Using XLReporter with Microsoft SQL Server

Overview

Microsoft SQL Server is a relational database that is used to store data in tables. Microsoft SQL Server Express is a lightweight version of SQL Server which is free to download, distribute and use.

To use SQL Server, an instance needs to be created. This represents a complete SQL Server which contains its own copy of the server files, databases and security credentials. The Microsoft SQL Server Management Studio is used to create and manage SQL Server instances and so it is suggested that this is downloaded.

This document describes how to setup a database in either SQL Server or SQL Server Express 2008 and above.

Database Considerations

Server Specification

When connecting to Microsoft SQL Server, a list of available servers on the network is presented. If the server you are looking for is not listed, it may be because the **SQL Server Browser** service is not running on the server.

To enable, on the machine where SQL Server is installed, open the **SQL Server Configuration Manager**. Select **SQL Server Services**, right-click **SQL Server Browser** and select **Start**.

Alternatively, the SQL Server name can be entered manually.

OLEDB Driver

When a connector is configured to Microsoft SQL Server, by default it uses the Microsoft OLEDB Provider for SQL Server that is delivered with the Windows Operating System. However, this OLEDB provider does not support TLS 1.2 security which can be configured in later versions of SQL Server.

The telltale sign that TLS 1.2 security is enabled on the database is that when the Test Connection button is clicked, the error returned is: **SSL Security error**.

To combat this, Microsoft has released an updated driver (Microsoft OLEDB Driver for SQL Server). This can be downloaded from Microsoft's website and installed. Both the 32 (x86) and 64 (x64) drivers should be installed.

If this driver is installed on the PC, when a connector to SQL Server is configured, it will use the new driver rather than the one delivered by the Operating System.

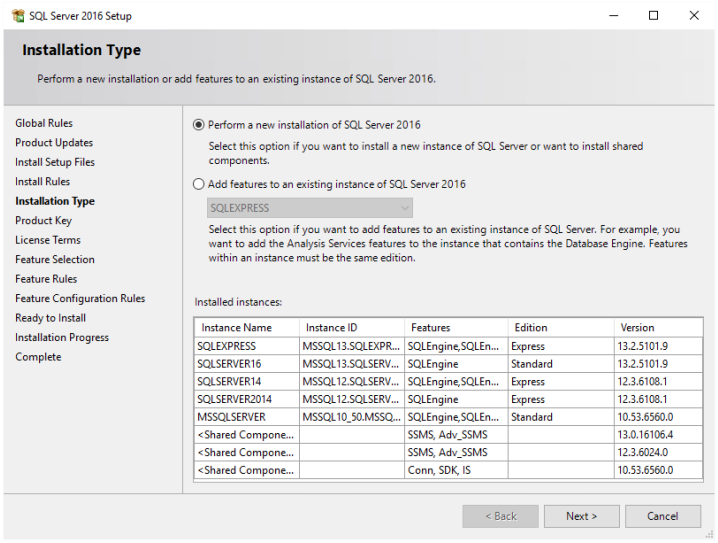
Setup

To use SQL Server, an instance needs to be created. This represents a complete SQL Server which contains its own copy of the server files, databases and security credentials. The Microsoft SQL Server Management Studio is used to create and manage SQL Server instances and so it is suggested that this is downloaded.

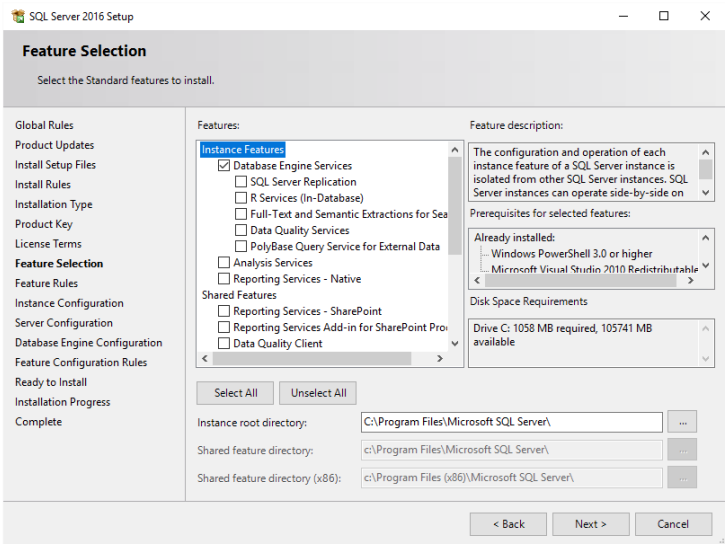
The following describes how to setup a database in either SQL Server or SQL Server Express 2008 and above.

Create a SQL Server Instance

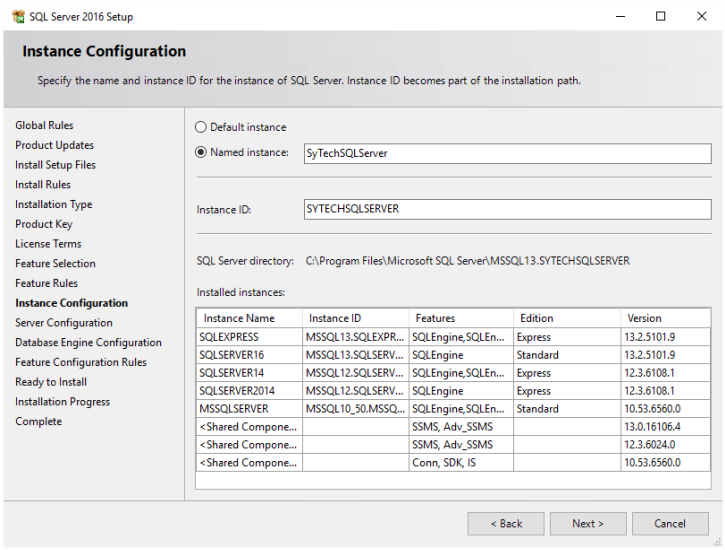
Start the SQL Server Installation Media and select Installation, **New SQL Server stand-alone installation or add features to an existing installation**.



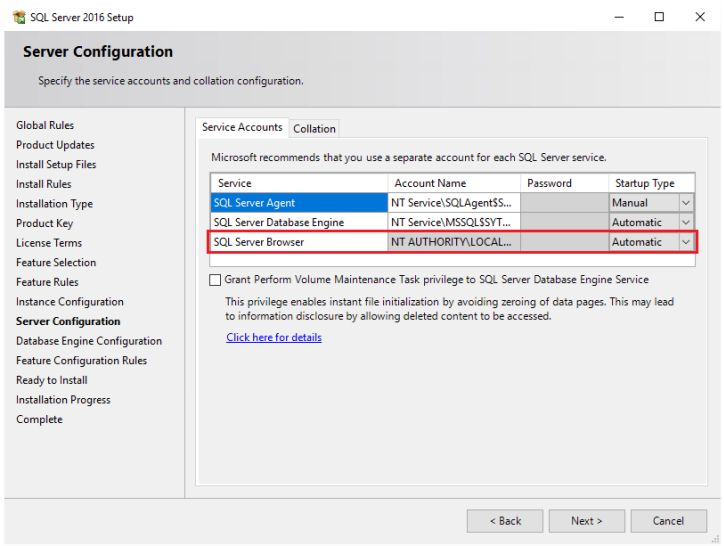
- Check **Perform a new installation of SQL Server**.



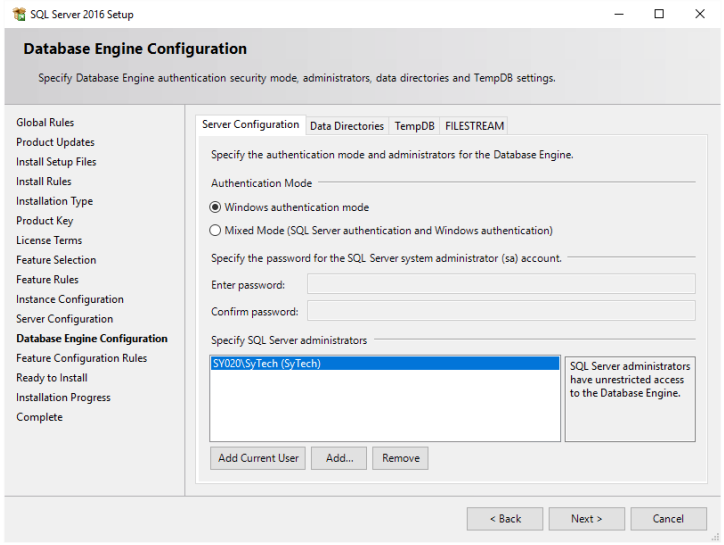
- Check the features to install. At a minimum check **Database Engine Services**.



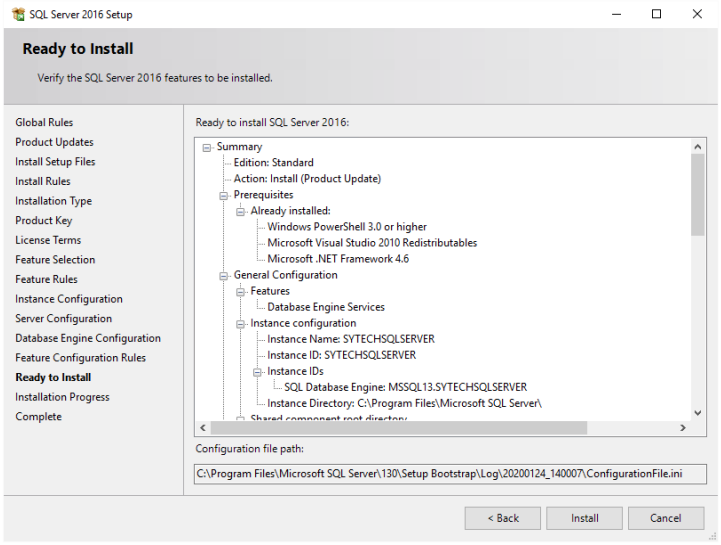
- Enter a **Named Instance**.



- Set **SQL Server Browser** to *Automatic*.



- Click **Add Current User** to give the current Windows account permission to the database.
- If the database will be accessed across the network enable **Mixed Mode**.



- Click **Install** to start the setup of the Instance.

SQL Server Applications

There are many applications that are or can be installed to help configure SQL Server. The following are the main ones referred to in this document.

SQL Server Management Studio

SQL Server Management Studio provides all the tools to create databases, tables, views as well as setting up users and security.

Management Studio is not installed by default with SQL Server. It must be downloaded and installed separately.

Once installed, SQL Server Management Studio is available from the Microsoft SQL Server program group.

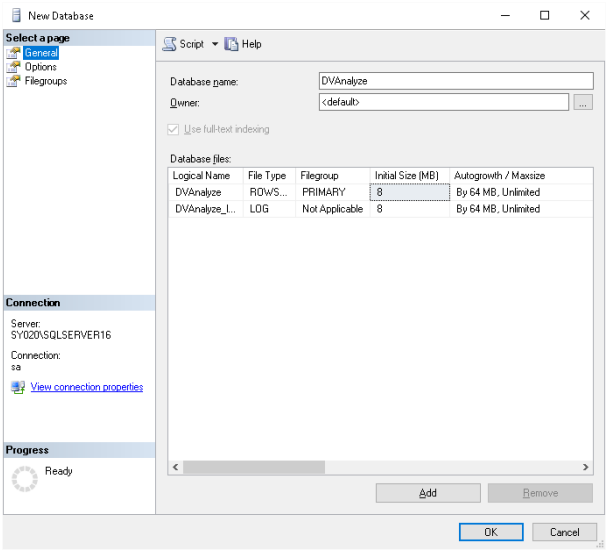
SQL Server Configuration Manager

SQL Server Configuration Manager provides the tools to manage the SQL Server instances in the Windows operating system.

This is installed with SQL Server. In some versions it is available in the Microsoft SQL Server program group by selecting **SQL Server Configuration Manager**. In other versions it needs to be launched manually. For example, to launch the Configuration Manager for SQL Server 2016, from **Start**, enter *SQLServerManager13.msc*. The *13* is the version of SQL Server so this will change based on what SQL Server version is installed.

Create a Database

Open **Microsoft SQL Server Management Studio** and connect to the SQL Server instance. In the Studio, expand the instance. Select **Databases**, right-click and select **New Database**.



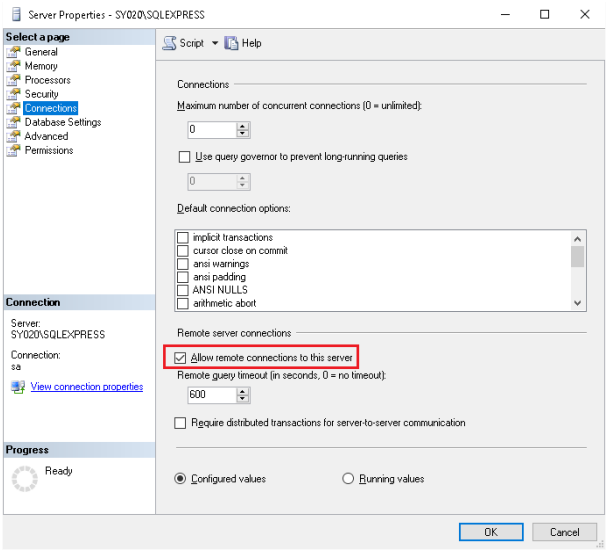
- Set the **Database name** to something meaningful.

Remote Connection

If the SQL Server is remote to where **XLReporter** is installed i.e., on a different workstation then remote connections must be enabled in SQL Server.

Open the **SQL Server Management Studio** and connect to the SQL Server instance.

- Right click the server at the top and select **Properties**.



- Under **Select a Page**, select **Connections**.
- In the **Remote server connections** section, check **Allow remote connections to this server.**

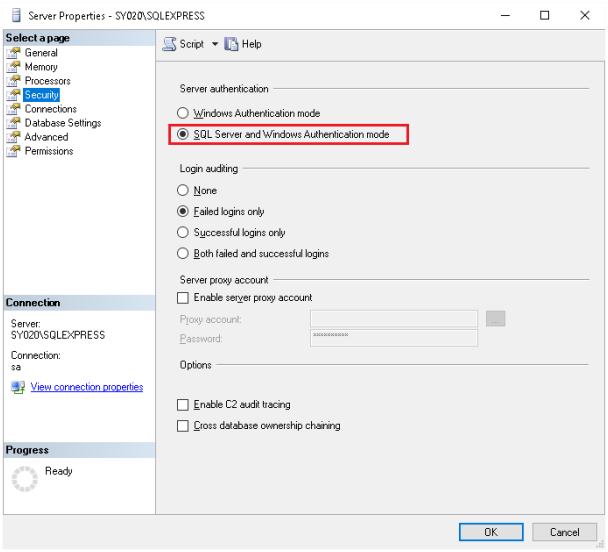
Authentication

When a XLReporter connects to SQL Server, it needs to provide authentication credentials. Two types of authentication are supported by SQL Server:

- Windows Authentication
  The credentials used will be those of the active Windows user. For a remote connection, this credential will fail unless an identical account is available on the remote system.
- SQL Server Authentication
  The credentials used will be those of users that have been created in the SQL Server database. This can be used anywhere on the network.

To use SQL Server authentication, open the **SQL Server Management Studio** and connect.

- Right click the server at the top and select **Properties**.

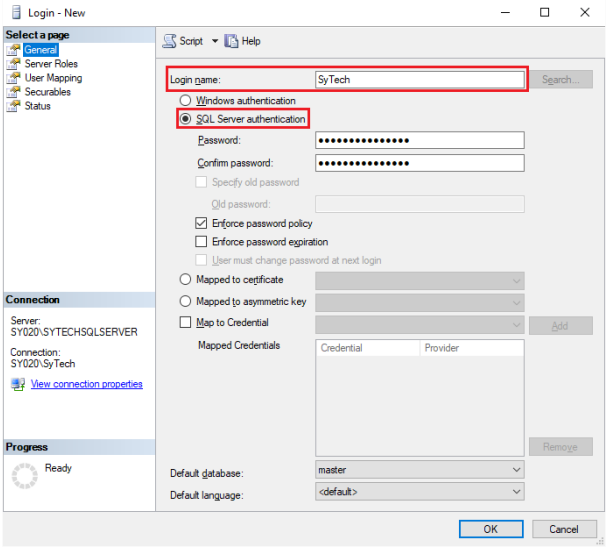


- Under **Select a Page**, select **Security**.
- In the **Server Authentication** section, select **SQL Server and Windows Authentication mode**.

Create SQL Server Users

If SQL Server authentication is the preferred way of connecting to the database, then users will be required.

Expand **Security** and right-clicking **Logins**, select **New Login**.



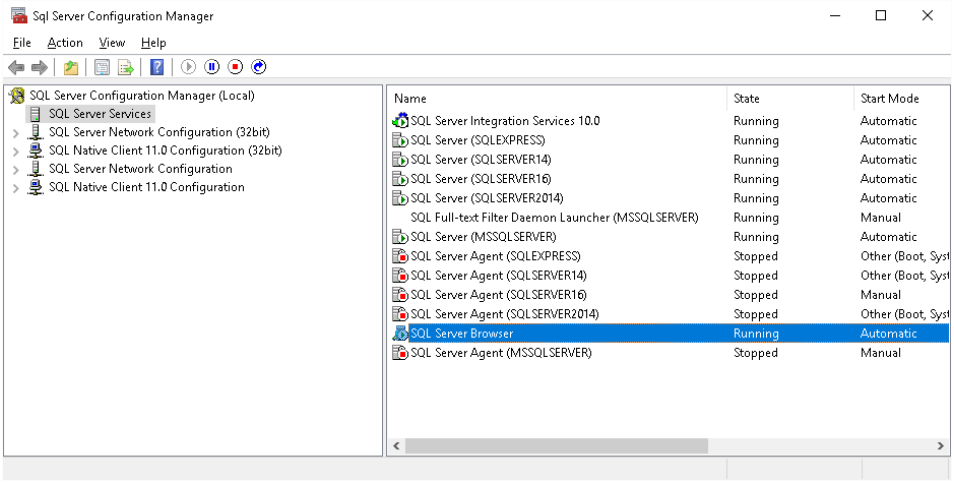
- Enter a **Login name**.

- Select **SQL Server Authentication**.

Browsing

To browse SQL Server names across the network, the **SQL Server Browser Service** needs to be enabled.

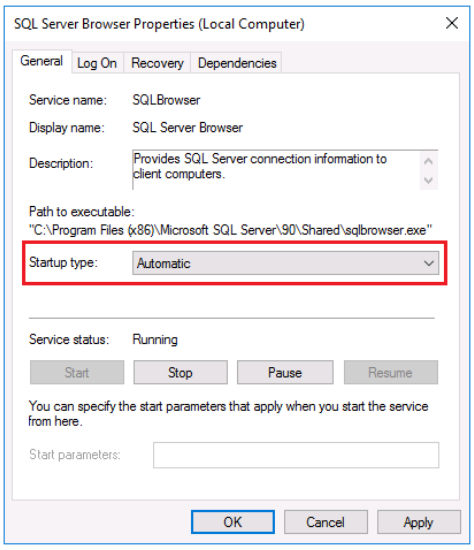
Open **SQL Server Configuration Manager**.



- Under **SQL Server Configuration Manager (Local)** select **SQL Server Services**.
- On the right, right-click **SQL Server Browser** and choose **Start** to start the service.

If Start and Stop are disabled, this means that the service itself is disabled. To enable, open the Windows **Control Panel** and open **Administrative Tools, Services**.

- Locate the **SQL Server Browser** service and double click on it to access **Properties**.



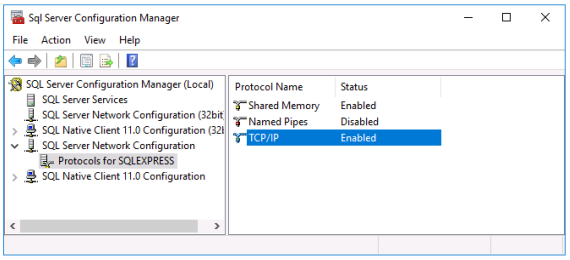
- In **Properties** set **Startup type** to *Automatic*.
- Click **OK**.

Return back to the **SQL Server Configuration Manager** and enable the **Browser** service.

Protocols

To access a SQL Server instance across the network it may be required to enable the TCP/IP protocol to make the connection.

Open **SQL Server Configuration Manager**.



- Under **SQL Server Configuration Manager (Local) expand SQL Server Network Configuration** and select **Protocols for** the instance configured.
- On the right, right-click **TCP/IP** and choose **Enable** to enable the protocol.

Windows Firewall

If the Windows Firewall is enabled on the machine where SQL Server is installed remote connections may still fail. This is because the **Port** that SQL Server is configured for is not opened in the Windows Firewall.

To identify the Port to open, open the **SQL Server Configuration Manager**.

- Under **SQL Server Configuration Manager (Local) expand SQL Server Network Configuration** and select **Protocols for** the instance configured.
- On the right, right-click **TCP/IP** and choose **Properties**.
- In **TCP/IP Properties** select the **IP Addresses** tab.
- Scroll to the **IPAll** section. If the SQL Server is running on a static port, **TCP Port** is the **Port** number to open. Otherwise **TCP Dynamic Ports** is the **Port** number to open.

Now that the **Port** is identified, it needs to be opened in the Windows Firewall.

- Open the Windows Firewall. Typically the easiest way to do this is by typing Firewall into the search bar at the bottom left of Windows.
- Click **Advanced Settings**
- Right-click **Inbound Rules** and select **New Rule**.
- For the rule type select **Port** and click **Next**.
- Apply the rule to **TCP**.
- For port, select **Specific local ports** and specify the port number identified previously. Click **Next**.
- Leave **Allow the connection** selected and click **Next**.
- Apply the rule for every network type required and click **Next**.
- Give the rule a **Name** and click **Finish**.

If a remote connection still fails, repeat the steps above opening **TCP Port** *1433* and **UDP Port** *1434*.

TLS Security

When a connector in **XLReporter** is configured to Microsoft SQL Server, by default it uses the Microsoft OLEDB Provider for SQL Server that is delivered with the Windows Operating System. However, this OLEDB provider does not support TLS 1.2 security which can be configured in later versions of SQL Server.

The telltale sign that TLS 1.2 security is enabled on the database is that when the Test Connection button is clicked, the error returned is: **SSL Security error**.

To combat this, Microsoft has released an updated driver (Microsoft OLEDB Driver for SQL Server). This can be downloaded from Microsoft's website and installed. Install either the 32 (x86) or 64 (x64) according to the System Type of the Windows Operating System (32 or 64 bit).

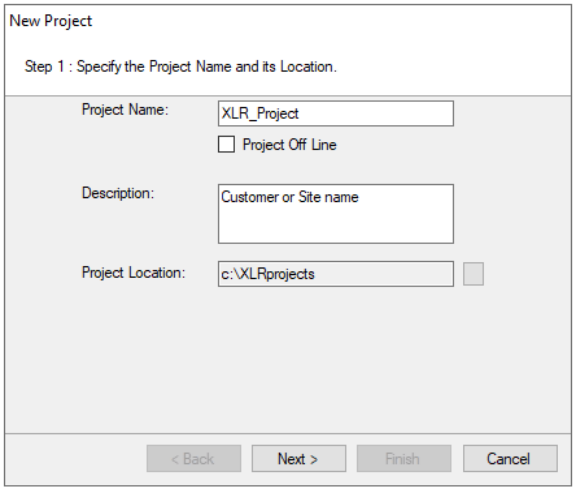
If this driver is installed on the PC, when a connector to SQL Server is configured, it will use the new driver rather than the one delivered by the Operating System.

Create a Project

From the **XLReporter Project Explorer** select **File, New** to start the **Project Wizard**. This will give step-by-step instructions on creating a project.

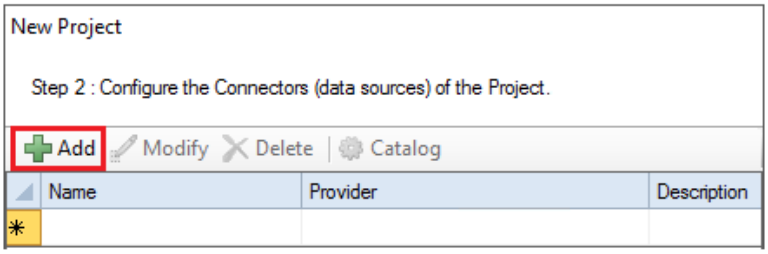
Step 1

- Enter a **Project Name** and **Description** (optional).



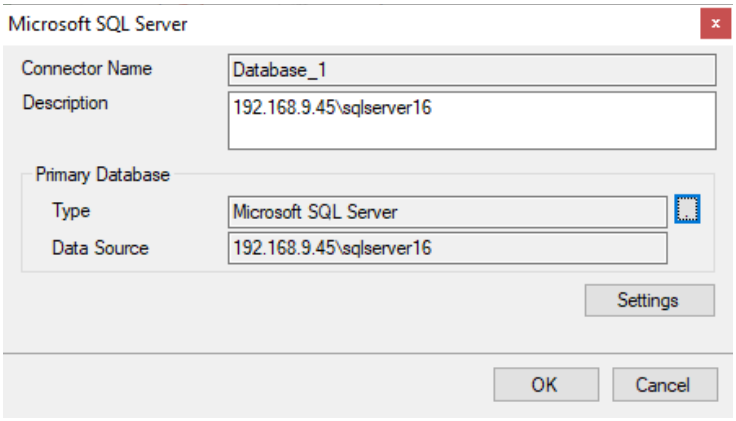
Step 2

- Configure the data connector, click **Add.**



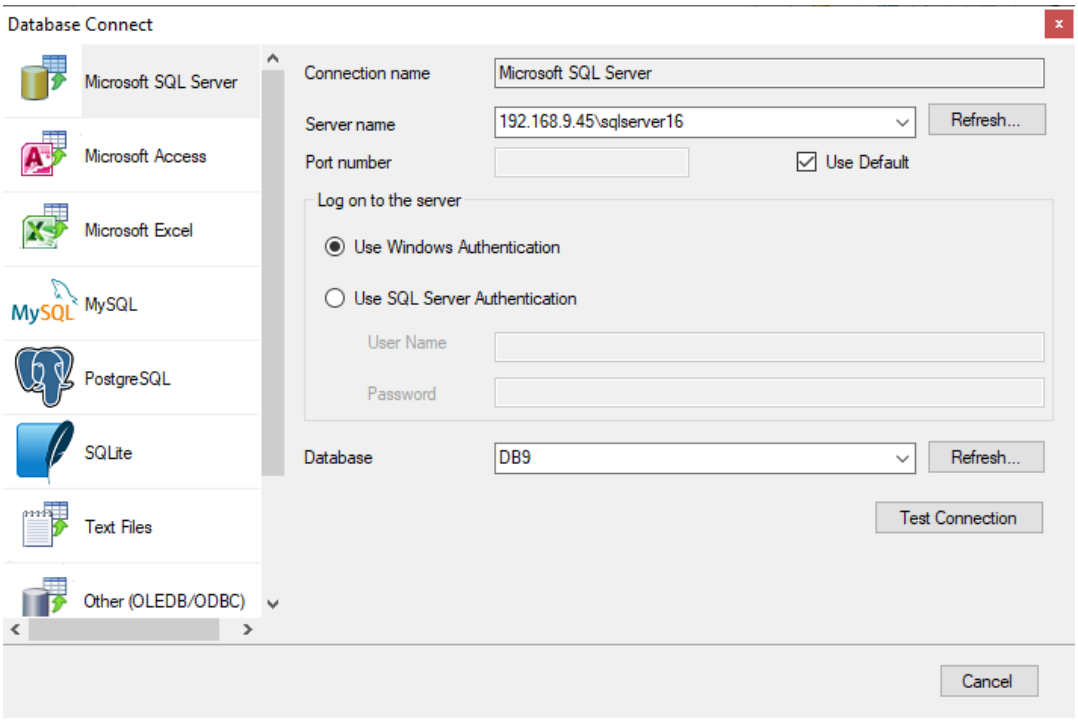
Select **Database**, **Microsoft SQL Server**.

Connector



Primary Database

This defines a connection to the database. A browse button […] is provided to define.



Browse for a **server**.

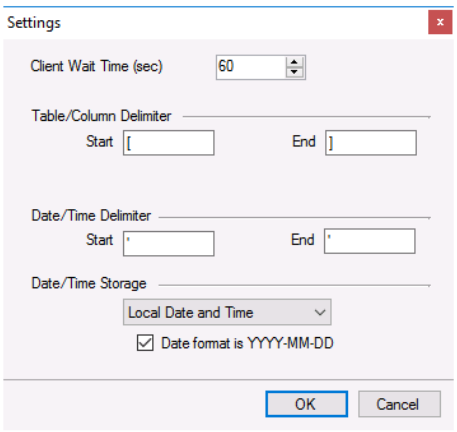
Select a **login** method.

Select a **Database.**

Click **Test Connection** to validate communication with the server.

Settings

The **Settings** button opens the **Settings** dialog that defines characteristics of the database that are used to retrieve data.

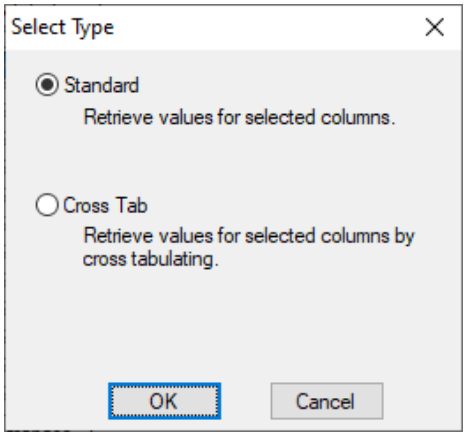


Typically, these settings are defaulted correctly based on the **Primary Server**.

Verify Data Communication

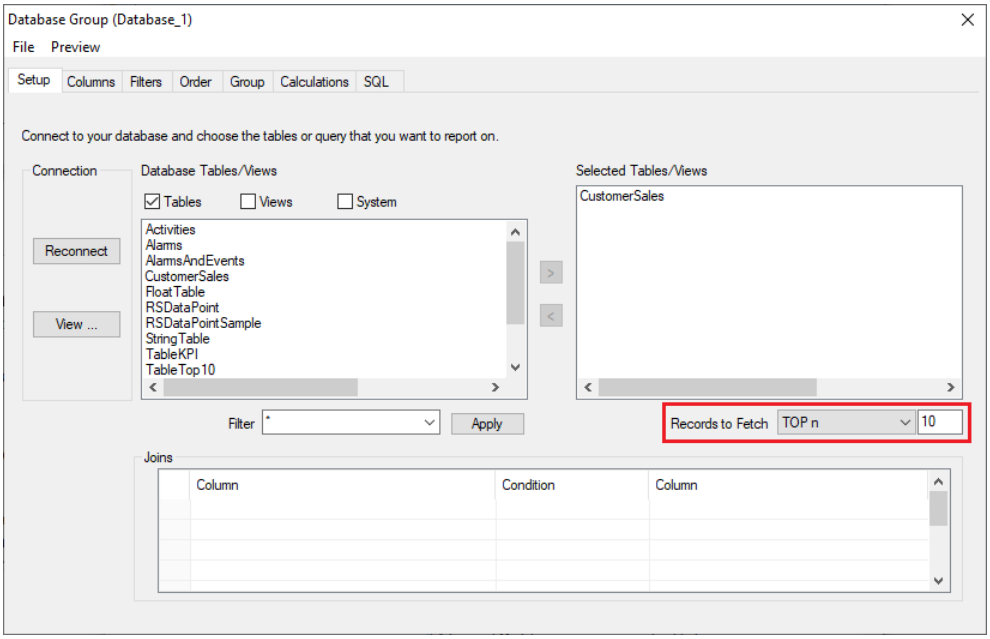
To verify communication with Microsoft SQL Server, open the **Project Explorer** and select the **Tools** tab**.** Open **Connector Groups.** Select your Microsoft SQL Server connector and then select **Add**.

For more information on database groups settings, see the document on **Relational Database Data Groups**.



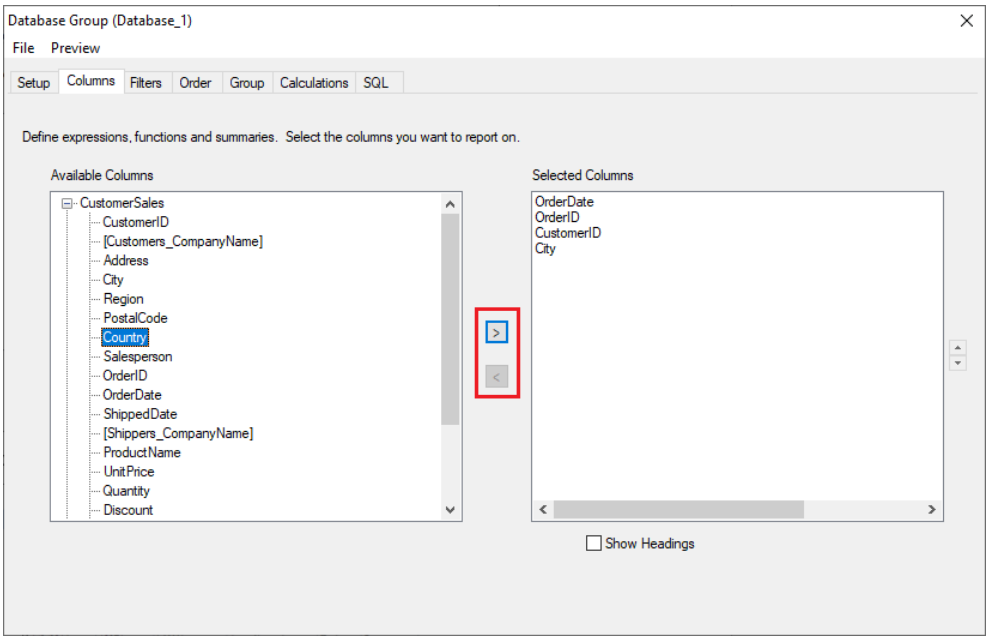
Select the **Standard** group type and click **OK**.

Under the **Setup** tab:

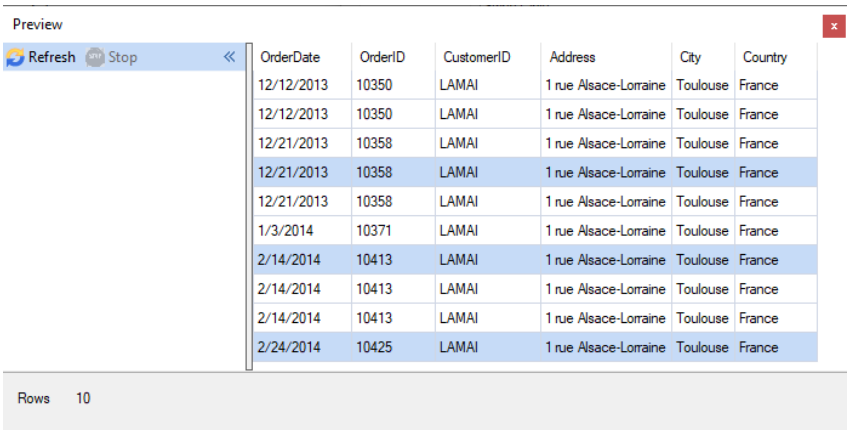


- Select a table from the database.
- Change **Records to Fetch** to *TOP n* and set the number of records to *10*. This will limit the number of records in the output to 10.

Under the **Columns** tab:



- Use the arrow buttons to move columns from the **Available Columns** list to the **Selected Columns** list.



Preview is opened from the **Preview** menu option. To preview the output of the group settings, click the **Refresh** pushbutton. The first 10 records will be displayed.