

1. SQL Data Generator 2 documentation	2
1.1 Requirements	3
1.2 Installing	4
1.3 Licensing	5
1.3.1 Activating	6
1.3.2 Deactivating	12
1.3.3 Troubleshooting licensing and activation	15
1.4 Upgrading	18
1.4.1 Using Check for Updates	19
1.4.2 Troubleshooting Check for Updates errors	21
1.5 Technical notes	22
1.6 Worked example - setting up the data generator	23
1.7 Using generators	30
1.7.1 Generic generators	33
1.7.2 Customizing existing generators	36
1.7.3 Creating new generators	38
1.8 Using existing sources	41
1.8.1 Mapping SQL tables or views	42
1.8.2 Mapping CSV files	43
1.8.3 Cleansing or removing sensitive data from an existing data source	45
1.9 Using the command line	47
1.9.1 Exit codes used in the command line	49
1.10 Troubleshooting	50
1.10.1 Self-referencing table constraints can cause generation to stop	51
1.10.2 Conforming to unique constraints in SQL Data Generator	52
1.10.3 Generating data in improperly-named column causes System.InvalidOperationException	53
1.10.4 Creating random GUIDs in SQL Data Generator	54
1.10.5 Logging and log files	55
1.11 Release notes and other versions	57
1.11.1 SQL Data Generator 2.0 release notes	58
1.11.2 SQL Data Generator 1.2 release notes	59
1.11.3 SQL Data Generator 1.1 release notes	60

SQL Data Generator 2 documentation

About SQL Data Generator

With SQL Data Generator, you can populate selected tables and entire databases with realistic data. You can populate empty tables, or add extra rows to your existing data.

Quick links

[Release notes for Data Generator 2.0](#)

[Worked example - Setting up the data generator](#)

Requirements

To use SQL Data Generator you need:

- One of the following Microsoft Windows operating systems:
 - Windows 2000
 - Windows XP
 - Windows Server 2003
 - Windows Vista
 - Windows 7
 - Windows Server 2008
 - Windows 8
- SQL Server client-side tools
- [Microsoft .NET Framework 2.0 or later](#)
- [MDAC 2.8 or later](#)
- 256 MB RAM
- 200 MB hard disk space on the client
- Sufficient hard disk space on the server to store all the generated data

The following versions of SQL Server are supported:

- SQL Server 2005
- SQL Server 2008
- SQL Server 2008 R2
- SQL Server 2012
- SQL Server on Amazon RDS

Please note that this product has not been tested on an Itanium server.

Installing

Most Redgate products are available as part of a bundle. You can select which individual products to install when you run the installer.

When you install a non-free product, you have 14 days to evaluate the product. For the DLM Automation Suite, DLM Automation Suite for Oracle, SQL Source Control, Schema Compare for Oracle, Data Compare for Oracle, and Source Control for Oracle, you have 28 days. For more information, see [Licensing](#).

To install a Redgate product:

1. Download the product from the [website](#).
2. Run the installer and follow the instructions.

The product is listed on the **Start** menu under **Red Gate**.

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

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.

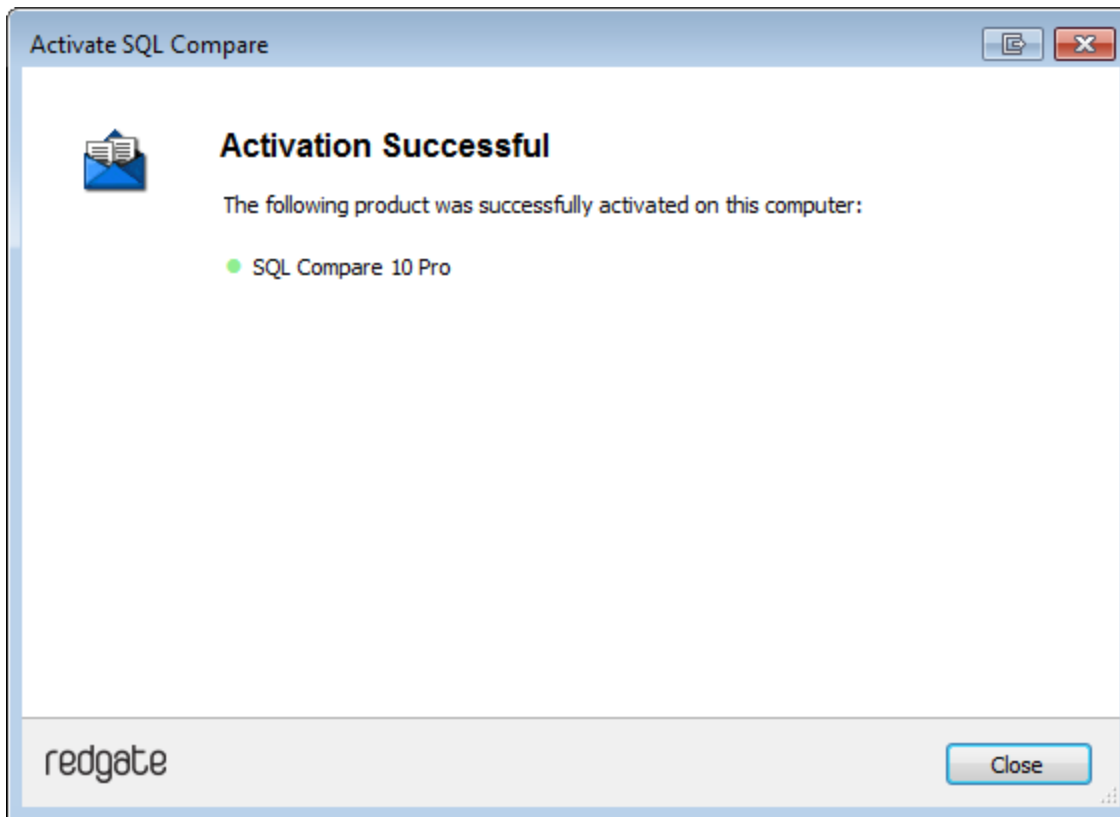
E-mail (optional)
Please provide the email address you would like us to send update notifications to:
user@example.com

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:

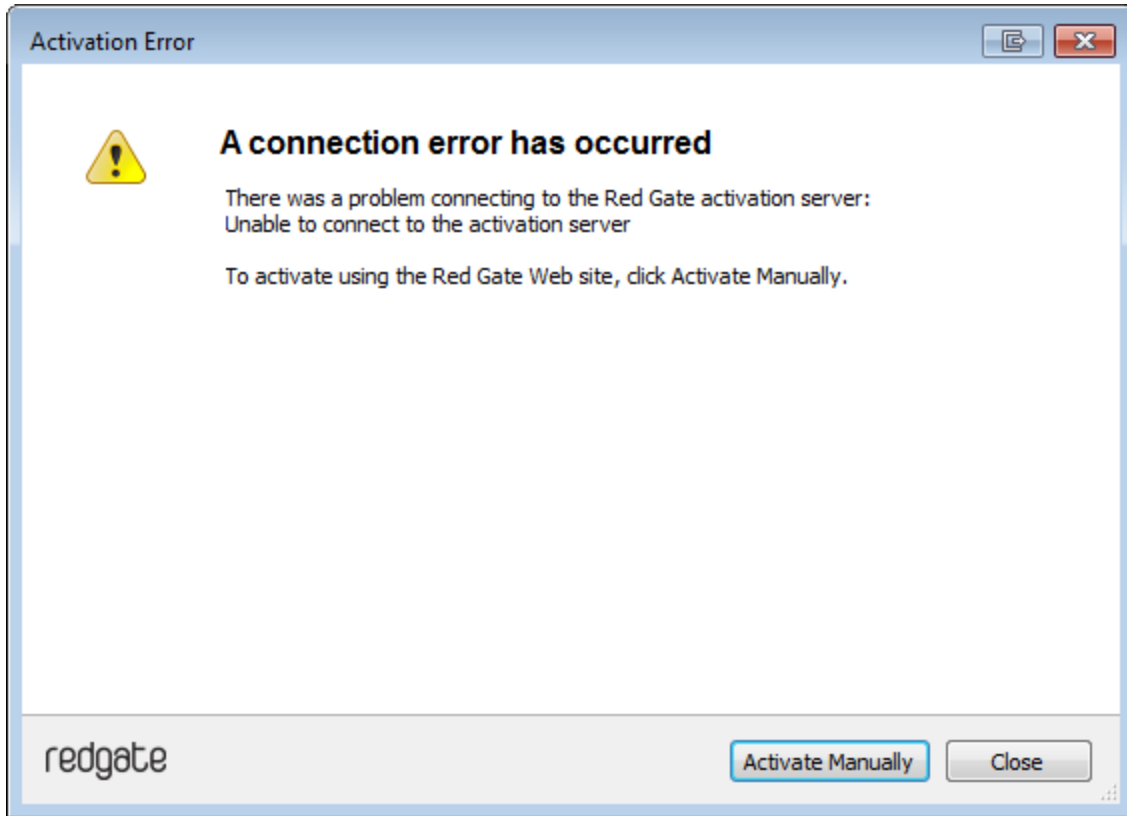
```
sqlcompare /activateSerial:123-456-789012-ABCD
```

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.

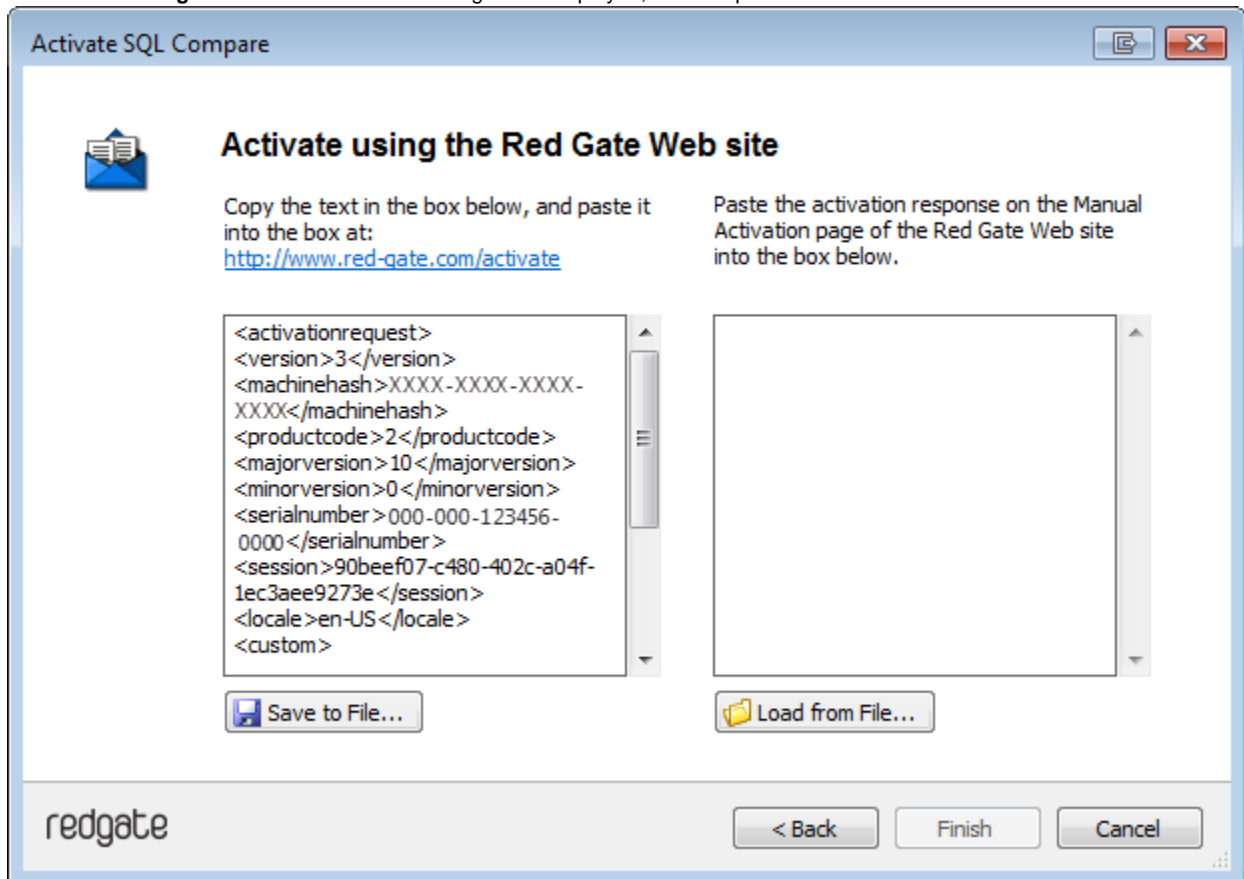
You can use manual activation whenever the **Activation Error** dialog box is displayed and the **Activate Manually** button is available, for example:



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:



- 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.
- 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>90bef07-c480-402c-a04f-1ec3aee9273e</session>
<locale>en-US</locale>
<custom>
```

Get Activation Response

Step 2

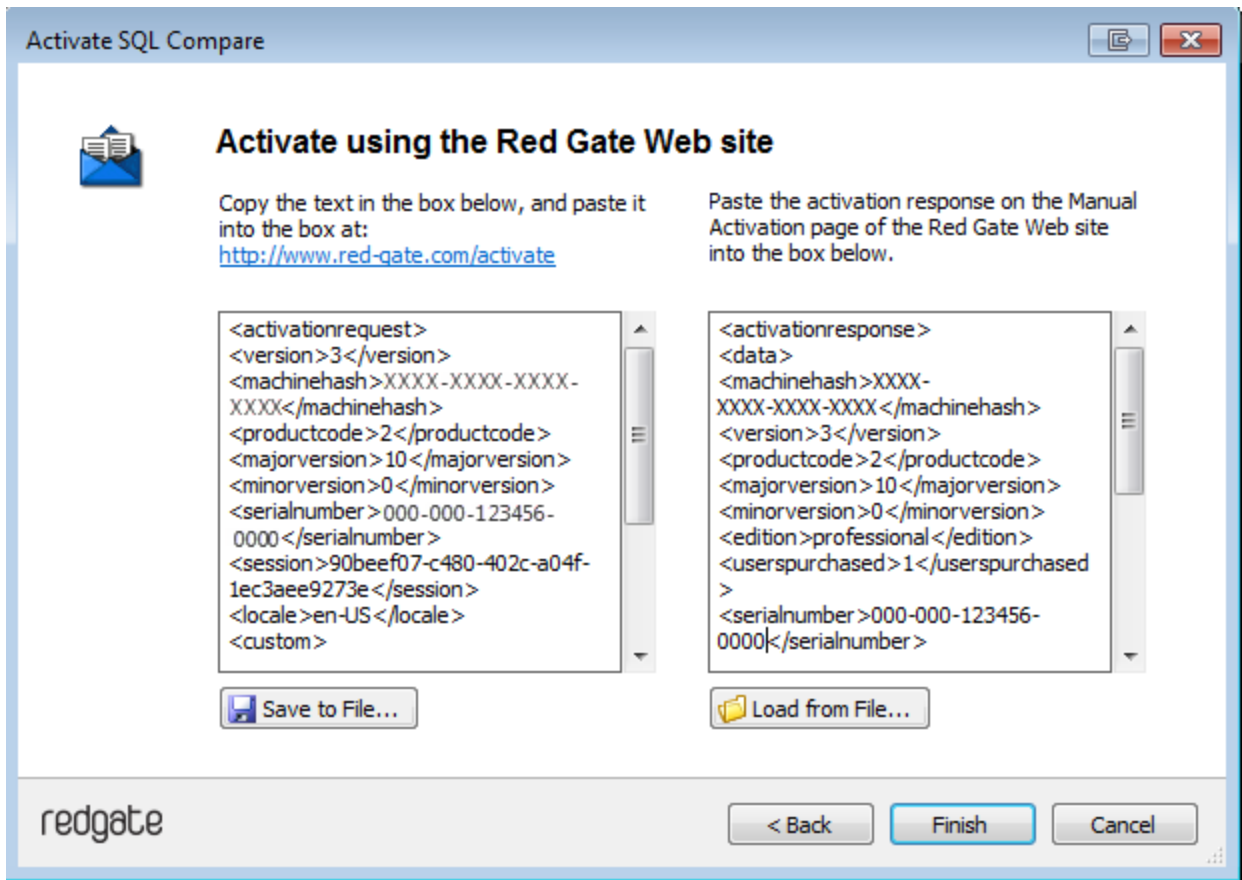
Copy the contents of this box into your product activation dialog box.

Save to File...

Got a question?

0800 169 7433
shop@red-gate.com


- Click **Get Activation Response**.
- When the activation response is displayed under **Step 2**, copy all of it. Alternatively you can save the activation response to a .txt file.
- 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.



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
```

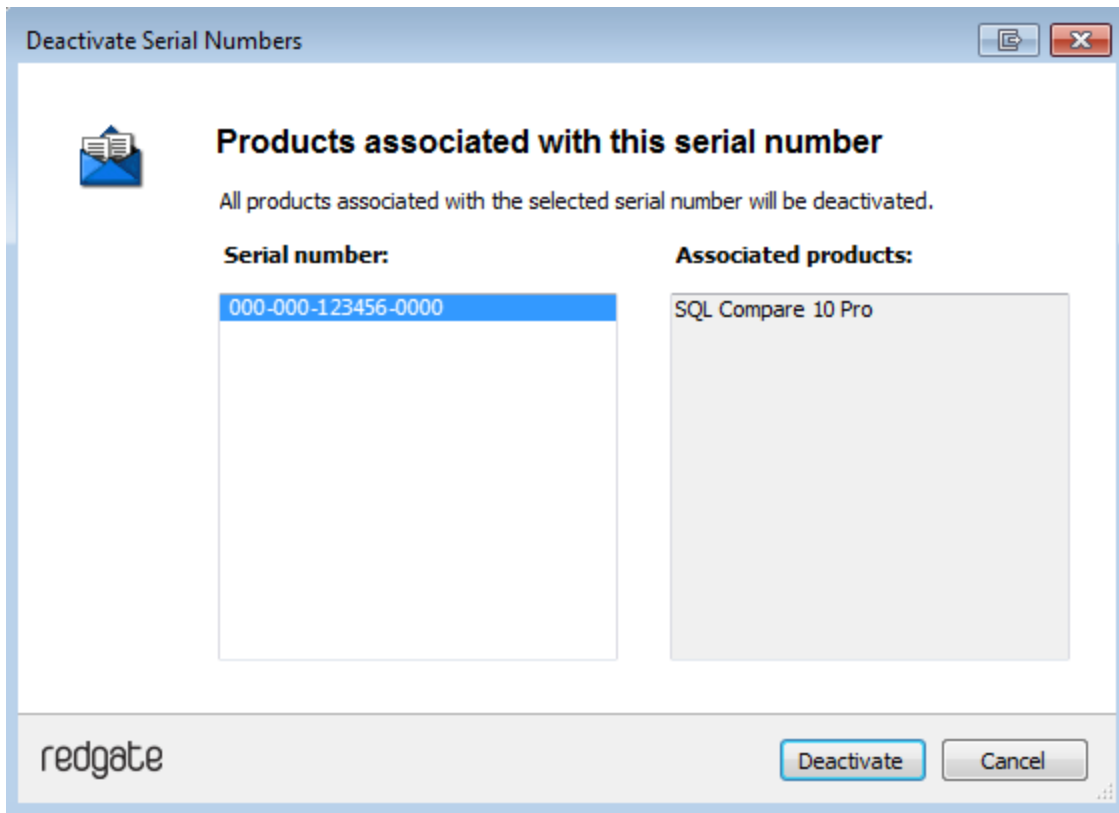
The **Deactivate Serial Numbers** dialog box is displayed. Follow the instructions below.

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:



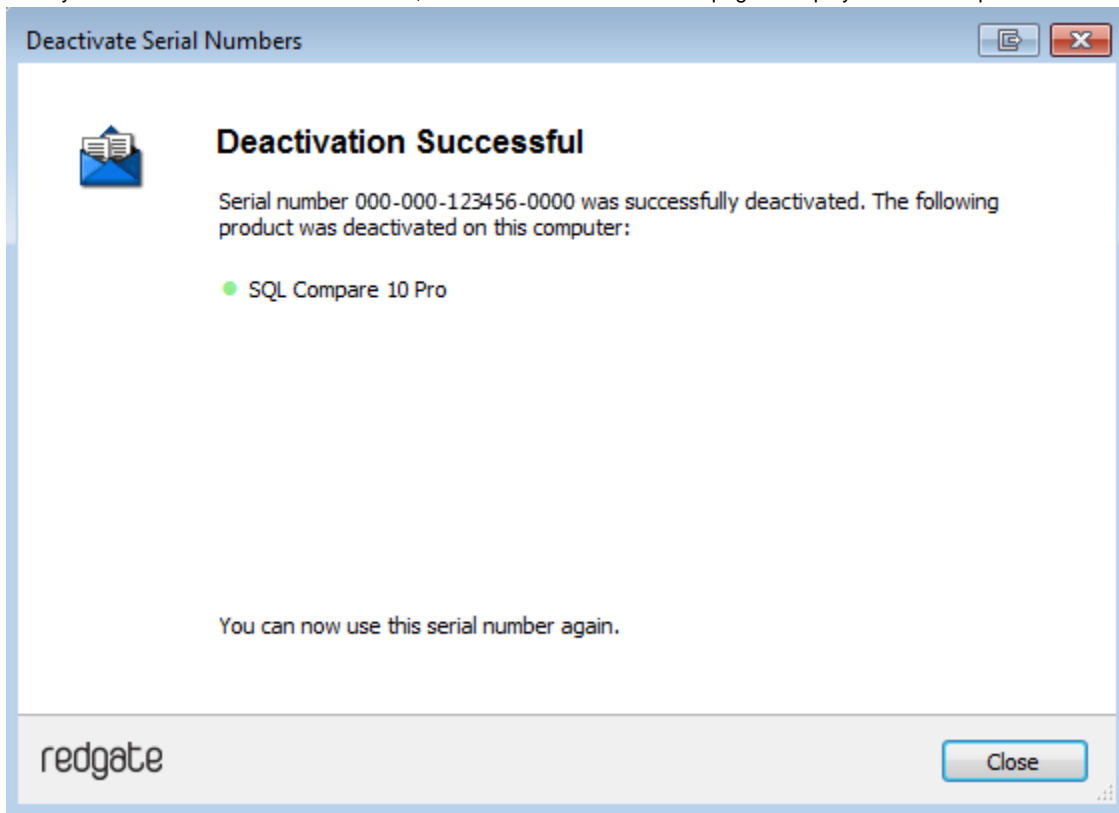
If you're running the executable file, the dialog box displays all the serial numbers for Red Gate products that have been activated on your computer.

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.

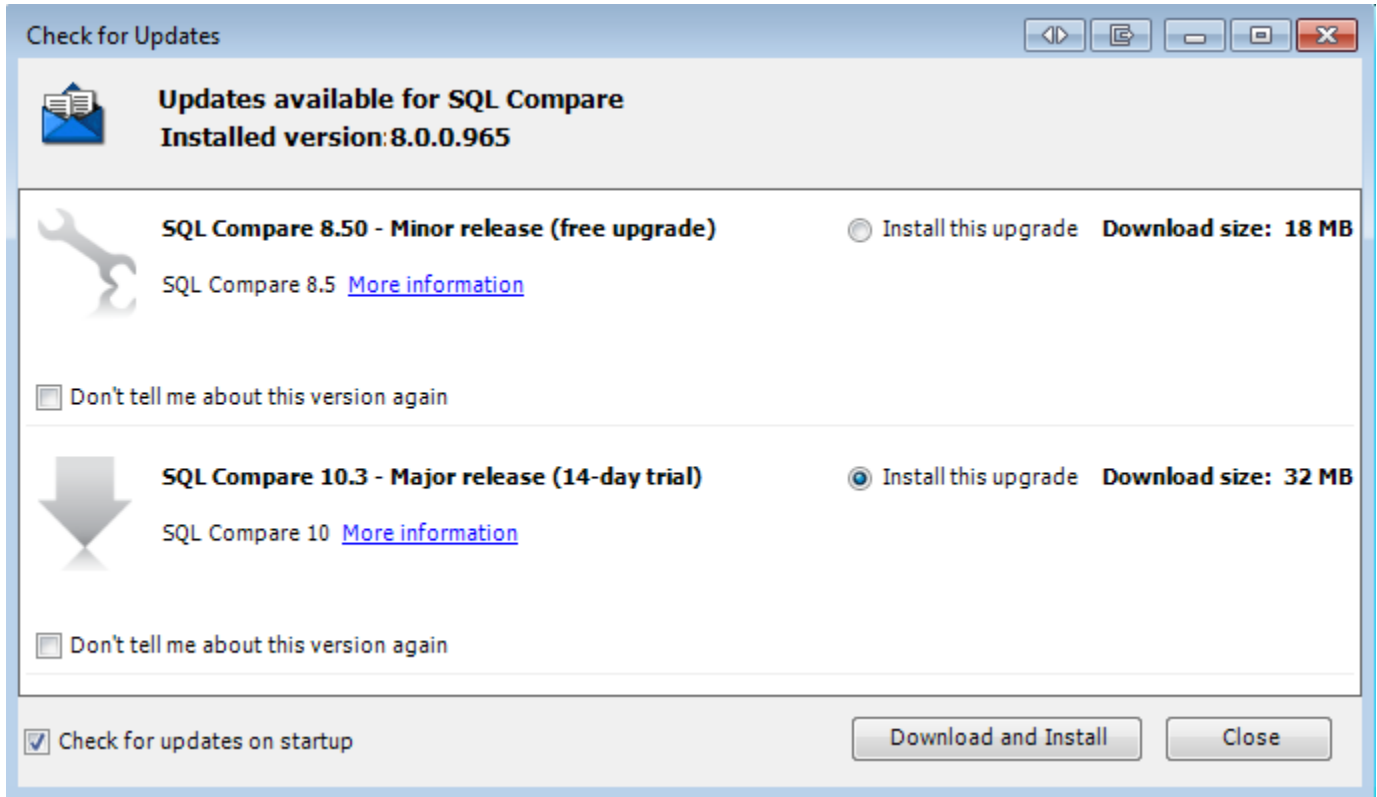
Using Check for Updates

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:



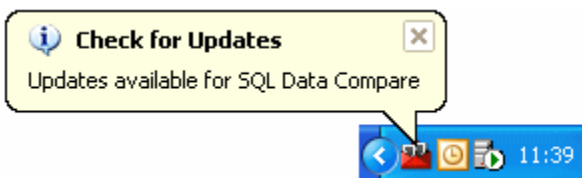
To view the full release details in your default web browser, click **More information**.

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](#).

Technical notes

SQL Data Generator automatically assigns a generator to each column based on information such as table name, column name, data type, and any constraints; otherwise the [Regular Expressions Generator](#) is assigned.

There are over 80 pre-configured generators, supporting all SQL Server 2008 data types. These are detailed in the list of [generators and the data types they support](#) (PDF).

You must have administrator privileges for the database that you want to populate.

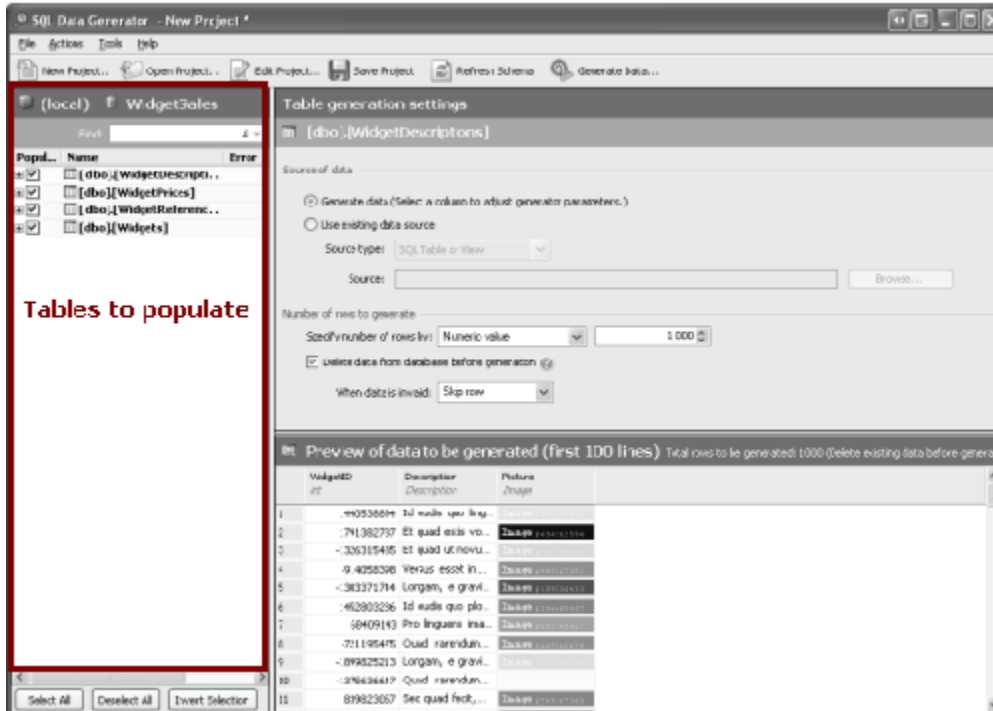
Worked example - setting up the data generator

This page provides an overview of how you set up SQL Data Generator to generate data.

You are recommended to back up the database that you are going to populate before you generate the data; you can then adjust the settings and repeat the data generation if you are not happy with the results.

To generate data, first create a project by selecting the SQL Server and database you want to populate. The project also defines some options for the data generation, and you can specify any number of SQL scripts that you want SQL Data Generator to run automatically before or after generating the data.

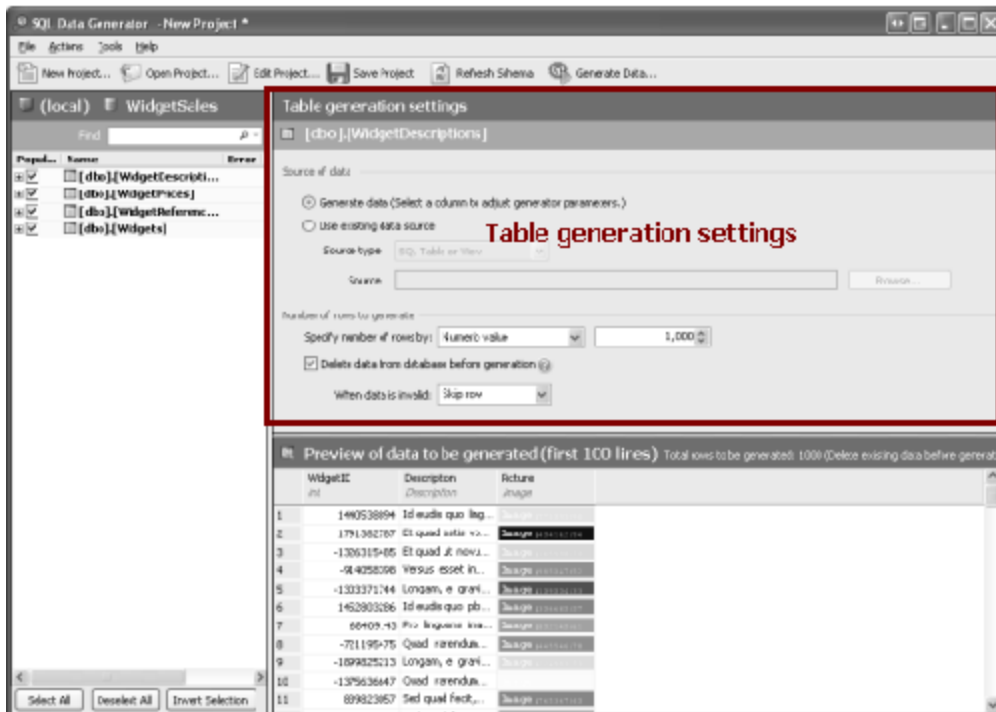
When you have created a project, the schema of the database you selected is listed in a tree view in the **Tables to populate** pane.



You specify the tables that you want to populate by selecting the **Populate** check box. By default, these are all selected, but you can change this option for both new projects and new tables in your application options (accessed from the **Tools** menu).

To see the creation SQL script for a table, right-click the table or column name in the tree view and click **Show Schema Creation Script**.

When you have selected the **Populate** check box for a table, you can define how you want the data to be generated: click the table name in the **Tables to populate** pane, and specify the details in the **Table generation settings** pane.



You can choose to:

- **create data using generators**

SQL Data Generator automatically assigns a generator to each column based on its table name, column name, data type, and length. If the column has constraints, SQL Data Generator uses these to set the generator parameters for the column; if the constraints cannot be complied with in this way, the **RegexGenerator** is assigned instead and an appropriate regular expression is set up. You can change the generator used by a particular column later if required. For detailed information, see [Using generators](#).

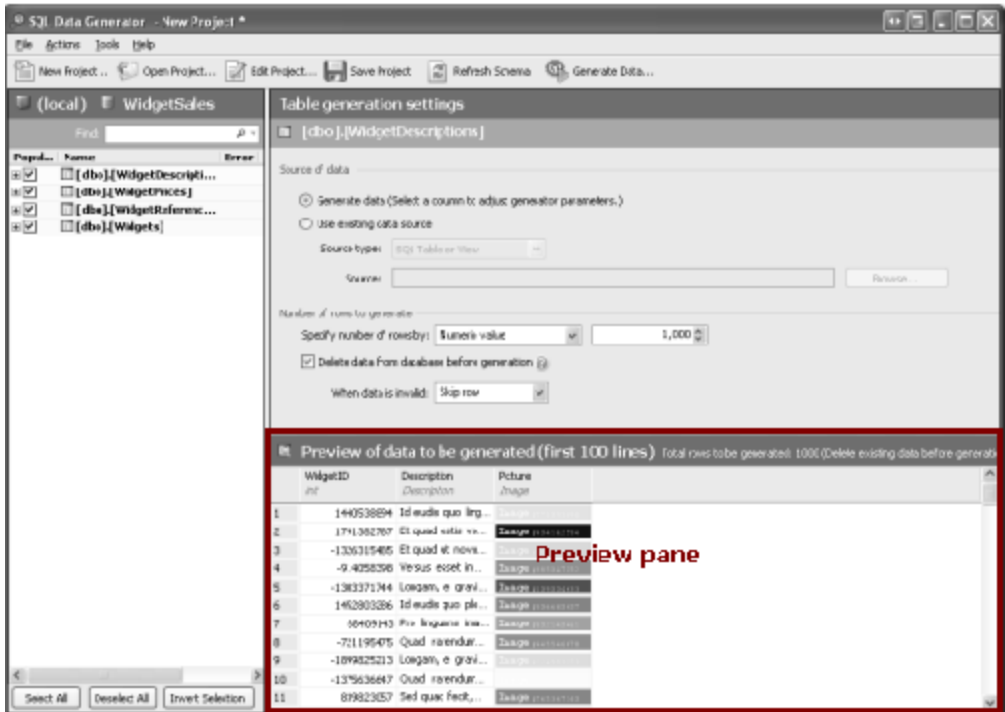
- **import data**

You can import a table or view from an existing database, or an existing CSV file. SQL Data Generator maps the columns based on name and data type. If any columns cannot be mapped, SQL Data Generator assigns a generator. You can change the mappings later if required. For detailed information, see [Mapping SQL tables or views](#) and [Mapping CSV files](#).

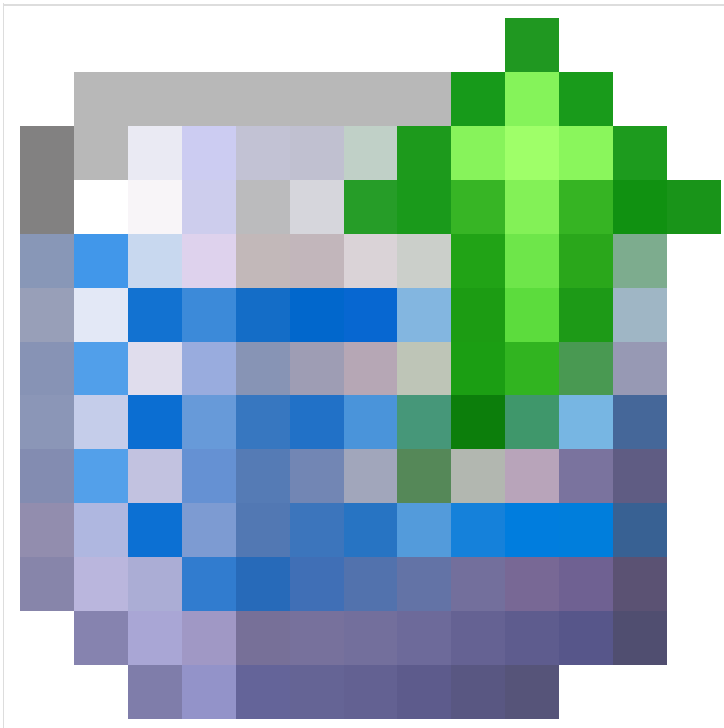
- Foreign keys are automatically assigned the **Foreign Key generator**; this cannot be changed to a different generator, but you can customize its settings.
- Columns for which data is auto-generated display *Server Assigned* in the **Generator** box; this cannot be changed. For example, identity columns and computed columns are server assigned.

In the table generation settings, you can also specify the number of rows that you want to be generated, and whether you want existing data to be deleted prior to populating the table.

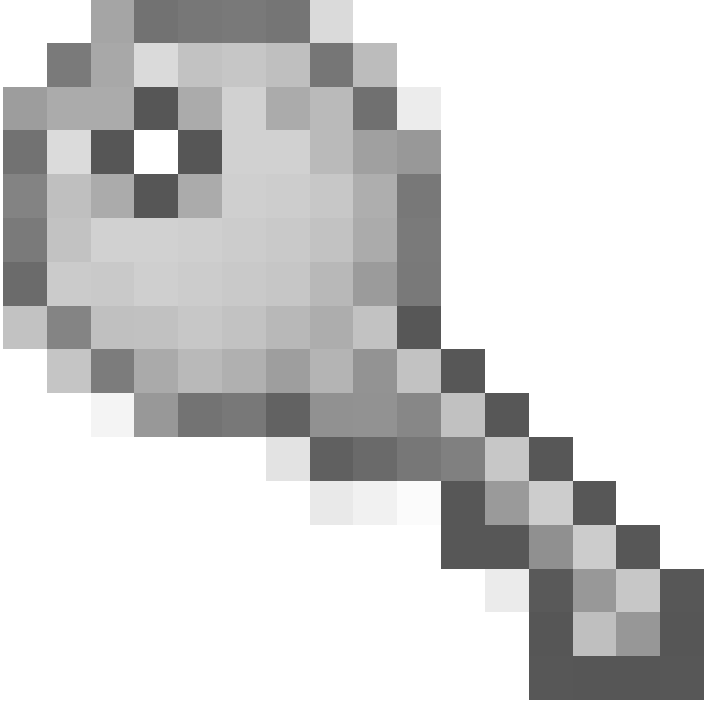
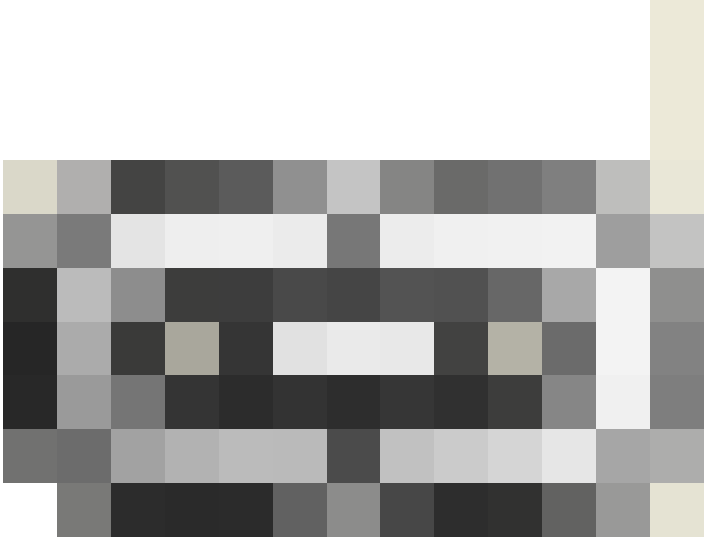
You can preview the data that will be generated for each table in the **Preview** pane.



You may see the following icons when the values for a column cannot be previewed prior to generation:



server-assigned column

	foreign key column
	computed column or a manual foreign key column

You may also see [warnings or errors](#) in the preview.

When you have set the table-level parameters, you can check the settings for each column in the table, and customize them if required. To select a column for customization, click the name of the column in the **Tables to populate** pane, or click the column in the **Preview** pane.

When you are happy with the settings for all the columns that you want to populate, click



Generate Data. An action plan provides a summary of the data generation for you to review before you generate.

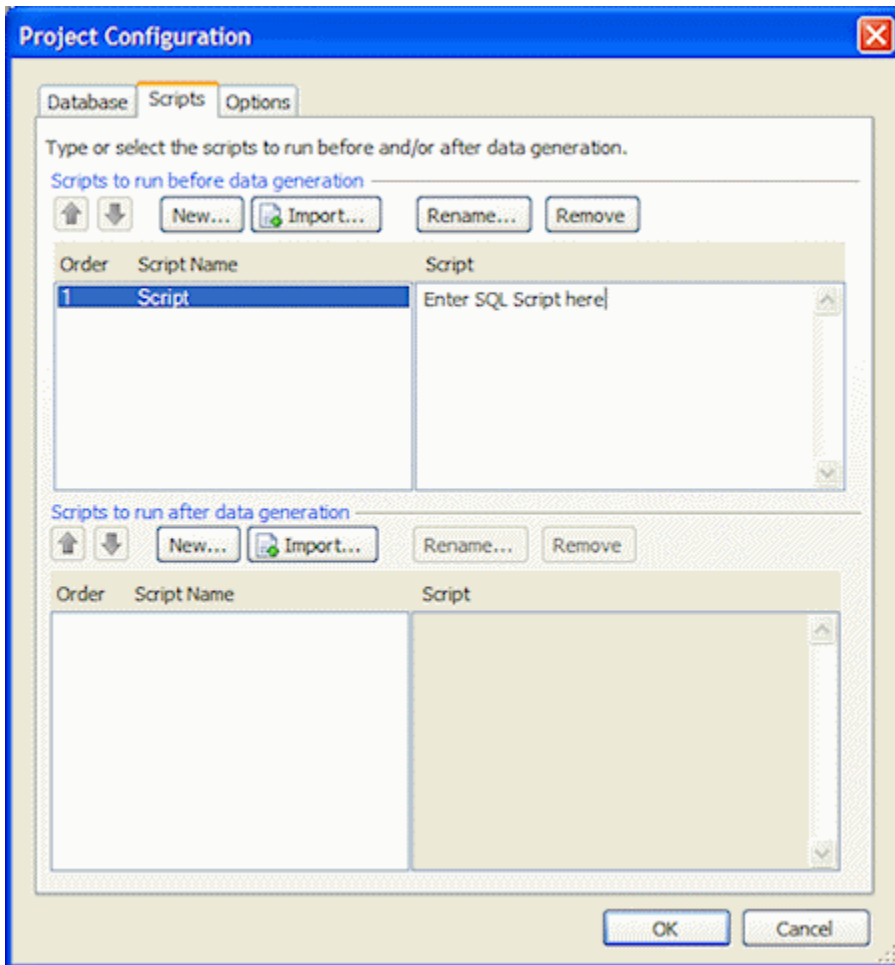
Scripts

You can run SQL scripts before or after the data generation. For example, you could use scripts to add custom data to the database. SQL Data Generator can run these scripts automatically.

You set up the scripts you want to run in the **Project Configuration** dialog box (click



Edit Project and select the **Scripts** tab).



You can link to external script files, or you can embed scripts by typing in the **Project Configuration** dialog box, (or a combination of the two). Scripts are run in the order in which you list them in the project configuration.

If a script that is run before data generation fails, the generation process is stopped.

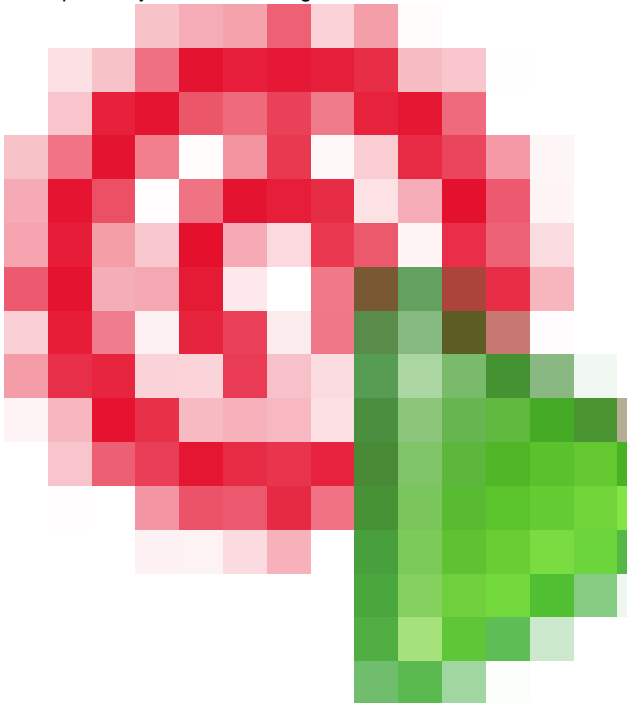
If a script that is run after data generation fails, the process continues with the next script.

Warnings and errors

When you are setting up the tables and columns that you want to populate, SQL Data Generator displays warning and error messages to inform you when there may be a problem with the data generation.



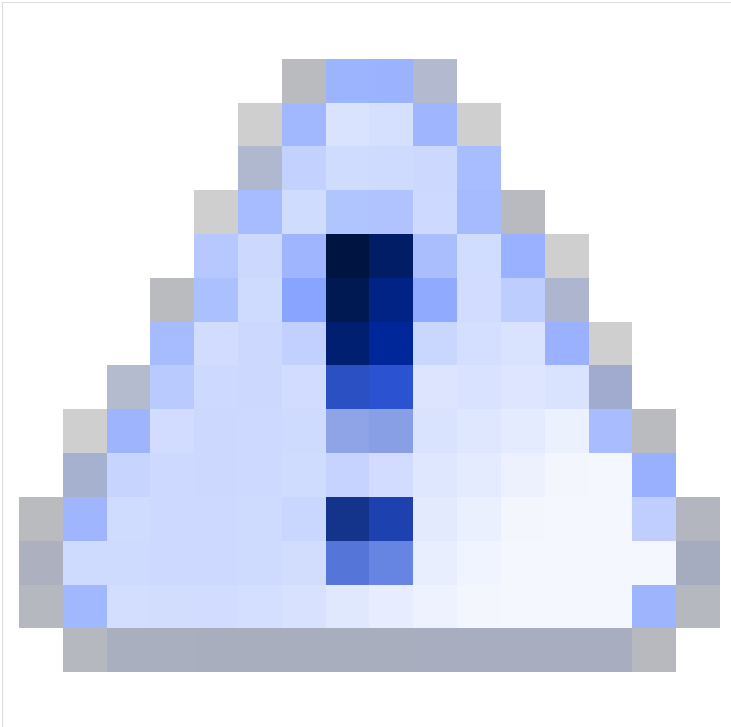
Errors prevent you from selecting



Generate Data. For example, an error is displayed if there are circ dependencies.



Warnings inform you when a problem may arise during data gener: The problem does not prevent you from generating data, but it may the data generation for a particular table. For example, a warning is displayed if you have chosen to delete rows in a table, but another references it.



Information messages provide you with information about issues that you may want to rectify, but which will not prevent data generation. For example, you may see an information message if SQL Data Generator cannot create enough rows for the preview.

To see the details of an error or warning, hover your mouse pointer over the icon to display a tooltip, or click the column name to see the details in the **Column generation settings** pane.

Refreshing the schema

When you open a project, if the database schema has changed since you created the project, the schema that has changed is flagged with



in the **Tables to populate** pane. For example, a changed column is indicated by



and a changed primary key column is indicated by



To clear the flags, click



Refresh Schema.

You can also click



Refresh Schema to see any changes that have been made to the schema since you opened it. Click



Refresh Schema again to clear the flags.

Command line

When you have set up a SQL Data Generator project, you can use the command line to run the project. For more information, see: [Using the command line.](#)

Using generators

SQL Data Generator uses generators to create the data for the tables that you choose to populate. Different generators are used to create different types of values, and to enable you to define specific parameters for the values.

When you select a column in the **Tables to populate** pane, **Generators** lists only the generators that create data of the same data type as the column. For example, if the column type is *int*, only generators that create integer values are available in the list.

SQL Data Generator provides a number of pre-defined generators, such as *FirstName*, *WorkingAge*, *Country*, and so on. These generators are grouped by subject area in the **Generator** list. You can change the settings for these generators as required.

In addition, SQL Data Generator provides some non-specific generators for you to customize:

- *SQL Type* lists a generator for each SQL data type (except CLR)
- *Generic* lists some basic generators
For information about the generic generators, see: [Generic generators](#).

For full details about customizing generators, see: [Customizing existing generators](#).

For information about how you can create your own generators, see: [Creating new generators](#).

Data types supported by the supplied generators

To see a matrix of the data types that are supported by the supplied generators, see the table of [supported data types by generator \(PDF\)](#).

Uniqueness

Many of the generators have a **Set unique** setting. When this check box is selected, SQL Data Generator makes the values that are generated for the column unique.

If the column schema has a uniqueness constraint (such as a unique index or primary key), **Set unique** is selected by default. However, you can override the uniqueness for the column by clearing the check box. For example, you may want to do this if the uniqueness constraint applies across multiple columns, and you know that another of the columns is unique. A warning is displayed, but you can proceed with the generation.

Generators that do not offer the **Set unique** option are not available for columns that have a uniqueness constraint, except for the [SQL statement](#) generator.

If **Set unique** is selected but there are not enough unique values to display in the preview, a warning is displayed. However, you can proceed with the generation.

You can change the number of values to be displayed in the preview by changing your application options from the **Tools** menu.

Check constraints

When SQL Data Generator automatically assigns generators to the columns in a new project or new schema, it sets the generator parameters to take account of any check constraints.

However, it is not always possible to set the generator parameters appropriately. When you generate data, if the values generated do not comply with a check constraint, data generation for that table is stopped and an error is reported.

You can set up the project so that check constraints are not enforced when the data is generated. To do this, clear the **Enforce check constraints** check box in the project configuration options (click



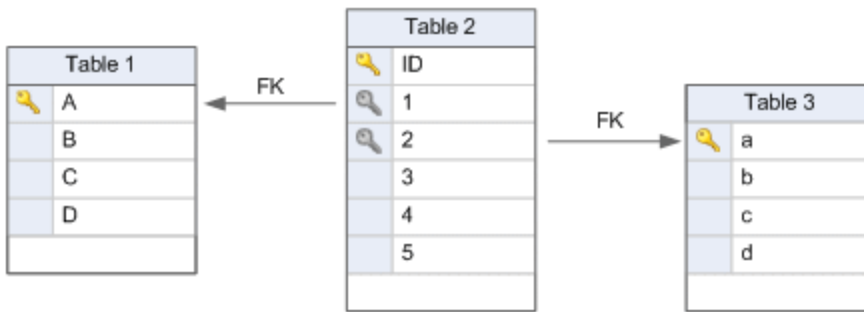
Edit Project and select the **Options** tab).

Foreign keys

When SQL Data Generator automatically assigns generators to the columns in a new project or new schema, the Foreign Key generator is assigned to all columns that have foreign key constraints.

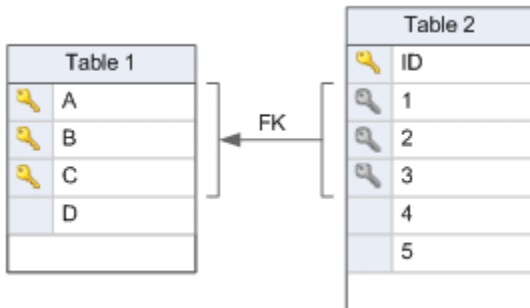
You cannot change the generator, but you can change the settings for the Foreign Key generator.

In the example below, Table 2 Column 1 references Table 1 Column A, and Table 2 Column 2 references Table 3 Column a.

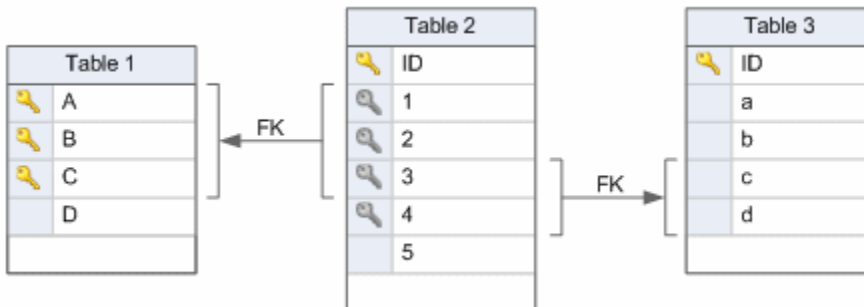


SQL Data Generator assigns the Foreign Key generator to Column 1 and Column 2. You can change the settings for these columns individually.

For a composite foreign key, the generator settings are the same for each of the columns; if you change the settings on one column, they are automatically changed on the others. In the example below, changing the generator settings for Column 2 in Table 2 also changes the settings for columns 1 and 3.



Similarly, if two or more composite foreign keys overlap, the generator settings are the same for each of the foreign keys. In the example below, changing the generator settings for Column 4 in Table 2 also changes the settings for Columns 1, 2, and 3 in Table 2.



In addition, when two composite foreign keys overlap, for the overlapping column(s) SQL Data Generator uses values that appear in both referenced tables; that is, if a value appears in one referenced table but not in the other, that value will not appear in the generated data. In the example above, only values that appear in both Table 1 Column C and Table 3 Column c will be used for Table 2 Column 3.

A NULL value in a composite foreign key is NULL across all of the columns in the foreign key.

SQL Data Generator cannot display preview values for the Foreign Key generator;



is displayed instead.

Foreign Key (manual) generator

You can create a single-column foreign key by using the Foreign Key (manual) generator, which is available under the SQL Types category.

There is no restriction on the data type of the column you select. However, if possible, you should select a column with the same data type. If you select a column with a different data type, SQL Data Generator attempts to convert the values when the data is generated; if SQL Data Generator is unable to convert the data, the data generation may fail.

You cannot create a self-referential foreign key using this generator.

SQL Data Generator cannot display preview values for the Foreign Key (manual) generator;



is displayed instead.

Dependencies

SQL Data Generator takes dependencies into account when defining the order in which tables' data will be generated.

If there are any circular dependencies,



is displayed next to the relevant columns in the **Tables to populate** pane, and



Generate Data is not available.

Computed columns

If your database contains computed columns, SQL Data Generator generates data for the referenced columns. The values for the computed column are then calculated by SQL Server.

Generating XML

There are a number of ways in which you can generate XML values:

- use the [XML generator](#) to generate XML strings
- use the [Regex generator](#) and write a regular expression that obeys the XML definition
- use the [File Import generator](#) to import XML files
- use the [SQL Statement generator](#) to retrieve values from another database that contains schema-validated XML

Generating real numbers

When you use the *real*/SQL type generator, if you set **Min** or **Max** to be a large value, sequential distribution will not produce sequential values because the increment cannot be set high enough.

Generic generators

SQL Data Generator provides the following generators in the Generic category for you to customize:

- [CSV](#)
- [File Import](#)
- [File List](#)
- [Regexp](#)
- [SQL Statement](#)
- [Text Shuffler](#)
- [Weighted List](#)

Information about each of these generators is provided below.

For information about how to customize the generators, see: [Customizing existing generators](#).

Data types supported by the supplied generators

To see a matrix of the data types that are supported by the supplied generators, see the [table of supported data types by generator \(PDF\)](#).

CSV generator

Use the CSV generator when you want to import data from a CSV file into a single column. (If you want to import data from a CSV file into an entire table or multiple columns in a table, you can use the **Use existing data source** table generation setting instead; for details, see: [Mapping CSV files](#).)

Click **Browse** to select the CSV file you want to use; you then specify the delimiters to be used when importing the data, and select the column in the CSV file that you want to import.

When you select **Shuffle data**, changing the **Seed** value in the CSV generator settings changes only the position of any null values.

File Import generator

Use the File Import generator to import the contents of files in a specified folder.

For example, if you specify a folder containing a number of images, each image is imported into a new row. You can specify a search string to identify the files within the specified folder you want to use.

If you specify large files, or if you specify a large number of files, performance will be reduced.

File List generator

Use the File List generator to import values from a text file.

You must first create a text file containing the list of values, with each value on a new line. The values will be imported from the list in a random order. You can then browse to this file when you select the File List generator.

If you have a very long list of values, you may want to consider creating a CSV file with the list of values and then importing the values using the [CSV generator](#) to import the values.

Regexp generator

Use the Regexp generator to define the generated data using a regular expression.

In the basic syntax, most characters are treated as literals (for example, `a` generates "a"). Below is a list of syntax elements.

Syntax	Example	Generates
ordinary chars	bob	bob
[chars]character set	[A-Z0- 9]	eg. 5 or G
individual chars	[FM]	F or M

initial] in char set	[[]]
[x-y] range	[0-9]	eg. 3 or 9
complement	[^abc]	eg. d or #
* zero or more	abc*	eg. abcccccc or ab
+ 1 or more	abc+	eg. abcccc or abc
? Include or not	abc?	ab or abc
(regexp) grouping	(abc)*d	eg., abcabcd or d
{num} repeat	a{4}	aaaa
{min,max} repeat	a{2,3}	aa or aaa
{min,} at least min repeats	a{3,}	eg. aaa or aaaaaaaaa
() empty string	()	
alternatives	Yes No	Yes or No
Empty Alternative	(some- often-)time	eg. some-time or often-time

Escapes

Syntax	Generates
\xdd hex char (8- bit)	eg. \x21 generates !
\xdddd hex unicode	eg. \u0021 generates !
\\	\
\.	.
\^	^
\\$	\$
\{	{
\[[
\]	\]
\((
\	
\))
*	*
\+	+
\?	?
\a	alarm character
\b	backspace
\d	digit
\e	escape
\f	formfeed
\n	newline

\t	tab
\r	carriage return
\s	space
\v	vertical tab
\w	[A-Za-z_0-9]

Use `$(file_name)` to import a text file containing a list of values.

Use `$(column_name)` to import the values from another column in the same table.

When you import values from another column:

- for repeating values, you must specify a fixed range of repetitions using `{1,10}`; you cannot use `*` when you import values from a column
- changing the **Seed** value in the RegexpGenerator settings changes only the position of any null values; to shuffle the order of the values, use the settings on the referenced column
- you shouldn't select **Set unique** in the RegexpGenerator settings, because only one row will be generated

You can use the buttons below the **Regular Expression** box to add commonly-used components for a regular expression, file lists, and table columns.

SQL Statement generator

Use the SQL Statement generator to define data to import from an external database using a SQL statement. If you want to import data from an external database into an entire table or multiple columns in a table, you can use the **Use existing data source** table generation setting instead; for details, see: [Mapping SQL tables or views](#).

Click **Edit** to specify the SQL Server and database, and then click **Next** to enter the SQL statement. The SQL statement must select a single column of values which are of the correct data type.

If the column you are populating has a unique constraint, make sure the SQL statement returns unique values; if it does not, the data generation will fail.

When you select **Shuffle data**, changing the **Seed** value in the SQL Statement settings changes only the position of any null values.

Text Shuffler generator

Use the Text Shuffler generator when you want to create values that contain words randomly selected from a pre-defined list. For example, you may want to use this to check the performance of a full text index.

You can type the text, or you can import it from a text file. You are recommended to use text that contains a high variety of words that are similar to your real data.

SQL Data Generator shuffles the text using the spaces as delimiters for the words. Therefore, if there is punctuation in the text, this will be included in the values.

Weighted List generator

Use this generator when you want to specify the percentage for the number of occurrences of each value in the column. For example, you may want to use this to check that your indexing strategy will work on the table.

Customizing existing generators

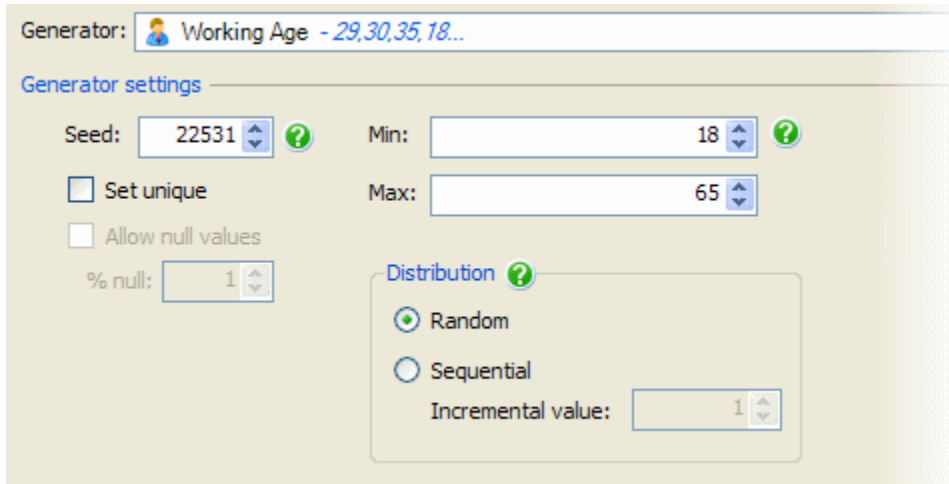
You can customize the generators in the following ways:


- change the settings for each column
- for generators that use lists, create a custom list of values

Changing the settings


When you select a generator for a column, you can customize the generator by changing the default settings.


For example, in the Personal category, *Working Age* is pre-configured to have a minimum value of 18 and a maximum value of 65. However, in your company the minimum age may be 21, so you can change this value in **Min**.



Generator:  Working Age - 29,30,35,18...

Generator settings

Seed: 


Min: 

Set unique

Max:

Allow null values

% null:

Distribution 

Random

Sequential

Incremental value:

Similarly, if you have selected the *Title* generator, you may want to add the title *Prof*. To do this, you can edit the expression in **Regular expression**. Change:

```
Mr | ( (Mrs | Miss | Ms) | Dr ) ( \. ? )
```

To:

```
Mr | ( (Mrs | Miss | Ms) | Dr | Prof ) ( \. ? )
```

For more information about regular expression generators, see: [Regexp generator](#).

This is a good way to customize a generator for an individual column.

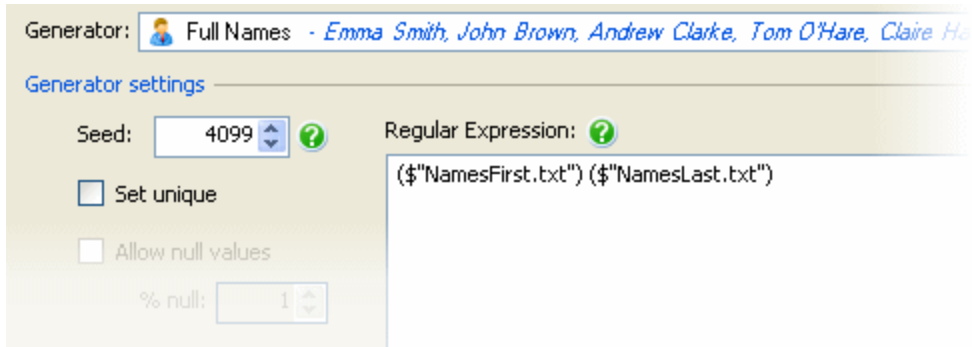
Creating custom lists

Some of the generators use lists of values (dictionaries) to supply the data. For example, *First Name (Female)* uses the list *NamesFirstFemale.txt*

The lists are located in `%ProgramFiles%\Red Gate\SQL Data Generator 2\Config` (on 32-bit machines) or `%ProgramFiles(x86)%\Red Gate\SQL Data Generator 2\Config` (on 64-bit machines).

You can create your own lists by creating a text file with each value on a new line. You can then amend the generator settings so that it uses your file. For example, for *First Name (Female)*, click **Browse...** to select your file. Alternatively, you can use the generic [File List generator](#).

For generators that use regular expressions such as *Full Names*, edit the regular expression to specify the name of the new text file.



For more information about regular expression generators, see: [Regexp generator](#).

If you will want to use the generator with the custom list again, you can create a new generator that references the custom list file, see: [Creating a custom generator](#).

Creating a custom generator

You may find that you want to make the same customizations for a number of columns. In this case, you can customize a generator and add it to the list of available generators so that you can re-use it.

To do this, find the SQL type or generic generator that you want to base the custom generator on, and then find its corresponding `.xml` file in `%ProgramFiles%\Red Gate\SQL Data Generator 2\UserExample\Config` (on 32-bit machines) or `%ProgramFiles(x86)%\Red Gate\SQL Data Generator 2\UserExample\Config` (on 64-bit machines).

Create a copy of the `.xml` file and name it appropriately, then edit the contents to specify the properties as required.

The example below shows the code for the *Working Age* generator which generates random integers between 18 and 65. It is based on `Int32Generator.xml`.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<generators>
  <generator
    /* Specify the class. */
    type="RedGate.SQLEngine.SQLDataGenerator.Generators.Number.Int32Generator"
    /* The name and description to be displayed in the generators list, */
    /* and its category */
    name="Working Age"
    description="29,30,35,18..."
    category="Personal">

    /* Set the properties for the generator. */

    <property name="MinValue">18</property>
    <property name="MaxValue">65</property>

    /* Specify the columns that match, and their score. */
    <matches field="Age" score="50"/>

    /* Define the data types for which the generator is valid. */
    <type type="Int64"/>
    <type type="Int32"/>
    <type type="Int16"/>
    <type type="Byte"/>
  </generator>
```

SQL Data Generator uses the matches field string when it allocates the generators to columns. You can specify a regular expression search string. You then specify a score; if more than one generator matches a particular column, SQL Data Generator uses the one with the highest score.

Creating new generators

You may want to create a new generator if the supplied generators don't meet your requirements and you can't [customize](#) them to suit your needs.

To write your own generator, you must be proficient at a .NET 2.0 language, have a good understanding of .NET, and have access to SQL Data Generator on your computer.

The procedure is summarized below:

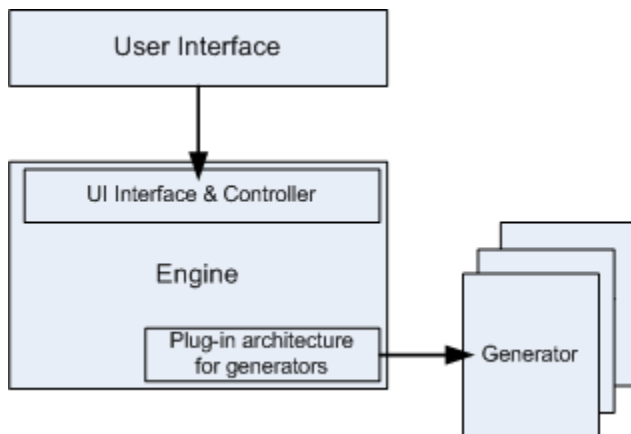
1. In Microsoft Visual Studio, create a Class Library .NET project.
2. Add references to RedGate.SQLDataGenerator.Engine and RedGate.SQLCompare.Engine
3. Create a public class that implements IGenerator.
4. Add the class attributes.
5. Implement the constructor.
6. Implement the method GetEnumerator.
7. Copy the output DLL to %ProgramFiles%\Red Gate\SQL Data Generator 2\Generators (on 32-bit machines) or %ProgramFiles(x86)\Red Gate\SQL Data Generator 2\Generators (on 64-bit machines).

Example Microsoft Visual Studio 2005 project files are provided in %ProgramFiles%\Red Gate\SQL Data Generator 2\UserExample\Generator (on 32-bit machines) or %ProgramFiles(x86)\Red Gate\SQL Data Generator 2\UserExample\Generator (on 64-bit machines).

The examples are written in C#. If you want to build the projects, you must first add the project references for the SQL Compare Engine and the SQL Data Compare Engine. You may also want to change the output path. When you have built the project, copy the output DLL to %ProgramFiles%\Red Gate\SQL Data Generator 2\Generators (on 32-bit machines) or %ProgramFiles(x86)\Red Gate\SQL Data Generator 2\Generators (on 64-bit machines).

Architecture

A simple diagram of the architecture is shown below.



The SQL Data Generator engine defines a series of interfaces. Each interface is very lightweight.

A generator must implement a series of interfaces in order that the engine considers it to be a generator. At startup, a specified folder is scanned for DLLs. Each DLL is loaded in turn, and reflection is used to check whether any public classes implement these interfaces. If they do, the class is considered to be a generator and is made accessible to the rest of the system.

By default, parameters for the generator are displayed in a standard Microsoft Grid Control. To override this default functionality, implement interfaces IGeneratorUIStyle and IGeneratorUI. See the *FullDemo* project for an example.

Basic interface: IGenerator

The IGenerator interface must be implemented for your class to be considered a generator.

The IGenerator interface is defined as:

```

namespace RedGate.SQLDataGenerator.Engine.Generators
public interface IGenerator
{
    IEnumerator GetEnumerator(
        GenerationSession session);
}
}

```

You must also implement a special constructor that takes a single parameter of type `GeneratorParameters`. This parameter describes the SQL field in the Table that is being assigned. If necessary, your code can throw exceptions and so on.

To display your generator in the graphical user interface (GUI), you must add a simple `Generator` attribute to your class. The following example code produces random values between 0 and 1024 for the 8 times table.

```

namespace Basic
{
    [Generator(typeof(int), "Generic", "8 times table",
        "8, 16, 0, 256, ...")]
    public class Basic : IGenerator
    {
        public Basic(GeneratorParameters parameters)
        {
        }

        public System.Collections.IEnumerator GetEnumerator(
            GenerationSession session)
        {
            Random r = new Random(0);
            while (true)
            {
                yield return r.Next(0, 1024) * 8;
            }
        }
    }
}

```

The `Generator` attribute defines the type of .NET result, the Category that the generator is to be placed in, and the name and description to be displayed in the GUI. It must be defined only once per class.

SQL Data Generator assigns the SQL data type that corresponds to the specified type of .NET result. To create a generator that supports multiple SQL data types, add `SupportSQLType`. SQL Data Generator will add SQL data types based on `SqlTypes` defined in the SQL Compare engine.

Make sure the class is public, and the `Generator` class exists.

Constructor

`GeneratorParameters` provides access to the field. This enables your code to verify that lengths and types are consistent.

GetEnumerator

The easiest way to implement this code is by using the `Yield` statement; the above example never runs out of values. However, it is not always possible to do this. The engine is designed to cater for a limited number of values from the `GetEnumerator`. If necessary, the `GetEnumerator` can throw exceptions.

Interface: ISeedableGenerator

This interface enables the generator to specify a seed. The generator can then generate random data that is different each time.

The ISeedableGenerator is defined as:

```
public interface ISeedableGenerator
{
    int Seed { get; set; }
}
```

Use the seed to initialize the Random class in the GetEnumerator.

The engine will automatically give a value to Seed at initialization. Each column will have its own seed, therefore the same generator can be assigned multiple times within a table and different values will be produced for each column.

A typical implementation is:

```
public int Seed
{
    get { return m_Seed; }
    set { m_Seed = value; }
}
```

For a complete example, see *%ProgramFiles%\Red Gate\SQL Data Generator 2\UserExample\Generator\Seedable\Seedable.cs* (on 32-bit machines) or *%ProgramFiles(x86)%\Red Gate\SQL Data Generator 2\UserExample\Generator\Seedable\Seedable.cs* (on 64-bit machines).

Interface: IUniqueableGenerator

This interface enables the generator to specify whether the data generated is unique so that the generator can then generate a unique value.

The IUniqueableGenerator is defined as follows:

```
public interface IUniqueableGenerator
{
    bool Unique { get; set; }
}
```

A typical implementation is:

```
public bool Unique
{
    get { return m_Unique; }
    set { m_Unique = value; }
}
```

For a complete example, see *%ProgramFiles%\Red Gate\SQL Data Generator 2\UserExample\Generator\Uniqueable\Uniqueable.cs* (on 32-bit machines) or *%ProgramFiles(x86)%\Red Gate\SQL Data Generator 2\UserExample\Generator\Uniqueable\Uniqueable.cs* (on 64-bit machines).

The generator can now be assigned to unique fields. The engine automatically configures the Unique flag as on when the generator is assigned.

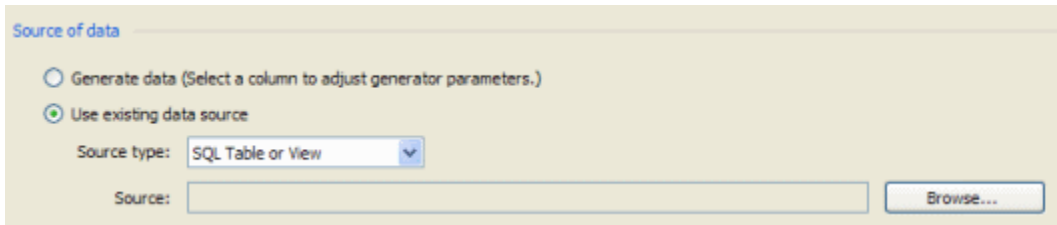
Using existing sources

- Mapping SQL tables or views
- Mapping CSV files
- Cleansing or removing sensitive data from an existing data source

Mapping SQL tables or views

You can populate an entire table, or multiple columns in a table, by mapping to another SQL table or view. For example, this may be useful if you want to import master data or lookup data into your schema.

With the table selected in **Tables to populate**, click **Use existing data source**, select **SQL Table or View** and click **Browse**.

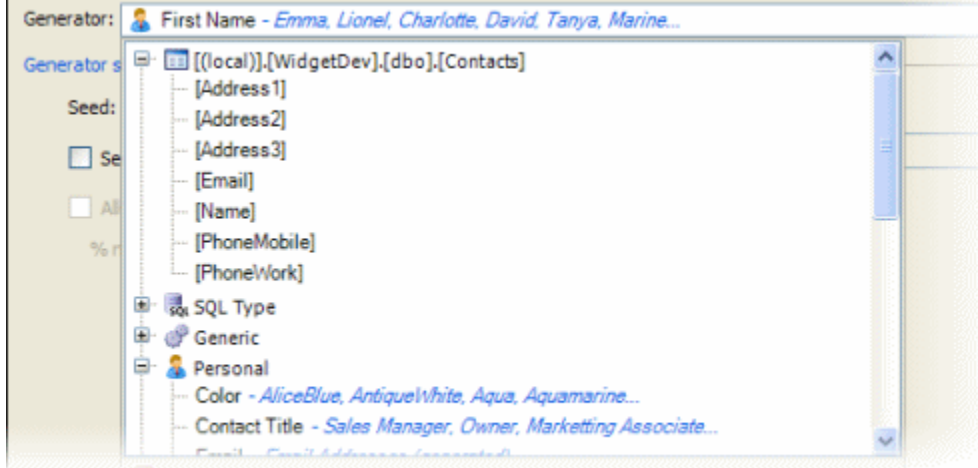


You can then select the SQL Server and database, and the table you want to use from the Select SQL Table or View wizard. When you click **Finish** on the wizard, SQL Data Generator matches columns in the two tables based on data type and column name.

Any columns in the table for which SQL Data Generator does not find a match will have a generator assigned to them instead. However, you can still map the schema column to a column in the external SQL table as long as the data types match:

1. Select the column in the **Tables to populate** pane.
2. Click the **Generator** list.

The list displays the table name, with any columns for which the data types match.



3. Click the name of the column you want to use.

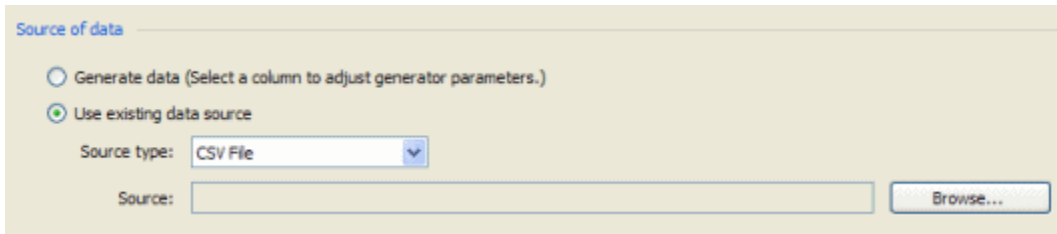
Similarly, if SQL Data Generator maps a schema column to an external SQL table column but you do not want to use it, you can select a different generator for that column.

If you want to define data to import from an external database using a SQL statement, use the SQL Statement generator.

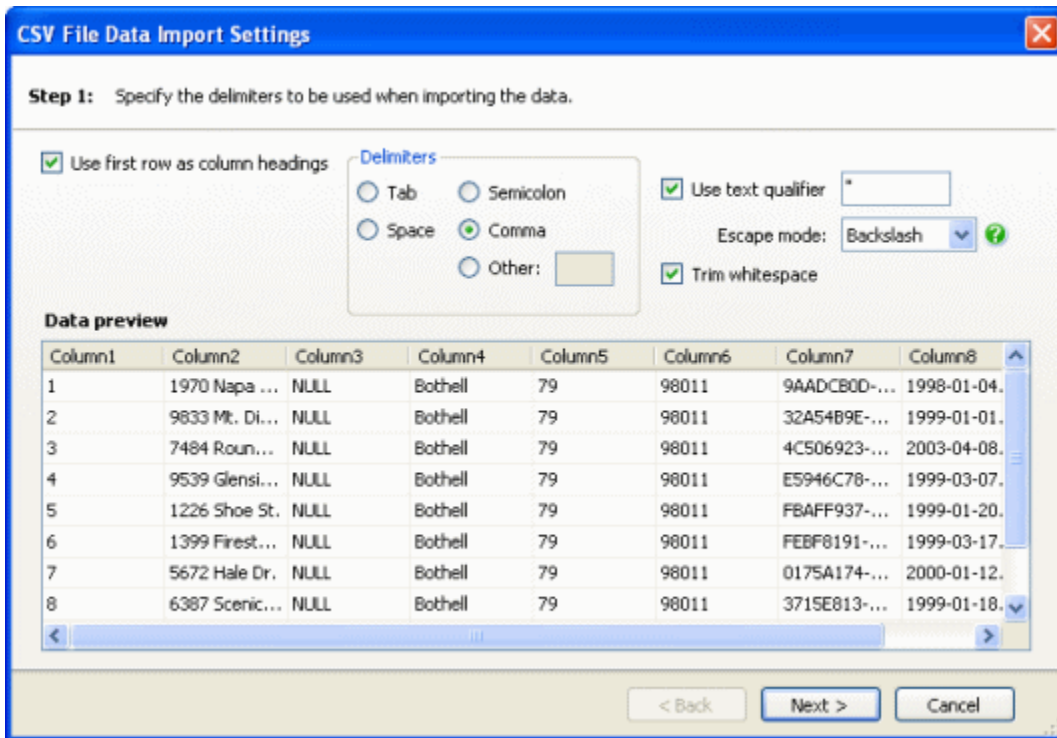
Mapping CSV files

You can populate an entire table, or multiple columns in a table, by mapping an imported CSV file to the table. For example, this may be useful if you want to import master data or lookup data into your schema.

With the table selected in **Tables to populate**, click **Use existing data source**, select **CSV File** and click **Browse**.

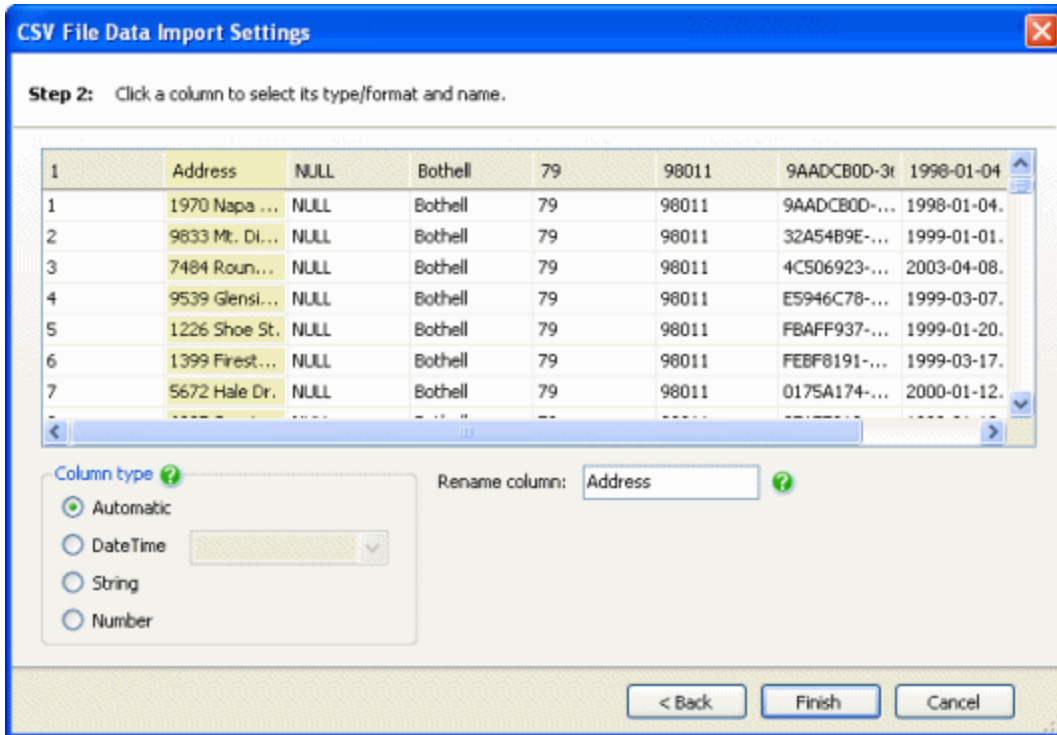


You can then select the file, and define the import settings.



Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8
1	1970 Napa ...	NULL	Bothell	79	98011	9AADC80D-...	1998-01-04.
2	9833 Mt. Di...	NULL	Bothell	79	98011	32A54B9E-...	1999-01-01.
3	7484 Roun...	NULL	Bothell	79	98011	4C506923-...	2003-04-08.
4	9539 Glensi...	NULL	Bothell	79	98011	E5946C78-...	1999-03-07.
5	1226 Shoe St.	NULL	Bothell	79	98011	FBAFF937-...	1999-01-20.
6	1399 Firest...	NULL	Bothell	79	98011	FEBF8191-...	1999-03-17.
7	5672 Hale Dr.	NULL	Bothell	79	98011	0175A174-...	2000-01-12.
8	6387 Scenic...	NULL	Bothell	79	98011	3715E813-...	1999-01-18.

SQL Data Generator matches columns in the CSV file to columns in the table based on data type and column name. If the data type and/or column name for a column in the CSV file is not the same as in the table, you can specify these in the data import settings.



When you click **Finish**, SQL Data Generator maps the columns.

Any columns in the table for which SQL Data Generator does not find a match will have a generator assigned to them instead. However, you can still map the schema column to a column in the CSV file as long as the data types match:

1. Select the column in the **Tables to populate** pane.
2. Click the **Generator** list.

The list displays the CSV file path, with any columns for which the data types match.



3. Click the name of the column you want to use.

Similarly, if SQL Data Generator maps a schema column to a CSV file column but you do not want to use it, you can select a different generator for that column.

If you want to import data from a single column in a CSV file into a single column in a table, use the **CSV generator**.

Cleansing or removing sensitive data from an existing data source

There may be occasions when it would be useful to use existing data, but the data source contains sensitive data such as dates of birth, personal email addresses or credit card numbers. For example, you may want to send a database to a third party application vendor for troubleshooting. SQL Data Generator can help in this situation by enabling you to use most of the existing data but replacing the personal information.

This page explains how you can set up a SQL Data Generator project to use existing data and then replace the sensitive data with randomly-generated but realistic data.

Preparing the data

The data must exist in a SQL Server database. You must create a copy of the database because you will be removing the sensitive data permanently.

To create the copy, either:

- restore a backup of the database to a new location, or
- use [SQL Packager](#) to make a .NET executable of the existing database and then run the executable to create a new database

Setting up the SQL Data Generator project

When you have created a copy of the the database, create a new SQL Data Generator project using the copy. You must use the copy because you will be using the original database as the data source but permanently replacing the columns containing sensitive data in the copy.

For further information about how to create a SQL Data Generator Project, see: [Setting up the data generator](#).

To create a new project:

1. Click **New Project**.
2. In the **Project Configuration** dialog box, select the SQL Server and name of the database you just created and click **OK**.
3. In the **Tables to populate** pane, click **Deselect all** and then select only the tables containing the columns with sensitive data that you want to replace.

In the **Table generation settings** pane, for each table that contains data to be cleansed:

1. Under **Source of data**, select **Use existing data source**.
2. Next to **Source type**, make sure *SQL Table or View* is displayed.
3. Next to **Source**, click **Browse**.
4. In the **Select SQL Table or View** dialog box, select the original SQL Server and database and click **Next**.
5. Select the table containing the columns that you want to replace and click **Finish**.
6. Make sure the **Delete data from table before generation** check box is selected.

You have now mapped the existing database tables to the tables that contain the data to be replaced so that any columns within that table that you do not want to change retain the original data. Now you need to select the columns containing sensitive data within these tables and choose the generators to generate random data.

To select the column and associated generator:

1. In the **Tables to populate** pane, expand the table tree.
2. Select the column you wish to replace.
3. In the **Column generation settings** pane, select an appropriate generator from the drop-down list.
4. The preview pane is updated dynamically showing that the original data will be replaced with randomly-generated data.

Repeat the above steps for each column with sensitive data.

Generating the data

When you are happy with the preview, click



Generate Data above the **Table generation settings** pane.

The **Data Population Summary** dialog box displays an action plan for the data generation.

Click **Generate Data**. Data is generated according to the options specified in your project, using the existing data for columns you did not want to change, and randomly-generated data for those you did.

A data generation report in PDF format is then displayed, confirming the number of rows inserted into each table. You can export and save this file if required. Close the report to return to SQL Data Generator.

If you are likely to perform the same data generation task, save and name the project: click



Save Project.

Packaging the cleansed database

Now that the data is ready for distribution, you can decide how to despatch it. You can either create a backup file or use SQL Packager to create a .NET executable.

You can then send the data to the recipient with no personal information at risk of compromise.

Using the command line

SQL Data Generator provides a command line interface for you to use with SQL Data Generator projects that you have already created using the graphical user interface.

This page describes how to use the basic features of the command line.

Getting help from the command line

To display help, enter:

```
SQLDataGenerator /help
```

This displays a brief description, and help on all the command line switches.

For more detailed help enter:

```
SQLDataGenerator /help /verbose
```

This displays a detailed description of each switch and the values it can accept (where applicable), and all exit codes. To output the help in HTML format, enter:

```
SQLDataGenerator /help /verbose /html
```

Entering a command

When you enter a command line, the order of switches is unimportant. You are recommended to follow the Microsoft convention of separating a switch from its values using a colon as shown below.

```
/out:output.txt
```

(You can separate a switch that accepts a single value from its value using a space, but this is not recommended.) Values that include spaces must be delimited by double quotation marks ("). For example:

```
/out:"c:\output file.txt"
```

Aliases

Many of the switches have an alias. The alias provides a convenient shorthand way to specify the switch. For example, `/?` is the alias for the `/help` switch, and `/v` is the alias for the `/verbose` switch.

```
Switches and aliases aren't case-sensitive.
```

`/verbose` and `/quiet` switches

The standard output mode prints basic information about what the tool is doing while it is executing. You can specify verbose and quiet modes using the `/verbose` and `/quiet` switches respectively: in verbose mode, detailed output is printed; in quiet mode, output is printed only if an error occurs.

Redirecting command output

Output from all commands can be redirected to a file by one of several methods:

- Use the `/out` switch to specify the file to which you want output directed:

```
SQLDataGenerator ... /out:outputlog.txt
```

where *outputlog.txt* is the name of the file. If the file exists already, you must also use the `/force` switch to force the tool to overwrite the file, otherwise an error will occur.

- Use the output redirection features that are provided by the shell in which you are executing the command. From the standard command prompt provided by Windows, you can redirect output to a file as follows:

```
SQLDataGenerator ... > outputlog.txt
```

The redirection operator (`>`) and file name must be the last items on the command line.

If the specified file exists already, it will be overwritten.

To append output from the tool to an existing file, for example to append to a log without losing the data already present in the log, enter the following:

```
SQLDataGenerator ... >> existinglog.txt
```

If you are scripting using a language such as VBScript, JScript, PHP, Perl, or Python, or if you want to access the tool from Web pages using ASP.NET, refer to the documentation for the language.

Exit codes used in the command line

Code	Description	Additional information
0	Success	
1	General error code	
3	Illegal argument duplication	Some arguments may not appear more than once in a command-line. If such arguments appear more than once this exit code will be returned.
8	Unsatisfied argument dependency or violated exclusion when user runs command line	For example, <i>/arg2</i> depends on <i>/arg1</i> but you have specified <i>/arg2</i> without specifying <i>/arg1</i> , or alternatively <i>/arg2</i> cannot be used with <i>/arg1</i> but you have tried to use them both.
32	Value out of range	Numeric value supplied for an argument that is outside the range of valid values for that argument.
33	Value overflow	The magnitude of a value supplied for an argument is too large and causes an overflow.
34	Invalid value	The value supplied for an argument is invalid.
35	No / invalid software license or trial period has expired	
63	Error	
64	General command-line usage error	
65	Data error	Some input data required by the tool is invalid or corrupt.
69	A resource or service required to run the tool is unavailable	
73	Failed to create report	
74	IO error occurred	Generally returned if the program attempts to write to a file that already exists without the user having specified the <i>/force</i> option.
77	Action cannot be completed because the user does not have permission	
126	Execution failed because of an error	
130	Execution stopped because Ctrl+Break	

Troubleshooting

Common issues

- Self-referencing table constraints can cause generation to stop
- Conforming to unique constraints in SQL Data Generator

Error messages

- Generating data in improperly-named column causes System.InvalidOperationException

Technical questions

- Creating random GUIDs in SQL Data Generator

Self-referencing table constraints can cause generation to stop

When generating data against a table which has foreign key references to itself, the following error may occur:

Generation stopped. The generator for column [COLUMN] could not generate any more values

The following example illustrates a reproduction for this issue:

```
CREATE TABLE [dbo].[Account]
(
  [AccountID] [uniqueidentifier] NOT NULL,
  [AccountNumber] [nvarchar] (12) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
  [AccountName] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
  [AccountType] [int] NOT NULL,
  [LedgerID] [uniqueidentifier] NULL,
  [UniqueID] [int] NOT NULL IDENTITY(1, 1),
  [Version] [datetime] NOT NULL,
  [FinYearID] [uniqueidentifier] NOT NULL
)

GO
ALTER TABLE [dbo].[Account] ADD CONSTRAINT [AccountTypeRule] CHECK (([AccountType]=(1)
OR [AccountType]=(2) OR [AccountType]=(3) OR [AccountType]=(4) OR [AccountType]=(5)))
GO
ALTER TABLE [dbo].[Account] ADD CONSTRAINT [PK_Account] PRIMARY KEY CLUSTERED
([UniqueID])
GO
ALTER TABLE [dbo].[Account] ADD CONSTRAINT [UK_AccountID_FinYearID] UNIQUE
NONCLUSTERED ([AccountID], [FinYearID])
GO
ALTER TABLE [dbo].[Account] ADD CONSTRAINT [UK_AccountName_FinYearID] UNIQUE
NONCLUSTERED ([AccountName], [FinYearID])
GO
ALTER TABLE [dbo].[Account] ADD CONSTRAINT [FK_Account_FinancialYear] FOREIGN KEY
([FinYearID]) REFERENCES [dbo].[FinancialYear] ([FinYearID])
GO
ALTER TABLE [dbo].[Account] ADD CONSTRAINT [FK_LedgerID_AccountID] FOREIGN KEY
([LedgerID], [FinYearID]) REFERENCES [dbo].[Account] ([AccountID], [FinYearID])
GO
```

This is a known issue and we are currently evaluating support for this scenario.

Conforming to unique constraints in SQL Data Generator

Tables that have columns with unique constraints and indexes may require special attention when generating data using SQL Data Generator.

For example, the following table contains a unique index on the `TicketTypeCode` column. SQL Data Generator may generate non-unique and NULL data for this column by default, violating the constraint:

```
CREATE TABLE [dbo].[TicketType](
  [TicketTypeID] [int] IDENTITY(1,1) NOT NULL,
  [TicketTypeCode] [nvarchar](10) NOT NULL,
  [TicketTypeName] [nvarchar](50) NOT NULL,
  [RowVersion] [timestamp] NOT NULL,

  CONSTRAINT [PK_TicketType] PRIMARY KEY CLUSTERED
  (
    [TicketTypeID] ASC
  )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
  ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

CREATE UNIQUE NONCLUSTERED INDEX [IX_TicketType] ON [dbo].[TicketType]
(
  [TicketTypeCode] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
  IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
  ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

In order for data in the above example to be generated, change the setting for the column `[TicketTypeCode]` in SQL Data Generator to an Account Number or 5 Digit ID, and make sure the "Set unique" option is selected. This will allow the data to be generated.

Generating data in improperly-named column causes System.InvalidOperationException

When generating data for a table whose name contains a dot (.), the following exception message will be displayed and data generation will stop:

```
System.InvalidOperationException: Failed to obtain column collation information for the destination table.
```

The cause of the problem is that the Microsoft ADO .NET SqlClient's bulk insert functionality contains a bug that will cause a collation error if there is a dot in the table name. For instance, a table called *[dbo].[My.Table]* would cause this error to occur.

There is no known workaround for this error at this time except to rename the table or exclude it from data generation.

For more information, see: <http://support.microsoft.com/kb/944389> (Microsoft documentation).

Creating random GUIDs in SQL Data Generator

You may want to task SQL Data Generator with creating a Globally-Unique Identifier (GUID) to populate a text column.

It is possible to create a GUID in a string field by using the RegexpGenerator and entering the following in the Regular Expression field:

```
[0-9a-f]{8}\-[0-9a-f]{4}\-[0-9a-f]{4}\-[0-9a-f]{4}\-[0-9a-f]{12}
```

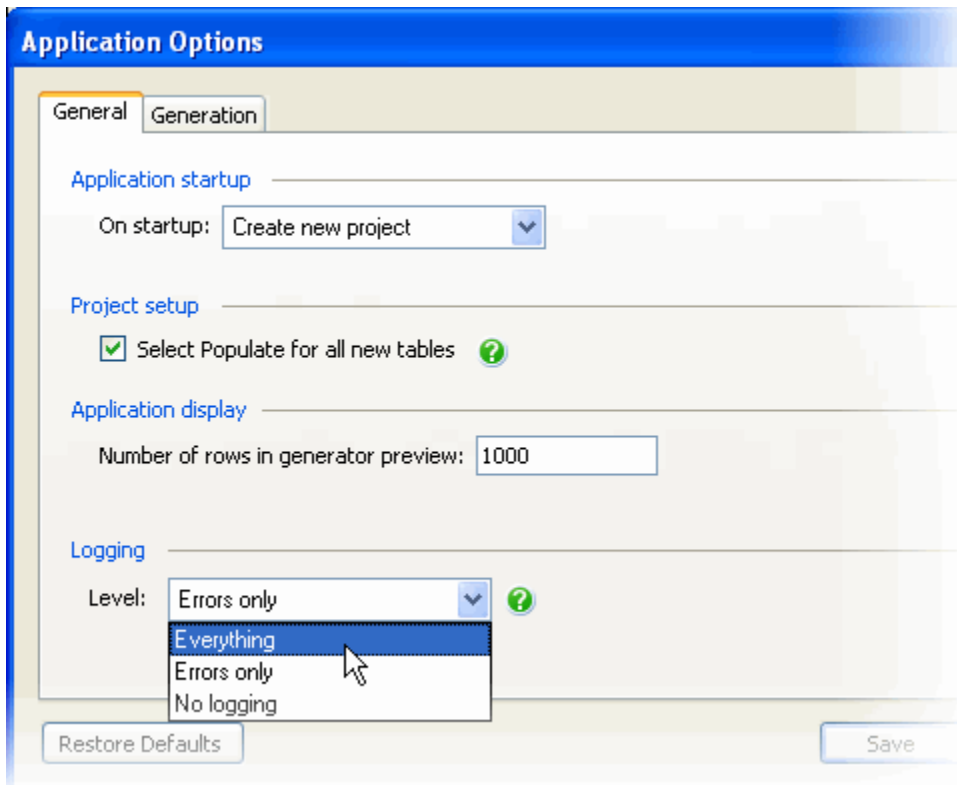
Logging and log files

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

Selecting a logging level

To select a log level:

1. On the **Tools** menu, click **Application Options**.
2. In the **General** tab, under **Logging**, click the required log level from the **Level** list.



Everything	Reports all messages in the log file.
Errors only	Reports serious errors and core information. For example, an error message might report a failed operation.
No Logging	Disables logging.

By default, the logging level is set to **Errors only**.

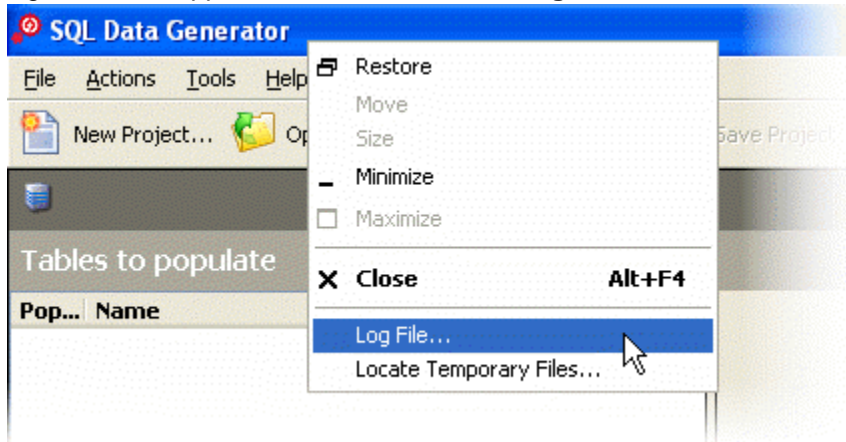
The selected logging level may affect performance. **Everything** reports all messages, and so writes the most information to disk and produces the largest log files.

Log files are automatically deleted after 10 days.

Viewing and locating the log files

To view the log file for the current session in your default text editor:

- Right-click the application title bar and click **Log File**:



To open the folder where the log files are stored, click **Locate Temporary Files**. By default the log files are stored in:

%ALLUSERSPROFILE%\Local Settings\Temp\Red Gate\SDG

Release notes and other versions

Version 3.1 (current)	November 17th, 2015 (latest)	Release notes	Documentation
Version 3.0	January 2nd, 2014	Release notes	
Version 2.0	October 2011	Release notes	Documentation
Version 1.2	July 28th, 2008	Release notes	Documentation (PDF)
Version 1.1	May 7th, 2008	Release notes	<i>See version 1.2 documentation</i>
Version 1.0	March 31st, 2008	Original release	<i>See version 1.2 documentation</i>

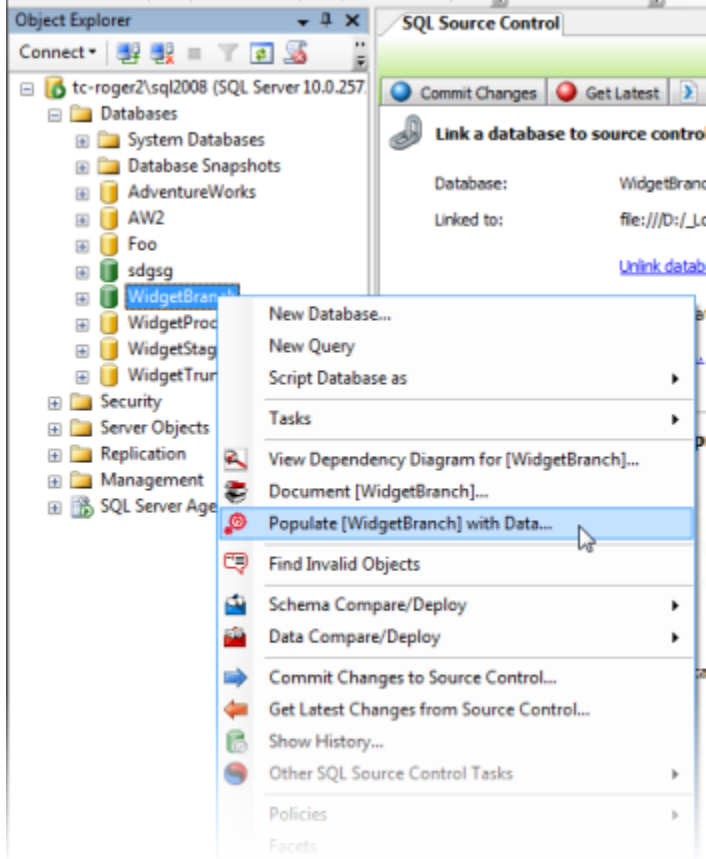
If you need to install an old version of SQL Data Generator, go to [Download old versions of products](#).

SQL Data Generator 2.0 release notes

October 2011

- **SQL Server 2008 R2 support**
- **Launch SQL Data Generator from within SSMS**

You can now generate data by right-clicking a database in the Object Explorer:



SQL Data Generator launches with the database details pre-populated and a generator selected.

- **Scriptable Python generator**
Version 2 introduces a new Python generator, so you can create custom test data to meet your needs.
- **Bug fixes**

SQL Data Generator 1.2 release notes

July 28th, 2008

- SQL Server 2008 support, including generators for the new SQL Server data types

SQL Data Generator 1.1 release notes

May 7th, 2008

- Improved manual key generator so it matches the automatic foreign key generator
- Improved support for generating more rows over composite foreign keys
- Bug fix where the population type on foreign keys was being reset every time you loaded the project
- Fixed a crash that was caused by certain constraints
- Scripts can now be renamed
- Bug fix with user defined types where the length of the strings was detected incorrectly
- Improved command line so you can see a generation report or what errors stopped generation
- Fixed a bug with the constraint detection where case insensitivity was not taken into account
- Visible generation start and end time on the generation report
- Fixed a focus issue that means clicking on the generate data button would not save the changes to settings e.g. row count