehash* element in the activation request. The *machinehash* is a checksum of attributes from your computer. We use the *machinehash* to identify computers on which our products have been activated. If the format of the *machinehash* element is incorrect, the Redgate activation server cannot process the request.
Check that you copied and pasted the activation request correctly.

The activation request contains an invalid session

This error message is displayed:

- if your internet connection does not allow SOAP requests.
Try using manual activation; on the error dialog box, click **Activate Manually**, and then follow the instructions that are displayed. For more information, see [Manual activation](#).
- when you are using manual activation and there is a problem with the format of the activation request. If the format of the *session* element is incorrect, the Redgate activation server cannot process the request.
Check that you copied and pasted the activation request correctly.

The activation request contains an invalid serial number

This error message is displayed:

- if your internet connection does not allow SOAP requests.
Try using manual activation; on the error dialog box, click **Activate Manually**, and then follow the instructions that are displayed. For more information, see [Manual activation](#).
- when you are using manual activation and there is a problem with the format of the activation request. If the format of the serial number is incorrect, the Redgate activation server cannot process the request.
Check that you copied and pasted the activation request correctly.

The activation request contains an invalid product code or version number

This error message is displayed:

- if your internet connection does not allow SOAP requests.
Try using manual activation; on the error dialog box, click **Activate Manually**, and then follow the instructions that are displayed. For more information, see [Manual activation](#).
- when you are using manual activation and there is a problem with the format of the activation request. If the product code or version numbers are incorrect, the Redgate activation server cannot process the request.
Check that you copied and pasted the activation request correctly.

There's a problem deactivating your serial number

This error message is displayed if your computer is not connected to the internet or your internet connection does not allow SOAP requests. You cannot deactivate a serial number if your computer does not have an internet connection.

Try deactivating again later. If the problem persists, contact your system administrator.

If you require more activations because you cannot deactivate your serial number, you can request them on the [Request Extra Activations](#) page.

This serial number is not activated on this computer

This error message is displayed when you try to deactivate a serial number that has not been activated on your computer.

If you think the product installation on your computer is corrupt, you can try re-activating the product, and then deactivating the product again.

If you require more activations because you cannot deactivate your serial number, you can request them on the [Request Extra Activations](#) page.

Products not activated on this computer

This error message is displayed when you try to deactivate a serial number for a bundle of Redgate products and those products were not activated on your computer.

If you think the product installation on your computer is corrupt, you can try re-activating the product, and then deactivating the product again.

If you require more activations because you cannot deactivate your serial number, you can request them on the [Request Extra Activations](#) page.

Upgrading

Minor releases are free for all users. For example, if you have a license for version 7.0 of a product, you can upgrade to version 7.1 at no cost. When you download and install a minor release, the product is licensed with your existing serial number automatically.

Major releases are free for users with a current Support and Upgrades contract. For example, if you have a license for version 7 of a product, you can upgrade to version 8 at no cost. When you download and install a major release, the product is licensed with your existing serial number automatically.

If you don't have a current Support and Upgrades contract, installing a major release will start a free 14-day trial. You'll need to buy a new license and activate the product with your new serial number.

To check whether you have a current Support and Upgrades contract or see the cost of upgrading to the latest major version of a product:

- visit the [Upgrade Center](#)
- email sales@red-gate.com
- call:
 - 1 866 733 4283 (toll free USA and Canada)
 - 0800 169 7433 (UK freephone)
 - +44 (0)870 160 0037 (rest of world)

To check the latest version of a product, see [Current versions](#).

How to upgrade

You can download the latest version of a product using [Check for Updates](#), the [Upgrade Center](#), or the [Redgate website](#).

- If you download the latest version from the Upgrade Center or our website, you need to run the installer to upgrade the product.

Some Redgate products are available as part of bundle. You can select which products you want to upgrade when you run the installer.

- If you use Check for Updates, the installer runs automatically.

You can install the latest *major* version of any product (other than SQL Backup Pro) on the same machine as the previous version. For example, you can run version 9 and version 10 in parallel. However, installing a *minor* release will upgrade the existing installation.

To revert to an earlier version, uninstall the later version, then download and install the version you want from the Release notes and other versions page. You can use a serial number for a later version to activate an earlier version.

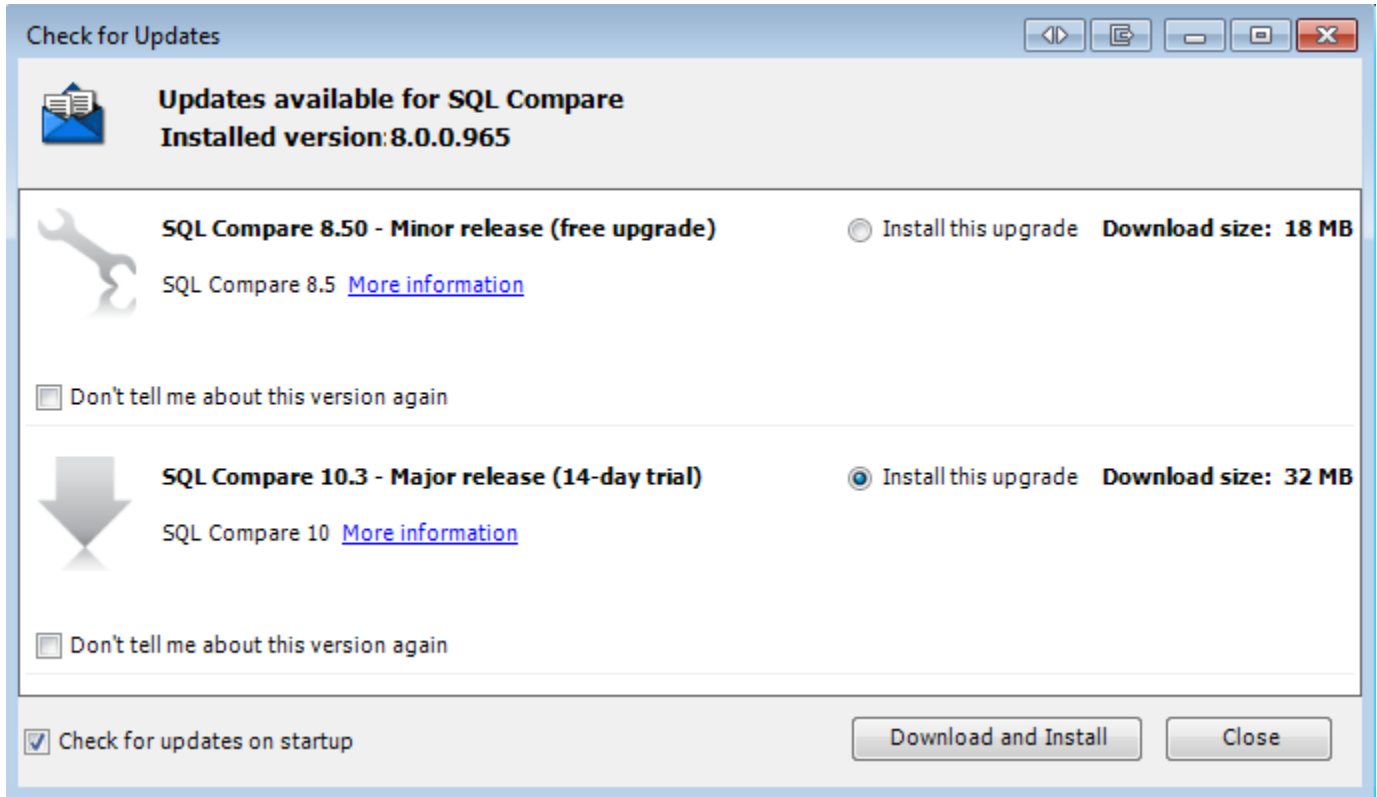
Using Check for Updates

This page applies to several Redgate products, so the screenshots below may not match your product.

The Check for Updates service checks whether a more recent version of the product is available to download. To use the service, your computer must have a connection to the internet. If your internet connection uses a proxy server, make sure your web browser connection settings are configured correctly.

The Check for Updates service doesn't work with automatic configuration scripts.

To check for updates for a Redgate product, on the **Help** menu, click **Check for Updates**. Any available updates are listed:



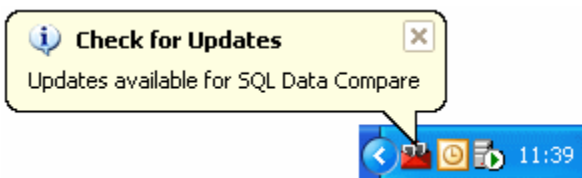
To view the full release details in your default web browser, click **More information**.

To get the update, click **Download and Install**. If you have a choice of updates, choose by selecting **Install this upgrade**, and then click **Download and Install**.

The installer will ask you to close the program. If you're upgrading an add-in, you'll also be asked to close the host program (SQL Server Management Studio, Visual Studio or Query Analyzer).

About the Check for Updates service

When you start the application, the Check for Updates service informs you automatically when there are updates available:



If you don't want to receive these notifications for the product, clear the **Check for updates on startup** check box.

If you don't want the Check for Updates service to inform you about a particular update again, select the **Don't tell me about this version again** check box. The Check for Updates service will still inform you of new updates when they become available.

Troubleshooting Check for Updates errors

For details about how to use the Check for Updates service, see [Using Check for Updates](#).

Error: There is a problem saving the download file to your computer

This error message is displayed if:

You don't have enough disk space

The Check for Updates service downloads the updates to the location defined by the *RGTEMP* environment variable, or the *TMP* variable if the *RGTEMP* variable doesn't exist.

If you don't have enough disk space, you can change the environment variable to a location with more space.

Changing the *RGTEMP* or the *TMP* variables will affect other programs that use those variables. The *RGTEMP* variable affects only Redgate programs. For information about environment variables, see your Windows documentation.

There's a problem with permissions on your computer

The Check for Updates service downloads the updates to the location defined by the *RGTEMP* environment variable, or the *TMP* variable if the *RGTEMP* variable does not exist. If your user account doesn't have permissions to write to the location specified by these environment variables, contact your system administrator.

There's a problem with the download file on the Redgate web server

Contact [Redgate support](#).

Error: There is a problem with the network connection

This error message is displayed if:

Your internet connection dropped while the Check for Updates service was downloading the updates

Try checking for updates again later.

Proxy authentication failed

Check your user name and password.

Your computer can't connect to the Check for Updates service.

Contact your system administrator. If you're using a proxy server, check it's configured correctly (see Control Panel > Internet Options > Connections).

The Check for Updates service doesn't work with automatic configuration scripts.

There's a problem with the download file on the Redgate web server

Contact [Redgate support](#).

Speeding up your queries

There are several features in SQL Prompt to help you write faster, more accurate queries:

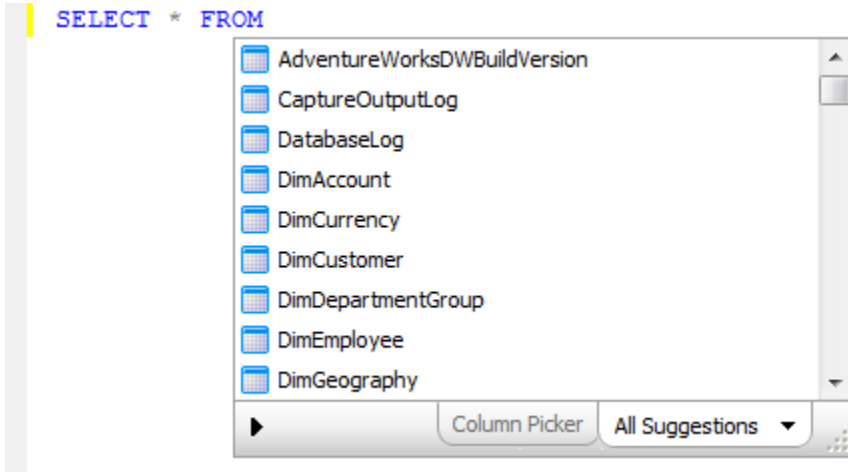
- [Inserting suggestions into your code](#)
- [Inserting columns with the column picker](#)
- [Working with aliases](#)
- [Inserting snippets](#)
- [Managing snippets](#)
- [Using SSMS templates in SQL Prompt snippets](#)
- [Using keyboard shortcuts](#)

If you want to start using SQL Prompt, [try out the worked example](#).

Inserting suggestions into your code

SQL Prompt provides SQL code auto-completion using the *suggestions box*. The suggestions box pops up to display items based on what you type in your query editor.

For example, when you type `SELECT * FROM`, the suggestions box displays a list of tables first, then views, and further down the list, owners, databases, functions, and so on.



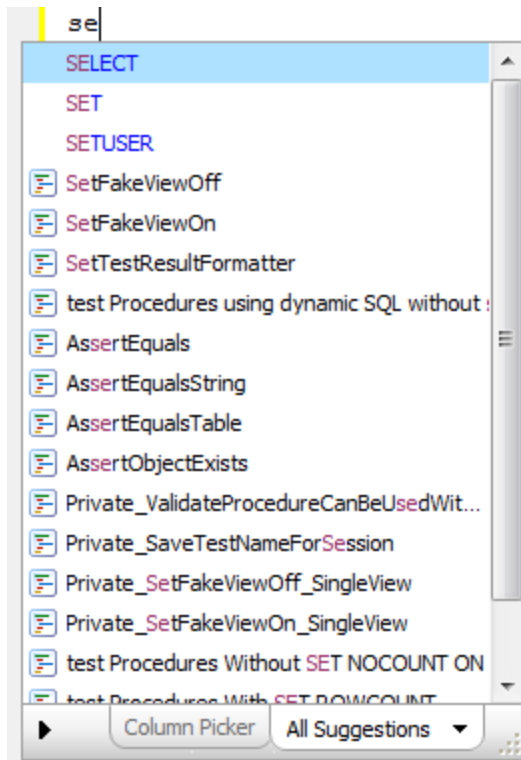
The suggestions box is displayed automatically when you type in a query window. As you type, the suggestions are filtered to match the typed characters.

Types of suggestions

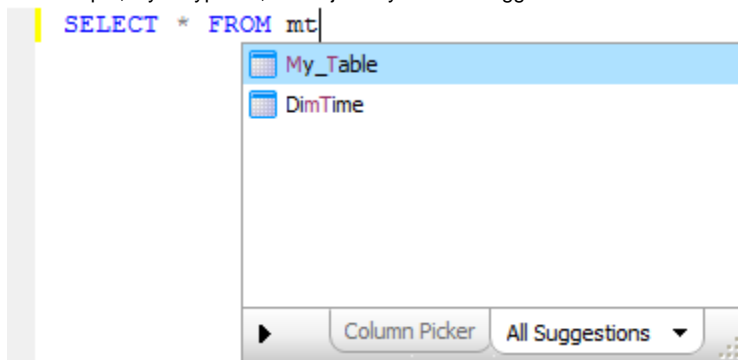
Only prefix suggestions are available before SQL Prompt 5.3.

Suggestions are listed in the following order in the suggestions box:

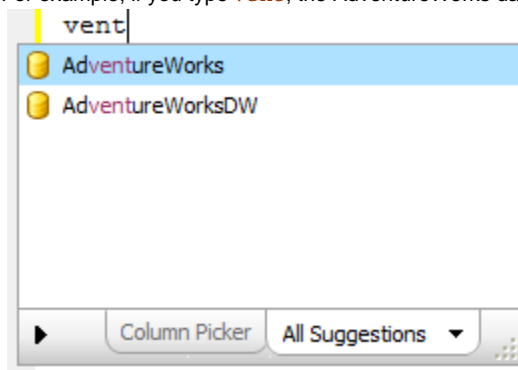
1. Prefix suggestions
For example, if you type `se`, the `SELECT` keyword is suggested:



2. CamelCase, or compound word suggestions
For example, if you type `mt`, the object `MyTable` is suggested:



3. Mid-string suggestions
For example, if you type `vent`, the AdventureWorks database is suggested:



Mid-string suggestions are particularly useful if you can only remember part of an object name.

To insert the currently highlighted suggestion, press the Tab or Enter key, or any of the defined insertion keys. You can customize which keys insert suggestions, see: [Managing SQL Prompt behavior](#).

Closing the suggestions box

To close the suggestions box without inserting anything, press the Esc key. If nothing is selected in the suggestions box, you can also press the Enter key to close it.

You can also click anywhere in your query window to close the suggestions box.

Order of suggestions

The order in which suggestions are listed depends on the context of your query. For example:

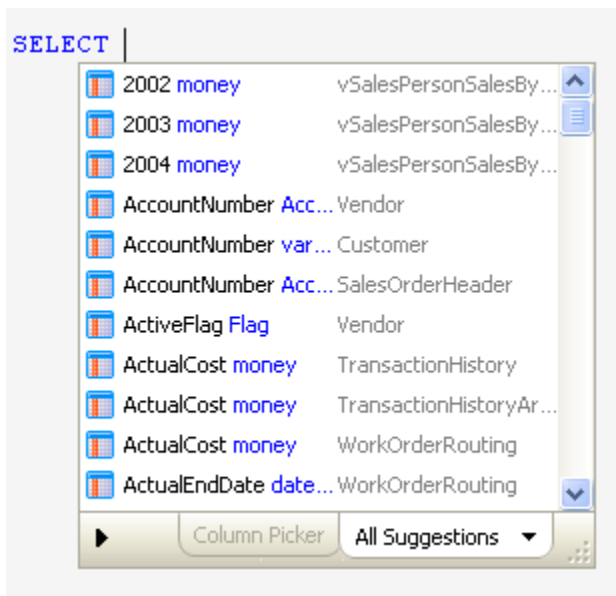
- After typing **USE**, databases are listed at the top of the suggestions box.
- After typing **SELECT * FROM**, tables are listed at the top of the suggestions box, followed by views, schemas and then database names.
- In a **CREATE TABLE** statement, after typing a column name, data types are listed at the top of the suggestions box.

Viewing all columns after a SELECT statement

After typing **SELECT**, SQL Prompt does not by default list all the possible columns from all tables. If you want to see all column names after typing **SELECT**:

1. From the **SQL Prompt** menu, click **Options**.
2. Go to the **Suggestions > Types of suggestion** page.
3. Select **List all database columns after a SELECT statement**.

When you type **SELECT**, all column names will now be displayed in the suggestions box, listed alphabetically:



Only displaying the suggestions box manually

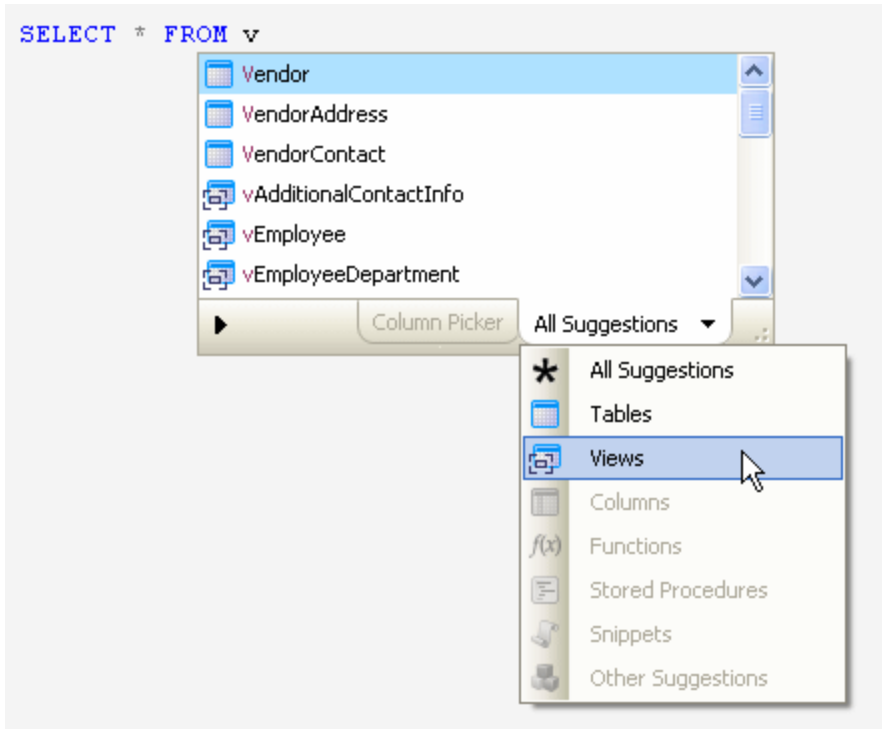
To turn off the automatic display of the suggestions box:

1. On the **SQL Prompt** menu, select **Options**.
2. On the **Main > Behavior** page (displayed by default when you first display the SQL Prompt Options dialog box), clear the **Automatically trigger suggestions** check box.

When automatic display of the suggestions box is turned off, you can press Ctrl + Space at any time to display it manually. The type of suggestions displayed will depend on where in the query you display the suggestions box.

Using categories

The suggestions box displays items grouped by category. Categories are a way of organizing suggestions so you can find the item you require more easily:



Only the categories that contain suggestions are available to select.

To switch between categories, click the **All Suggestions** drop-down or press Ctrl + Up arrow or Ctrl + Down arrow.

The following categories are shown:



Tables lists tables from the current database.



Views lists views from the current database.



Columns lists column names, with their data types and associated table names (or table aliases). Primary key



and foreign key



columns are also shown.

You can also use the [column picker](#) to insert columns.



Functions lists user-defined functions in the current database, and built-in functions.



Stored Procedures lists stored procedures from the current database.



Snippets lists shortcuts for inserting pre-defined SQL fragments or statements.

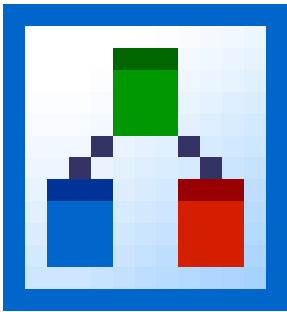

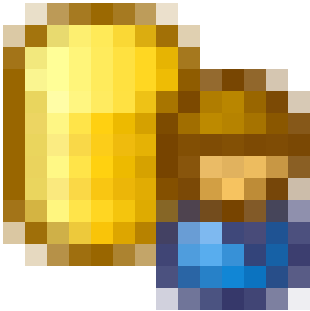
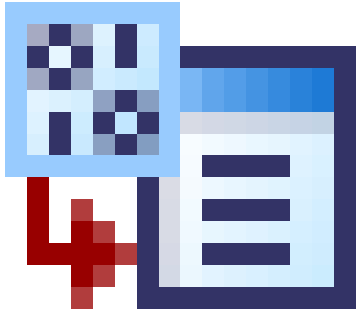

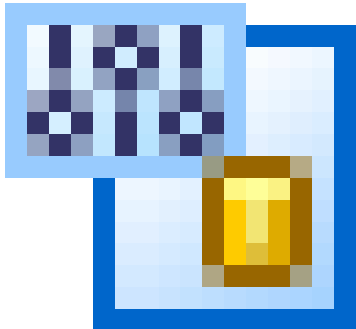

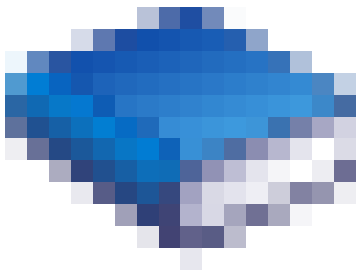

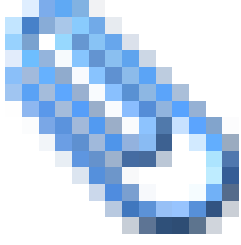
For more information about snippets, see [Inserting snippets](#).

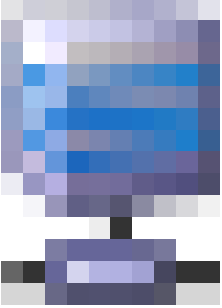


Other Suggestions lists keywords, data types, and objects such as users and roles.


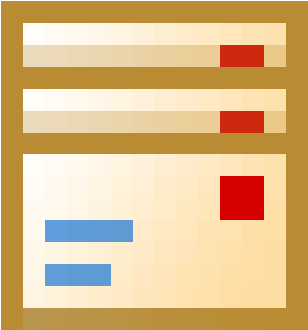
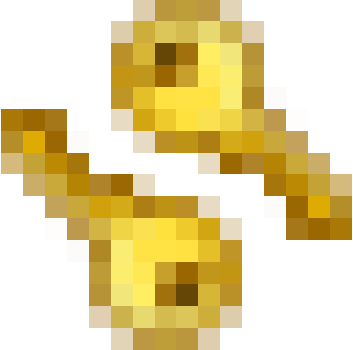
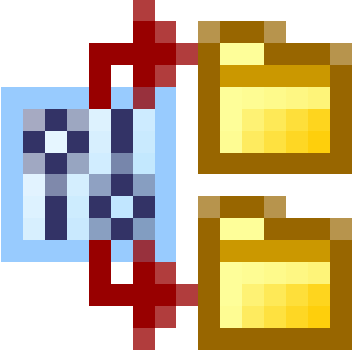

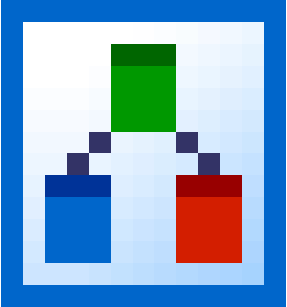
Object types

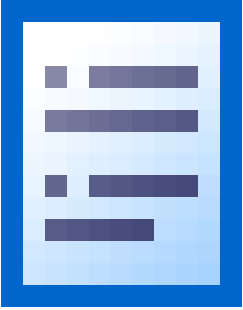

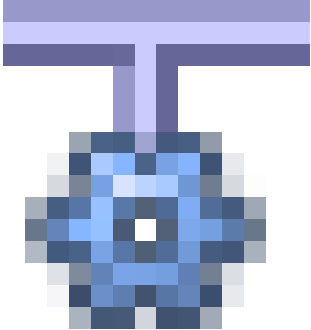
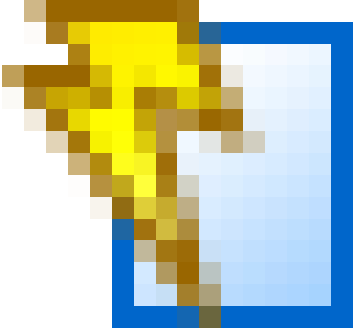
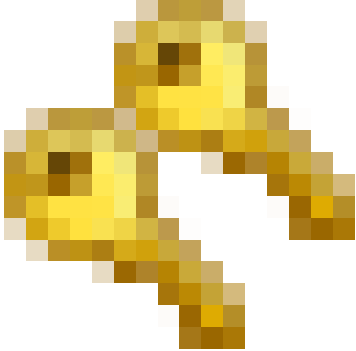
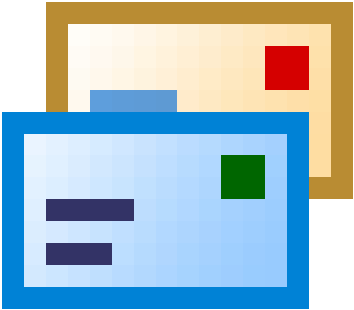
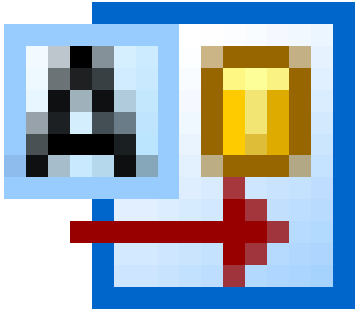
You may see the following types of object or suggestion listed in **Other Suggestions**:

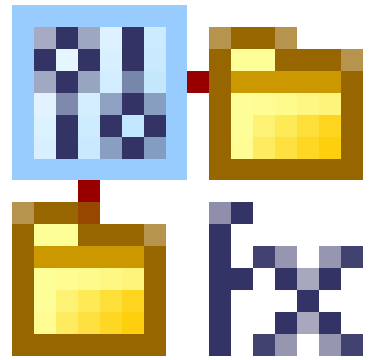
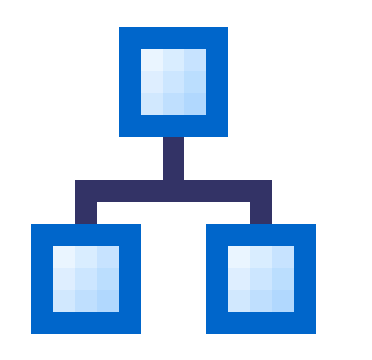
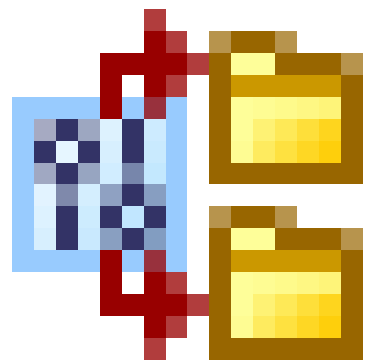

	DML triggers		Rules
	Users		Defaults
	Roles		User defined types
	Functions		Full text catalogs
	System variables		Join suggestions

	Linked server objects		
---	-----------------------	--	--


For SQL Server 2005 and 2008:

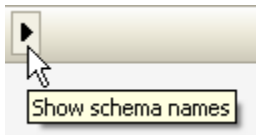
	Assemblies		Queues
	Asymmetric Keys		Routes
	Certificates		Schemas

	Contracts		Services
	DDL Triggers		Service Bindings
	Event Notifications		Symmetric Keys
	Message Types		Synonyms

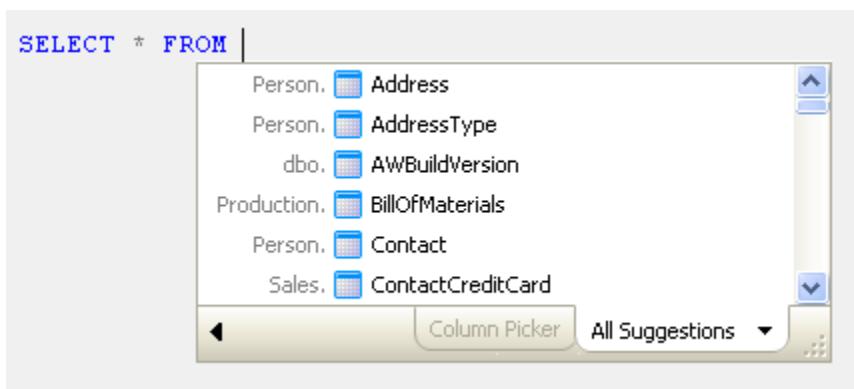
	Partition Functions		XML Schema Collections
	Partition Schemes		Full Text Stoplist (SQL Server 2008 only)


Displaying schema (owner) names

You can display schema (owner) names in the suggestions box by clicking the right arrow  in the bottom left corner of the suggestions box.



The suggestions box is automatically widened to allow space for the schema names, shown in grey to the left of the suggestions:



To hide the list of schema (owner) names, click the left arrow .

The arrows only affect how object names are displayed in the suggestions box. The schema (owner) name is only inserted into your code when **Qualify object names with owner name** is selected on the **Inserted Code > Qualification** options page.

Changing the size of the suggestions box

You can resize the suggestions box by dragging the resize handle



The new size is remembered the next time the suggestions box is displayed.

The size of the suggestions box is remembered between sessions in SQL Server Management Studio and Visual Studio.

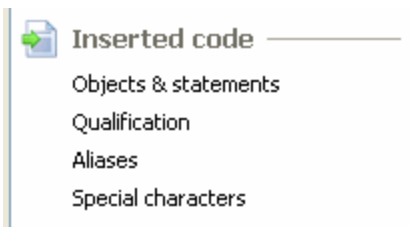
Moving through the list of suggestions

You can move through the list of items in the suggestions box by pressing:

- Up / Down arrow keys to move up or down the list one item at a time. If you are at the top of the list, pressing the Up arrow key takes you to the bottom of the list; if you are at the bottom of the list, pressing the Down arrow key takes you to the top of the list.
- Page Up or Page Down to move up or down the list one page at a time.
- Ctrl + Page Up or Ctrl + Page Down to move up or down the list one page at a time.

Changing how suggestions are inserted

When you select an item from the suggestions box, SQL Prompt inserts the suggestion according to the options you have set in the **Inserted code** pages of the **Options** dialog box:



For example, you can specify how ALTER and INSERT statements are inserted, and whether object names are qualified. For more information, see [Customizing inserted code](#).

Encrypted objects

SQL Prompt can automatically decrypt encrypted objects to show their creation script in the object definition box. (Available in SQL Prompt Pro Edition only.)

If you do not have permission to decrypt objects, you can still view them in the suggestions box and insert them into your query.

By default, SQL Prompt will decrypt encrypted objects. To turn this option off:

1. On the **SQL Prompt** menu, select **Options**.
2. Go to the **Suggestions > Types of suggestion** page.
3. Turn off **Decrypt encrypted** objects.

Expanding SELECT * to insert all columns

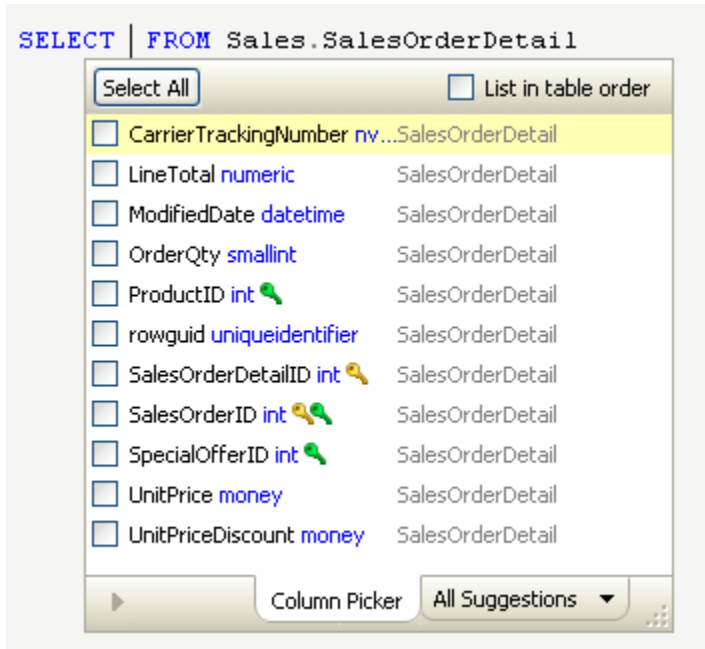
When you use a * wildcard in your query, you can expand it to explicitly list all the columns in the table or view.

Press Tab after the * wildcard to insert all the columns. The columns are formatted based on the settings in the Format pages in the Options dialog box.

Inserting columns with the column picker

The Column Picker enables you to select multiple columns for tables or views specified in a **SELECT** statement or similar syntax.

When the suggestions box is displayed, press Ctrl + Left arrow or click the **Column Picker** tab to display the Column Picker:



Primary key



and foreign key



columns are indicated by key icons.

The Column Picker is never automatically selected.

- The columns are listed in the order in which they are defined in the table. Clear the **List in table order** check box to list them alphabetically.
- The first column in the list is highlighted, but it is not selected. Only columns that you specifically check will be inserted.

The Column Picker is only available when it is valid syntax to insert more than one column into your code at this point in the query, and there are columns available to insert. For example, the column picker is available for **SELECT** statements and after **ORDER BY**, but not in a **WHERE** clause.

Selecting columns

As you start typing, the list of columns is filtered to show only those that match the typed characters.

- Use the Up and Down arrow keys to navigate through the list.
- Press Space to select a column to be inserted.
- You can select all the columns in the list by pressing Ctrl + A or clicking the **Select All** button.

If you have already selected some columns and then filtered the list by typing some letters, the previously selected columns are listed at the bottom:

```
SELECT u FROM Sales.SalesOrderDetail
```

<input type="button" value="Select All"/>	<input type="checkbox"/> List in table order	3 columns selected
<input type="checkbox"/> UnitPrice money	SalesOrderDetail	
<input type="checkbox"/> UnitPriceDiscount money	SalesOrderDetail	
<input checked="" type="checkbox"/> OrderQty smallint	SalesOrderDetail	
<input checked="" type="checkbox"/> ProductID int	SalesOrderDetail	
<input checked="" type="checkbox"/> SalesOrderID int	SalesOrderDetail	

Inserting selected columns

Press the Enter or Tab key to insert the selected columns. The columns are inserted in the order in which you selected them.

You can also insert the selected columns using any of the other insertion keys you have specified, for example Space or Dot (.) or Comma (,) - see [Managing SQL Prompt behavior](#).

The inserted columns are automatically formatted depending on the options specified in the SQL Prompt Options dialog box:

- For SQL Prompt Pro Edition, under **New lines** on the **Format > Data Statements** page.
- For SQL Prompt Standard Edition, under **Column Placement** on the **Format > Basic (Standard Edition)** page.

If the format of the inserted columns doesn't look the way you want, you can change the formatting options and then apply the new format to the selected code:

From the **SQL Prompt** menu, select **Format SQL**.

The **Format SQL** command is only available in SQL Prompt Pro Edition.

Qualified column names

Inserted column names are qualified when:

- you have selected **Qualify column names with table name** on the **Inserted code > Qualification** options page, or
- you have specified more than one table in the FROM clause, and the column names would be ambiguous without also specifying their table

In the example below, both qualification options have been turned on:

```
SELECT  Production.ProductCategory.ProductCategoryID ,
        Production.ProductCategory.Name ,
        Production.ProductCategory.rowguid ,
        Production.ProductCategory.ModifiedDate
FROM    Production.ProductCategory
```

In the example below, SQL Prompt has automatically qualified certain column names to avoid ambiguity:

```

SELECT  ProductCategoryID ,
        Name ,
        ProductCategory.rowguid ,
        ProductCategory.ModifiedDate ,
        CustomerID ,
        TerritoryID ,
        AccountNumber ,
        CustomerType ,
        Customer.rowguid ,
        Customer.ModifiedDate
FROM    Production.ProductCategory ,
        Sales.Customer

```

Only those columns that appear in more than one table are qualified.

When **Assign** aliases is turned on as well as **Qualify column names with table name**, columns are qualified with the table alias. For example:

```

SELECT  a.AddressID ,
        a.AddressLine1 ,
        a.AddressLine2 ,
        a.City ,
        a.StateProvinceID ,
        a.PostalCode ,
        a.rowguid ,
        a.ModifiedDate
FROM    Person.Address AS a

```

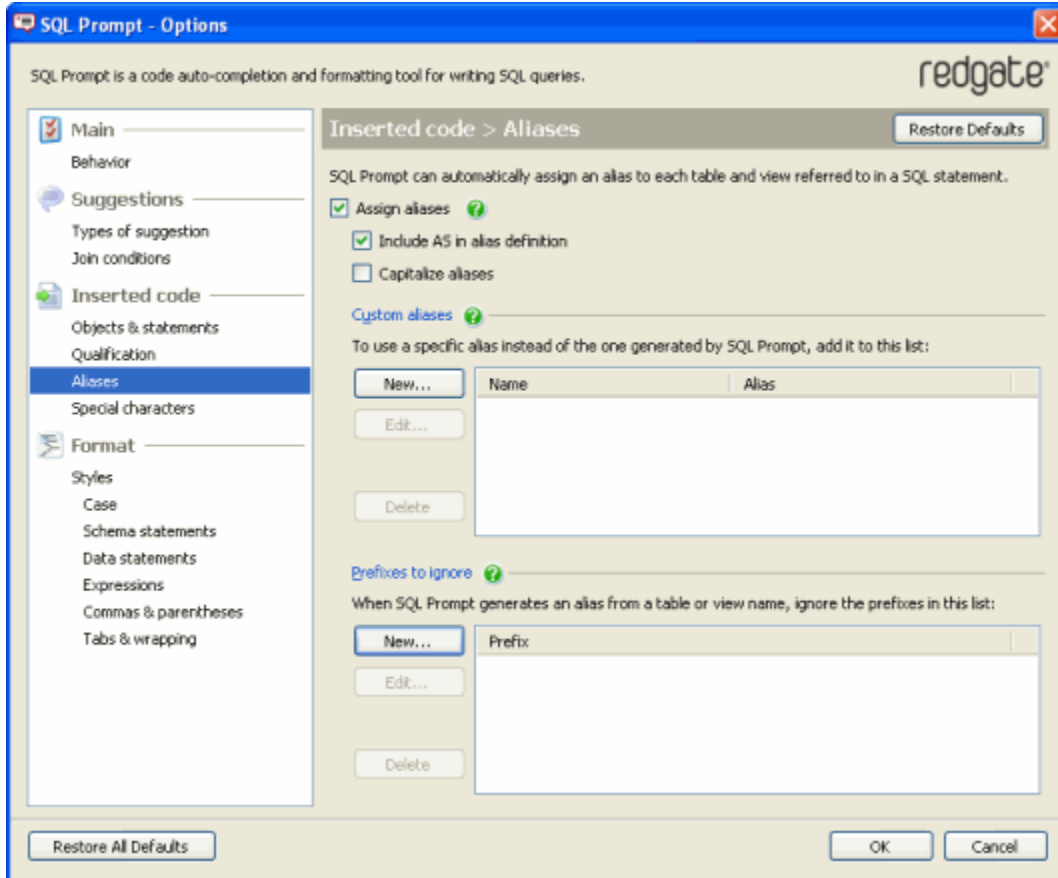
You can customize how aliases are created and inserted. For more information, see [Working with aliases](#).

Working with aliases

You can set SQL Prompt to automatically assign an alias to each table and view that is referenced in a SQL statement. You can also define custom aliases for tables and views.

To manage aliases:

1. On the **SQL Prompt** menu, click **Options**.
2. On the SQL Prompt Options dialog box, select the **Inserted code > Aliases** page.



Alias assignment

When the **Assign aliases** check box is selected, SQL Prompt assigns an alias to tables and views that are referenced in a SQL statement, provided that you have specified a list of columns or used * to select all columns.

For example, if you select all columns from the table *Contact*, SQL Prompt creates the alias *c* to represent the table name *Contact*.

```
SELECT * FROM Person.Contact AS c
```

If you do not want SQL Prompt to include the AS keyword when it assigns aliases, clear the **Include AS in alias definition** check box.

Where possible, SQL Prompt generates aliases using the first letter of the table or view name. SQL Prompt also takes into account:

- underscores
TBL_Contact is assigned the alias *tc*
- hyphens
hyphenated-tablename is assigned the alias *ht*
- case
MixedCase is assigned the alias *mc*

The suggestion box displays the learned aliases at appropriate points in your query, for example when you are typing a WHERE clause or adding additional columns to your query.

SQL Prompt creates additional aliases whenever there is ambiguity, for example in self-joins:

```
SELECT DISTINCT
    pv.VendorID ,
    pv2.ProductID ,
    pv.ProductID
FROM    Purchasing.ProductVendor AS pv
        INNER JOIN Purchasing.ProductVendor AS pv2
ON pv.ProductID = pv2.ProductID
WHERE   pv.VendorID <> pv2.VendorID
```

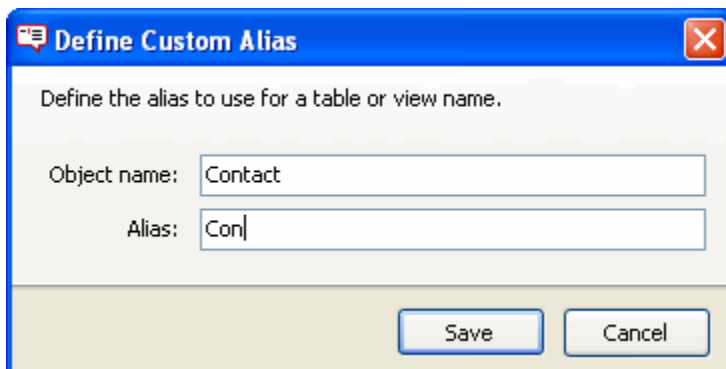
Custom aliases

If the aliases that SQL Prompt automatically generates do not satisfy your naming conventions, you can specify user-defined aliases for table or view names.

To add a user-defined alias:

1. Under **Custom aliases**, click **New**.
2. In the **Alias Define Custom Alias** dialog box, type the name of the table or view in the **Object name** box and the alias in the **Alias** box.
3. Click **Save**.

For example, to specify the user-defined alias *Con* for the Contact table, type the following in the **Define Custom Alias** dialog box:



SQL Prompt will then assign the alias as follows:

```
SELECT * FROM Person.Contact AS Con
```

To delete a custom alias, select the alias that you want to delete, and click **Delete**.

Prefixes to ignore

You can specify that SQL Prompt should ignore a prefix when assigning an alias for a column, table, or view name.

To add a prefix to ignore when generating an alias:

1. Under **Prefixes to ignore**, click **New**.
2. In the **Prefix to Ignore** dialog box, type the name of the prefix.
3. Click **Save**.

For example, if you specify **TBL** as a prefix to ignore and there is a table called **TBL_Orders**, SQL Prompt considers only **Orders** when assigning an alias for the table name:

```
SELECT * FROM dbo.tbl_orders AS o
```

- Prefixes are not case-sensitive. Specifying **tbl**, **TBL**, or **Tbl** as the prefix will result in the SQL Prompt behavior shown above.
- You do not need to include the underscore in the prefix name.

Inserting snippets

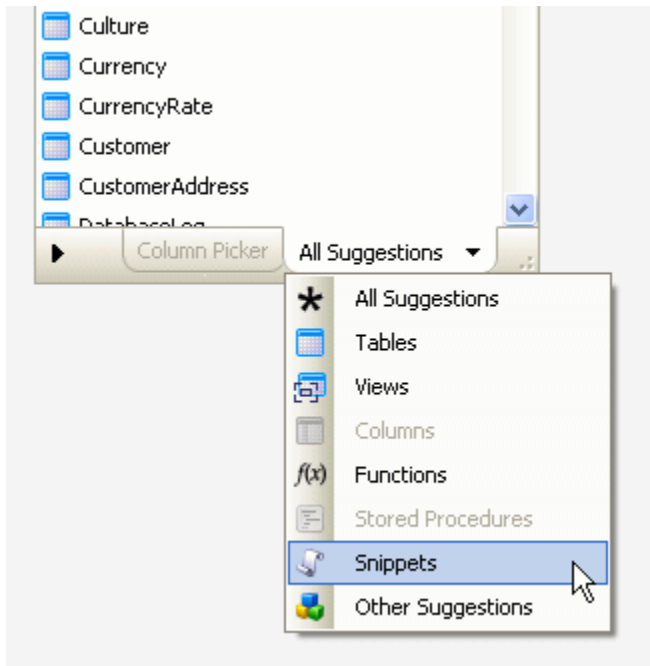
Snippets enable you to insert pre-defined code into your query editor.

To insert snippet code, type the snippet name and press Tab or any other specified insertion key. In some cases, only Tab is available to insert the snippet (for example when the snippet name could also be the name of an alias you are typing).

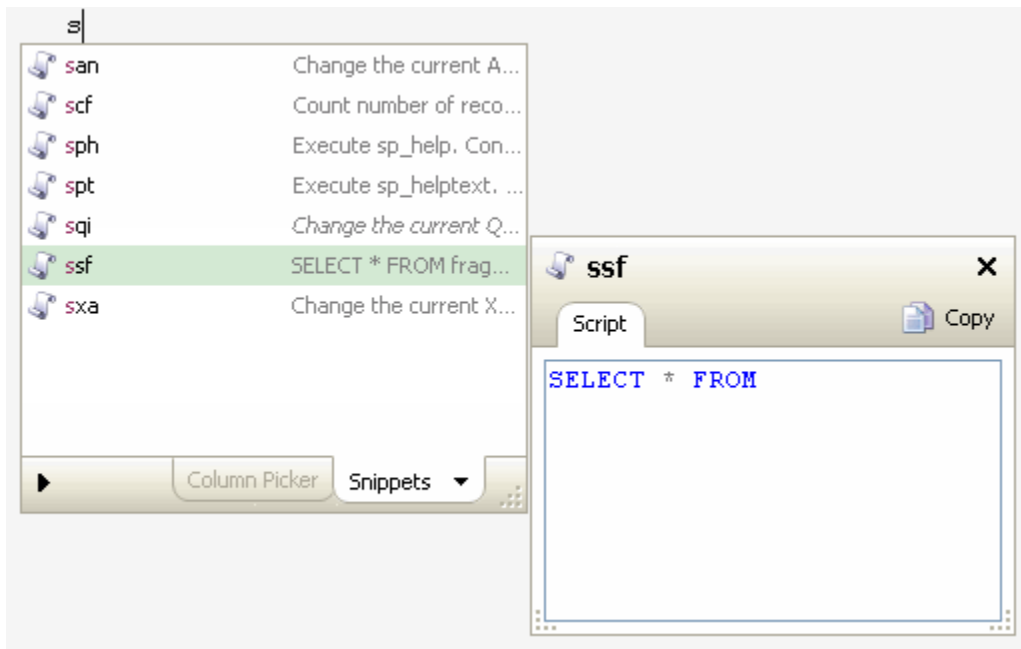
To see a list of available snippets, press Ctrl + Space to display the suggestions box, and select the



Snippets category. You can also press Ctrl + Down arrow to move down the types of suggestion to Snippets.



A preview of the snippet code is displayed in the object definition box. (You can turn off the display of the object definition box - see [Managing SQL Prompt behavior](#).)



The snippet code is inserted at the same indentation level as your SQL code at the point of insertion.

Using default snippets

SQL Prompt is pre-configured with a number of snippets. For example:

- **ssf** inserts a SELECT * FROM fragment
- **cdb** inserts a CREATE DATABASE statement and places the insertion point after the CREATE DATABASE fragment.
- **cuci** creates a unique clustered index

You can edit or delete the default snippets in the **Snippet Manager** dialog box.

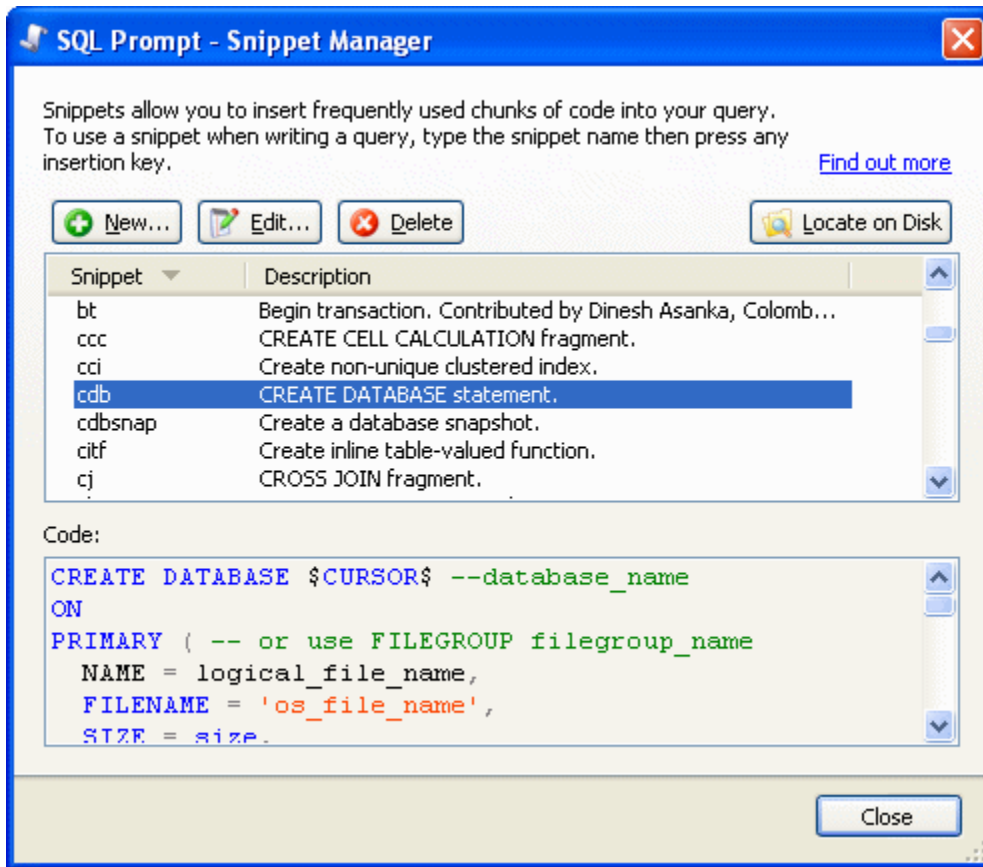
Creating your own snippets

You can create your own snippets, including specifying the position at which SQL Prompt will place the cursor after it has inserted the snippet of code.

See [Managing snippets](#).

Managing snippets

Snippets enable you to insert pre-defined code into your query editor. To view a list of snippets, on the **SQL Prompt** menu, select **Snippet Manager**.



The **Snippet Manager** dialog box displays a list of snippets, sorted alphabetically. To sort by description, click the **Description** column header.

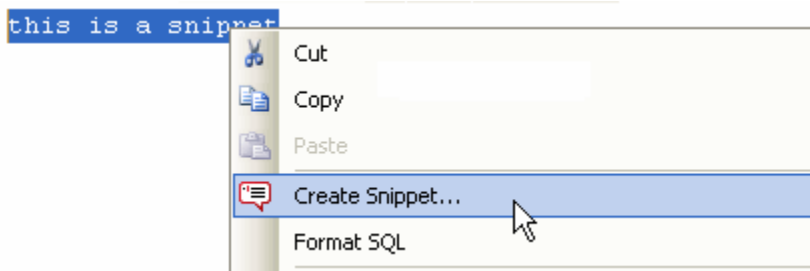
Click on a snippet to view the code for that snippet in the **Code** area.

The snippet code is inserted at the indentation level of your SQL code at the point of insertion.

Creating a snippet from code you have typed

You can create a snippet from code you have typed into the query editor.

Select the code so that it is highlighted, then right-click and select **Create Snippet**.



The Create New Snippet dialog box is displayed, so that you can edit the snippet name or text if required.

The default snippet name is created automatically from the initial letters of the selected text. You can edit this if you want to use a different name.

Creating a new snippet

1. On the **SQL Prompt** menu, select **Snippet Manager**.
2. Click **New**.
3. In the **Snippet** box, type the text that will insert the snippet.
4. Optionally, type a short description of your snippet in the **Description** box.
The description helps you to identify a snippet if you are unsure of the snippet name. You can leave the Description box blank if required.
5. Type or paste the SQL code in the **Code** box.
The SQL you enter can contain the following placeholders: `$DATE$`, `$TIME$`, `$USER$`, `$MACHINE$`.
To specify the insertion point at which you want the cursor to be placed when the snippet is inserted, type `$CURSOR$`. For example, `SELECT $CURSOR$ FROM` places the insertion point after `SELECT`.
In SQL Prompt 5.1 and later, you can also specify parameters, in the format `<parameter_name, data_type, value>`. For full details, see [Using SSMS templates in SQL Prompt snippets](#).
6. Click **Save**.

Editing snippets

In the **Snippet Manager** dialog box, select the snippet that you want to edit and click **Edit**, or double-click the snippet. Then, in the **Edit Snippet** dialog box, change the details as required and click **Save**.

Sharing snippets

You can send your snippets to other SQL Prompt users, or use snippets they send you.

To use a snippet that someone has emailed you, you need to copy the individual snippet file (*.sqlpromptsnippet* extension) to the Snippets folder:

1. From the **SQL Prompt** menu, select **Snippet Manager**.
2. In the Snippet Manager dialog box, click **Locate on Disk** to view the path to the Snippets folder (by default, this is `C:\Documents and Settings\<your user name>\Local Settings\Application Data\Red Gate\SQL Prompt 5\Snippets`).
3. Close the Snippet Manager dialog box.
4. Copy the *.sqlpromptsnippet* file to the Snippets folder.
5. Re-open the Snippet Manager. The new snippet will be displayed in the list.

Dealing with duplicate snippets

Snippet names are defined by the `<Shortcut>` tag in the *.sqlpromptsnippet* file.

If a snippet file you have copied to the Snippets folder uses the same name as an existing snippet, a warning is displayed:



Use the green arrow buttons to locate the next/previous duplicate snippet in the list, then do one of the following:

- Click **Delete** to remove the duplicate. (Use the Code preview box to check which snippet you are deleting).
- Click **Edit** and change the name in the **Snippet** box to a unique name.

You can also edit the *.sqlpromptsnippet* file in a text editor to change the value of the `<Shortcut>` tag.

Using SSMS templates in SQL Prompt snippets

SQL Server Management Studio template parameters are only available in SQL Prompt 5.1 and later.

SQL Prompt enables you to specify SQL Server Management Studio template parameters in [SQL Prompt snippets](#).

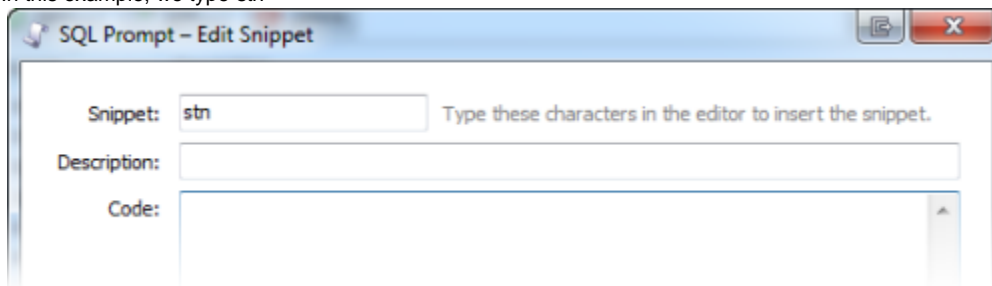
Example: Creating and inserting a snippet with template parameters

In this example, we will create a snippet to select the following from a table:

- the top N rows
- a title displaying the total number of rows

To create the snippet:

1. On the **SQL Prompt** menu, select **Snippet Manager**.
2. Click **New**.
3. In the **Snippet** box, type the text that will insert the snippet.
In this example, we type *stn*



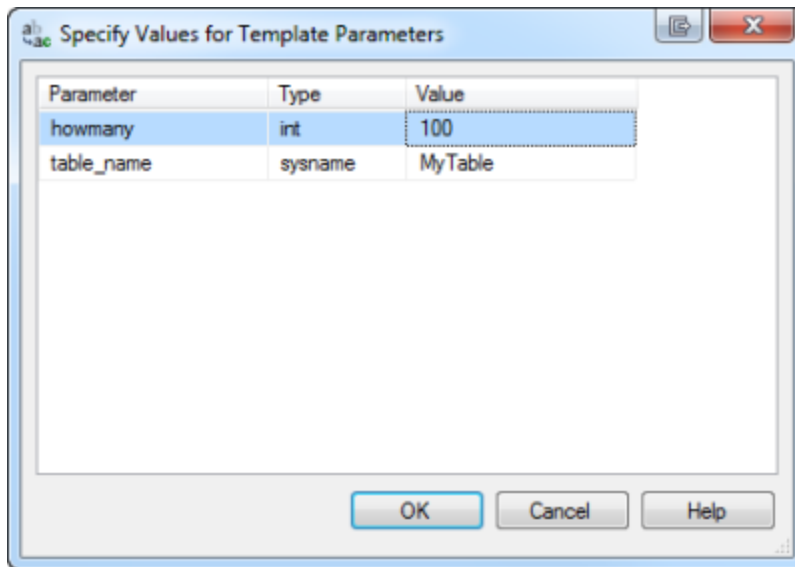
4. Type or paste SQL code that includes template parameters in the **Code** box.
You specify template parameters using the format `<parameter_name, data_type, value>`. For more information, see [Using SQL Server Management Studio Templates \(Microsoft documentation\)](#).
In this example, we use the following snippet code:

```
SELECT 'Top <howmany,int,100> rows of '+ CONVERT(NVARCHAR(MAX), COUNT(*))+ ' in  
<table_name,sysname,MyTable>' AS TitleFROM    <table_name,sysname,MyTable>SELECT  
TOP <howmany,int,100>*FROM    <table_name,sysname,MyTable>
```

5. Click **Save**.

We can now insert the snippet into a query, and replace the template parameters with the values we want to use:

1. In a query window, type *stn*, and then press Tab.
The snippet is inserted into the query.
2. Press Ctrl + Shift + M to open the **Specify Values for Template Parameters** dialog box:



A screenshot of a Windows-style dialog box titled "Specify Values for Template Parameters". The dialog has a standard title bar with a minimize button, a maximize button, and a close button. Inside the dialog, there is a table with three columns: "Parameter", "Type", and "Value". The table contains two rows of data. The first row has "howmany" in the "Parameter" column, "int" in the "Type" column, and "100" in the "Value" column. The second row has "table_name" in the "Parameter" column, "sysname" in the "Type" column, and "MyTable" in the "Value" column. The first row is highlighted with a blue background. Below the table, there are three buttons: "OK", "Cancel", and "Help".

Parameter	Type	Value
howmany	int	100
table_name	sysname	MyTable

OK Cancel Help

3. Specify the values you want to use, and click **OK**.

Using keyboard shortcuts

Using keyboard shortcuts to select and insert code suggestions in SQL Prompt can dramatically increase the speed of writing your SQL code. You can use the following keyboard shortcuts:

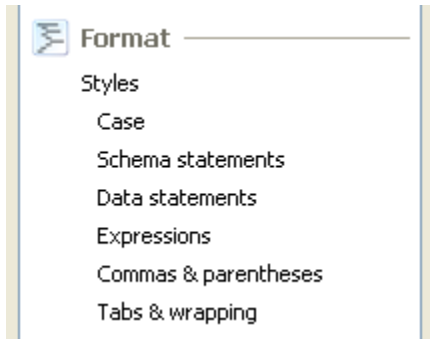
Action	Shortcut
Switch to/from column picker	Ctrl + Left arrow / Ctrl + Right arrow
Move up/down suggestions list	Up arrow / Down arrow
Select column in column picker for insertion	Space
Move up/down the suggestions box filters	Ctrl + Up arrow / Ctrl + Down arrow
Apply format	Ctrl + K + Y (Hold Ctrl and press K then Y)
Turn on/off code auto-completion (SQL Prompt 5.2 and later)	Ctrl + Shift + P
Refresh suggestions (SQL Prompt 5.3 only)	Ctrl + Shift + D

Formatting your code

Formatting your code using SQL Prompt is only available in **SQL Prompt Pro Edition**.

Setting the format options

To set formatting options, on the **SQL Prompt** menu, select **Options**. On the SQL Prompt Options dialog box, formatting options are all available in the **Format** group:



Click the page containing the type of format options you want to set.

These settings will affect the appearance of your code only after you close the Options dialog box and then insert suggestions or select **Format SQL**.

For more information about formatting options, see [Options for formatting your code](#).

Previewing the format options

There are two ways to preview the effect of applying the current format options to your code:

- The **Sample SQL** view shows how an example query would be formatted using the current options.
- The **Current Query** view shows how the first few lines of the query that is currently active in the editor window of your SQL editing application (eg SSMS) would be laid out using the current options.

Current Query is available only when you are currently editing a query.

Formatting your SQL

To format the SQL code in the current query editor or a highlighted SQL code fragment, do one of the following:

- on the **SQL Prompt** menu, click **Format SQL**
- right-click in your query editor window and click **Format SQL**
- press Ctrl + K, Y (In Query Analyzer, press Ctrl + Shift + Y)

Not all SQL commands can be refactored by SQL Prompt. For unsupported commands only the wrapping option is applied.

Saving your format options as a style

When you have set up all your format options, you can save them as a style. Styles allow you to reapply all your format options without having to change each one again.

Expanding wildcards

SQL Prompt enables you to expand SELECT * and SELECT *table*.* statements so that they list all of the columns that exist in the referenced tables.

Expanding all wildcards in a SQL script is available only in SQL Prompt Pro Edition.

You can only expand individual wildcards in a script using SQL Prompt Standard Edition; for details see [Expanding SELECT * to insert all columns](#).

To expand all wildcards in a SQL script:

1. Open the script you want to modify in a SQL Server Management Studio query window.
2. Ensure you are connected to the SQL Server and database that contains the objects referenced in the SELECT statements.
3. If required, select the portion of SQL that contains the SELECT statements that you want to expand. If you do not select any SQL, all SELECT statements in the script are expanded.
4. On the **SQL Prompt** menu, click **Expand Wildcards**.

You can undo the changes to the script using the standard SQL Server Management Studio Undo features.

To keep the changes, save the script.

Keyboard shortcut: hold Ctrl and press B then W to expand all wildcards in a script, or highlight a particular wildcard to expand only that one. (SQL Prompt 5.3 only)

Qualifying object names

SQL Prompt enables you to modify a SQL script so that all object names are qualified in the format:

`owner.object`

Column names are qualified in the format:

`table.column`

Note

- SQL Server name and database name qualifiers are not added to the names.
- Qualifying object names is available only in SQL Prompt Pro Edition.

To qualify object names:

1. Open the script you want to modify in a SQL Server Management Studio query window.
2. If required, select a portion of SQL in the script that contains the object names that you want to qualify.
If you do not select any SQL, all object names in the script are qualified.
3. On the **SQL Prompt** menu, click **Qualify Object Names**.

You can undo the changes to the script using the standard SQL Server Management Studio Undo features.

To keep the changes, save the script.

Keyboard shortcut: hold Ctrl and press B then Q to qualify all object names in a script, or highlight a particular object to qualify only that one. (SQL Prompt 5.3 only)

Uppercasing keywords

Uppercasing keywords is only available in SQL Prompt 5.1 and later.

SQL Prompt allows you to uppercase selected keywords in a script. For example, the following line of SQL:

```
select * from Person.Contact as c
```

can easily be modified to:

```
SELECT * FROM Person.Contact AS c
```

To uppercase keywords:

1. Open the script you want to modify in a SQL Server Management Studio query window.
2. If required, select a portion of SQL in the script that contains the keywords that you want to uppercase.
If you do not select any SQL, all keywords in the script are uppercased.
3. On the **SQL Prompt** menu, click **Uppercase Keywords**.

You can undo the changes to the script using the standard SQL Server Management Studio Undo features.

To keep the changes, save the script.

Keyboard shortcut: hold Ctrl and press B then U to uppercase all keywords in a script, or highlight a keyword to uppercase only that one. (SQL Prompt 5.3 only)

Using styles to format your code

A style contains all your preferred settings for formatting SQL code. Saving your formatting options as a style allows you to reapply them to a query at a later date, or share them with other SQL Prompt users so they can use the same conventions for formatting their queries.

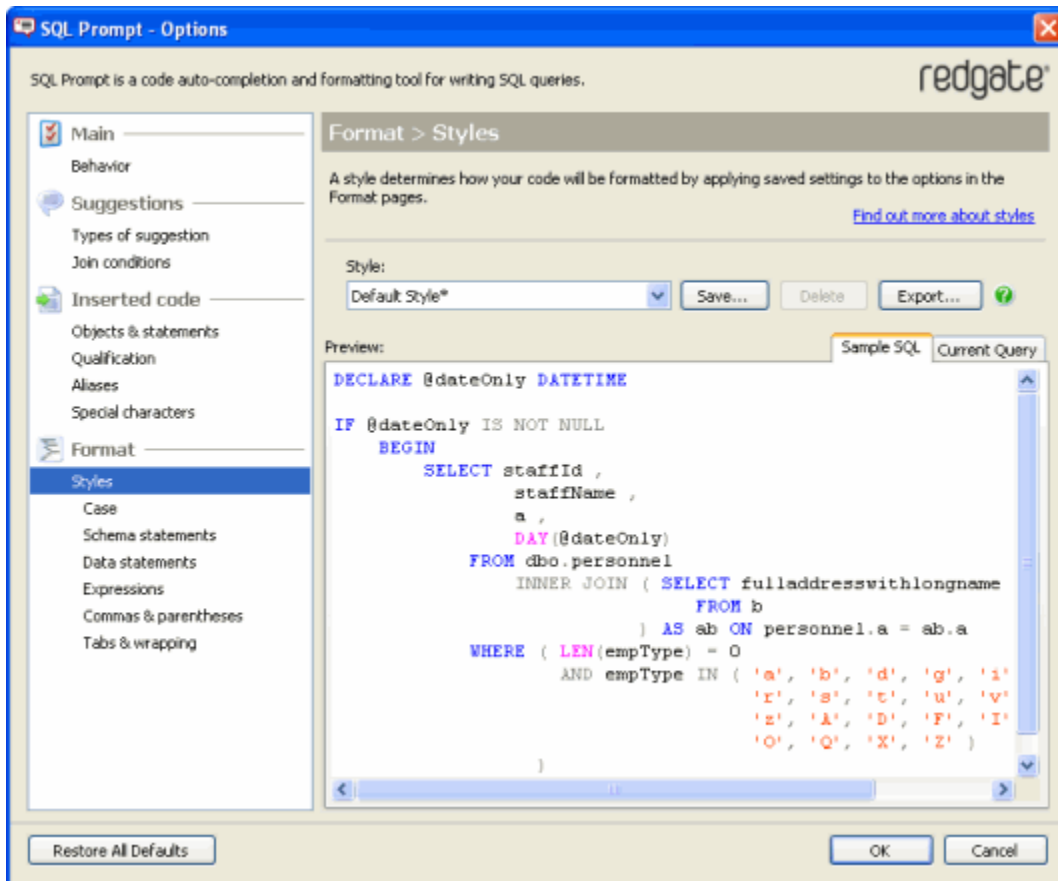
Overview: applying a style to your code

1. On the **Format > Styles** page of the SQL Prompt Options dialog box, select the style you want to use from the **Styles** drop-down list. Before you apply the style, you can preview how it will lay out the first few lines of your current query - select the **Current Query** preview tab. The style is not applied to your code until you specifically choose to apply it in your code editor.
2. Click **OK** to close the SQL Prompt Options dialog box.
3. Select the code you want to format in your SQL editor, right-click and click **Format SQL**.
If you don't select any code, the entire query will be formatted.

SQL Prompt cannot format your query if it contains syntax errors. A message will be displayed to indicate when there are syntax errors.

Choosing a style

From the SQL Prompt Options dialog box, select **Format > Styles**.



Select a style you have previously created from the **Style** drop-down list.

An asterisk * next to a style name indicates that some format options currently selected in SQL Prompt do not match those previously saved as this style.

If you close the SQL Prompt Options dialog box and then format the code, it will use the current settings rather than the settings saved in the style.

If the style you want to use has been emailed to you, or is available on your network, you will first need to import it into SQL Prompt. See [Importing and exporting styles](#).

Previewing the effect of the style

There are two ways to preview the effect of applying the current format options to your code:

- The **Sample SQL** view shows how an example query would be formatted using the current options.
- The **Current Query** view shows how the first few lines of the query that is currently active in the editor window of your SQL editing application (eg SQL Server Management Studio) would be laid out using the current options.

Editing the format settings

See [Options for formatting your code](#).

Applying a style to your code

1. Select the style you want to use, then click **OK** to close the SQL Prompt options dialog box.
2. In your query editor, highlight the lines of code you want to format, and do one of the following:
 - press Ctrl + K, Y (hold down Ctrl and press K then Y)
 - right-click and click **Format SQL**
 - on the **SQL Prompt** menu, select **Format SQL**

If you don't select any code, the formatting will be applied to the entire query.

Importing and exporting styles

You can share formatting styles with other people. SQL Prompt can export all the format settings as a single file, which can then be imported by another SQL Prompt user. This allows you to ensure consistent conventions within your organisation. See [Importing and exporting styles](#).

Importing and exporting styles

If you have saved your code formatting options as a style, you can share that style with other people in your organization by exporting it.

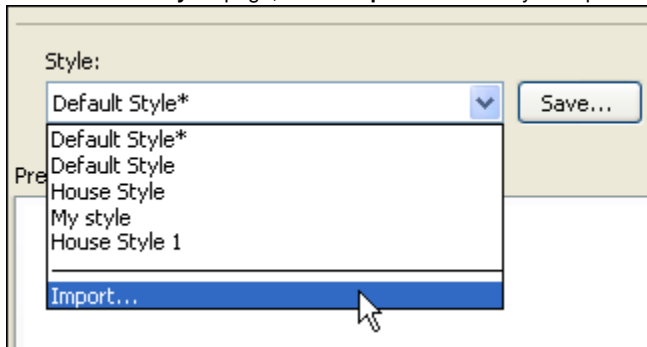
To export a style

SQL Prompt exports the style as a single file with a *.sqlpromptstyle* extension.

1. Set all the [formatting options](#) that you want:
On the SQL Prompt Options dialog box, use the options on the various Format pages to lay out the code the way you want it. Select the **Current Query** tab to preview how the code currently in your SQL editor would be laid out using these settings.
2. On the **Format > Styles** page, click **Export**.
3. Browse to where you want to store the file and click **Save**.

To import a style

1. In the **Format > Styles** page, select **Import** from the Style drop-down list:



2. Browse to the folder where the *.sqlpromptstyle* file is located, then click **Open**.
3. All the format options will be updated. Your code will not be formatted until you click **OK** to close the SQL Prompt Options dialog box and then apply the style.

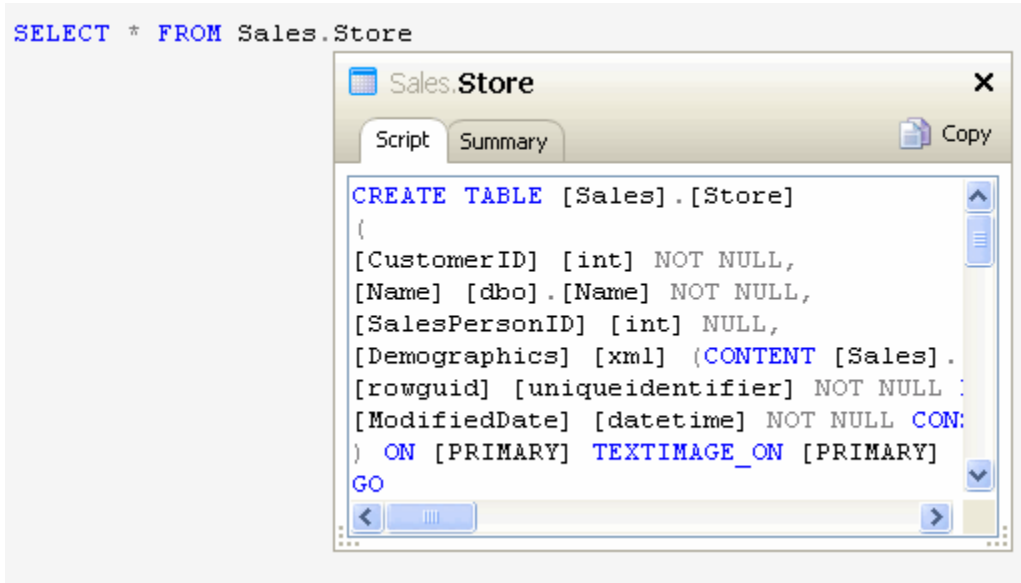
Navigating your code

The following features help you understand and tidy up your code:

- [Using the object definition box and object tooltips](#)
- [Scripting an object as ALTER](#)
- [Finding invalid objects](#)
- [Finding unused variables and parameters](#)
- [Summarizing a script](#)

Using the object definition box and object tooltips

SQL Prompt displays object definitions in an *object definition box*. For example, when you select a table in the suggestions box, the object definition box pops up to show the creation script for that table, and a summary of the object.



- For tables and views, the **Summary** tab shows the column names and data types.
- For stored procedures and functions, the **Summary** tab displays parameters and their data types, and for functions the data type that is returned.
- For snippets, the snippet code is shown on the **Script** tab. The **Summary** tab is not available for snippets.
- For temporary tables, CTEs and table valued variables, only the **Summary** tab is available.

To resize the object definition box, drag the resize handle



at the bottom left or right corner.

Showing or hiding the object definition box

When you first install SQL Prompt, the object definition box will always be displayed for any suitable object selected in the suggestions box. To turn it off, so that it isn't displayed automatically:

1. On the **SQL Prompt** menu, select **Options**.
2. On the **Main > Behavior** page of the SQL Prompt Options dialog box (this page is the first page displayed by default), turn off **Display object definitions**.

If you turn off the object definition box, you can still view the object creation script for any object in your script by hovering over the object name to display a tooltip. Click the tooltip to display the object definition.

You can also increase the time lag before the object definition box is displayed. This may be useful if you don't want it to appear when you are typing code quickly, but you do want to see definitions for certain tables after a slight pause:

Enter the time in milliseconds. 500 milliseconds is the default time.

Copying object definitions


You can copy the information in the object definition box to the clipboard:

- To copy all the information, ensure that there is no text selected, and click **Copy**.
- To copy part of the information, select the required text, and click **Copy**.

Using object definition tooltips

When you move the mouse pointer over an object in your query, SQL Prompt displays a tooltip containing the fully qualified object name:

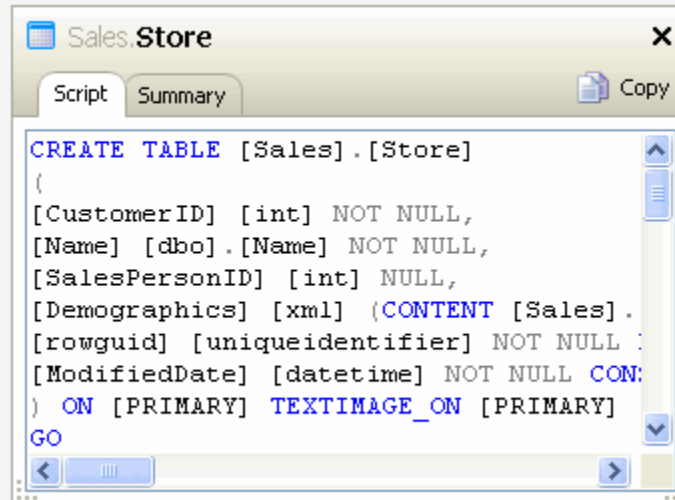

```
SELECT * FROM Sales.Store
```

 **Sales.Store (Table)**
Customers (resellers) of Adventure Works products.

The tooltip also displays the MS_Description extended property for the object, in this case a brief explanation of the table's contents.

If the object name is underlined, you can click it to display the object definition box for the object:

```
SELECT * FROM Sales.Store
```




For functions, the tooltip displays the possible parameters for that function.

Using tooltips for more help on system functions and parameters

Move the mouse pointer over a system (built-in) function to display an interactive tooltip. You can click on the tooltip to access the SQL Server online help for that function:

```
SELECT CHARINDEX( )
```

 **CHARINDEX(expression1, expression2, [start_location])** RETURNS *bigint*
Returns the starting position of the specified expression in a character string.

You can move the mouse over the function parameters to display a tooltip specifically listing the valid parameters for this function:

```
SELECT CHARINDEX( )
```

CHARINDEX(expression1, expression2, [start_location])
expression1 *char*

As you enter the parameters, the next expected parameter is highlighted in blue.

Click either type of tooltip to read more about the current function in Books Online:

CHARINDEX (Transact-SQL)

Searches *expression2* for *expression1* and returns its starting position if found. The search starts at *start_location*.



[Transact-SQL Syntax Conventions](#)

Syntax

```
CHARINDEX ( expression1 ,expression2 [ , start_location
n ] )
```

Arguments

expression1

Is a character [expression](#) that contains the sequence to be found. *expression1* is limited to 8000 characters.

expression2

Is a character expression to be searched.

Showing or hiding tooltips

When you first install SQL Prompt, tooltips are enabled by default. To turn them off:

1. On the **SQL Prompt** menu, select **Options**.
2. On the **Main > Behavior** page of the SQL Prompt Options dialog box (this page is the first page displayed by default), turn off **Objects** or **Parameters** under **Tooltips**.

Tooltips are displayed for these objects:

- tables
- views
- stored procedures
- databases
- data types
- snippets
- triggers
- local variables (type)
- local parameters (type)
- columns (table name and type)
- built-in functions (function signature)
- user-defined functions (function signature)

Viewing tooltips for aliases

SQL Prompt displays a tooltip when you have created an alias for a table or view. Aliases are identified by the



icon in the tooltip:

```
FROM Sales.Customer AS cust
```



You can click the tooltip to view the summary of columns in that table or view. Aliases for views are also identified as "Table" in the object tooltip.

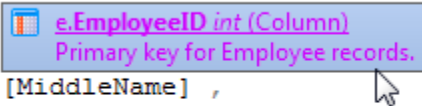
Viewing column dependencies

You can use SQL Prompt to view a list of objects that reference a column; this can be useful when you want to modify an object.

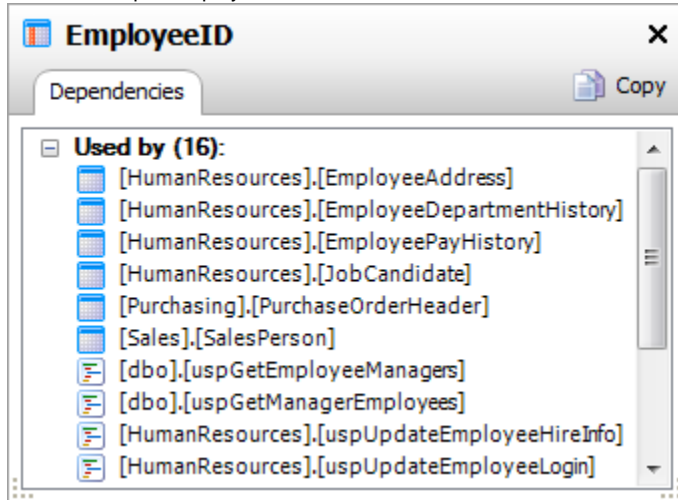
To view column dependencies:

1. In a SQL Server Management Studio query window, move the mouse pointer over a column in your query. SQL Prompt displays a tooltip containing the fully qualified column name:

```
SELECT e.[EmployeeID] ,  
c.[EmployeeID] int (Column)  
c.[MiddleName] ,  
c.[LastName] ,  
c.[Suffix] ,  
e.[Title] AS [JobTitle] ,  
c.[Phone] ,
```



2. Click the tooltip to display the column definition box:



All objects that reference the selected column are listed.

Scripting an object as ALTER

If you are navigating a query that contains a reference to an object, SQL Prompt enables you to script out that object directly from the query window.

You can script out the following objects:



Views



Stored Procedures



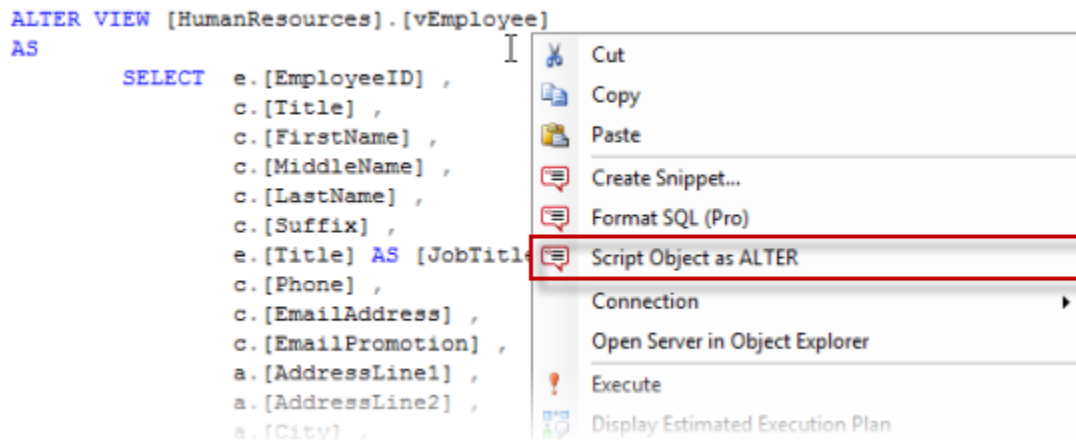
Triggers



Functions (scalar-valued and table-valued)

To script out an object from a query window:

1. In the query, right-click the object you want to script out:



2. On the context menu, click **Script Object as ALTER**.
The object is scripted as an ALTER statement in a new query window.

Keyboard shortcut: with the mouse insertion point on the name of the object you want to script out, press **F12**.

Finding invalid objects

SQL Prompt enables you to find invalid objects in your database. This can be useful, for example, if you have inherited a legacy database and you want to know if any objects are referencing objects that have been dropped, and no longer exist in the database.

Objects are listed as invalid by SQL Prompt if they:

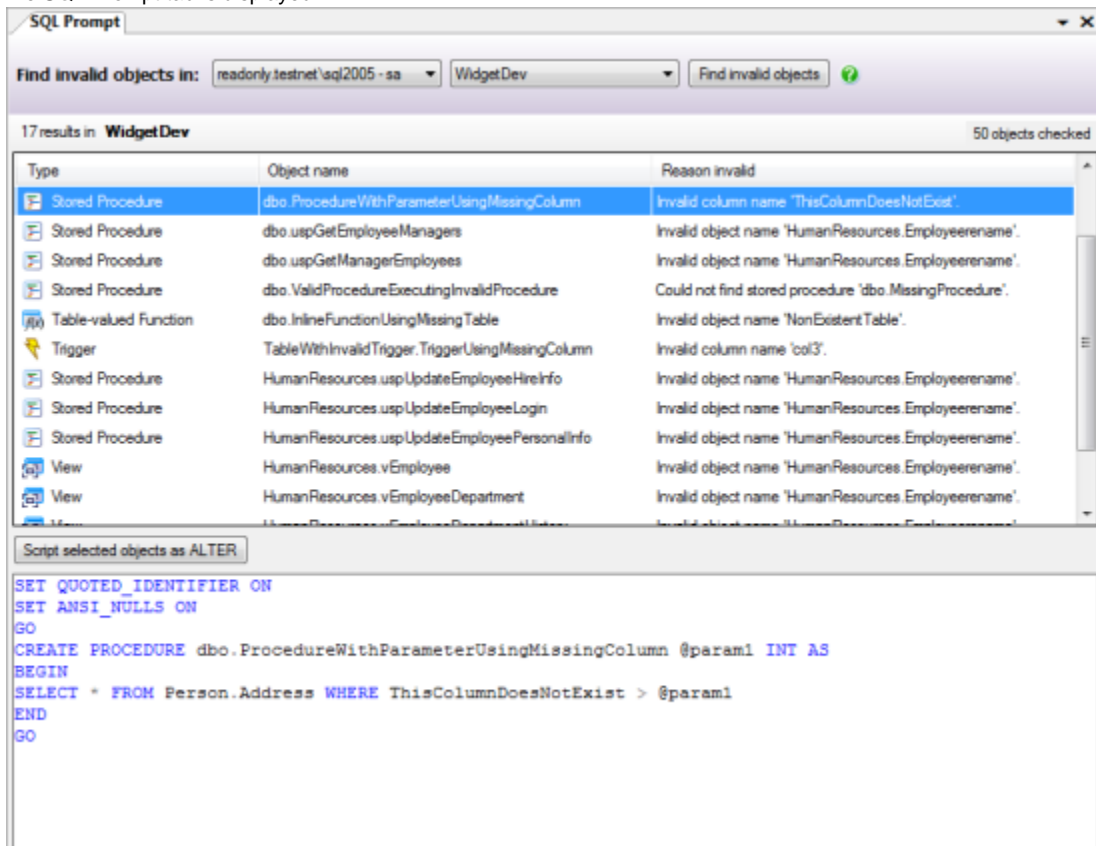
- cannot be compiled
- can be compiled, but cannot be executed

For example, a stored procedure that references an object that no longer exists in the database is listed as invalid.

To find invalid objects in a database:

1. In the SQL Server Management Studio Object Explorer, right-click the database you want to find invalid objects in, and click **Find Invalid Objects**.

The SQL Prompt tab is displayed:



In SQL Prompt 5.0, this is the Invalid Objects tab.

Invalid objects are listed in the upper pane.

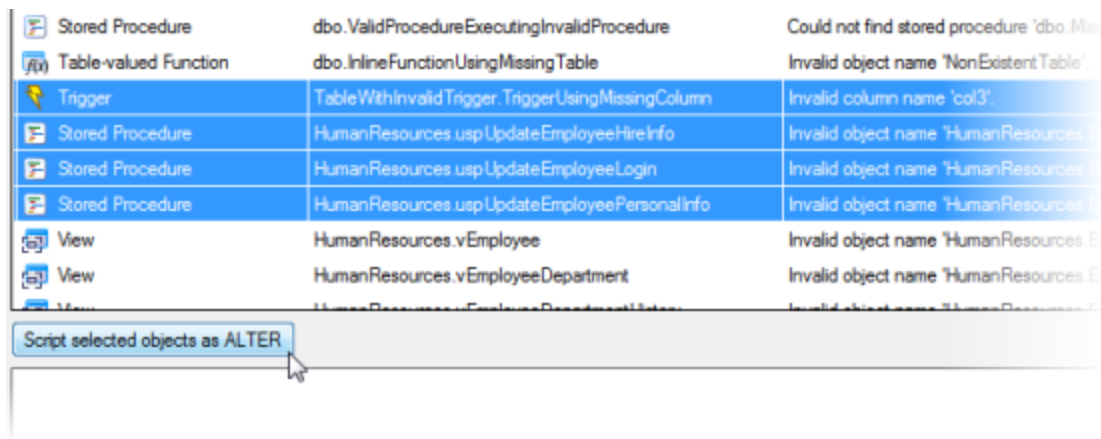
2. Click an object in the list to view its SQL creation script in the lower pane.

To open the script as an ALTER statement in a new query window, click **Script selected objects as ALTER**.

Tips

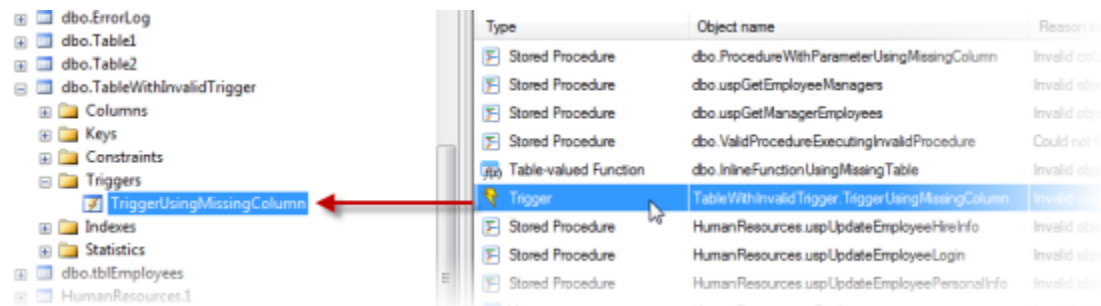
You can:

- script multiple invalid objects as ALTER by selecting more than one object in the upper pane, and then clicking **Script selected objects as ALTER**:

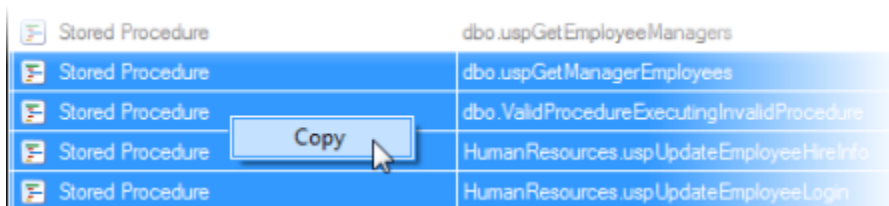


This only creates one new query; each object is scripted as a separate ALTER action.

- locate an invalid object in the Object Explorer by double-clicking it in the upper pane:



- copy a list of invalid objects you select in the upper pane by right-clicking a selected object, and clicking **Copy**:



- copy any SQL you select in the lower pane by pressing Ctrl + C.

Finding unused variables and parameters

Finding unused variables and parameters is available only in SQL Prompt Pro Edition.

SQL Prompt allows you to find any parameters or variables that are unused in a script.

A parameter or variable is considered to be unused if it has been declared or assigned a value, but not queried or used in any control statements, and not used in INSERT or UPDATE statements, WHERE clauses, PRINT or EXECUTE statements, loops or case statements, and so on.

To find unused variables and parameters:

1. Open the script you want to use in a SQL Server Management Studio query window.
2. If required, select the block of SQL that contains the parameters and variables that you want to check.
If you do not select any SQL, all parameters and variables in the script are checked.
3. On the **SQL Prompt** menu, click **Find Unused Variables and Parameters**.
Unused parameters, variables, and values are underlined in the query:

```
BEGIN
  DECLARE @i AS int
  SET @i = NULL
  ... The value assigned here is never used
  SET @i = 4
  SELECT @i
  ...
END
```

Keyboard shortcut: hold Ctrl and press B then F to find unused variables and parameters in the selected script. (SQL Prompt 5.3 only)

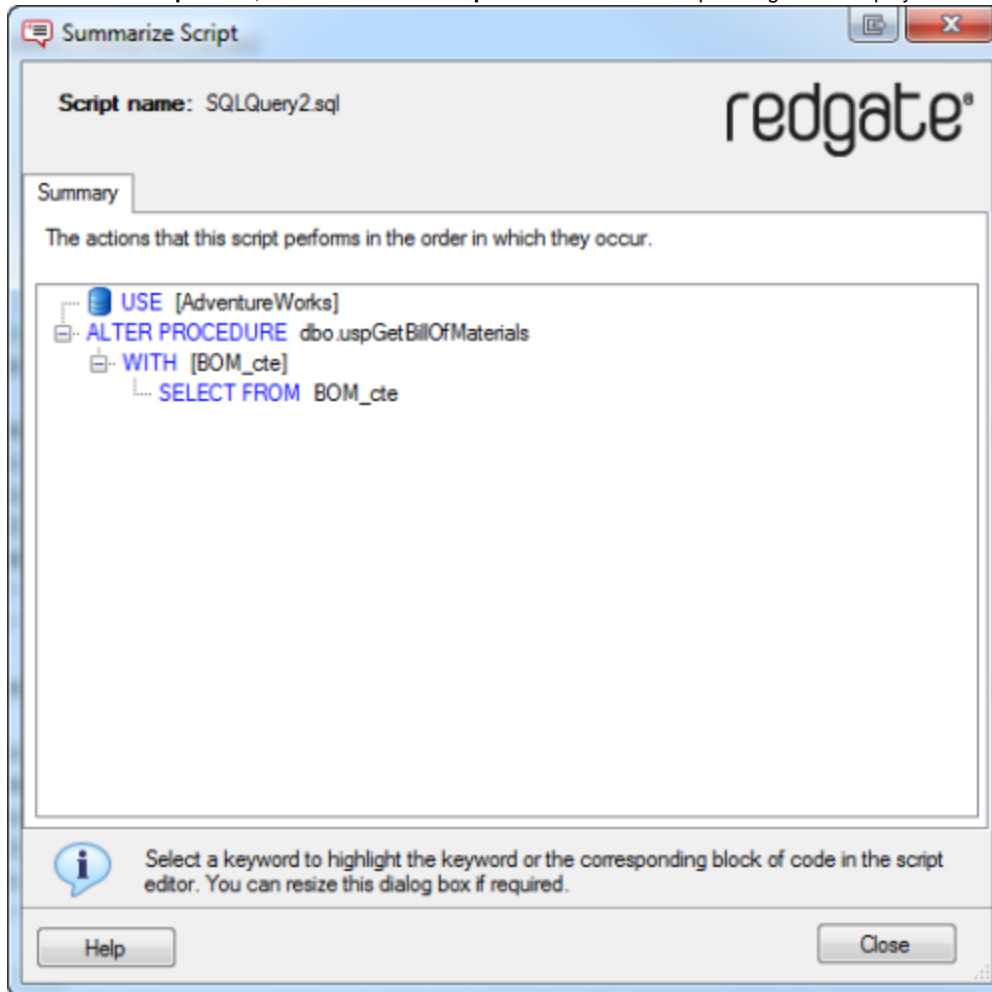
Summarizing a script

Summarizing a script is available only in SQL Prompt Pro Edition.

SQL Prompt allows you to view a summary of the actions that a SQL script performs, in the order in which they occur.

To view the script summary:

1. Open the script you want to summarize in a query window.
2. On the **SQL Prompt** menu, click **Summarize Script**. The Summarize Script dialog box is displayed:



The script is displayed in grouped blocks of SQL. For example, the keywords within a CREATE statement are grouped into a block. To expand a block, click



or double-click the keyword.

The following symbols are used to draw your attention to commands that change the context:



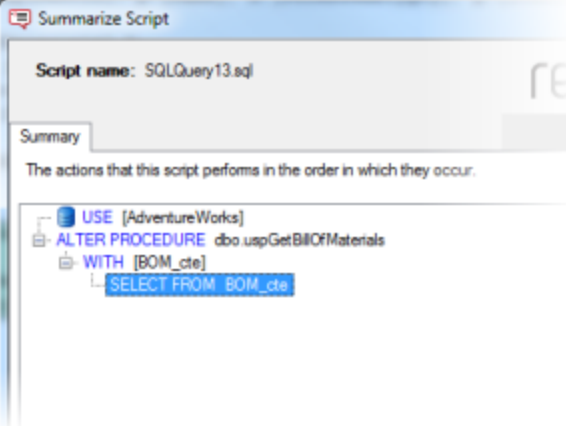
USE commands



SETUSER, EXECUTE AS LOGIN, and REVERT commands

You can click any keyword in the summary to highlight it in your query editor:


```
SELECT b.[ProductAssemblyID], b.[ComponentID], p.[Name], b.[PerAssemblyQty], p.[StartDate]
FROM [BOM_cte] cte
INNER JOIN [Production].[BillOfMaterials] p
ON b.[ProductAssemblyID] = cte.[ProductAssemblyID]
INNER JOIN [Production].[Product] p
ON b.[ComponentID] = p.[ProductID]
WHERE @CheckDate >= b.[StartDate]
AND @CheckDate <= ISNULL(b.[EndDate], @CheckDate)
)
-- Outer select from the CTE
SELECT b.[ProductAssemblyID], b.[ComponentID], p.[Name], b.[PerAssemblyQty], p.[StartDate]
FROM [BOM_cte] b
GROUP BY b.[ComponentID], b.[ProductAssemblyID], b.[PerAssemblyQty], b.[StartDate]
ORDER BY b.[BOMLevel], b.[ProductAssemblyID]
OPTION (MAXRECURSION 25)
END;
```



If you click on the main keyword for a block, the block is highlighted.

To more easily view the script in your query editor as you navigate the summary, it can be useful to move the Summarize Script dialog box onto another monitor. Alternatively, you can resize the dialog box as required.

Keyboard shortcut: hold Ctrl and press B then S to open the Summarize Script dialog. (SQL Prompt 5.3 only)

Refactoring your code

SQL Prompt allows you to perform the following code refactorings:

- [Renaming an object](#)
- [Encapsulating SQL as a new stored procedure](#)
- [Splitting a table](#)

These features are available from the **SQL Prompt 5** menu in SQL Server Management Studio.

Renaming an object

Renaming an object with SQL Prompt is available only in SQL Prompt Pro Edition.

SQL Prompt can create a script that allows you to rename objects in your database without breaking dependencies. You can rename the following:

- Tables (including columns)
- Views (including columns)
- Stored procedures (including parameters)
- Functions (including parameters)

When an object is renamed:

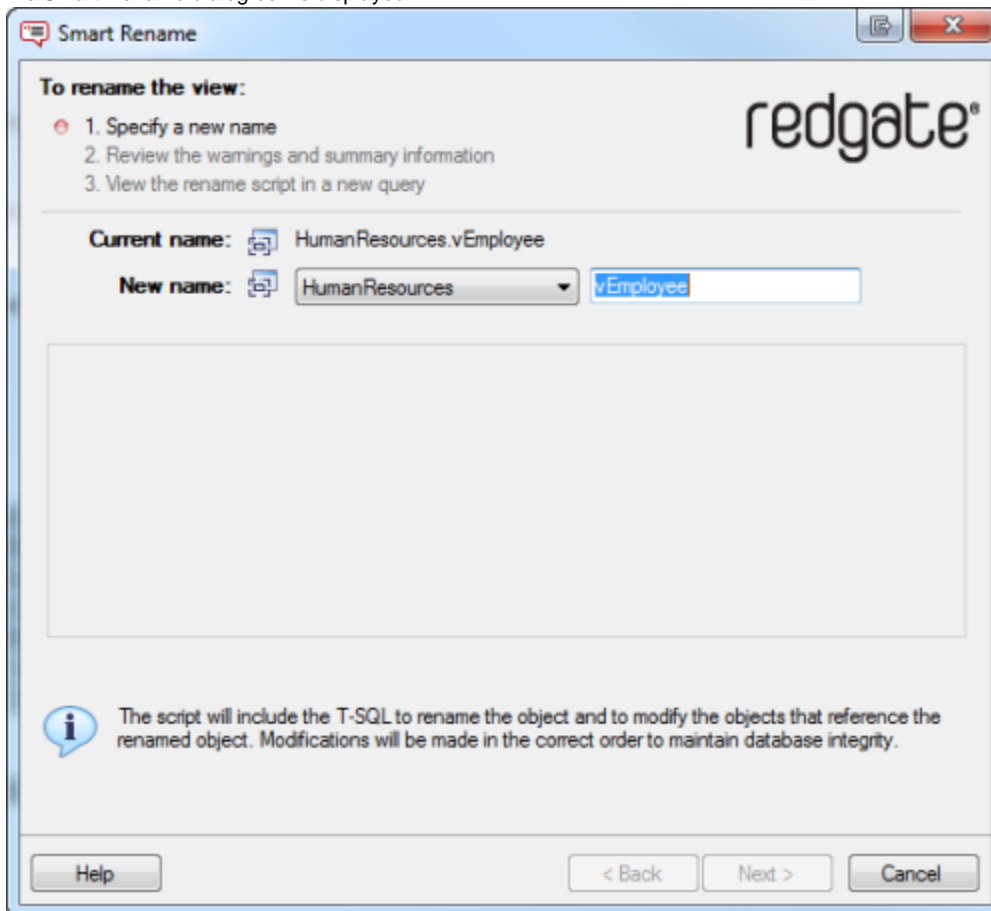
- SQL Prompt also modifies any objects that reference, or are referenced by, the renamed object to ensure that dependency links are not broken.
If you have previously renamed an object using SQL Server Management Studio or Enterprise Manager **Rename**, or the T-SQL **sp_rename** command, the object definition will contain the original name.
Any objects that reference this original name are *not* updated.
To help you locate objects that reference objects that no longer exist, see [Finding invalid objects](#).
- The original permissions and extended properties of the object are preserved.

You can rename encrypted objects in SQL Server 2000 if you are a member of the *sysadmin* server role; you cannot rename encrypted objects in SQL Server 2005 or 2008.

To rename an object:

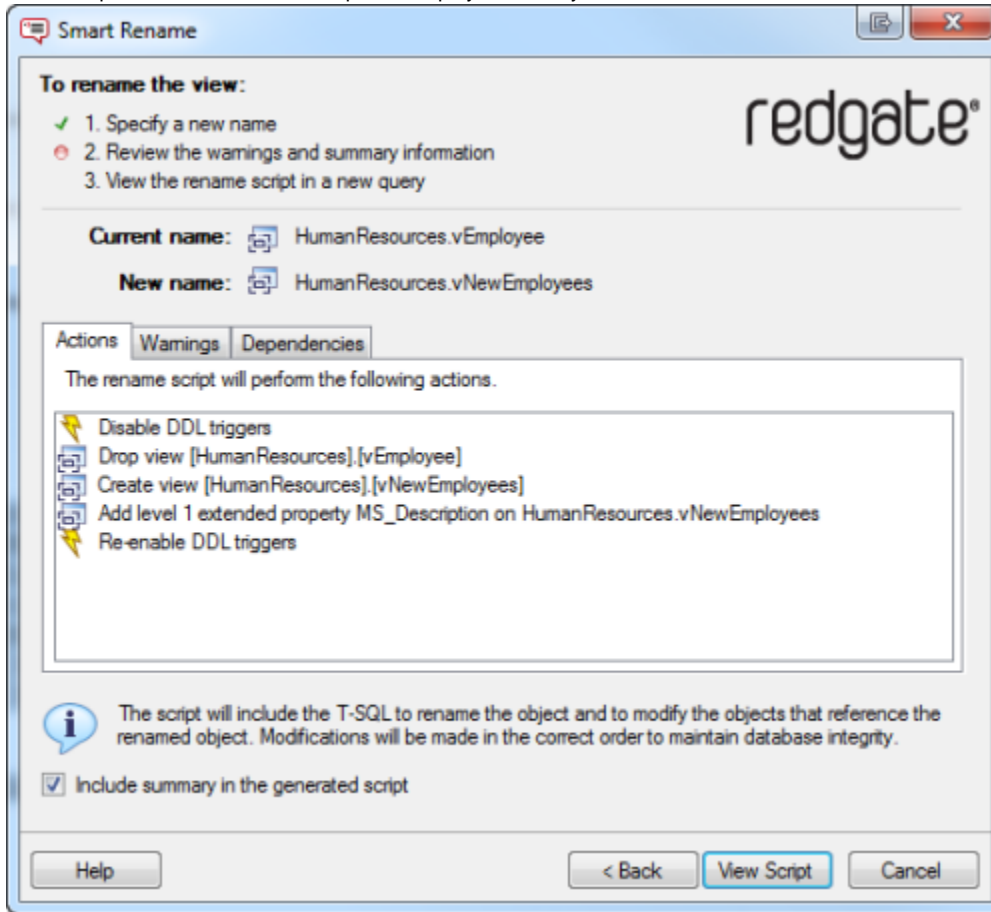
1. On the **SQL Prompt** menu, click **Smart Rename**, or in the SQL Server Management Studio Object Explorer pane, right-click the object you want to rename and select **Smart Rename**.

The Smart Rename dialog box is displayed:



2. If you want to change the object owner, select a new owner.
3. Type a new name for the object.
The name you enter must not already be in use.
4. Click **Next**.

SQL Prompt creates the rename script and displays summary information:



- **Actions** is a summary of the actions that the script will perform, in the order in which they will occur.
- **Warnings** displays information that you should consider prior to running the script, including any reasons the script might fail.
- **Dependencies** lists objects that will be modified by the script because they reference, or are referenced by, the primary table.

The actions and warnings are included in the split table script, below the header. If you do not want to include this information in the script, clear the **Include summary in the script** check box.

5. Click **View Script**.

The dialog box is closed, and the rename script is displayed in a new SQL Server Management Studio query window.

You are recommended to back up your database before you run the script.

6. Once you have reviewed the script, run it.

You must refresh the Object Explorer list before it will display the new name.

Encapsulating SQL as a new stored procedure

Encapsulating SQL as a new stored procedure is available only in SQL Prompt Pro Edition.

SQL Prompt allows you to create a new stored procedure from any block of valid SQL. To encapsulate SQL as a new stored procedure:

1. Open the SQL script you want to encapsulate in a query window.
2. Select the section of the script that you want to use for the new stored procedure. Parameters are created from variables in the script, therefore it is recommended that you declare any variables in the SQL you select.

If there are syntax errors in the SQL you select, SQL Prompt cannot create a new stored procedure.

3. On the **SQL Prompt** menu, click **Encapsulate As New Stored Procedure**.
The Encapsulate As New Stored Procedure wizard is displayed:

Encapsulate As New Stored Procedure

To encapsulate the SQL as a new stored procedure:

1. Specify a name for the new stored procedure
2. Choose whether to replace the source SQL
3. View the new stored procedure script in a new query

New stored procedure

Owner:

Name:

Parameters:

@CurrentFlag	[dbo].[Flag]	input
@EmployeeID	[int]	input
@HireDate	[datetime]	input
@PayFrequency	[tinyint]	input
@Rate	[money]	input
@RateChangeDate	[datetime]	input
@Title	[nvarchar] (50)	input

Preview:

```
CREATE PROCEDURE dbo.EncapsulatedProcedure
(
    @CurrentFlag AS [dbo].[Flag],
    @EmployeeID AS [int],
    @HireDate AS [datetime],
    @PayFrequency AS [tinyint],
    @Rate AS [money],
    @RateChangeDate AS [datetime],
    @Title AS [nvarchar] (50)
)
```

Note that SQL Prompt does not check for duplicate names. If a stored procedure with the owner and name you enter already exists, the new stored procedure script will fail when you run it.

Help < Back Next > Cancel

4. In the **Owner** box, type the name of the owner for the new stored procedure.
5. In the **Name** box, type a name for the new stored procedure.
Note that SQL Prompt does not check for duplicate names. If a stored procedure with this owner and name already exists, the SQL script to create the new stored procedure will fail when you run it.
Variables that will be used to create parameters are listed in the **Parameters** box. SQL Prompt automatically determines whether each variable will be an input parameter or an output parameter.
6. In the **Parameters** box, reorder the parameters as required using the



and



buttons.

If a



warning triangle is displayed by a parameter, move your mouse over the warning triangle to see details of the warning. The SQL script to create the new stored procedure is displayed in the **Preview** box.

7. Click **Next**.

The second page of the wizard is displayed:

8. If you do not want the new stored procedure to replace the SQL you selected, under **Source SQL**, select **Do not replace**. The **Preview** box now displays the SQL you selected, and shows how it will be modified if you chose to replace it with a reference to the new stored procedure.
9. Click **View Script**. The SQL script to create the stored procedure is displayed in a new query window. If you selected **Replace with reference to new stored procedure**, SQL Prompt replaces the SQL you selected with a reference to the new stored procedure.
10. Edit the stored procedure creation script if required, and run it. The new stored procedure is created.

Keyboard shortcut: hold Ctrl and press B then E to encapsulate the selected script as a new stored procedure. (SQL Prompt 5.3 only)

Splitting a table

Splitting a table is available only in SQL Prompt Pro Edition.

SQL Prompt enables you to split a table in your database into two tables.

Splitting a table can be useful if you want to:

- normalize a database
For example, by moving columns to a new secondary table to avoid insert anomalies.
- improve database performance
For example, by moving large but rarely used columns to a secondary table.
- change the database design
For example, to enable storing multiple customer addresses.

Example: splitting a simple table

The following example demonstrates how SQL Prompt splits a table. In the example, we want to split a table with three columns, *Name*, *ID* and *Category*:

Name	ID	Category
dog	1	mammal
ant	2	insect
cat	1	mammal
eagle	3	bird
rabbit	1	mammal
fly	2	insect

In the table, the value of *ID* determines the value of *Category*. When splitting the table, we can use SQL Prompt to:

- copy *ID* to a new secondary table
- move *Category* to a new secondary table

Error rendering macro 'multimedia' : null

Preparation

If you copy any of the following to a secondary table, the split table script will fail:

- XML columns
- Columns that allow NULL values
- Timestamp columns
- Partitioned columns

DML triggers that access data in the primary table are dropped when the table is split. You are strongly recommended to back up these triggers prior to launching the split table wizard.

For more detailed information, see [Preparing the table in detail](#).

Splitting the table

To split a table:

1. In the Object Explorer pane, right-click the table you want to split, and then click **Split Table**.
The Split Table wizard is displayed:

Split Table

To split the table:

1. Specify a name for the new secondary table
2. Assign columns
3. Create foreign key relationship
4. Review the summary information
5. View the split table script in a new query

Primary table name
Person.Contact

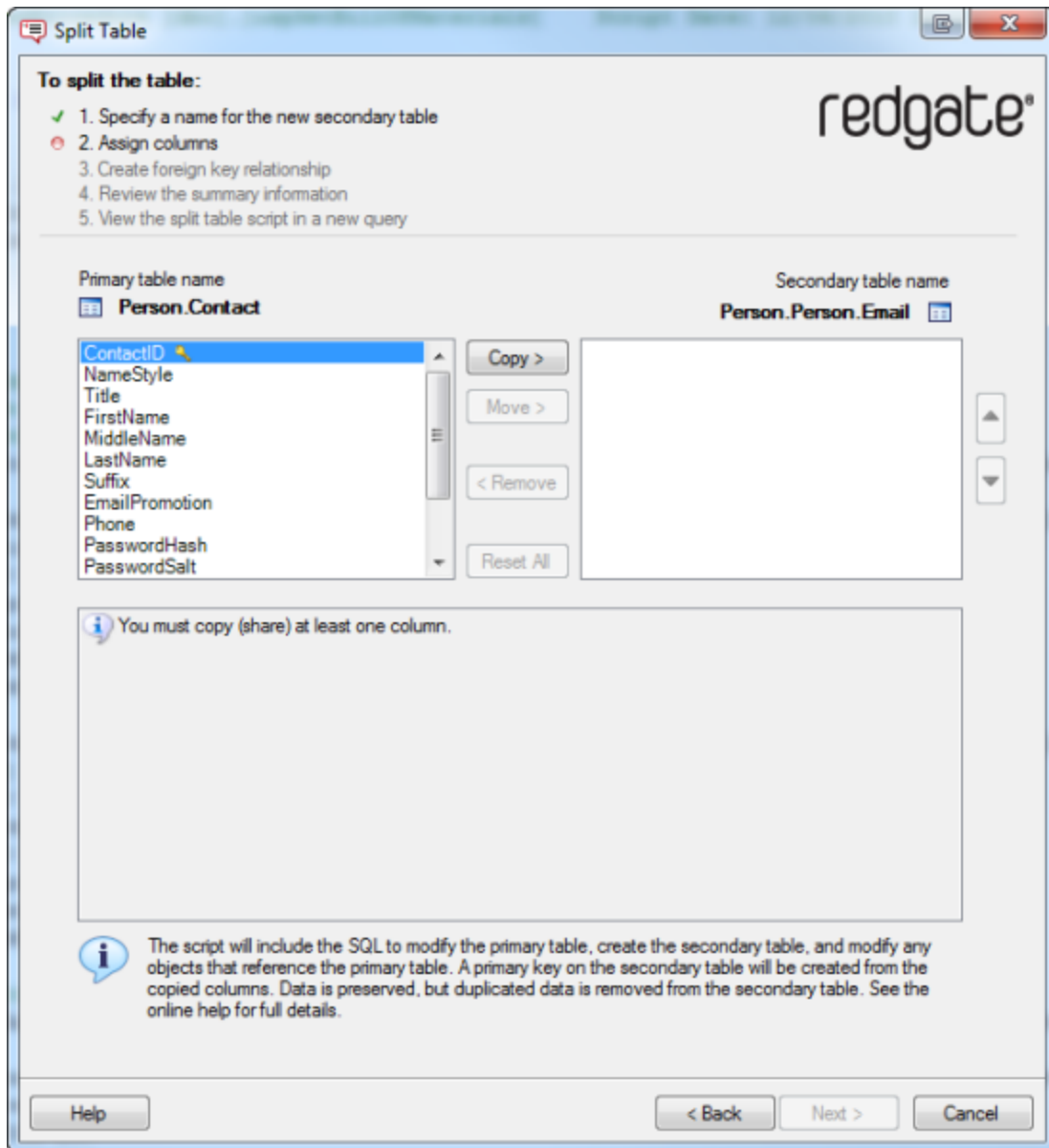
Secondary table name
Person Contact

ContactID
NameStyle
Title
FirstName
MiddleName
LastName
Suffix
EmailAddress
EmailPromotion
Phone
PasswordHash


The script will include the SQL to modify the primary table, create the secondary table, and modify any objects that reference the primary table. A primary key on the secondary table will be created from the copied columns. Data is preserved, but duplicated data is removed from the secondary table. See the online help for full details.

Help < Back Next > Cancel

2. In **Secondary table name**, type a name for the new secondary table.
You must enter a name that is not already in use.
If required, you can also select a new owner for the table.
3. Click **Next**.



4. To assign columns from the primary to the secondary table, select the column(s) and click the **Copy >** or **Move >** buttons. Note that:

- you must copy at least one column to the secondary table so the tables can maintain a relationship
- you cannot move primary key columns
 to the secondary table
- you can select multiple columns to move, copy, or remove using Ctrl or Shift
 To remove an assigned column from the secondary table, select the column and click **< Remove**.
 To remove all of the assigned columns from the secondary table, click **Reset All**.

5. If required, reorder the assigned columns in the secondary table using the



and



buttons.

You can reorder multiple columns simultaneously.

6. Click **Next**.

Split Table

To split the table:

- ✓ 1. Specify a name for the new secondary table
- ✓ 2. Assign columns
- ⊗ 3. Create foreign key relationship
- 4. Review the summary information
- 5. View the split table script in a new query

Primary table name
Person.Contact

Secondary table name
Person.Person.Email

Primary table columns:

- ContactID
- NameStyle
- Title
- FirstName
- MiddleName
- LastName
- Suffix
- Phone
- PasswordHash
- PasswordSalt
- AdditionalContactInfo

Secondary table columns:

- ContactID
- EmailAddress
- EmailPromotion

Create foreign key on:

☒ Primary table
☐ Secondary table

Person.Contact → **Person.Person.Email**

A foreign key will be created on **Person.Contact** that references the primary key on **Person.Person.Email**. This is for 1:1 and m:1 relationships.

Information: The script will include the SQL to modify the primary table, create the secondary table, and modify any objects that reference the primary table. A primary key on the secondary table will be created from the copied columns. Data is preserved, but duplicated data is removed from the secondary table. See the online help for full details.

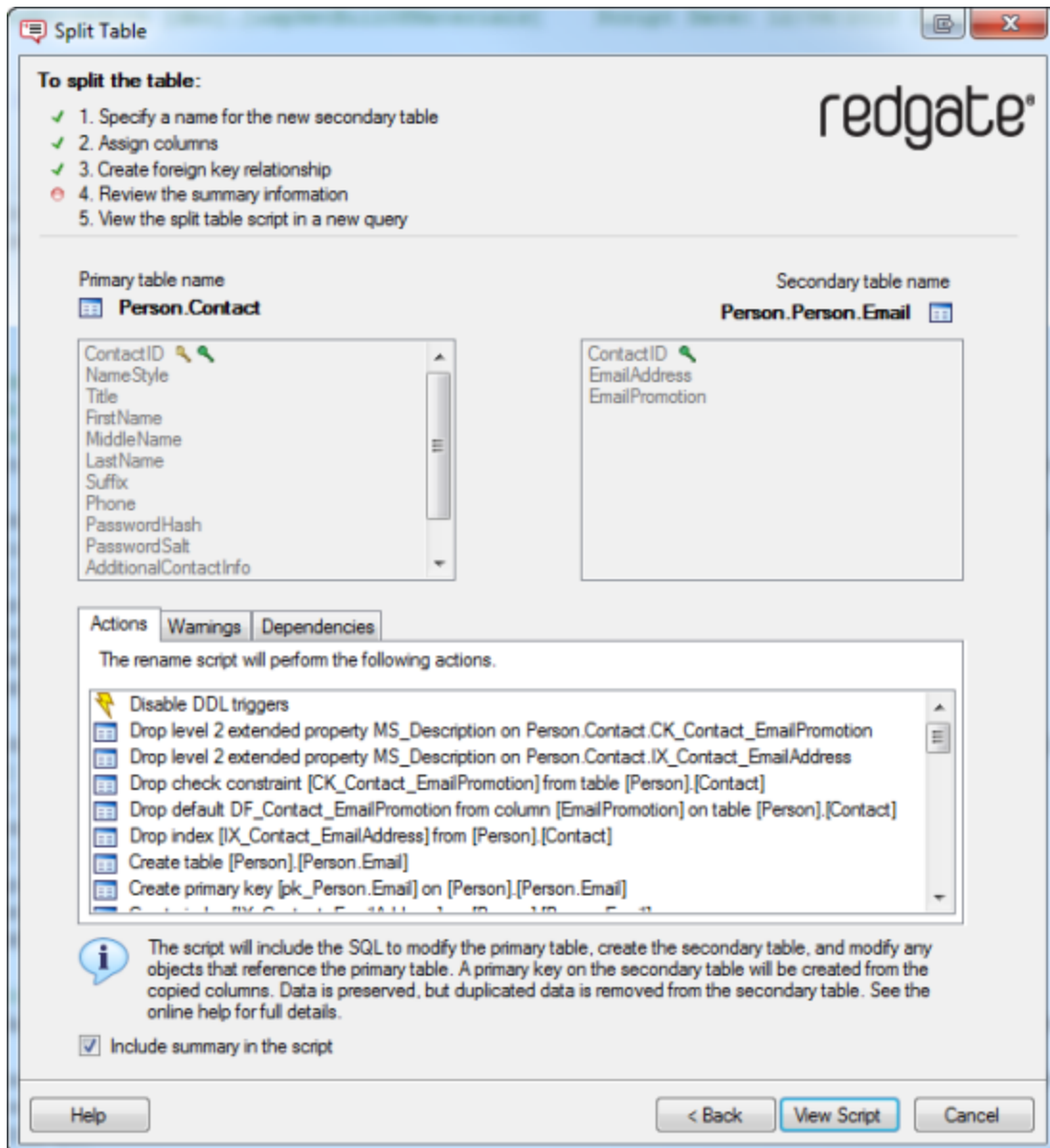
Buttons: Help, < Back, Next >, Cancel

By default, a foreign key is created on the primary table, to reference the primary key on the secondary table. This is for 1:1 or m:1 relationships.

If you have copied all the primary key columns from the primary table to the secondary table, you can choose to create the foreign key on the secondary table, to reference the primary key on the primary table. This is for 1:1 or 1:n relationships.

7. Click **Next**.

SQL Prompt creates the split table script and displays summary information:



- **Actions** is a summary of the actions that the script will perform, in the order in which they will occur.
- **Warnings** displays information that you should consider prior to running the script, including any reasons the script might fail.
- **Dependencies** lists objects that will be modified by the script because they reference, or are referenced by, the primary table.

The actions and warnings are included in the split table script, below the header. If you do not want to include this information in the script, clear the **Include summary in the script** check box.

8. Click **View Script**.

The wizard is closed, and the split table script is displayed in a new SQL Server Management Studio query window.

You are recommended to back up your database before you run the script

For detailed information describing what happens to your objects when you run the script, see [What happens when I run the split table script?](#)

9. Once you have reviewed the script, run it.

Script failure or cancellation

If a script fails, or if it is cancelled, in most cases SQL Prompt uses transactions to roll back any changes.

SQL Prompt displays a warning message if it will be unable to roll back all the changes; in these cases, your database will be in an unknown state.

Preparing the table in detail

To plan how the columns in the table are to be split and to prepare your data appropriately, consider the following points.

Copying columns

You must ensure that at least one column is copied to the secondary table so that it exists in both tables. SQL Prompt uses the copied columns to create the foreign key, and the primary key for the secondary table. Therefore, before you open the Split Table wizard, you must ensure that the primary table contains the columns that you want to copy. The values contained within the copied columns must uniquely identify each row of data in the secondary table.

Data

For columns that remain in the primary table, all data is retained.

For columns that are copied or moved to the secondary table, SQL Prompt uses the DISTINCT keyword when possible to ensure that only unique data is copied or moved.

However, for the following data types the DISTINCT keyword cannot be used:

- XML data
- SQL Server 2000 large object (LOB) data (*text*, *ntext*, *image*)

Therefore, if you choose to move or copy any of these data types, SQL Prompt cannot use the DISTINCT keyword when the secondary table is populated. This means that all data is moved or copied to the secondary table, including any duplicate data. If the copied columns contain duplicate data, the primary key cannot be created on these columns, and the script will fail.

Computed columns

If you move or copy a computed column, you must ensure that any columns that are referenced by the computed column are also moved or copied. If you do not do this, SQL Prompt displays a warning, and you cannot generate the script.

If you copy a computed column and all its referenced columns, and the computed column is persisted, SQL Prompt converts the computed column to a normal column in the secondary table so that data is retained. For example, a column that computes two columns that have data type *char(50)* is changed to a normal column of data type *char(100)*.

If you copy a computed column and all its referenced columns, and the computed column is *not* persisted, SQL Prompt displays a warning; if you have chosen to create the foreign key on the secondary table, the script will fail because it is not possible to create a foreign key on a column that is not persisted.

If you move a computed column and all its referenced columns, the computed column remains a computed column in the secondary table.

You cannot move a column from the primary table if it is required by a computed column in the primary table (but you can copy it).

XML columns

XML columns cannot be copied because a primary key cannot be created on an XML column.

Large object data

Columns that contain large object (LOB) data cannot be copied because a primary key cannot be created on LOB data.

Timestamp columns

Timestamp columns cannot be copied or moved. This is because data cannot be inserted into timestamp columns and the original values cannot, therefore, be inserted into the secondary table.

Common language runtime data

If a common language runtime (CLR) data column is copied or moved to the secondary table and the CLR data is not byte ordered (*IsByteOrder* is *false*), the script will fail. This is because the DISTINCT keyword cannot be used on a CLR column that is not byte ordered.

NULL values

NULL values are not allowed in copied columns in the secondary table. This is because the copied columns are used to create the primary key on the secondary table, and primary key columns cannot contain NULL values.

If any columns that you copy allow NULL values, a warning is displayed. When the script is run, these columns will be modified to disallow NULLs (set to NOT NULL).

If any of the data in the copied columns contains NULL values, the script will fail.

Identities

If you copy or move an identity column, the identity is copied or moved, and data will be inserted using the IDENTITY_INSERT setting.

Partition schemes

You cannot move a column over which the primary table is partitioned to the secondary table. If you attempt to do this, SQL Prompt displays a warning. You can, however, copy the column if required.

DML triggers

All DML triggers that access the data in the primary table are dropped. You are recommended to save these triggers prior to running the split table script if you will want to recreate them following the split.

Triggers that do not access the primary table are preserved, but are not duplicated on the secondary table.

What happens when I run the split table script?

When you run the script to split a table, the secondary table is created and populated with data. The primary table, and any objects referenced by the primary table, are modified.

The following describes the actions of the split table script in more detail.

Primary keys

The primary key in the primary table is unchanged.

A primary key for the secondary table is created based on the copied columns. SQL Prompt generates a name for the primary key on the secondary table automatically.

The primary key columns cannot contain NULL values. Therefore, if any of the copied columns allow NULL values, a warning is displayed. When the script is run, these columns will be changed to NOT NULL columns. If any of the data in the copied columns contains NULL values, the script will fail when it is run.

Note that if you copy the primary key columns from the primary table to the secondary table, the primary key itself is *not* copied; a new primary key is always created on the secondary table. Similarly, any options set on the primary table's primary key (using the WITH clause, such as clustering) are not copied.

Foreign keys

By default, a foreign key is created on the primary table to reference the copied columns in the secondary table (which are used as the secondary table's primary key). SQL Prompt generates a name for the foreign key automatically.

If you chose to create the foreign key on the secondary table, SQL Prompt creates the foreign key to reference the primary key columns in the primary table.

Existing foreign keys that reference other tables:

- on columns in only the primary table, are preserved on the primary table
- on columns in only the secondary table, are created on the secondary table
- on columns that are copied, are preserved on the primary table and duplicated on the secondary table
- on columns in both tables that are not copied, are deleted

Permissions

Table-level permissions on the primary table are duplicated on the secondary table.

Column-level permissions:

- on columns in only the primary table, are preserved on the primary table
- on columns in only the secondary table, are created on the secondary table
- on copied columns, are preserved on the primary table and duplicated on the secondary table

Indexes

Indexes that reference:

- columns in only the primary table, are preserved on the primary table
- columns in only the secondary table, are created on the secondary table
- copied columns, are preserved on the primary table and duplicated on the secondary table
- columns from both tables that are not copied, are dropped

Note that clustered indexes are created as nonclustered indexes on the secondary table. This is because the primary key on the secondary table is clustered to improve access speed when the tables are joined in queries (for more information, see [Dependencies](#)).

Constraints

Table-level constraints:

- that reference columns in only the primary table, are preserved on the primary table
- that reference columns in only the secondary table, are created on the secondary table
- that reference copied columns, are preserved on the primary table and duplicated on the secondary table
- that reference columns from both tables that are not copied, are dropped

Column-level constraints:

- on columns in only the primary table, are preserved on the primary table
- on columns in only the secondary table, are created on the secondary table
- on copied columns, are preserved on the primary table and duplicated on the secondary table

Note that DEFAULT constraints are renamed when they are created on the secondary table; the name of the secondary table is appended to the original constraint name. For example, a DEFAULT constraint called *ConstraintA* that is created on a secondary table called *TableB* is created as *ConstraintA_TableB*.

Extended properties

Table-level extended properties are preserved on the primary table and duplicated on the secondary table.

Extended properties on level 2 objects (columns, constraints, triggers, and indexes):

- on objects in only the primary table, are preserved on the primary table
- on objects in only the secondary table, are created on the secondary table
- on copied objects, are preserved on the primary table and duplicated on the secondary table
- on objects from both tables that are not copied, are dropped (because the objects are dropped)

Filegroups

If the primary table was created on a filegroup, the secondary table is created on the same filegroup.

Partition schemes

If the primary table has been partitioned over a column, and that column stays in the primary table following the split, the primary table remains partitioned.

If the column over which the table is partitioned is moved to the secondary table, a warning is displayed and SQL Prompt does not allow you to generate the script.

Full-text indexes

If a full-text index exists on a column that is moved to the secondary table, the full-text index is added to the column in the secondary table.

However, note that full-text indexing cannot be added to a column from within a transaction. Therefore, if the script fails, SQL Prompt will not be able to roll back the script, and your database will be in an undetermined state. If you choose to move a column that has a full-text index, SQL Prompt displays a warning.

Dependencies

If you have only *copied* columns from the primary table to the secondary table, any objects referenced by the primary table are not modified.

If you have *moved* any columns from the primary table to the secondary table, any objects referenced by the primary table are modified as follows.

For the object types listed below, objects that referenced the primary table before it was split are modified so that they reference both tables:

- functions
- stored procedures
- views
- DML triggers
- DDL triggers

The following modifications are made to SELECT statements:

- the table reference is replaced with a JOIN subquery that joins the primary and secondary tables based on the copied columns
- an alias is created for the JOIN subquery
- the SELECT column list and the following clauses are modified to reference the appropriate columns
 - WHERE
 - GROUP BY
 - ORDER BY
 - HAVING
- SELECT * clauses are expanded
- wherever possible, fully-qualified names are used
- table hints are preserved for the primary table and duplicated for the secondary table

- table samples are removed

The following modifications are made to INSERT, UPDATE, and DELETE statements:

- INSERT statements are split into two: one for the primary table, and one for the secondary table
- the column list and WHERE clause are modified to reference the appropriate columns
- for statements that include a FROM clause, the table reference is replaced with a JOIN statement that joins the primary and secondary tables based on the copied columns

Note that the generated script for INSERT, UPDATE, and DELETE statements may not precisely reflect your intentions for the data, particularly for complex requirements. Therefore, you should review these statements in detail before you run the script. Foreign keys that reference the primary key on the primary table are not changed.

All DML triggers that accessed columns on the primary table before the split are dropped.

Managing SQL Prompt behavior

You can set various options for how SQL Prompt behaves in the SQL Prompt Options dialog box. This page describes the options available on the first page of the SQL Prompt Options dialog, the **Behavior** page; for other more specific options, see the following:

- [Customizing inserted code](#)
- [Options for formatting your code](#)
- [Managing connections and memory](#)
- [Changing the location of the settings, cache files and snippets folders](#)
- [SQL Prompt Labs](#)

To display the SQL Prompt Options dialog box, on the **SQL Prompt** menu, select **Options**.

Turning off SQL Prompt code suggestions

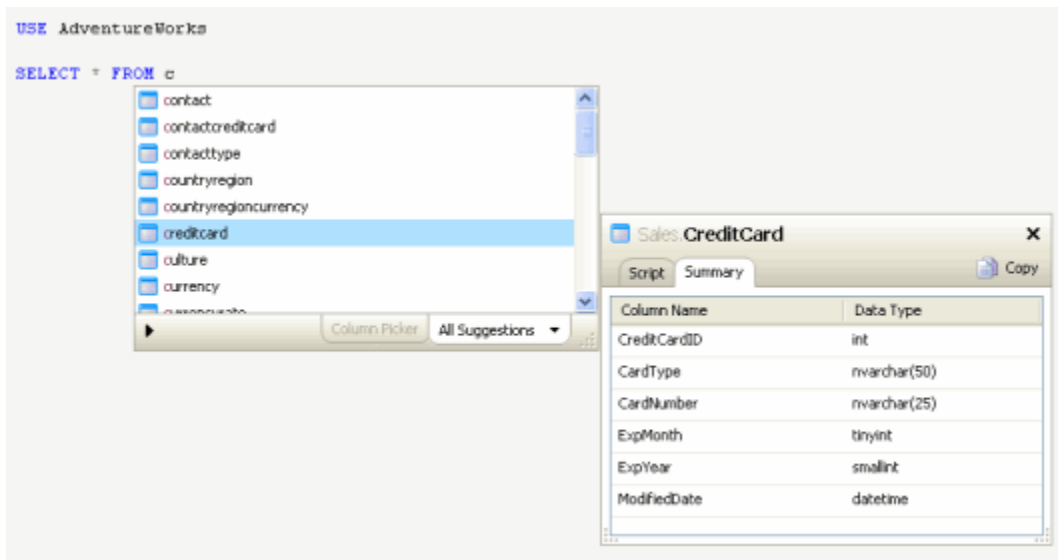
To stop SQL Prompt offering code suggestions:

- In SQL Prompt 5.2 and later, clear the **Enable code suggestions** check box.
- In SQL Prompt 5.1 and earlier, clear the **Use SQL Prompt** check box.

The suggestions box will not be displayed until you select the check box again.

Turning off the object definition box

When you select a suggestion, the object definition box is displayed, containing information about the selected object. By default, for tables and views, a summary is displayed, listing the column names and data types:



To view the object creation script instead, click the **Script** tab.

If you don't want the object definition box to be automatically displayed, clear **Display object definitions**.

You can view the object definition for a specific object in your query by clicking its tooltip.

Viewing tooltips


Object tooltips display the fully qualified object name for an object when you move the mouse pointer over the object name in your SQL script.



For tables, views, and stored procedures, you can click this tooltip to view the object definition.

Tooltips can also be displayed for parameters, including parameters for built-in functions.

```
SELECT substr( )
```



If you don't want tooltips to be displayed, clear **Objects** or **Parameters** under **Show tooltips for** in the Tooltips section.

Changing the insertion keys

Insertion keys are the keys that insert the currently selected suggestion into your code:

Insertion keys

Insert selected suggestions into your code when any of the following keys are pressed:

<input checked="" type="checkbox"/> Enter	<input checked="" type="checkbox"/> Space bar	<input checked="" type="checkbox"/> Parentheses ()	<input checked="" type="checkbox"/> Closing square bracket (])
<input checked="" type="checkbox"/> Tab	<input checked="" type="checkbox"/> Dot (.)	<input checked="" type="checkbox"/> Comma (,)	

Insertion keys are also used to insert snippet code when you type a snippet into your query. The default insertion keys are ENTER and TAB.

Restoring options to their default setting

You can reset the options in SQL Prompt in two ways:

- for a specific page in the Options dialog box
- for all options

To restore the options on a single page

Click **Restore Defaults** in the top right corner:

Inserted code > Special characters Restore Defaults

Brackets

<input type="checkbox"/> Enclose identifiers within square brackets []
<input checked="" type="checkbox"/> Add parentheses () when inserting a function or data type

Closing characters

Automatically insert the corresponding closing character when you type any of the following:

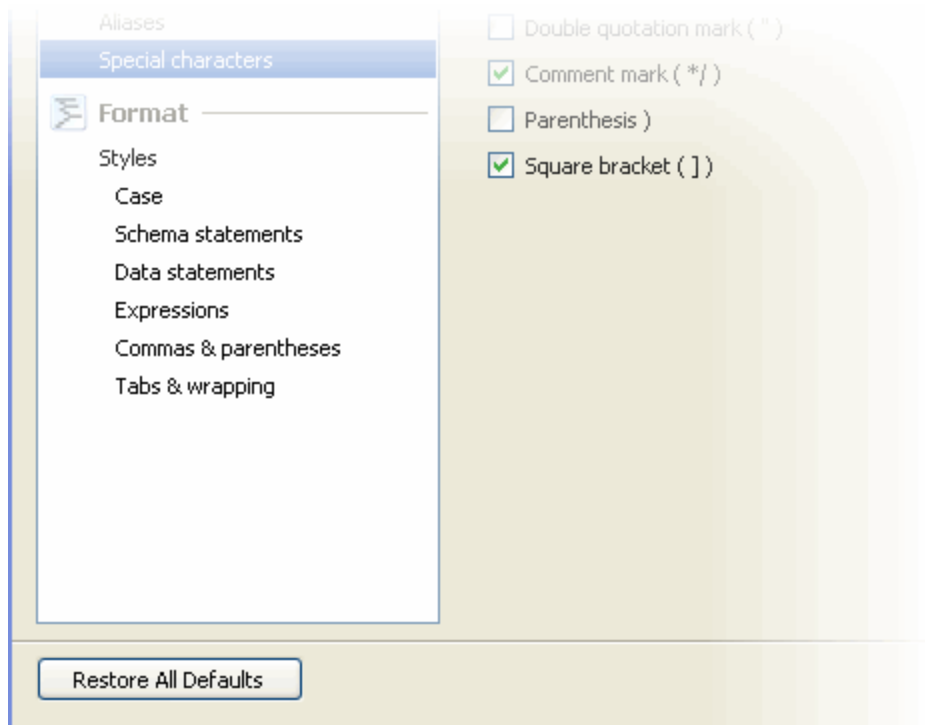
<input checked="" type="checkbox"/> Single quotation mark (')
<input type="checkbox"/> Double quotation mark (")
<input checked="" type="checkbox"/> Comment mark (/*)
<input type="checkbox"/> Parenthesis)
<input checked="" type="checkbox"/> Square bracket (])

This will restore only the options on this page to the SQL Prompt defaults.

Options on all other pages are not affected.

To restore all SQL Prompt options

Click **Restore All Defaults** in the bottom left corner:

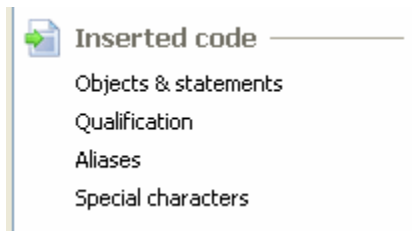


Options on all pages are reset to the SQL Prompt defaults.

Restore All Defaults also resets all formatting options. If you want to save your formatting options, you should save them as a style. This allows you to reapply them, even if you have restored all defaults. See [Using styles to format your code](#).

Customizing inserted code

To customize how SQL Prompt inserts suggestions into your code, use the **Inserted code** pages on the **SQL Prompt Options** dialog box:



To display the SQL Prompt Options dialog box, on the **SQL Prompt** menu, select **Options**.

Objects & statements

Contains options for inserting full syntax for various kinds of statement.

For example, you can specify that SQL Prompt inserts column names, data types and default values for ALTER, EXEC and INSERT statements:

```
INSERT INTO Sales.Customer
( TerritoryID ,
  AccountNumber ,
  CustomerType ,
  rowguid ,
  ModifiedDate
)
VALUES ( 0 , -- TerritoryID - int
  '' , -- AccountNumber - varchar(10)
  '' , -- CustomerType - nchar(1)
  NULL , -- rowguid - uniqueidentifier
  '2009-08-03 15:35:51' -- ModifiedDate - datetime
)
```

Qualification

Contains options for specifying whether object or column names are qualified when inserted.

Object and column names are automatically qualified in certain cases, for example:

- When columns in more than one table have the same name, and they need to be qualified to avoid ambiguity (only the affected column names are qualified).
- When you insert JOIN conditions from the suggestions box, both column names are qualified.
- For cross-database and linked server queries, qualified column names may also include the owner name.

Aliases

See [Working with aliases](#).

Special characters

Contains options for automatically inserting certain characters into your code:

- Enclose all identifiers in square brackets:

```
USE [AdventureWorks]

SELECT * FROM [Person].[Address]
```

- Add parentheses when inserting a function or data type:



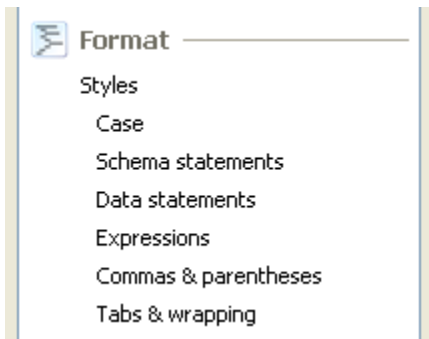
The screenshot shows a code editor with the text `SELECT COL_NAME()`. A tooltip is displayed over the opening parenthesis, showing the function signature `COL_NAME(table_id, column_id)` and the parameter `table_id`.

SQL Prompt displays a tooltip containing the expected parameters whenever you type or insert brackets after a function.

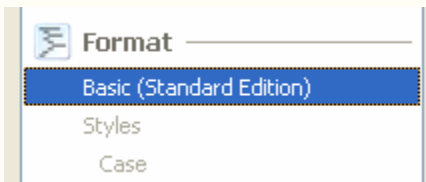
- Automatically insert closing characters
Insert the matching closing character when you type a quotation mark, comment character, or bracket.

Options for formatting your code

To set formatting options, on the **SQL Prompt** menu, select **Options**. The formatting options are all available in the **Format** group:



If you are using SQL Prompt Standard Edition, the format pages are unavailable; but you can set a number of options for how inserted code is formatted on the Basic (Standard Edition) page:



To review the effect of any changes you make, use the Preview tab:

- The **Sample SQL** view shows how an example query would be formatted using the current options.
- The **Current Query** view shows how the first few lines of the query that is currently active in the editor window of your SQL editing application (eg SQL Server Management Studio) would be laid out using the current options.

Case

Case options are available in both Standard and Pro Editions of SQL Prompt:

- In Standard Edition, case options are on the **Format > Basic (Standard Edition)** page
- In Pro Edition, case options are on the **Format > Case** page

Selecting **UPPERCASE** or **lowercase** will replace any words you have typed with the selected case, and also insert any suggestions using the selected case.

For example, if UPPERCASE is selected:

- If you have already typed `select` or `Select` in your query, SQL Prompt replaces the word with `SELECT`.
- If you type `sel` or `Sel` and then insert `SELECT` from the suggestions box, SQL Prompt inserts `SELECT`.

Selecting **Leave as is** will leave the case of any letters you have already typed, but insert the remainder of the word using the case of the last typed letter.

For example:

- If you type the entire word `Select`, SQL Prompt leaves the word exactly as typed.
- If you type `Sel`, SQL Prompt will insert `Select`.
- If you type `SEL`, SQL Prompt will insert `SELECT`.

Changing the case in pasted code

When you paste code into your query editor, you can apply your preferred case to that code. This is only available in SQL Prompt Pro Edition.

1. Set the case options for keywords, functions, and data types as required on the **Format > Case** page of the SQL Prompt Options dialog box.
2. Click **OK** to save these settings.
3. In your query editor, select all the lines of code to which want to apply the case.
4. On the **SQL Prompt** menu, select **Format SQL**.

This will also apply all other format settings (such as indentations, new line settings, and so on) to that code.

Schema statements

Options for formatting new lines and indents in CREATE and ALTER statements.

Data statements

Options for formatting new lines, parentheses and indents in SELECT, INSERT, UPDATE and DELETE statements.

Expressions

Options for formatting parts of statements, for example how Boolean and other types of operators should appear.

Commas & parentheses

Options for formatting where commas should appear in your SQL statements (at the end or beginning of lines), and how they should align with other lines of code.

Tabs & wrapping

Options for tabs and spaces and when to wrap lines.

Formatting SQL in your current editor window

To format the SQL code in the current query editor or the highlighted SQL code fragment, do any of the following:

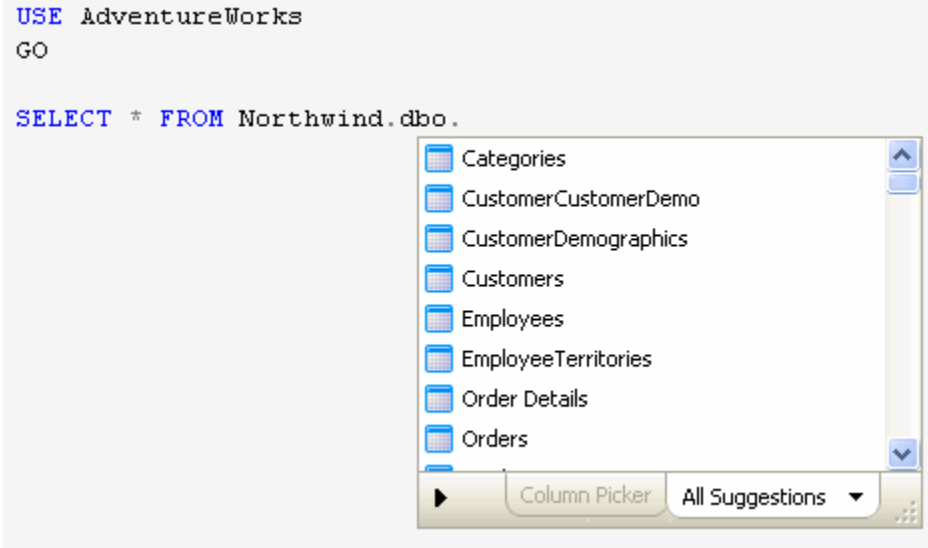
- on the **SQL Prompt** menu, select **Format SQL**
- press Ctrl + K,Y (hold down Ctrl and press K then Y)

Saving format settings

You can use styles to save your preferences for formatting code. See [Using styles to format your code](#).

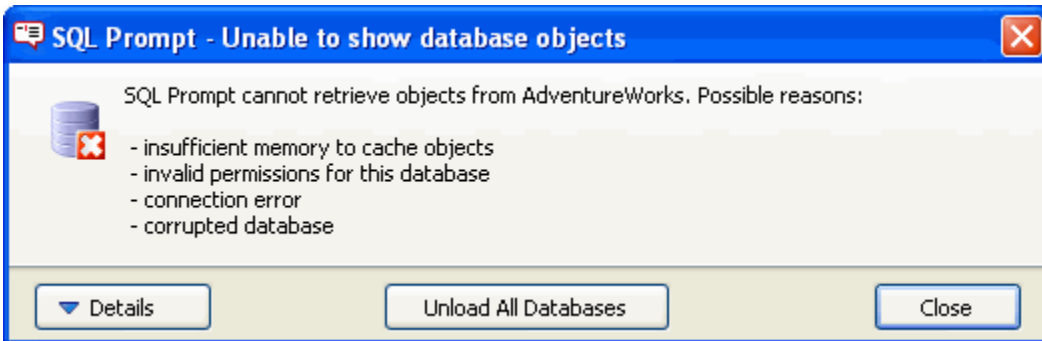
Managing connections and memory

SQL Prompt will automatically connect to other databases specified in your query. For example:

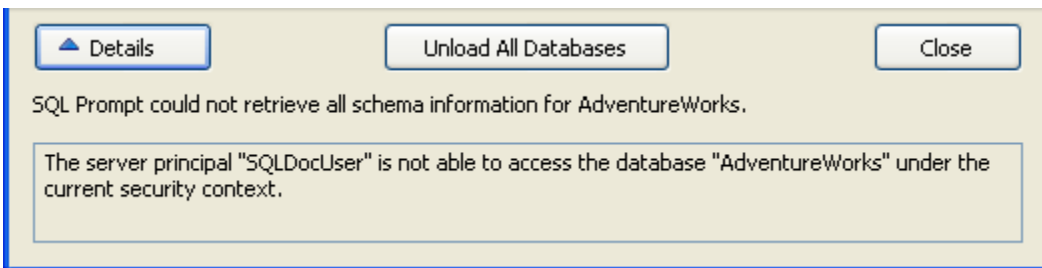


In this example, SQL Prompt loads objects from the Northwind database as soon as you type the database name and then continue with your query. For large databases, you may have to wait a few seconds while SQL Prompt loads the objects.

If you do not have permissions to connect to that database, then a message is displayed:



Click **Details** to see more information about the connection error:



Once you close this message, you can continue to write your query, but no suggestions from that database will appear in the suggestions box.

This message may also be displayed for reasons other than permissions issues, for example, when SQL Prompt doesn't have enough memory to store the objects for this database.

Connecting to different databases in the same session

If you connect to a number of different databases during the same session, SQL Prompt will load all the objects for each database as required, without unloading any of the previously stored objects. This means that you can go back to previous queries and continue to see suggestions for any database to which you have previously connected.

Refreshing suggestions when a database schema changes

If you are writing a query on a database while someone else is making changes to the structure of that database, the suggestions offered by SQL Prompt may become out of date.

For example, if someone has added several tables to the database while you have been connected to that database, these tables will not appear in the suggestions box.

To reload the database objects, on the **SQL Prompt** menu, select **Refresh Suggestions** or press Ctrl + Shift + D. Any changes to the database structure will now be reflected in the suggestions box.

You should also refresh suggestions when you have created a new object in the current query, and executed that block of code. For example, if you create a table `NewObject` and execute the statement, then continue to write the query, the `NewObject` table will not appear in the suggestions box until you refresh the suggestions.

Notes

- Refreshing suggestions applies only to the database to which you are currently connected.
- SQL Prompt automatically refreshes a database's objects when you close down your query editor application, then re-open it and re-connect to that database.

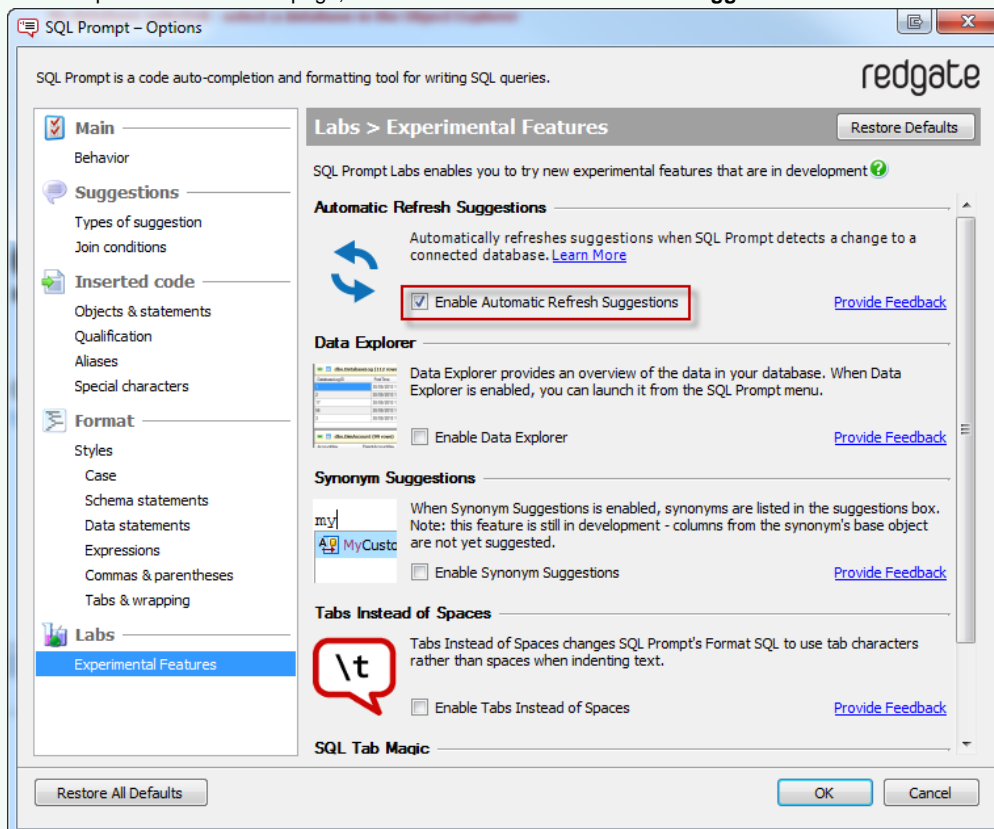
Experimental feature in SQL Prompt 5.3: Automatic Refresh Suggestions

If the SQL Prompt Labs feature **Automatic Refresh Suggestions** is enabled, SQL Prompt automatically refreshes the suggestion list when a change to a connected database is detected.

⚠ SQL Prompt Labs features are still in development, so might run slowly, have bugs, or even crash.

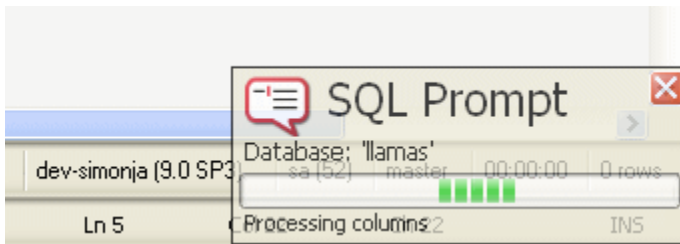
To enable Automatic Refresh Suggestions:

1. In SQL Server Management Studio or Visual Studio, click the SQL Prompt 5 menu, and then click **Experimental Features**.
2. On the Experimental Features page, select **Enable Automatic Refresh Suggestions**:



Managing the SQL Prompt memory

SQL Prompt loads database objects into its memory whenever you connect to a database in SQL Server Management Studio or Visual Studio, in order to display suggestions from that database. For databases with large schemas, a progress dialog may be displayed while SQL Prompt loads the objects into memory:



If you connect to several databases with large schemas, this may eventually result in SQL Prompt running out of memory. In this case, a warning message will be displayed.

If this message is displayed, you can clear the SQL Prompt memory by clicking **Unload All Databases**. SQL Prompt will load each database's objects as required.

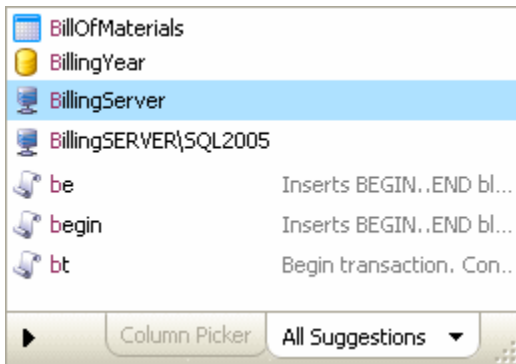
If you close the message, you can continue to write your query, but no objects from that database will appear in the suggestions box.

Clearing the SQL Prompt memory

To clear the SQL Prompt memory at any time, close your editor application. This unloads all objects from all databases from memory.

Connecting to a linked server

Linked server objects for the current instance are displayed in the suggestions box:



When you select a linked server, SQL Prompt attempts to connect automatically to that server, using the mapped credentials for that linked server object.

- If you have permissions to connect to the server, suggestions from that server will be loaded into SQL Prompt's memory.
- If you do not have permissions to connect to that server, a message is displayed. Click **Close** to continue writing your query; no suggestions from the linked server will appear in the suggestions box.

To stop SQL Prompt attempting to connect to linked servers:

1. On the **SQL Prompt** menu, select **Options**.
2. Go to the **Suggestions > Types of suggestions** page.
3. Turn off **Enable linked server support**.

Changing the location of the settings, cache files and snippets folders

By default, the SQL Prompt settings, cache and snippets files are stored in:

- *%LOCALAPPDATA%\Red Gate\SQL Prompt 5* (Windows Vista, Windows 2008 and later)
- *%UserProfile%\Local Settings\Application Data\Red Gate\SQL Prompt 5* (Windows XP and Windows 2003)

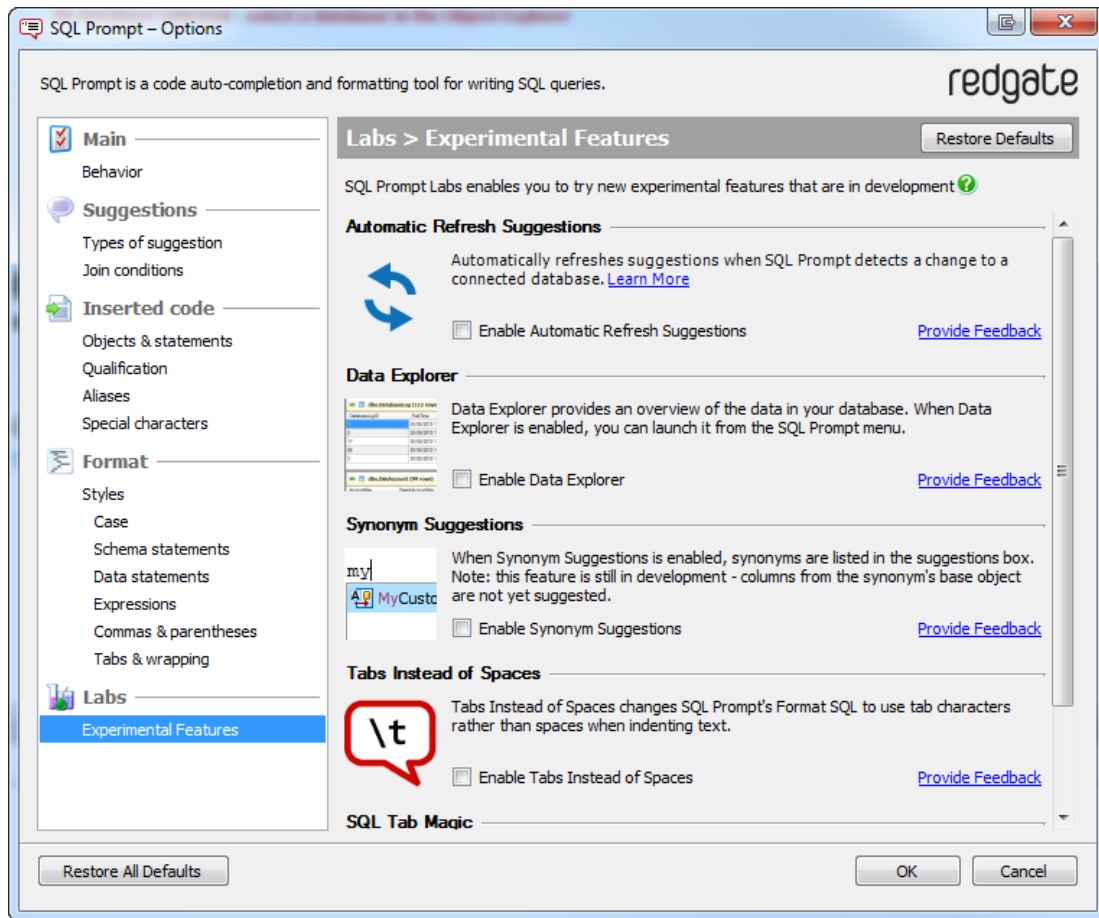
To change the locations of these files so they aren't stored in your user profile:

1. Open Registry Editor (regedit.exe).
2. Navigate to the SQL Prompt registry key: *HKEY_CURRENT_USER\Software\Red Gate\SQL Prompt <version number>*
3. Right-click the key and select **New > String Value**.
4. Set the string name to *Options Folder*, *Cache Folder* or *Snippets Folder* as appropriate.
5. Right-click the string and select **Modify** to open the Edit String dialog.
6. In the **Value data** box, type the location in which you want to store the files.

If you want to store existing snippet files in the new location, you will need to move them manually.

SQL Prompt Labs

SQL Prompt 5.2 and later includes "Labs", a set of experimental features you can enable from the SQL Prompt options:



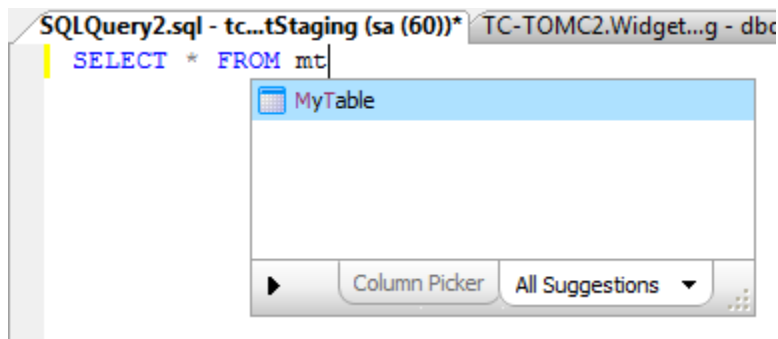
Partial Match Suggestions

Partial Match Suggestions are a full feature of SQL Prompt 5.3.

Partial Match Suggestions adds two new types of suggestion to SQL Prompt's code suggestions:

CamelCase, or compound word suggestions

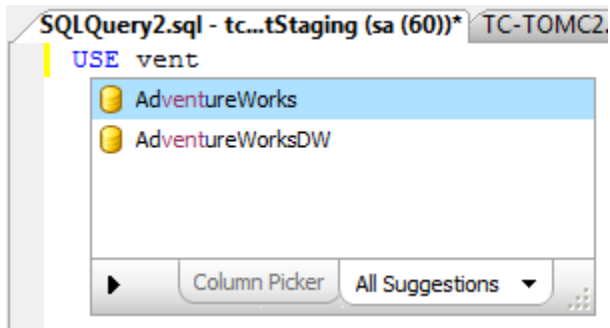
For example, if you type *mt*, the object *MyTable* is suggested:



(This also works on names with underscores, such as *my_table*)

Mid-string suggestions

For example, if you type *vent*, *Ad*vent*ureWorks* is suggested:



This is particularly useful if you can only remember part of an object name.

These new types of suggestion will appear below the standard suggestions you are used to in the suggestions list - the order is as follows:

1. Standard (prefix)
2. CamelCase
3. Mid-string

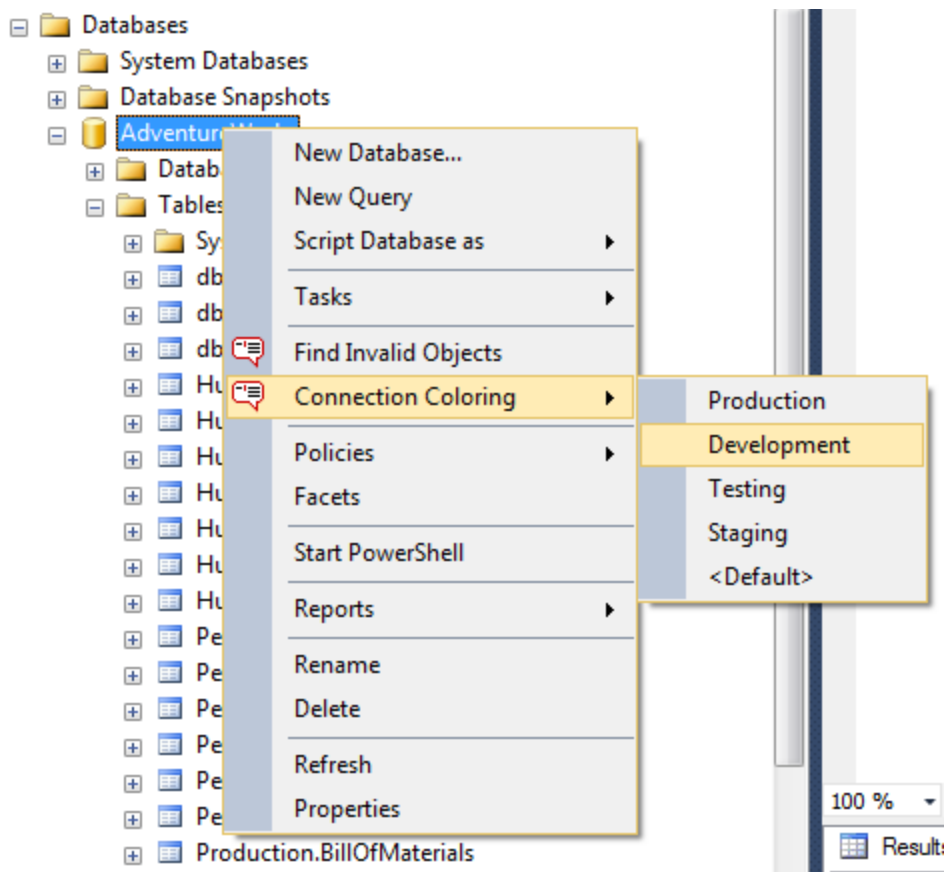
Tabs Instead of Spaces

Tabs Instead of Spaces changes the behavior of the Format SQL command so that it uses tab characters rather than spaces to indent text.

Query Tab Coloring (only in SSMS 2012)

Query Tab Coloring allows you to color query tabs by server connection in SQL Server Management Studio 2012.

Once you've enabled Query Tab Coloring, you can set a color category for a server by right-clicking it in the Object Explorer:



The default color categories are:

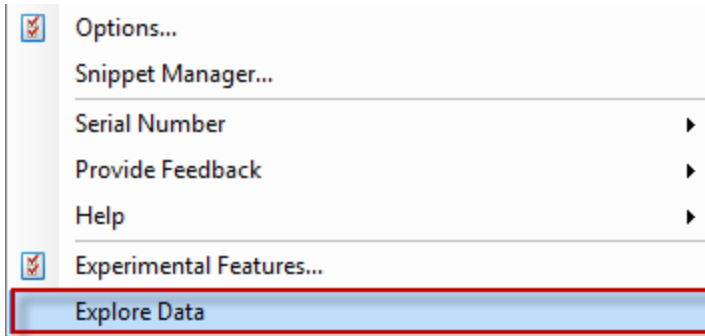
Production: **Red**
 Development: **Green**
 Testing: **Orange**
 Staging: **Blue**

Once you set a color category for a server, each query tab connected to the server will be colored automatically:

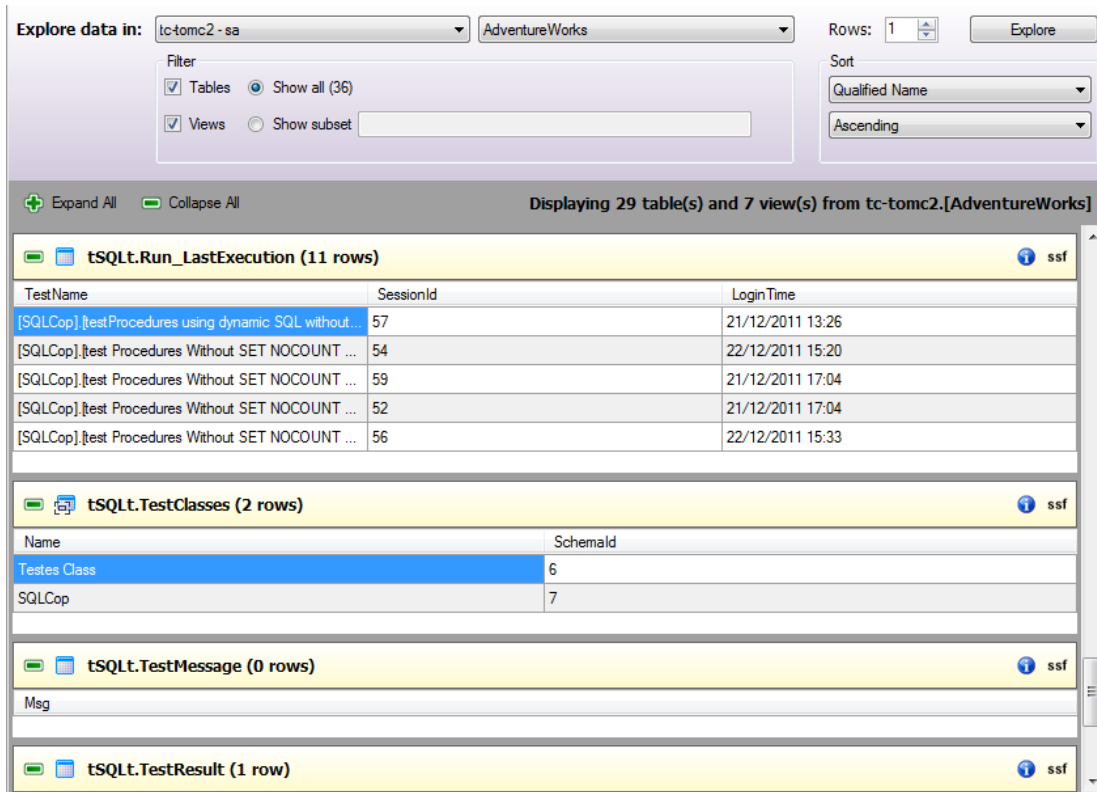


Data Explorer

Data Explorer provides an overview of the data in your database. Once you've enabled Data Explorer, it appears as a new SQL Prompt menu item:



Data Explorer displays rows of data in the tables and views of a database:

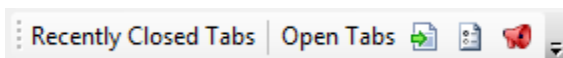


You can use Data Explorer to:

- filter and sort the data using the upper pane controls
- view the creation script for an object, as well as a summary of its columns by clicking the blue information icon
- create a `SELECT * FROM` query for a table or view by clicking **ssf**

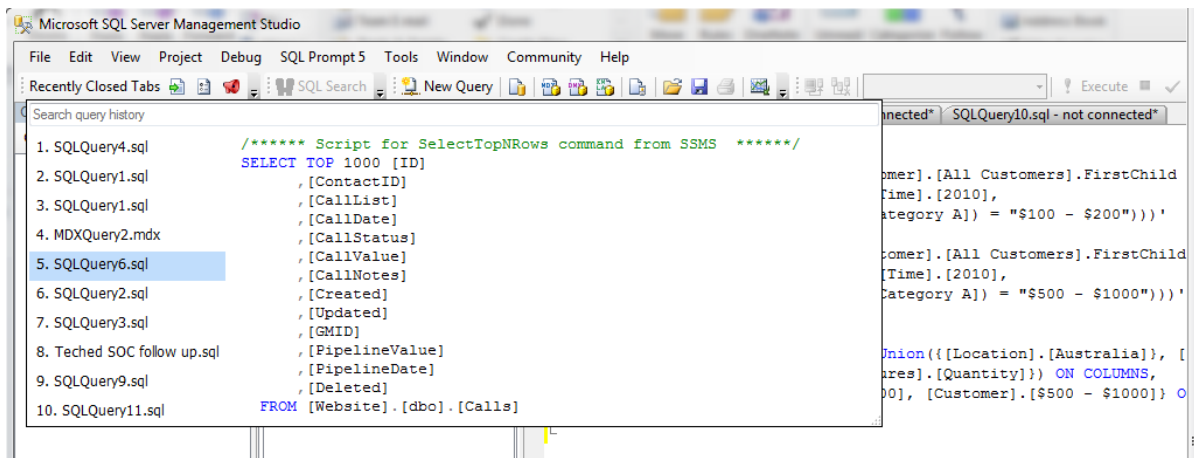
SQL Tab Magic

SQL Tab Magic is an add-in for SSMS. Once installed, you'll get some new buttons on your SSMS toolbar:



Recently Closed Tabs

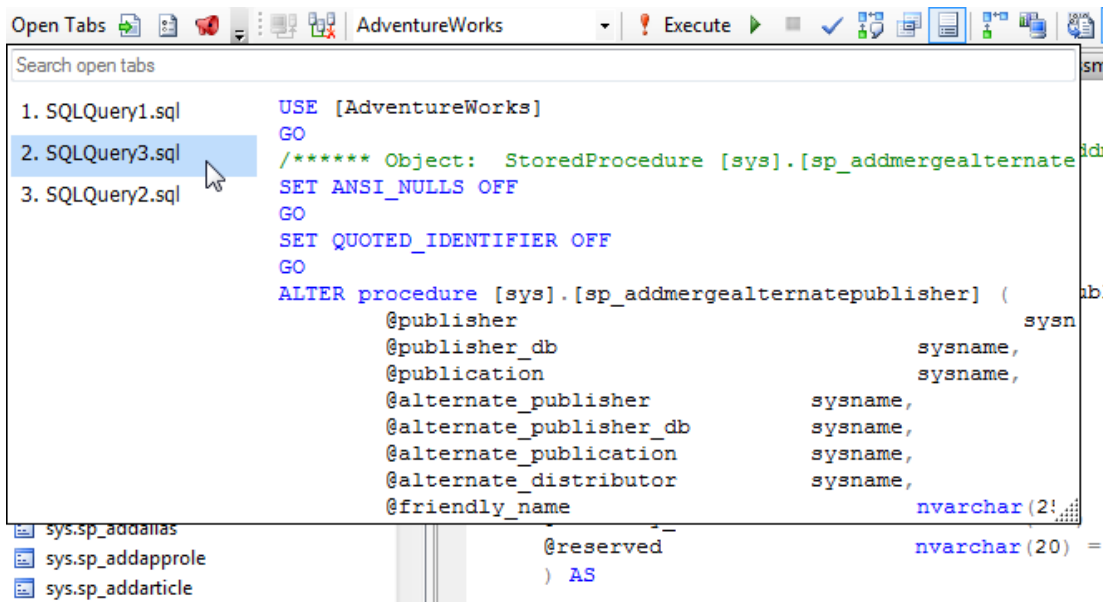
This feature shows you a list of tabs you've recently closed:



You can also click a tab in the list to re-open it. This is very useful if you accidentally close an unsaved query.

Open Tabs

This feature shows you a list of all your open tabs, with a useful preview of their contents. This makes it much easier to quickly scan and find the query you're looking for:



Find Open Tab

This feature allows you to search your open tabs. The search looks both at a tab's title and its contents:


```

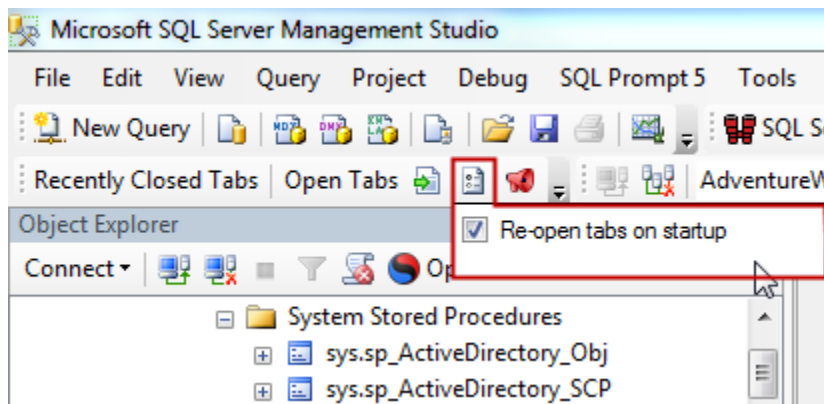
_publisher_db      sysname,          /* Alternate publisher_db */
_publication        sysname,          /* Alternate publication */
_distributor        sysname,          /* Alternate publisher's Distr
name
Search names and contents of open tabs for:
sql|
SQLQuery1.sql
USE [AdventureWorks] GO /***** Object: Table [dbo].[DatabaseLog]  Script Date:
SQLQuery2.sql
CREATE DATABASE bob
SQLQuery3.sql
USE [AdventureWorks] GO /***** Object: StoredProcedure [sys].[sp_addmergealter
etcode
ubid                uniqueidentifier
ubid                uniqueidentifier
lternate_subid      uniqueidentifier
escription          nvarchar(255)

```

You can use the keyboard shortcut **Ctrl+T** to quickly open the search box.

Re-open tabs on startup

This feature allows you to restore all the tabs that were open when you last closed SQL Server Management Studio. You can enable/disable this feature from the SQL Tab Magic toolbar:



Worked examples

The following examples show how SQL Prompt can help you to write queries more quickly. All these examples use the AdventureWorks database for SQL Server 2005.

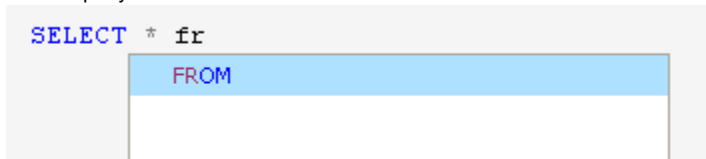
- Example 1: Writing a simple query
- Example 2: Using the Column Picker to select and insert columns
- Example 3: Using a snippet to insert a block of code
- Example 4: Writing a full INSERT statement
- Example 5: Executing functions and stored procedures
- Example 6: Writing a cross-database query

Example 1: Writing a simple query

1. In your query editor, start a new query on the AdventureWorks database.
2. Type

se (press Tab) * fr

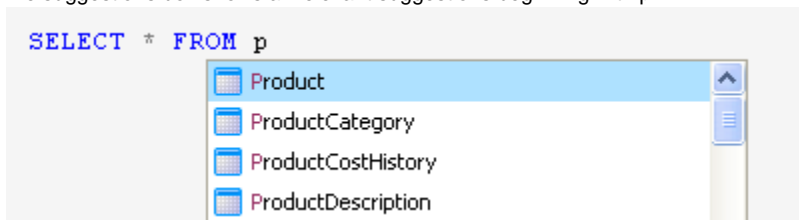
As you type, the suggestions box is displayed. Whenever you press the Tab key, SQL Prompt inserts the currently highlighted suggestion into the query:



Press Tab, then Space, then type:

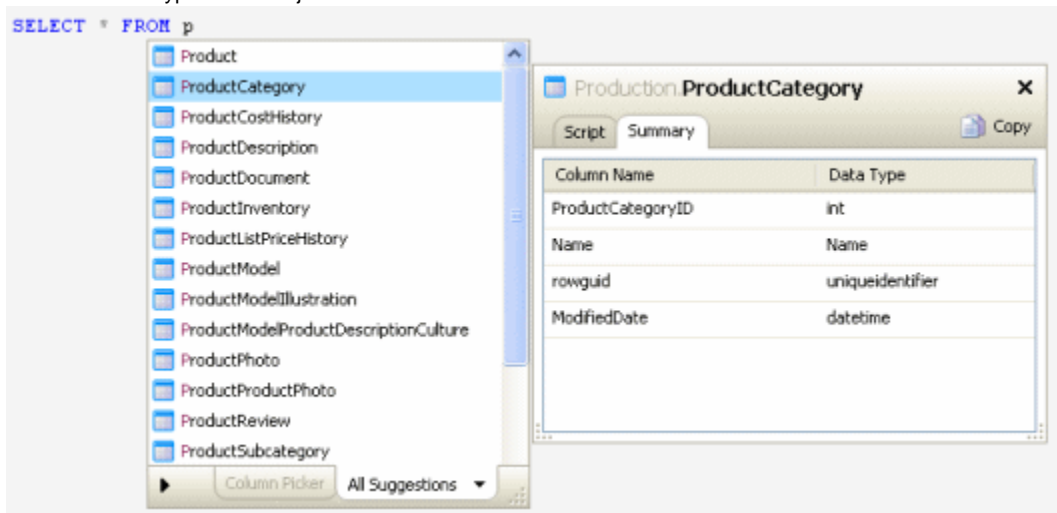
p

The suggestions box shows all relevant suggestions beginning with p:



SQL Prompt always places the most contextually relevant types of suggestions at the top of the list, depending on what you have typed in your query.

3. Press the Down arrow key once to highlight the ProductCategory table. You can check if this is the correct table by looking at the column names and data types in the object definition box:



4. Press Enter to insert the table and complete the query. Your query should now look like this:

```
SELECT * FROM Production.ProductCategory
```

In SQL Server 2005 and 2008, SQL Prompt automatically inserts the schema name for objects in non-default schemas.

Quick tips:

- Use the Up and Down arrow keys to move through the list of suggestions.
- Press Tab or Enter to insert the current suggestion. To set other insertion keys, for example Space or Dot (.), see [Managing SQL Prompt behavior](#).
- To view the object creation script for an object instead of a summary, click the **Script** tab in the object definition box. The Script tab will be the default view from this point onwards.
- To turn off the automatic display of the object definition box, see [Using the object definition box](#).
- Note that SQL Prompt automatically replaces lower case keywords with upper case as you type. You can change the case used for keywords. See [Options for formatting your code](#).
- You can also change how object names are qualified when inserted into your code. See [Inserting suggestions into your code](#).

Example 2: Using the Column Picker to select and insert columns

In this example, Space is used as an additional insertion key. To define Space as an insertion key:

1. On the **SQL Prompt** menu, select **Options**.
2. In the Options dialog box, go to the **Main > Behavior** page, then under **Insertion Keys**, select **Space bar**, then click **OK**.

From now on, each time you press Space, the currently highlighted suggestion will be inserted into your query.

1. Create a new query and type:
`se * fr` (include all the spaces)

SQL Prompt will insert the matching suggestion whenever you press Space, so your query should now look like this:

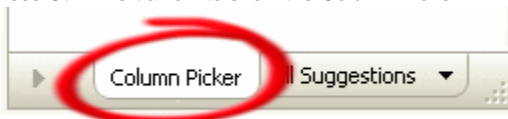
```
SELECT * FROM Sales.Customer
```

You can move the cursor to just after the * and press Tab to insert all columns:

```
SELECT * FROM Sales.Customer
```

Press TAB to expand wildcard

2. To insert specific columns from the CUSTOMER table, rather than all the columns, move the cursor to just after the * (asterisk) and click the Backspace button to delete *.
3. Press Ctrl + Space to display the suggestions box.
4. Press Ctrl + Left arrow to show the Column Picker:



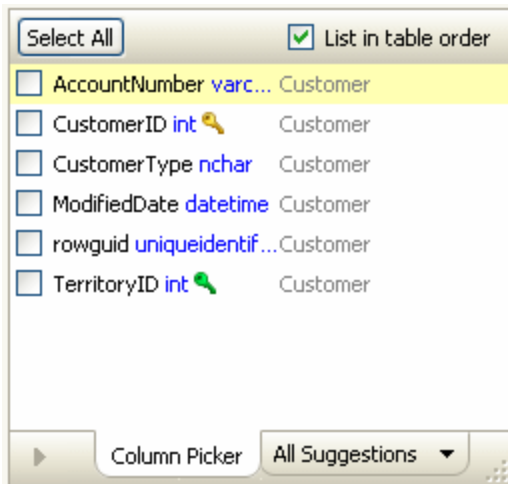
The column picker allows you to choose multiple columns from the table. Primary key



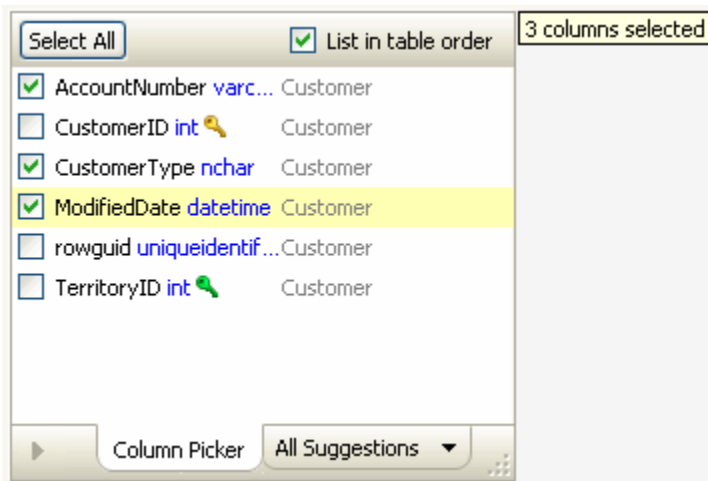
and foreign key



columns are indicated by key icons.



5. Use the Up and Down arrow keys to move through the list, then press Space to select a column for insertion. You can also use the mouse to select the check box for each column to be inserted:



6. Press Enter to insert all the columns into your query. Your query should now look like this:

```
SELECT  AccountNumber ,
        CustomerType ,
        ModifiedDate FROM Sales.Customer
```

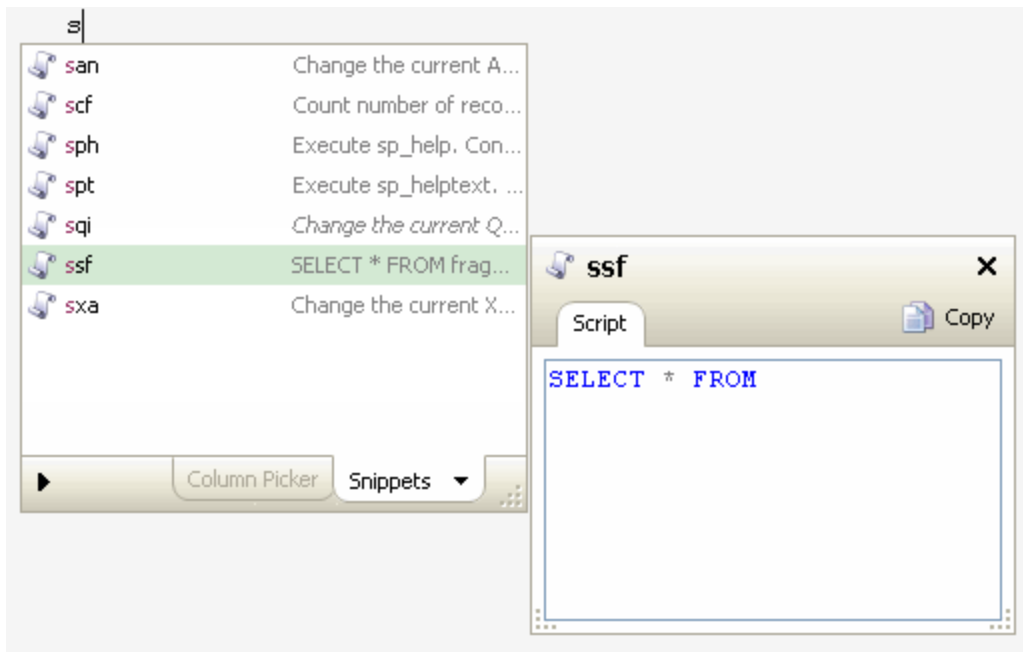
Columns are inserted in the order in which they were selected and automatically formatted based on the options set in the Format pages of the SQL Prompt Options dialog box.

Quick tips:

- By default, columns are listed in alphabetical order. To list them in the order they are defined in the table, select **List in table order**. When inserting columns from more than one table, **List in table order** groups columns by table rather than in a single alphabetical list.
- When the Column Picker is displayed, type the first few letters of a column name to filter the list. Any columns already selected are displayed at the bottom of the list and will still be inserted.
- Clicking **Select All** toggles between selecting all columns and selecting none.
- You can change how SQL Prompt lays out the inserted code (for example, to place each column on a new line). See [Formatting your code](#).
- The first time you display the column picker, a callout message box appears; click **X** to close it. Once you close it, the tooltip will not reappear.

Example 3: Using a snippet to insert a block of code

1. Create a new query and type s. The suggestions box shows all suggestions that start with s.
2. Press Ctrl + Down arrow to change the category of suggestions to **Snippets**.
3. Press the Down arrow key to move down the list of snippets to highlight **ssf**.



- Press any of the insertion keys (for example Space, Tab or Enter) to insert the snippet code. The position of the cursor after the snippet has been inserted is specified as part of the snippet definition.



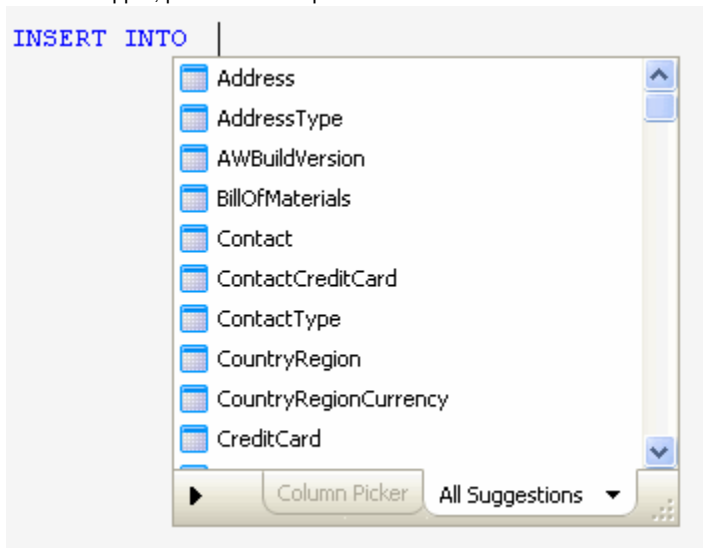
Quick tips:

- Use Ctrl + Down to change the category of suggestion (for example, to see only views or stored procedures in the suggestions box).
- To make a snippet from a block of code in your editor, highlight it, then right-click and select **Make Snippet**.
- SQL Prompt is pre-configured with a large number of default snippets. To see a list of all snippets, and to edit, delete or create new snippets, on the **SQL Prompt** menu, select **Snippet Manager**.

For more information on snippets, see [Managing snippets](#).

Example 4: Writing a full INSERT statement

- Create a new query and type **ii**.
- This is a snippet; press Tab to expand it to an INSERT INTO statement and press the Space to display the suggestions box again:



- Type **cus** to filter the list to the two customer tables, then press Enter to insert the first (highlighted) table: **Customer**. SQL Prompt completes the syntax of the INSERT statement, and inserts the name, data type, and default value for each column as comments:

```

INSERT INTO Sales.Customer
( TerritoryID ,
  AccountNumber ,
  CustomerType ,
  rowguid ,
  ModifiedDate
)
VALUES ( 0 , -- TerritoryID - int
  '' , -- AccountNumber - varchar(10)
  '' , -- CustomerType - nchar(2)
  NULL , -- rowguid - uniqueidentifier
  '2009-05-29 10:21:37' -- ModifiedDate - datetime
)


```

The insertion point is positioned for you to insert the values.

Quick tips:

- By default, column names, data types and default values are automatically inserted when you write an INSERT statement. You can turn off some or all of these defaults using the SQL Prompt options dialog. See [Customizing inserted code](#).
- To see a list of columns in a table, and their data type, move the mouse pointer over a table name to display the tooltip, then click the tooltip:

Sales.Customer

 [Sales.Customer \(Table\)](#)
Current customer information. Also see the Individual and Store tables.

The object definition box is displayed, showing the summary for the object. If the Script tab is displayed instead, click Summary:

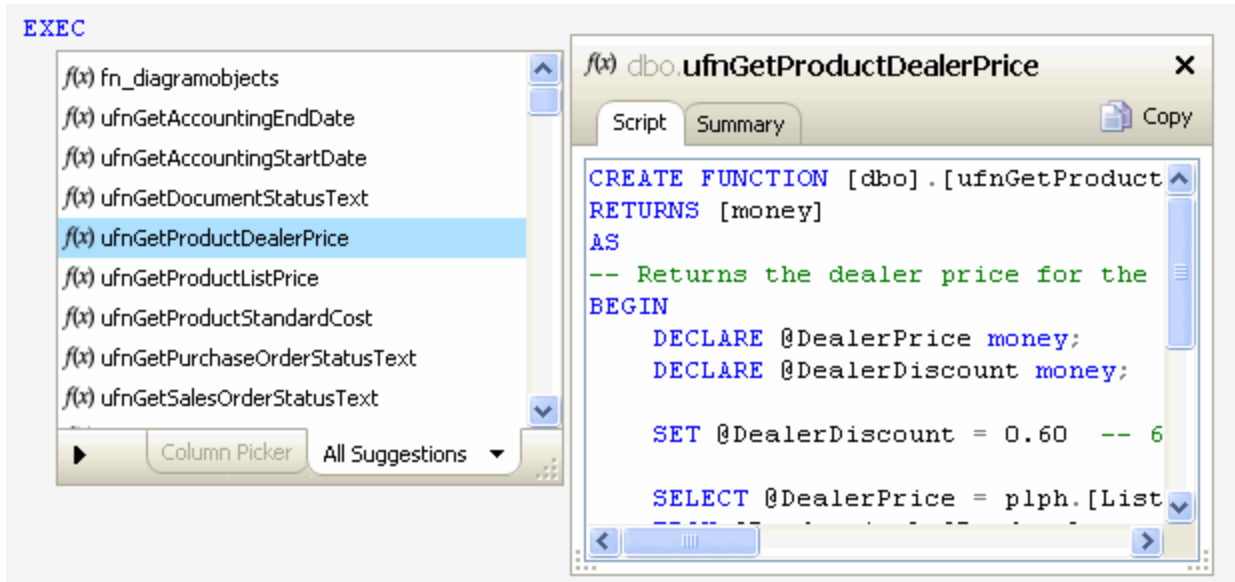
Sales.Customer

Sales.Customer	
<div> <div>Script</div> <div>Summary</div> <div>Copy</div> </div>	
Column Name	Data Type
AccountNumber	varchar(10)
CustomerID	int
CustomerType	nchar(2)
ModifiedDate	datetime
rowguid	uniqueidentifier
TerritoryID	int

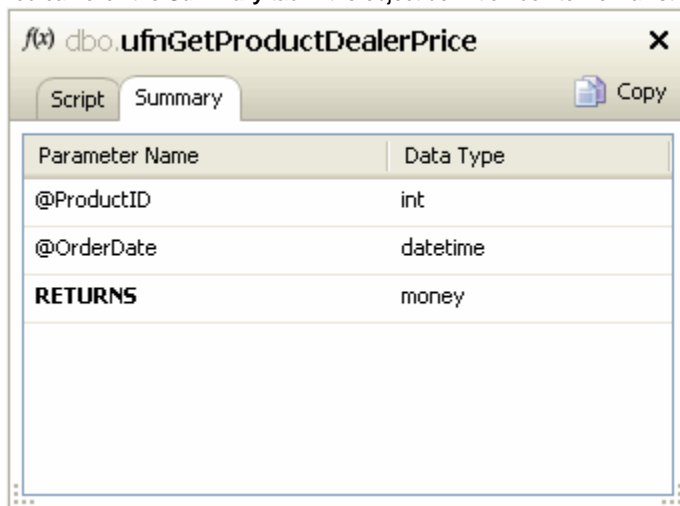
Example 5: Executing functions and stored procedures

1. Create a new query and type **EXEC**
2. Press Space.

The suggestions box is displayed, with all the functions and stored procedures at the top of the list. Select a function in the list, then click the Script tab to see a preview of the object creation SQL script in the object definition box for the selected object.



You can click the **Summary** tab in the object definition box to view a list of parameters for that function:

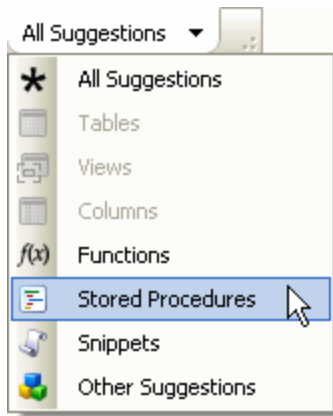


3. Press Enter to insert the function into your code:

```
EXEC dbo.ufnGetProductDealerPrice @ProductID = 0, -- int
    @OrderDate = '2009-05-29 10:29:45' -- datetime
```

Quick tips:

- By default, parameters are automatically inserted into functions and stored procedures, and the data type and default value is displayed for each parameter. You can change this behavior in the SQL Prompt Options dialog box. See [Customizing inserted code](#) for more details.
- To see only stored procedures in the suggestions box, click the **All Suggestions** tab at the bottom of the suggestions box, then click **Stored Procedures**:

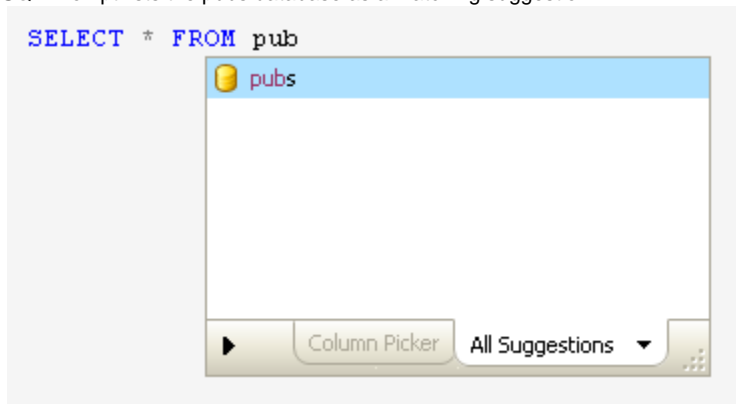


- To view system functions and stored procedures in the suggestions box, select **List system objects** on the **Suggestions > Types of suggestion** page in the SQL Prompt Options dialog box.

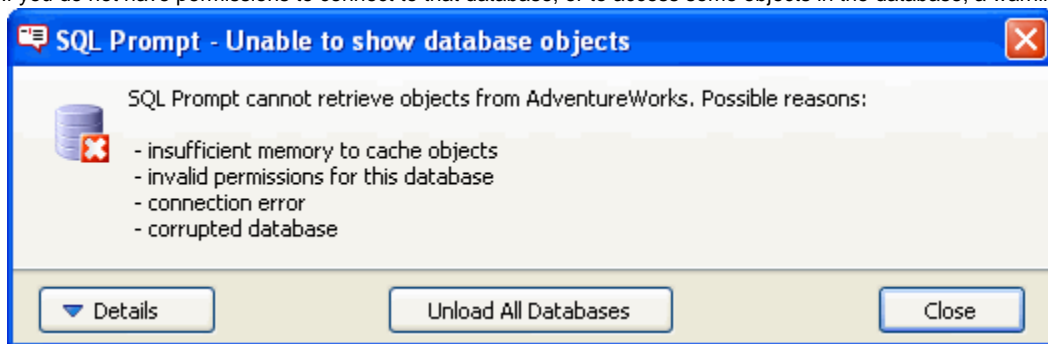
Example 6: Writing a cross-database query

Cross-database queries are automatically supported in SQL Prompt. You do not need to select an option to enable them.

1. Create a new query on the AdventureWorks database.
2. Type `SELECT * FROM pub`
3. SQL Prompt lists the pubs database as a matching suggestion:



4. Press Tab to insert the database name into your query, then type . (Dot). As soon as you press Dot, SQL Prompt reads all the database objects from the pubs database. For large databases, this may take a few minutes.
5. If you do not have permissions to connect to that database, or to access some objects in the database, a warning dialog is displayed:



This warning may also be displayed if SQL Prompt has run out of memory to store the objects from this database. See [Managing connections and memory](#).

Troubleshooting

If you are encountering an issue with SQL Prompt, the following pages may help:

- [Accessing the SQL Prompt log files](#)
- [Configuring permissions for SQL Prompt](#)

Error messages

- [Error 80070002 - The Add-in SQL Prompt failed to load or caused an exception](#)
- [Using Layout SQL crashes SQL Server Management Studio](#)

Common issues

- [SQL Prompt is not showing all linked servers](#)
- [Removing SQL Prompt menu item after uninstalling](#)
- [SQL Server Management Studio runs slowly after uninstalling SQL Prompt](#)

Previous versions

If you have an older version of SQL Prompt (prior to 5.x) these pages may help:

- [Unable to cast object of type 'System.DBNull' to type 'System.String'](#)
- [SQL Prompt is showing an old version of my stored procedure](#)
- [Add-in failing to load into SQL Server Management Studio on Vista](#)
- [Candidate suggestions being displayed slowly](#)
- [Suggestions to improve performance for pre-formatted SQL](#)

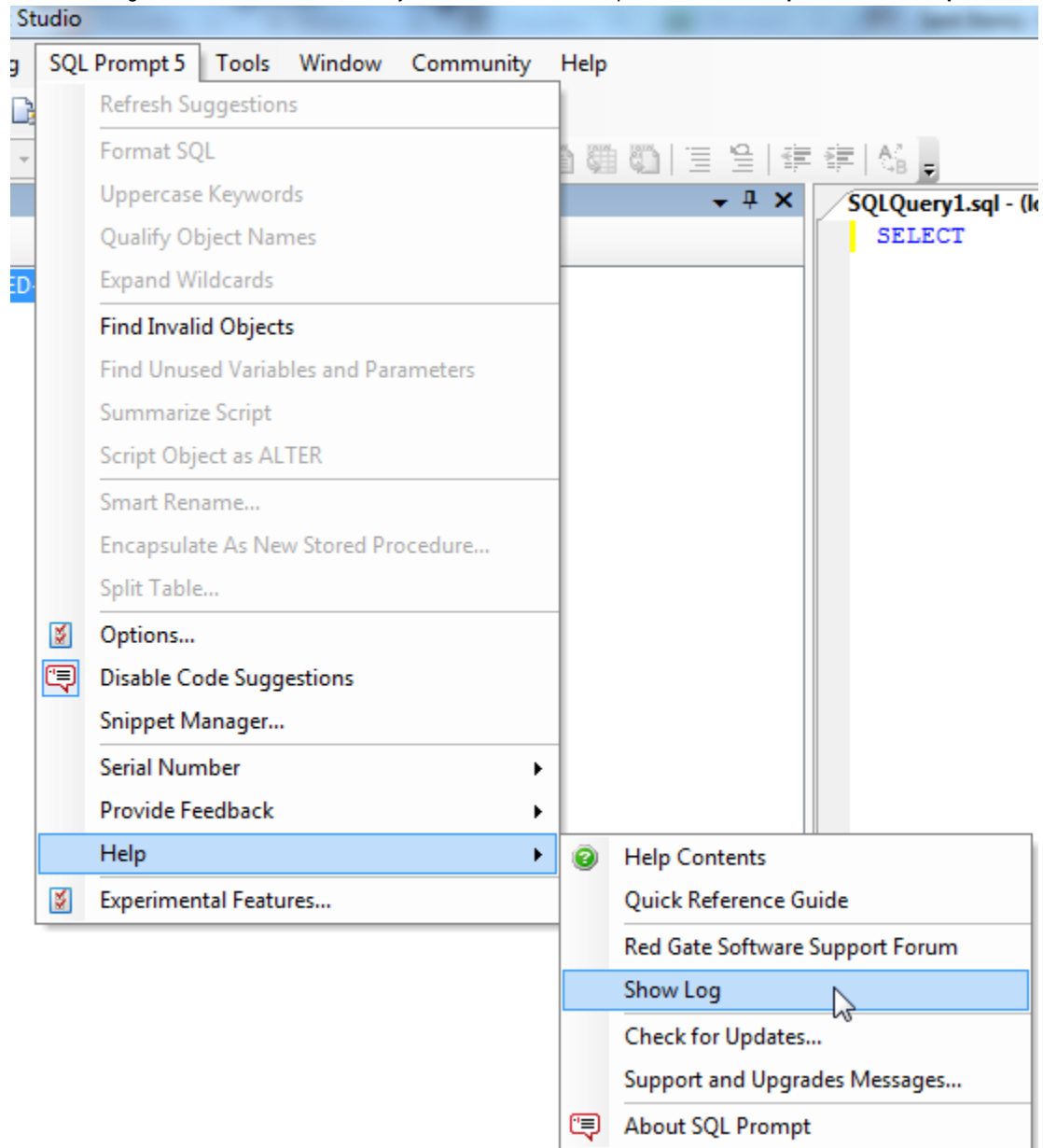
Further help

If you are experiencing an issue with SQL Prompt which is not listed above, try [the forums](#) or [contact our support team](#).

Accessing the SQL Prompt log files

Log files collect information about the application while you are using it. These files are useful to us if you have encountered a problem.

To view the log file for the current session in your default text editor, open the **SQL Prompt** menu, click **Help** and then click **Show Log**.



By default the log files are stored in:

- %LOCALAPPDATA%\Red Gate\SQL Prompt 5 (Windows Vista, Windows 2008 and later)
- %UserProfile%\Local Settings\Application Data\Red Gate\SQL Prompt 5 (Windows XP and Windows 2003)

Configuring permissions for SQL Prompt

The following account permissions are required by SQL Prompt:

- For SQL Server 7 or SQL Server 2000, SQL Prompt requires administrative or dbo or Sys Admin permissions.
- For SQL Server 2005 you can use the same permissions as for SQL Server 7/2000 or alternatively you can use the GRANT VIEW DEFINITION TO [{username}] which provides SQL Prompt with permission to retrieve the metadata without giving dbo access.
- For SQL Server 2008 and SQL Server 2012, it may be necessary to grant VIEW SERVER STATE to access information about encryption keys otherwise a 'User does not have permission to perform this action' message may prevent the display of candidates.

Error messages

- Error 80070002 - The Add-in SQL Prompt failed to load or caused an exception
- Using Layout SQL crashes SQL Server Management Studio

Error 80070002 - The Add-in SQL Prompt failed to load or caused an exception

When launching SQL Server Management Studio after installing SQL Prompt, the following error message may appear:

The Add-in 'SQL Prompt <version number>' failed to load or caused an exception. Would you like to remove this Add-in?

If you choose yes, you will need to reinstall the Add-in to use it again.

Error Message: The system cannot find the file specified.

Error number: 80070002

This error arises if *extensibility.dll* is missing.

To resolve this problem:

1. Open a command prompt and change directory to the SQL Prompt installation folder.
2. Run the following:
For SQL Prompt 3.x: `%SYSTEMROOT%\Microsoft.net\Framework\v2.0.50727\regasm "%ProgramFiles%\Red Gate\SQL Prompt 3\RedGate.SQLPrompt.SSMSUI.dll"`
For SQL Prompt 4.x: `%SYSTEMROOT%\Microsoft.net\Framework\v2.0.50727\regasm "%ProgramFiles%\Red Gate\SQL Prompt 4\RedGate.SQLPrompt.SSMSUI.dll"`
For SQL Prompt 5.x: `%SYSTEMROOT%\Microsoft.net\Framework\v2.0.50727\regasm "%ProgramFiles%\Red Gate\SQL Prompt 5\RedGate.SQLPrompt.SSMSUI.dll"`
This will register SQL Prompt as a Management Studio add-in.
3. If the command is successful, run `sqlwb.exe /setup` (SQL Server Management Studio 2005) `SSMS.exe /setup` (SQL Server Management Studio 2008, 2008 R2 and 2012) to re-initialize SQL Server Management Studio's interface.

If the problem persists, it may be necessary to reinstall SQL Prompt:

1. Open Add/Remove Programs or Programs and Features from the Windows Control Panel.
2. Select SQL Prompt then click **Repair**.

Using Layout SQL crashes SQL Server Management Studio

Using Layout SQL causes SQL Server Management Studio to hang and have to use Windows Task Manager to kill the process.

Error text

```
System.ApplicationException: Exception within progress task --->
System.InvalidCastException: Unable to cast COM object of type 'System.__ComObject' to
interface type 'Microsoft.VisualStudio.TextManager.Interop.IVsTextView'. This
operation failed because the QueryInterface call on the COM component for the
interface with IID '{BB23A14B-7C61-469A-9890-A95648CED5E6}' failed due to the
following error: No such interface supported (Exception from HRESULT: 0x80004002
(E_NOINTERFACE)).
    at Microsoft.VisualStudio.TextManager.Interop.IVsTextView.GetSelection(Int32&
piAnchorLine, Int32& piAnchorCol, Int32& piEndLine, Int32& piEndCol)
    at RedGate.SQLPrompt.CommonVS.Editor.VSEditorWindow.get_Selection()
    at RedGate.SQLPrompt.CommonUI.Refactor.Refactor.a(EditorWindowBase , b , c& )
    at RedGate.SQLPrompt.CommonUI.Refactor.Refactor.a(EditorWindowBase , c& , b[] )
    at RedGate.SQLPrompt.CommonUI.Refactor.Refactor.a.Run()
    at RedGate.SQLPrompt.CommonUI.Progress.TaskAggregator.Run()
    at RedGate.SQLPrompt.CommonUI.Forms.ProgressDialogue.a()
    --- End of inner exception stack trace ---
    at RedGate.SQLPrompt.CommonUI.Forms.ProgressDialogue.ShowProgress(ITaskRunner task)
    at RedGate.SQLPrompt.CommonUI.Refactor.Refactor.a(a )
    at RedGate.SQLPrompt.CommonUI.Refactor.Refactor.LayOutSql(EditorWindowBase editor,
SqlPromptOptions options)
    at b.Execute()
    at RedGate.SQLPrompt.CommonVS.Commands.VSCommandController.Exec(String CmdName,
vsCommandExecOption ExecuteOption, Object& VariantIn, Object& VariantOut, Boolean&
Handled)
```

Cause

Sometimes without apparent cause SQL Server Management Studio installations can delete the interface that Prompt and Refactor need to get access to the query window.

How to fix

Run the following command from a Command Prompt:

```
regsvr32 "%commonprogramfiles%\microsoft shared\msenv\textmgrp.dll"
```

If you continue to experience problems, please contact [Red Gate Support](#).

Common issues

- SQL Prompt is not showing all linked servers
- Removing SQL Prompt menu item after uninstalling
- SQL Server Management Studio runs slowly after uninstalling SQL Prompt

SQL Prompt is not showing all linked servers

SQL Prompt may not list all linked servers in the suggestions box.

- This problem may arise if the SQL Prompt user does not have the VIEW ANY DEFINITION permission, which allows the user to access the list of linked servers. Use `GRANT VIEW ANY DEFINITION TO <user>` to allow the user to view all metadata in the SQL Server instance. To check if the user has the correct permissions, run the following query:

```
SELECT srvname, srvproduct, providername FROM master..sys.servers
```

If the query fails, the user does not have the correct permissions.

- This problem may also arise if the linked server is not a Microsoft SQL Server. SQL Prompt cannot access OLE DB data sources other than SQL Server instances.

Removing SQL Prompt menu item after uninstalling

After uninstalling SQL Prompt, the SQL Prompt menu item may still be included in the SQL Server Management Studio and/or Visual Studio toolbar. If this happens, you will need to remove the menu item manually.

To remove the menu item from SQL Server Management Studio 2005 or 2008 or from Visual Studio 2008:

1. From the **Tools** menu, select **Customize**.
2. Select the **Commands** tab and click **Rearrange Commands**.
3. In the Rearrange Commands dialog, select **Toolbar**.
4. From the **Controls** list, select **SQL Prompt**, then click **Delete**.
5. Close the Rearrange Commands and Customize dialog boxes. The SQL Prompt menu item is no longer displayed on the toolbar.

To remove the menu item from Visual Studio 2010 and later:

1. From the **Tools** menu, select **Customize**.
2. Open the **Commands** tab.
3. From the **Controls** list, select **SQL Prompt**, then click **Delete**.
4. Click **Close** to close the dialog. The SQL Prompt menu item is no longer displayed on the toolbar.

SQL Server Management Studio runs slowly after uninstalling SQL Prompt

After uninstalling SQL Prompt, SQL Server Management Studio can run more slowly than before. This may occur if SQL Server Management Studio attempts to load the add-in after it has been uninstalled.

To resolve this problem:

1. Close SQL Server Management Studio.
2. Open Registry Editor (regedit.exe).
3. Navigate to the following registry key (where *version* is the SQL Server version): *HKEY_CURRENT_USER\Software\Microsoft\Microsoft SQL Server\<version>\Tools\Shell\PreloadAddinState*
4. Delete the *RedGate.SQLPrompt.SSMSUI* registry value.
5. Close Registry Editor.
6. Open a Command Prompt window and run `sqlwb.exe /setup` (SQL Server Management Studio 2005) or `ssms.exe /setup` (SQL Server Management Studio 2008, 2008 R2 and 2012) to re-initialize SQL Server Management Studio's interface.

Older versions of SQL Prompt

The following pages apply to non-current versions of SQL Prompt:

- [Unable to cast object of type 'System.DBNull' to type 'System.String'](#)
- [SQL Prompt is showing an old version of my stored procedure](#)
- [Add-in failing to load into SQL Server Management Studio on Vista](#)
- [Candidate suggestions being displayed slowly](#)
- [Suggestions to improve performance for pre-formatted SQL](#)

Unable to cast object of type 'System.DBNull' to type 'System.String'

This page applies to SQL Prompt versions 3.x

When caching a database in SQL Prompt, you may be presented with one of the following errors -

```
"Unable to cast object of type System.dbnull to type System.String."
```

```
"An unexpected null value has been encountered while querying the database system tables. Please make sure that the database is in a consistent state by running DBCC CheckDB."
```

Possible causes

Often these errors are reported because SQL Prompt is unable to retrieve certain metadata from the server using the account that SQL Prompt is logged in as.

How to fix

You will need to first check the account permissions and make sure that the login account meets the minimum requirements. Here is an article outlining the account permissions needed:

[Configuring permissions for SQL Prompt](#)

If the account is not the problem, then there may be an issue with the underlying schema. It is possible that there is some corruption or misconfiguration of an object or meta-data that SQL Prompt doesn't know how to handle. For example, we've seen this happen before when certain objects have been found to be without an owner. You may be able to find out more information if you use SQL Compare and compare the problem database with a non problematic database (SQL Prompt uses the SQL Compare engine to register the database). Hopefully SQL Compare should give you a better error message or point you towards the specific object that is causing the problem.

If SQL Compare doesn't present you with any extra information regarding the cause of the error, then you can do some further troubleshooting.

1. Start a default SQL Server Profiler Trace ([http://msdn.microsoft.com/en-us/library/ms187929\(SQL.90\).aspx](http://msdn.microsoft.com/en-us/library/ms187929(SQL.90).aspx)) on the server that is hosting the problem database.
2. Re-run the SQL Compare comparison to reproduce the error.
3. Stop the SQL Server Profiler trace and look through the results to find the last query run by the application "Red Gate Software - SQL Tools".
4. Copy this query into a new Management Studio query window opened against the original database, and execute it.
5. Have a look through the returned results for NULLs that exist in fields that shouldn't allow them. This should, hopefully, allow you to track down the object that is causing the issue. T
6. he results can often be hard to analyze and knowing where NULLs are allowed and where they're not sometimes comes down to experience. If you do require assistance analyzing the data then if you could send the query that you ran and your results set to support@red-gate.com, we'll look into it for you.

SQL Prompt is showing an old version of my stored procedure

This page applies to SQL Prompt versions 3.x

When viewing the creation SQL for an object using the candidate list or any of the features that insert SQL e.g. the insertion of the object definition after an ALTER statement. SQL Prompt will use the version of the SQL code in its cache file and not retrieve that latest version from the database.

This means that you will see the SQL code from the last time the cache was updated and not what is currently on the server. If you are using the auto insertion of object definition after the ALTER statement and you haven't updated your cache after somebody has changed a stored procedure, it is possible to accidentally overwrite the new version of the stored procedure with the old version stored in the cache file.

Cause

SQL Prompt is using the version of SQL code in its cache file.

How to fix

Update the cache if you know that someone might have modified the procedure. You may update the cache using the cache management option from the SQL Prompt menu.

If you are editing SQL on a database when you know the schema has been changed then make sure you refresh the cache before using the auto insertion of SQL features.

If you continue to experience problems, please contact [Red Gate Support](#).

Add-in failing to load into SQL Server Management Studio on Vista

This page applies to Windows Vista with Service Pack 1

After installing SQL Prompt or SQL Refactor and launching SQL Server Management Studio, the following error may appear:

Add-in failed to load: Class not registered

Possible cause

The service pack update to Windows Vista has been known to change permissions to the registry in a way that prevents user access to the information about installed classes: HKEY_CLASSES_ROOT. This prevents SQL Server Management Studio from loading the add-ins into the environment.

How to fix

To correct this problem, it's necessary to assign permissions to the registry for the logged-in user.

1. Open the Registry Editor (regedit.exe) using the "Run As Administrator" option.
2. Right-click HKEY_CLASSES_ROOT, select "Permissions", and give the logged-in user full control.
3. The following permissions needed to be added to HKEY_CLASSES_ROOT as well:

Users: read

System: full control

CREATOR OWNER: full control on subkeys only

Candidate suggestions being displayed slowly

This page applies to SQL Prompt versions 4.x and below

In some circumstances, the candidate list produced by SQL Prompt may be slow to load, and you may notice that the performance of SQL Server Management Studio deteriorates the longer you continue editing queries in the query windows.

Possible causes

A possible cause of the problem is that the video drivers are not functioning properly

How to fix

To troubleshoot, access the display properties of the Windows desktop. If an "acceleration" setting is available, try lowering it or turn it off altogether. If performance improves, consider leaving your display at this setting or update your graphics card drivers.

If you continue to experience problems, please contact [Red Gate Support](#).

Suggestions to improve performance for pre-formatted SQL

This page applies to SQL Prompt versions 3.x

You may experience performance issues when pasting SQL from notepad or other text editor into SQL Server Management Studio.

Cause

SQL Prompt is optimized for use with large scripts when you first install it. However, if you experience slow performance, for example because you have a slow processor, you can reduce the number of lines of SQL code that SQL Prompt parses when it populates the candidate list.

How to fix

To reduce the number of lines of SQL code that SQL Prompt parses when it populates the candidate list:

1. On the **Options** dialog box, click **Listed Candidates > Performance**
2. Ensure that **Search a fixed number of lines from the caret** is selected, and then reduce the number of lines to search for variables and parameters.

Decreasing the number of lines speeds up performance. However, if you reduce this value too much you may experience problems

When you open a SQL file or paste text into your query editor, SQL Prompt will automatically parse the new code for any scripted objects and assigned aliases. For large files and scripts, this may take a few seconds. If you do not want SQL Prompt to scan pasted text or opened SQL files, in the **Listed Candidates > Performance** page, clear the **Search for objects and aliases when opening or pasting text** check box option.

SQL Prompt stops searching an opened file or a pasted block of text after 5 seconds. For very large scripts, this may not be long enough to parse all the scripted objects and aliases. You can increase the length of time SQL Prompt scans the text; to do this, increase the value of the **MaximumScanTimeMilliseconds** property in the **EngineOptions.xml** file from its default value of 5000 milliseconds.

The EngineOptions.xml file is stored by default in your local user profile's application settings, for example `C:\Documents and Settings\<username>\Local Settings\Application Data\Red Gate\SQL Prompt 3`. To specify that SQL Prompt should always scan the whole file or block of text with no time limit, set the value to 0

If you continue to experience problems, please contact [Red Gate Support](#).

Release notes and other versions

Version 7.0 (latest)	August 18th, 2015	Release notes	Documentation
Version 6.5	March 26th, 2015	Release notes	Documentation
Version 6.4	September 16th, 2014	Release notes	
Version 6.3	April 8th, 2014	Release notes	
Version 6.2	December 18th, 2013	Release notes	
Version 6.1	September 23rd, 2013	Release notes	
Version 6.0	September 4th, 2013	Release notes	
Version 5.3	March 22nd, 2012	Release notes	Documentation
Version 5.2	November 3rd, 2011	Release notes	
Version 5.1	April 12th, 2011	Release notes	
Version 5.0	December 15th, 2010	Release notes	
Version 4.0	August 10th, 2009	Release notes	Documentation
Version 3.9	July 14th, 2008	Release notes	Documentation (PDF)
Version 3.8	April 23rd, 2008	Release notes	<i>See version 3.9 documentation</i>
Version 3.6	September 17th, 2007	Release notes	
Version 3.5	July 26th, 2007	Release notes	
Version 3.1	March 22nd, 2007	Release notes	
Version 3.0	January 22nd, 2007	Release notes	

If you need to install an old version of SQL Prompt, go to [Download old versions of products](#).

SQL Prompt 5.3 release notes

The following release notes are available for SQL Prompt 5.3:

- [5.3.8 release notes](#)
- [5.3.6 release notes](#)
- [5.3.2 release notes](#)
- [5.3.0 release notes](#)

Version 5.3.8 - May 13th, 2013

Bug fixes

- LingBridge.dll is now installed with SQL Prompt on systems with .NET 4. (SP-4537, SP-4548, SP-4554, SP-4655)
- Using Format SQL on code which includes ? as a parameter value no longer causes SQL Server Management Studio to crash.
- Improved resilience and reporting in the face of unexpected errors.
- Link to SQL Prompt Quick Reference Guide works for systems with .NET 4.5.

Version 5.3.6 - October 15th, 2012

Visual Studio 2012 support

- SQL Prompt now installs into Visual Studio 2012.

Bug fixes

- Restore the default SSMS and VS behavior whereby a middle-click on a tab closes it. (SP-4236 and SP-4501)
- Fixed bug whereby Find Invalid Objects fails completely on certain databases, incorrectly reporting a login error. (SP-4454)
- No longer throws error when formatting scripts containing SQL 2012 keywords.

Version 5.3.2 - May 9th, 2012

New features

- New SQL Prompt Quick Reference Guide, with handy tips and keyboard shortcuts, now installed into the Start Menu.

Bug fixes

- Fix error in Snippet Editor that removed newlines from the code in SQL Prompt 5.3.0. (SP-4358)
- Fix crash when disabling Connection Coloring experimental feature.

Version 5.3.0 - March 22nd, 2012

New features

- Partial match suggestions are now enabled for all users.
- New SQL Prompt Labs feature to refresh suggestions automatically when SQL Prompt detects a change to a connected database.
- Also refreshes SSMS's own IntelliSense local cache (equivalent to Ctrl+Shift+R).
- Basic support for most synonym types, so SQL Prompt notices when the base object is a table, stored procedure, view, function, etc.
- CREATE SYNONYM script is now shown in the object definition box for synonyms.

Bug fixes

- Fixed problem caching when base object for synonym is inaccessible (e.g. due to offline database or inadequate permissions).
- Fix crash related to scripts containing column list after table alias. (SP-4070)

Other

- Addition of MaxSupportedColumns parameter in the engine options, which will prevent SQL Prompt from trying to cache databases with a total number of columns exceeding this number. Default is -1, which means it's turned off. Related to SP-1207.

SQL Prompt 5.2 release notes

November 3rd, 2011

Improvements

- All SQL Prompt features are now enabled in Denali CTP3 SQL Server Management Studio.
- SQL Prompt now works with Denali servers, with improved support for new keywords and built-in functions.
- More of SQL Prompt's script refactoring features are now available inside Visual Studio for the first time.
- You can now turn code auto-completion on/off using a new SQL Prompt menu option, or a keyboard shortcut (Ctrl + Shift + P).
- You can now press Shift + Tab when editing the code for a snippet to get back to the Description field.

Fixes

A number of issues have been fixed, including:

- A SQL Azure connection issue has been fixed.
- Expand Wildcards now works for system tables.
- Uppercase Keywords could sometimes change white space inadvertently. It now doesn't.

Some resource leaks have been fixed, which should prevent some crashes in extreme cases.

SQL Prompt 5.1 release notes

April 12th, 2011

Find Invalid Objects improvements

SQL Prompt 5.1 includes many usability improvements to the Find Invalid Objects feature:

- you can do more with the list of invalid objects, including locating objects in the Object Explorer, copying the list of invalid objects, and scripting multiple invalid objects as ALTER
For full details, see: [Finding invalid objects: Tips](#)
- the number of invalid objects that have been found is displayed
- a progress bar is displayed when finding invalid objects
- you can change the server or database you want to find invalid objects in
- Timeout errors are also much reduced or eliminated.

SQL Refactor favorites

The following has been included from SQL Refactor:

- Uppercase Keywords, as a separate feature from Format SQL
For more information, see: [Uppercasing keywords](#)
- Ctrl+B Ctrl+L as a keyboard shortcut for Format SQL in SQL Server Management Studio (Ctrl+K Ctrl+Y is the primary shortcut)

Fixes

A number of bugs have been fixed, including:

- SQL Prompt 5.1 now works on SQL Server 2005 Express
- Fixed extreme slowdown with Common Table Expressions (CTEs)
- Prevented misleading *Thread was being aborted* message from being added to log
- Fixed a number of the most common crashes that cause a *Notify Red Gate of this error* dialog

SQL Prompt 5.0 release notes

December 15th, 2010

Find Invalid Objects

SQL Prompt 5 enables you to easily find invalid objects in your database. This can be useful, for example, if you have inherited a legacy database and you want to know if objects are referencing other objects that no longer exist in the database.

Script Object as ALTER from a query window

If you are navigating a query that contains a reference to an object, SQL Prompt 5 enables you to script out that object directly from the query window by pressing **F12**.

View column dependencies

SQL Prompt 5 enables you to view a list of objects that reference a column.

SQL Refactor functionality

SQL Refactor has been retired with the release of SQL Prompt 5. Much of the functionality of SQL Refactor is now available in SQL Prompt Pro Edition:

- Smart Rename
- Qualify Object Names
- Expand Wildcards
- Split Table
- Encapsulate as New Stored Procedure
- Find Unused Variables and Parameters
- Summarize Script

SQL Prompt 4.0 release notes

August 10th, 2009

- **Enhanced user interface**

The user interface for intellisense suggestions and object definitions has been redesigned to be less intrusive and to make SQL Prompt's features more easily discoverable.

The Options dialog box has also been completely reinvented, with better navigation and a rationalised set of options.

- **Improved SQL Grammar support**

Enhanced T-SQL grammar support for SQL 2000, SQL 2005 and SQL 2008 SQL server editions. Suggestions are more accurate and reliable, and cover a more extensive scope of T-SQL.

- **Performance improvements**

SQL Prompt 4 caches database objects dramatically faster than SQL Prompt 3, and does so much more quickly across high latency networks.

- **Snippet enhancements**

Snippets are now stored as separate files, and so can be easily shared among other SQL Prompt users.

There is a new, separate interface for creating and managing your snippets, including an option to go to the folder where snippets are stored.

- **Column picker**

The column picker has been radically improved, so that it is simpler to find and use, for example it is now clearly labelled, and shows the number of columns currently selected.

- **Options**

Many SQL Prompt 3 options have been removed and replaced with sensible defaults, to simplify the configuration and make more intuitive behavior. The Options dialog box itself has been drastically re-engineered so that it is much quicker to find the option you want, understand its purpose, and preview its effect.

- **Right-click context menus**

Right-click context menus have been implemented inside your query editor for creating SQL Prompt snippets and applying your preferred formatting to code.

- **Temporary storage only of database objects on disk**

Database objects are only cached during your current session. When closing down your query editor, the SQL Prompt object cache is emptied, to save disk space. Offline mode working is therefore no longer supported.

- **Format styles (Pro feature)**

Styles allow you to save and share your format options with other SQL Prompt users.

- **Decrypts encrypted objects: (Pro feature)**

SQL Prompt 4 can now decrypt encrypted objects from SQL 2000, SQL 2005 and SQL 2008 to allow you to view their creation script.

- **Importing snippets and aliases**

As a major new installation, existing user-created snippets and aliases or your old configuration options will not automatically be imported into v4. Red Gate offers two free tools that help migrate snippets and aliases from v3.9. For more information, please see the following forum posts:

Snippet migrator: <http://www.red-gate.com/MessageBoard/viewtopic.php?t=9361>

Alias importer: <http://www.red-gate.com/MessageBoard/viewtopic.php?t=9362>

Installing SQL Prompt 4 will disable SQL Prompt 3 and start a new trial period. If you wish to re-enable SQL Prompt 3 you will need to uninstall SQL Prompt 4. If you have user-defined aliases or snippets in SQL Prompt 3 which you wish to retain, we have a migration tool available.

SQL Prompt 3.9 release notes

July 14th, 2008

- SQL 2008 support including data types and new keywords
- Layout support for SQL Server 2008 syntax
- Fixed a problem with Visual Studio integration
- Fixed a possible exception with layout

SQL Prompt 3.8 release notes

April 23rd, 2008

- Pro and Std Editions
- Object definition hints
- Function parameter hints
- Layout (Pro Edition)
- Summarized object view in Schema Panel
- Improved logging support
- VS.NET 2008 support

SQL Prompt 3.6 release notes

September 17th, 2007

- Data Dude support
- Improved VS 2005 support
- CTEs
- Scripted tables, views, stored procs and functions
- GRANT, REVOKE, DENY permissions
- IF, WHILE and other control flow statements
- Transaction control statements (BEGIN, COMMIT, ROLLBACK)
- ENABLE/DISABLE TRIGGER

SQL Prompt 3.5 release notes

July 26th, 2007

- Visual Studio 2005 database project support
- Cross-database queries
- Distributed queries across linked SQL Server instances
- Correlated subqueries
- 3 – 5 times faster metadata retrieval
- Works well with large SQL scripts (>1000 lines)
- 40 – 95% decrease in memory footprint
- Bug fixes and other enhancements

SQL Prompt 3.1 release notes

March 22nd, 2007

- Support for SQL Server Management Studio Express
- Improved auto closing and auto casing
- Now prefers exact matches for the current filter string
- Bug fixes to improve stability of the product

SQL Prompt 3.0 release notes

January 22nd, 2007

- Direct integration into SQL Server Management Studio and Query Analyzer
- Full SELECT support: aliases, derived tables, subqueries, joins
- INSERT, UPDATE, DELETE support
- Variables
- Column picker
- Wildcard auto-expansion
- Advanced completion list
- User object schema definitions