

SOFTWARE
DESIGN SOLUTIONS

**CTC Microsoft® SQL Server®
Installation & Configuration Guide**

Contents

Overview	3
Preparation	4
Installing SQL Server	5
<i>Selecting the SQL Server Features to Install.....</i>	<i>8</i>
<i>Specifying the SQL Instance (Name)</i>	<i>9</i>
<i>Selecting Who the SQL Services Run As</i>	<i>10</i>
<i>Defining Database Security.....</i>	<i>11</i>
<i>Controlling the Locations of Data and Log Files</i>	<i>14</i>
Opening Firewall Ports	17
<i>Adding InBound Rules.....</i>	<i>18</i>
<i>Adding Outbound Rules</i>	<i>20</i>
<i>Enabling TCP/IP Protocol in SQL Server</i>	<i>23</i>
Installing SQL Server Management Studio.....	24
Additional Notes and Suggestions	28
<i>Turning on Backup Compression</i>	<i>28</i>
<i>Fixing SQL Server Agent</i>	<i>29</i>
<i>Scheduling Automatic Backups.....</i>	<i>30</i>

Overview

Some products from CTC Software ("CTC"), such as BIM Data Suite, may require you to have an instance of Microsoft SQL Server installed and configured in order to store the data that is captured from your design models.

If you already have Microsoft SQL Server (version 2008 R2 or later, NOT the free SQL Server Express edition) installed and configured, you do not need to read this document.

While CTC **strongly** recommends anyone purchasing these products have a Microsoft SQL Server Database Administrator (DBA) available, we recognize that for scenarios such as setting up a trial environment that sometimes a DBA may not be available. Therefore, this guide has been provided to assist in getting a small trial environment up and running for use with CTC products.

IMPORTANT: This guide is **NOT** a comprehensive source of information about, or complete reference for, Microsoft SQL Server. It is intended to outline the minimal procedures needed to install and configure Microsoft SQL Server specifically for use with CTC products.

For example, in the interests of simplicity default system accounts will be used instead of defining domain user accounts which the services should run as, even though using domain user accounts may normally be the most secure practice. Further, enabling SQL Authentication will be turned on, which allows defining user accounts in the database. While that may not normally be the most secure approach, it makes installation of the CTC services noticeably simpler.

Multiple copies ("instances") of SQL server can be installed on the same computer. Each instance can have different settings, store databases to different folders, etc. and is differentiated from the other instances by having a unique name. In the interests of simplicity, this install will presume no other instances of SQL server are installed on this computer, and that the default instance name will be used.

All other aspects of using SQL Server, such as scheduling backups, restoring from a backup, performance tuning and custom reporting are the responsibility of a DBA or other staff.

IMPORTANT: CTC is unable to provide report writing services or technical support for Microsoft SQL Server beyond the documentation that is provided with the software from CTC.

IMPORTANT: **SQL Server Express is not supported by Microsoft for production environments.** This is primarily due to limitations imposed by Microsoft on this free version of the SQL Server database engine. While SQL Server Express will physically function with software from CTC, and may be acceptable for a small trial environment, long-term production use of SQL Server Express is not supported.

Preparation

Microsoft SQL Server version **2008 R2 or later** is required.

Please consult Microsoft's documentation for the version of SQL Server being installed to ensure the prerequisites (such as the minimum Windows version required) are met before you begin the installation.

CTC strongly recommends installing SQL Server on a separate, dedicated physical computer which runs a server version of Windows (not a workstation version of Windows, such as Windows 10).

While installing SQL server on a virtual machine will function, the performance will almost certainly be significantly slower than if it were installed on an equivalent physical server.

In the example below, we will be installing Microsoft SQL Server 2016 on Windows Server 2012.

CTC also recommends having 3 separate disk drives (or partitions) defined on the server:

1. Operating System
2. Data
3. Logs

IMPORTANT: Neither database data nor log files should be stored on the operating system drive (e.g. drive C:).

This is because databases and log files can grow in size, sometimes quickly, and very bad things can happen to the entire server if the operating system drive gets full.

IMPORTANT: Starting with SQL Server 2016, **SQL Server Management Tools** are no longer installed from the main installer for SQL server itself. These tools are needed to manipulate the entire SQL Database system, such as for creating databases, changing databases, defining security, performing database backups and restores, scheduling database tasks etc. **These tools will be needed to configure and permission databases provided by CTC.** While it is not necessary to install them on the server, it can be very convenient to have them available on the server. They can also be installed on any workstation, which would be desirable for administering the databases without having to login to the server.

As of this writing, Microsoft makes these tools available from this URL:

<https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms>

Installing these tools will be explored in a later section of this document.

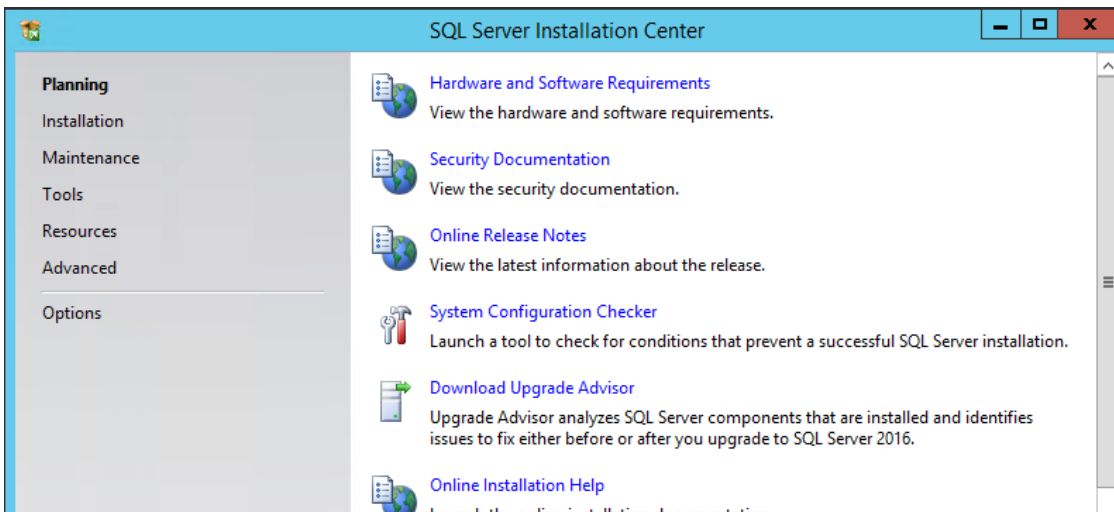
Once the server operating system has been configured, the installation can begin.

Installing SQL Server

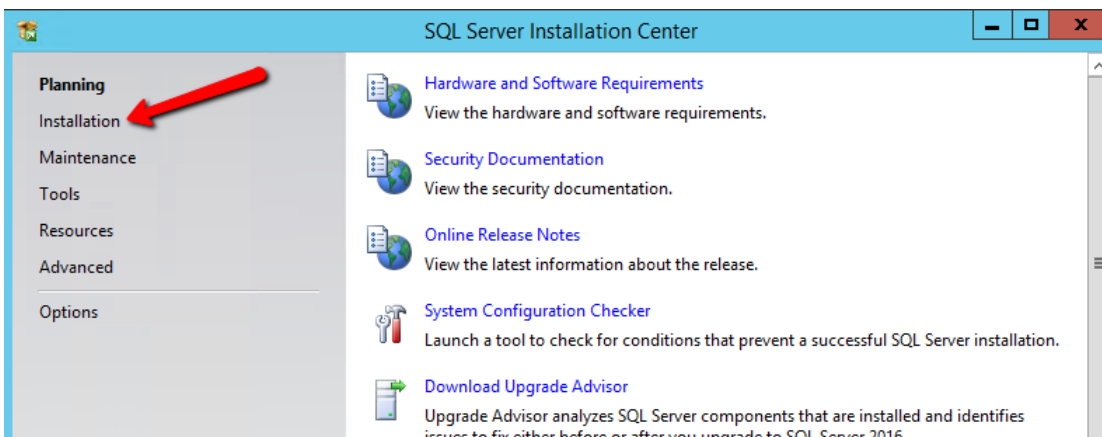
Begin by running the setup for the SQL Server installation.

IMPORTANT: The images shown below are from a SQL Server 2016 Standard edition installation. The installation screens for other flavors and versions of SQL server should be extremely similar, and should be easy to follow along with from this example.

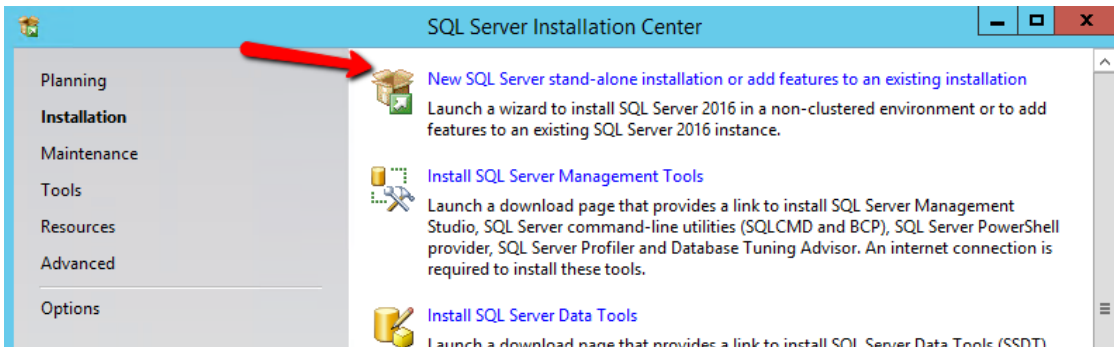
The default selection in the navigation pane on the left is the first choice, *Planning*. It would certainly not be a bad idea to peruse some of the planning choices on the right.



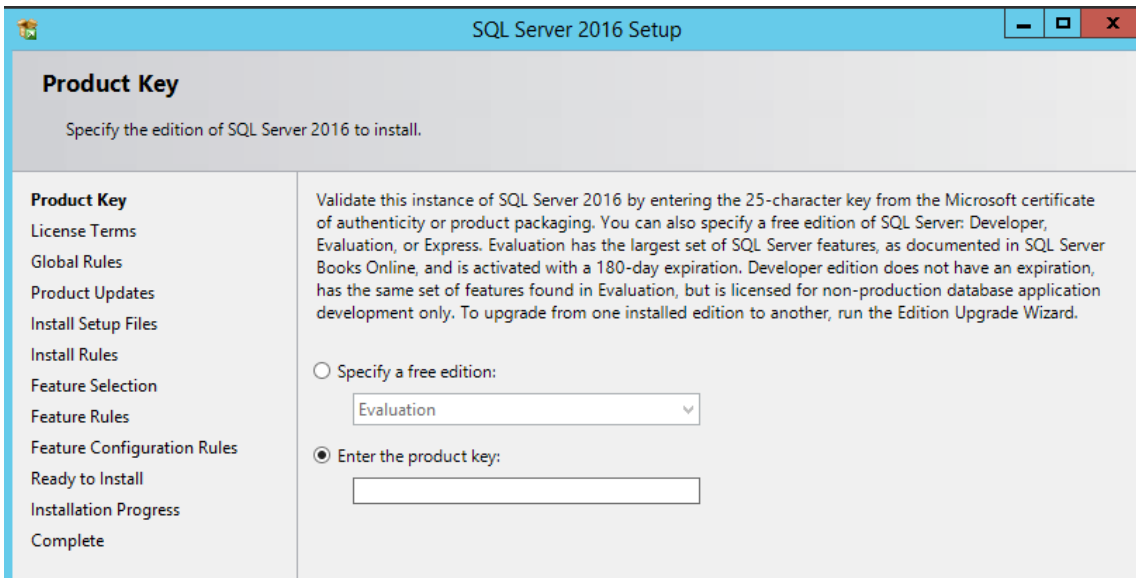
Once you're ready to begin the installation, in the navigation column on the left, select *Installation*:



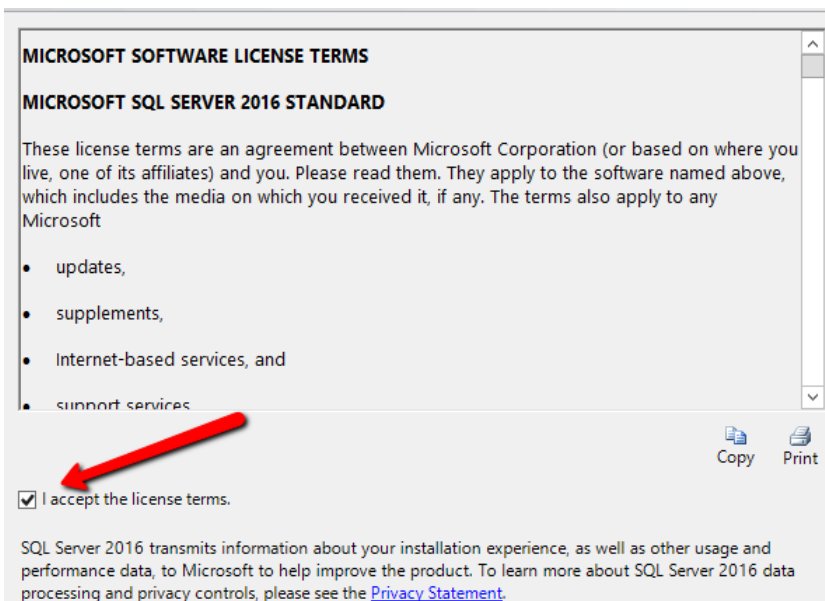
On the resulting pane on the right side, select *New SQL Server stand-alone installation or add features to an existing installation*:



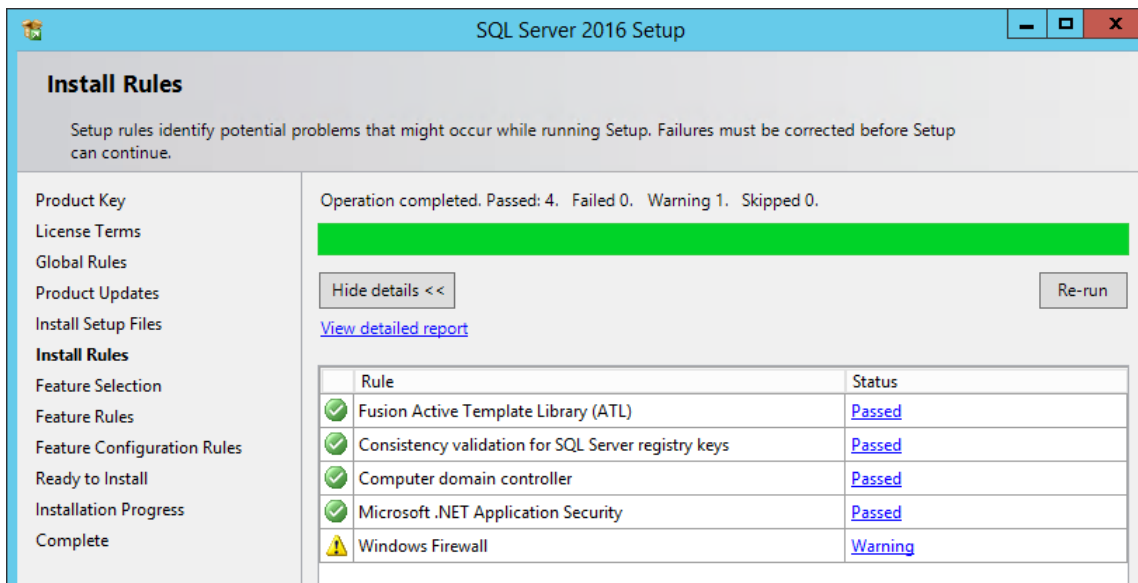
On the next screen, enter your product key, then click the Next button at the bottom of the window:



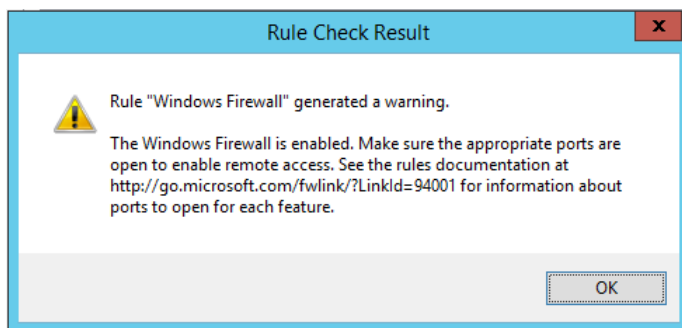
On the next screen, when ready check the checkbox to accept the agreement, then click the Next button.



Some pre-checks will be run, and a quick report will show the status:



Clicking on the warning link for the Windows Firewall shows this message:

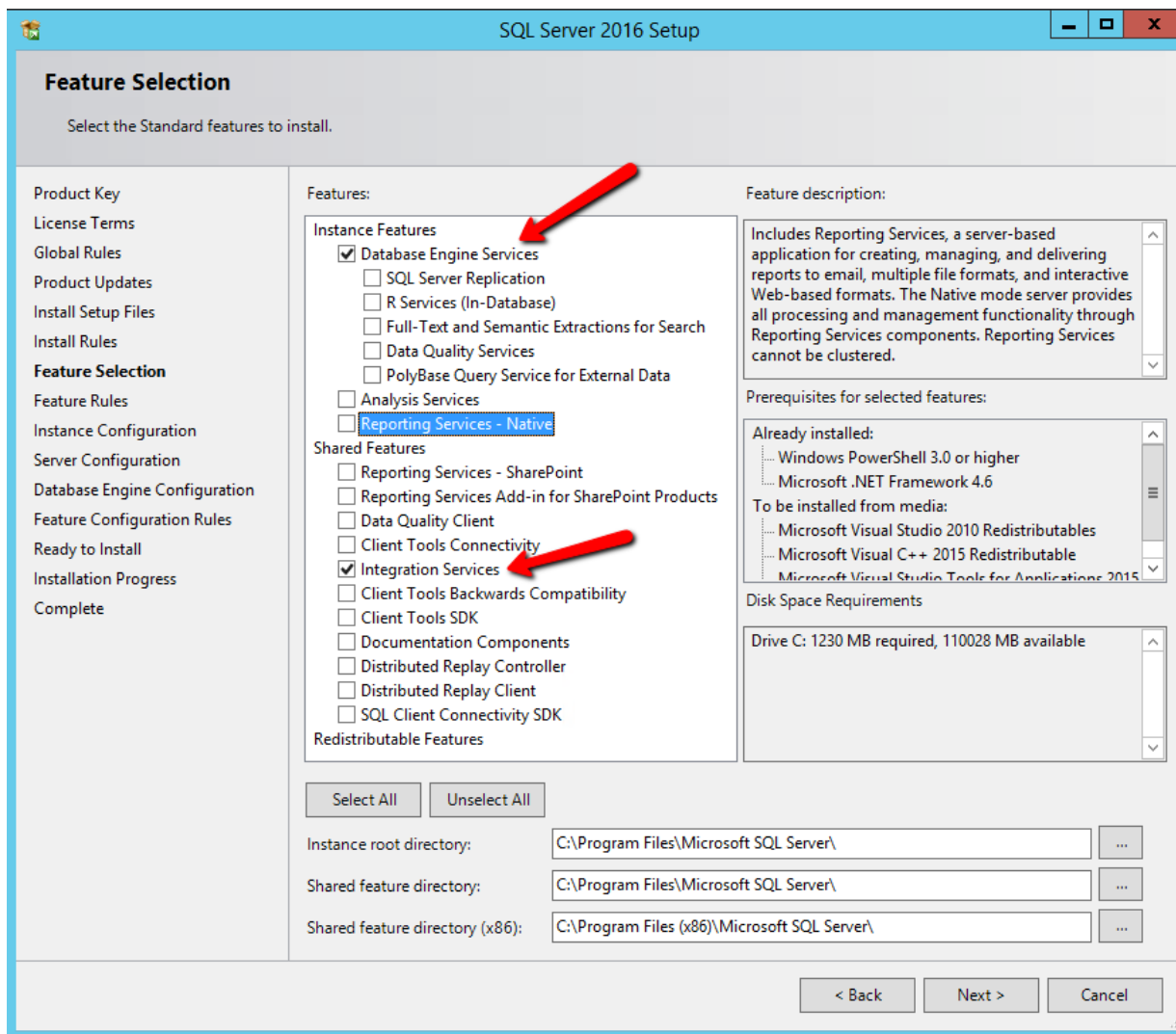


We will deal with firewall ports later in this process. For now, click the Next button at the bottom of the screen.

On the next screen, select the components to install. The minimal useful components are:

Database Engine Services
Integration Services

Selecting the SQL Server Features to Install



NOTE: *Reporting Services - Native* would be desirable if anyone plans to use the reporting functions that are built into SQL server for generating reports. CTC ships sample reports which use Microsoft's *PowerBI* software, but there are many other tools available for generating reports from data in a SQL Server database, including these which can be installed with SQL server itself.

For this example, in the interests of simplicity Reporting Services will NOT be selected for installation.

The directory folders seen at the bottom of the screen define where the server code should be placed for this instance of SQL server. Under normal circumstances the default folders will be fine, so we will leave them alone for this installation.

Click the Next button.

The following screen defines the instance name to use.

Specifying the SQL Instance (Name)

SQL Server 2016 Setup

Instance Configuration

Specify the name and instance ID for the instance of SQL Server. Instance ID becomes part of the installation path.

Product Key
License Terms
Global Rules
Product Updates
Install Setup Files
Install Rules
Feature Selection
Feature Rules
Instance Configuration
Server Configuration
Database Engine Configuration
Feature Configuration Rules
Ready to Install
Installation Progress
Complete

☒ Default instance
☐ Named instance: MSSQLSERVER

Instance ID: MSSQLSERVER

SQL Server directory: C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER

Installed instances:

Instance Name	Instance ID	Features	Edition	Version
---------------	-------------	----------	---------	---------

< Back Next > Cancel

IMPORTANT: The instance name chosen will affect the connection string that is used by the CTC services to control with which SQL server they communicate. The simplest connection string will only use the server computer's name, but to do that requires using the default instance name for versions of SQL server other than SQL Server Express. If either you are installing SQL Server Express or you choose a custom name, the instance name (SQLEXPRESS or your custom name, respectively) must be included in the connection string.

The list in the bottom half of the window will show any other instances of SQL Server that have already been installed on this computer. All instances must have a unique name. If an instance is already installed with either the default name (MSSQLSERVER) or the name you wish to use for this instance, you must specify the *Named instance* option and choose a different name for the new instance being installed.

In the interests of simplicity, just keep the *Default instance* choice selected and click the Next button at the bottom of the window.

If for any reason you can't use the default instance, then make a note of the instance name you chose. If you can use the default instance but are installing SQL Server Express, make a note of the SQLEXPRESS instance name.

If you had to make a note of the instance name used, you will need to know that name in the future when defining the connection string to use in the BIM Data Suite Servers configurations.

Configuring connection strings is explained in further detail in the CTC BIM Data Suite Servers Installation and Configuration guide.

Selecting Who the SQL Services Run As

The next screen controls who each service that will be installed runs as.

Server Configuration
Specify the service accounts and collation configuration.

Product Key
License Terms
Global Rules
Product Updates
Install Setup Files
Install Rules
Feature Selection
Feature Rules
Instance Configuration
Server Configuration
Database Engine Configuration
Feature Configuration Rules
Ready to Install
Installation Progress
Complete

Service Accounts Collation

Microsoft recommends that you use a separate account for each SQL Server service.

Service	Account Name	Passw...	Startup Type
SQL Server Agent	NT Service\SQLSERVERAGENT		Manual
SQL Server Database Engine	NT Service\MSSQLSERVER		Automatic
SQL Server Integration Services 13.0	NT Service\MsDtsServer130		Automatic
SQL Server Browser	NT AUTHORITY\LOCAL SERVICE		Disabled

☐ Grant Perform Volume Maintenance Task privilege to SQL Server Database Engine Service

This privilege enables instant file initialization by avoiding zeroing of data pages. This may lead to information disclosure by allowing deleted content to be accessed.

[Click here for details](#)

< Back Next > Cancel

For this installation we will keep things simple and use the default accounts, so just click the Next button.

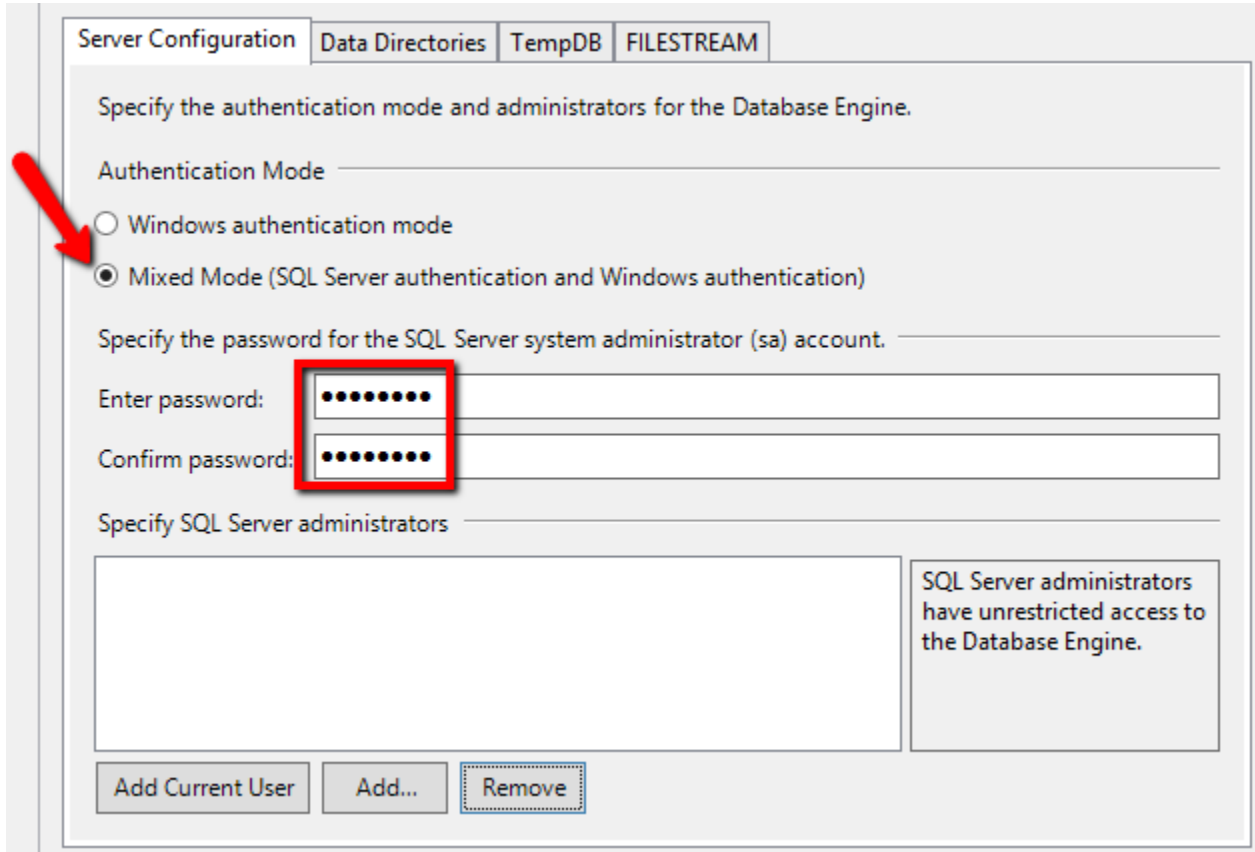
On the next screen we need to change items on the *Server Configuration* and *Data Directories* tabs.

Defining Database Security

On the *Server Configuration* tab, change the selection to Mixed Mode and enter a password for the built-in system administrator account.

This account is defined within the database instance itself, and has the user name of: sa

It is an account that can always be used to gain full administrative access to the database.



Server Configuration | Data Directories | TempDB | FILESTREAM

Specify the authentication mode and administrators for the Database Engine.

Authentication Mode

☐ Windows authentication mode

☒ Mixed Mode (SQL Server authentication and Windows authentication)

Specify the password for the SQL Server system administrator (sa) account.

Enter password: [password field]

Confirm password: [password field]

Specify SQL Server administrators

SQL Server administrators have unrestricted access to the Database Engine.

Add Current User | Add... | Remove

Turning on Mixed Mode allows defining user accounts other than sa to be defined in the database, which will make working with CTC software simpler. That process will be discussed in the installation and configuration guide for the CTC software, such as the BIM Data Suite Servers Installation and Configuration guide.

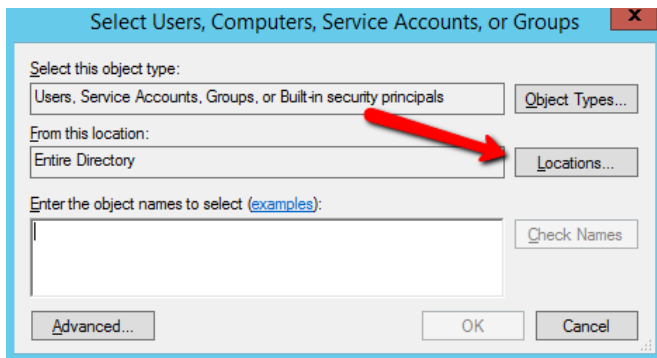
Also on this tab, it may not be a bad idea to add at least a few Windows users or groups to the list of “Specify SQL Server administrators” section. Any domain or local computer users or groups can be added to this list.

IMPORTANT: Be very careful who is added to this list. Anyone added to this list can do anything in this SQL server instance, including changing or deleting data, or even deleting entire databases.

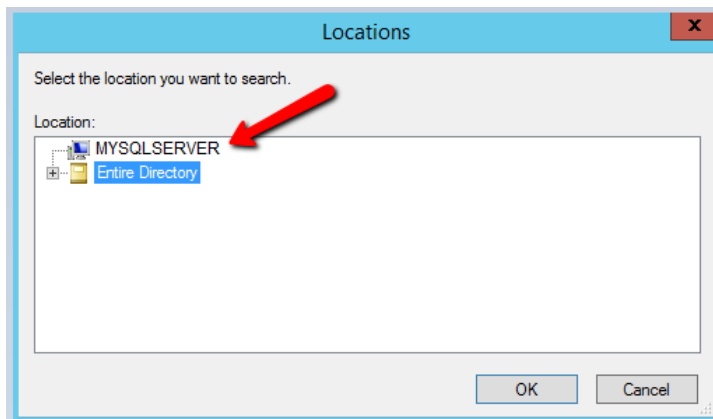
One approach is to specify that anyone who is an administrator on the server also has administrative access within SQL server, if that makes sense in your environment.

To do this, first click on the Add... button below the list.

In the resulting dialog, click the Locations... button:



Select the local computer's name, at the top of the list, then click OK:



Back on the previous screen, type into the list near the bottom:

Administrators

Then click the OK button. The *Server Configuration* tab should now look like this:

Server Configuration | Data Directories | TempDB | FILESTREAM

Specify the authentication mode and administrators for the Database Engine.

Authentication Mode

☐ Windows authentication mode

☒ Mixed Mode (SQL Server authentication and Windows authentication)

Specify the password for the SQL Server system administrator (sa) account.

Enter password:

Confirm password:

Specify SQL Server administrators

BUILTIN\Administrators (Administrators)

SQL Server administrators have unrestricted access to the Database Engine.

Add Current User | Add... | Remove

You may, of course, add other users or groups to this list as appropriate for your environment.

Controlling the Locations of Data and Log Files

Switch to the *Data Directories* tab. On this tab you specify the disc drives and folders to which things like data files or log files should be written.

IMPORTANT: Neither database data nor log files should be stored on the operating system drive (e.g. drive C:).

This is because databases and log files can grow in size, sometimes quickly, and very bad things can happen to the entire server if the operating system drive gets full.

In this example we will be storing databases on the D: drive and Log files on the F: drive.

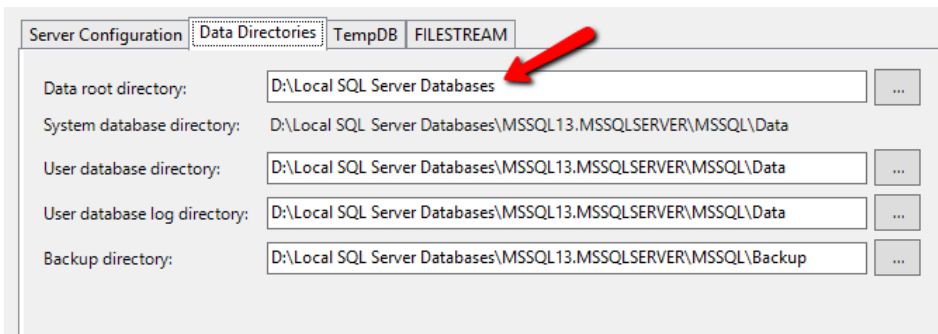
First, create the folders for data files and log files. In this example, we have created the folders:

D:\Local SQL Server Databases

F:\Local SQL Server Logs

IMPORTANT: If on the *Service Accounts* tab of the “Server Configuration” screen above you changed which user accounts the services will run-as, you must grant those user accounts read and write permissions on the folders to be used for storing data and log files.

When ONLY the “Data root directory” value is changed, the other fields update automatically:



The “MSSQL13.MSSQLSERVER” portion of the name indicates the version of SQL server being used, followed by the SQL instance name.

You can, of course, change any of these values to point to other folders, as appropriate for your environment.

Modify the “User database log directory” value by changing the first part of the path to the folder where the log files will be stored. In this example:

Server Configuration | **Data Directories** | TempDB | FILESTREAM

Data root directory: D:\Local SQL Server Databases

System database directory: D:\Local SQL Server Databases\MSSQL13.MSSQLSERVER\MSSQL\Data

User database directory: D:\Local SQL Server Databases\MSSQL13.MSSQLSERVER\MSSQL\Data

User database log directory: F:\Local SQL Server Logs\MSSQL13.MSSQLSERVER\MSSQL\Data

Backup directory: D:\Local SQL Server Databases\MSSQL13.MSSQLSERVER\MSSQL\Backup

Once all of the directory paths are configured correctly, click the *Next* button at the bottom of the screen.

The *Ready to Install* dialog appears, which lists a summary of the things to do. Just click the *Install* button at the bottom of the screen.

SQL Server 2016 Setup

Ready to Install

Verify the SQL Server 2016 features to be installed.

Product Key
License Terms
Global Rules
Product Updates
Install Setup Files
Install Rules
Feature Selection
Feature Rules
Instance Configuration
Server Configuration
Database Engine Configuration
Feature Configuration Rules
Ready to Install
Installation Progress
Complete

Ready to install SQL Server 2016:

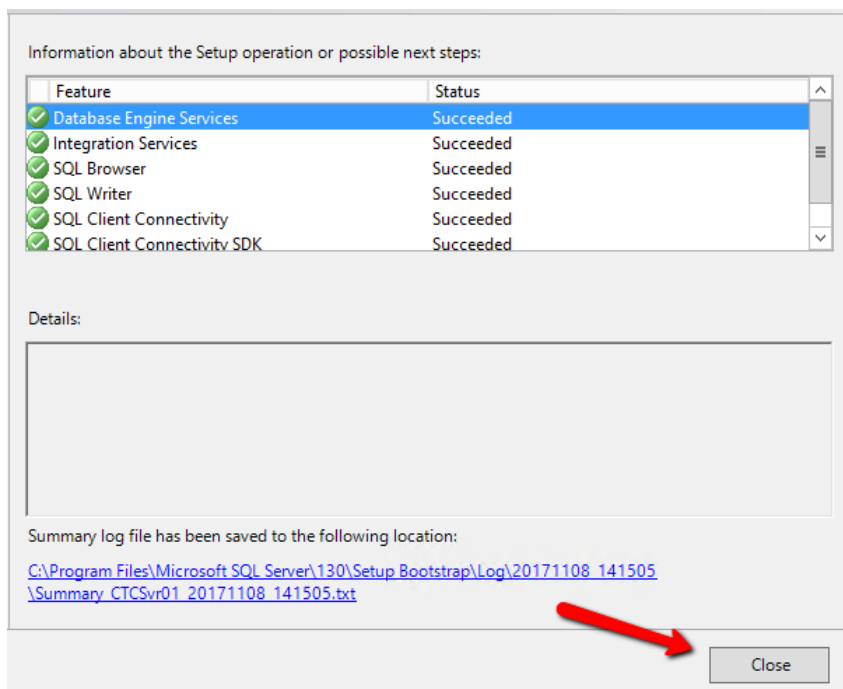
- Summary
 - Edition: Standard
 - Action: Install (Product Update)
 - Prerequisites
 - Already installed:
 - Windows PowerShell 3.0 or higher
 - Microsoft .NET Framework 4.6
 - To be installed from media:
 - Microsoft Visual Studio 2010 Redistributables
 - Microsoft Visual C++ 2015 Redistributable
 - Microsoft Visual Studio Tools for Applications 2015
 - General Configuration
 - Features
 - Database Engine Services
 - Integration Services
 - Instance configuration
 - Instance Name: MSSQLSERVER
 - Instance ID: MSSQLSERVER

Configuration file path:
C:\Program Files\Microsoft SQL Server\130\Setup Bootstrap\Log\20171108_141505\ConfigurationFile.ini

< Back Install Cancel

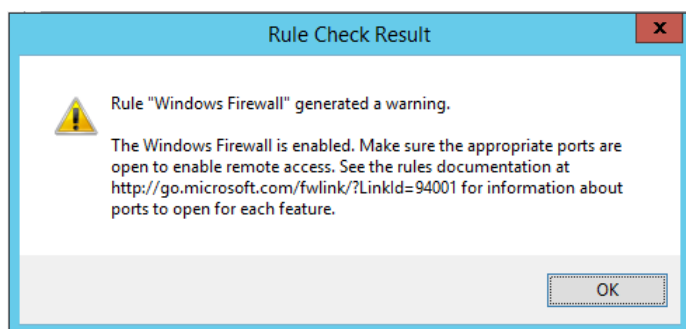
The installation process will begin. This may take several minutes.

Once complete, click the *Close* button at the bottom of the screen.



The core SQL Server engine has been installed, but there is still more to do because if you try to connect to the SQL server (e.g. using SQL Server Management Studio, discussed below), the connection will fail because firewall ports still need to be opened.

This issue was alluded to before when the prerequisites check showed the firewall warning:



Resolving this will be explained in the next section.

Opening Firewall Ports

As of this writing, the official documentation for configuring Windows firewall to allow SQL Server access can be found here:

<http://go.microsoft.com/fwlink/?LinkId=94001>

This documentation contains information for:

- Configure a Windows Firewall for Database Engine Access
- Configure the Windows Firewall to Allow Analysis Services Access
- Configure a Firewall for Report Server Access

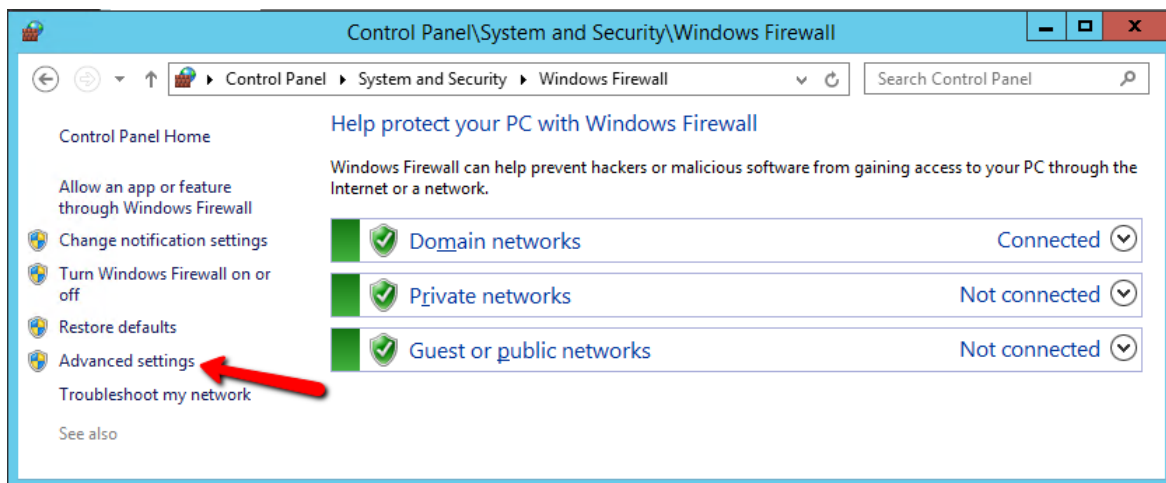
For this example we will only focus on the first item in the list. If you installed either Analysis Services or Reporting Services, please see the documentation from Microsoft about the other ports that need to be opened to use those features.

For this example, we will open the following ports:

TCP: 135, 1433, 1434, 2383, 4022

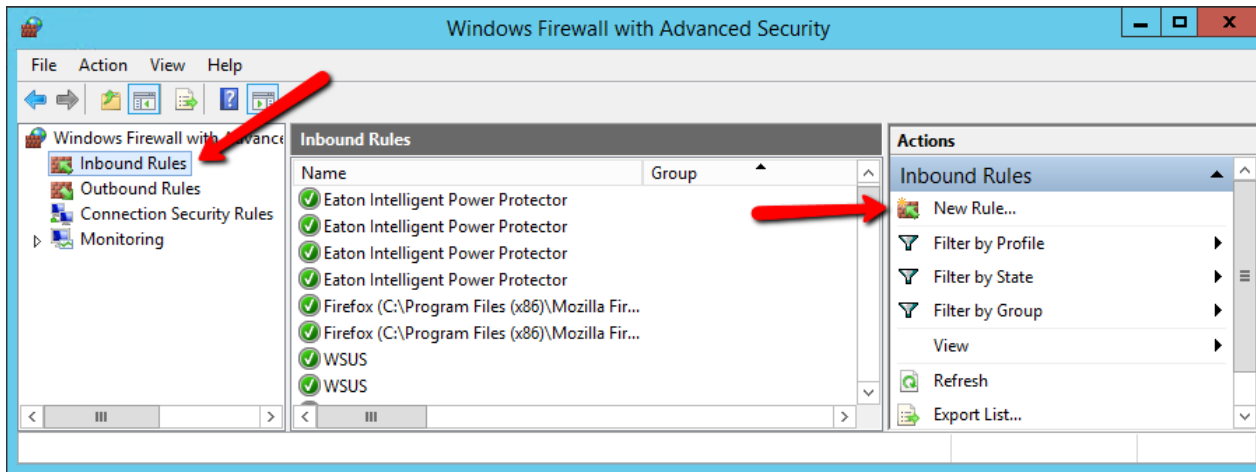
UDP: 1434

Open Control Panel, select System and Security, then Windows Firewall. Select “Advanced settings” from the list of choices on the left:

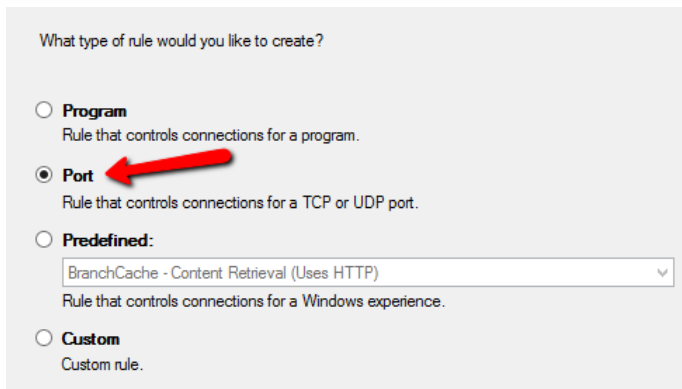


On the next screen, select *Inbound Rules* on the left, then *New Rule...* on the right:

Adding InBound Rules

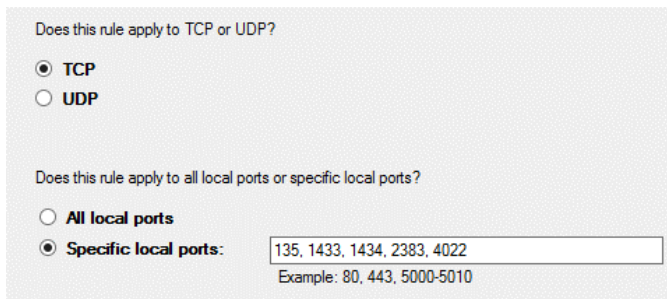


On the resulting *New Inbound Rule Wizard* dialog, select the *Port* choice, then click the Next button:



On the next screen, ensure *TCP* is selected as well as *Specific local ports*. Then enter the following in the specific local ports field:

135, 1433, 1434, 2383, 4022



Then click the Next button at the bottom of the screen.

On the next dialog, ensure *Allow the connection* is selected, and click the Next button:

What action should be taken when a connection matches the specified conditions?

☒ **Allow the connection**
This includes connections that are protected with IPsec as well as those are not.

☐ **Allow the connection if it is secure**
This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.

☐ **Block the connection**

On the next dialog, ensure *Domain* and *Private* are selected, then click the Next button:

When does this rule apply?

☒ **Domain**
Applies when a computer is connected to its corporate domain.

☒ **Private**
Applies when a computer is connected to a private network location, such as a home or work place.

☐ **Public**
Applies when a computer is connected to a public network location.

On the last dialog, provide at least a name for the rule. For example:

Open Inbound TCP Ports for SQL Server

Name:

Description (optional):

Then click the *Finish* button.

The new rule will appear on the list of Inbound rules.

We need to add another Inbound rule to open the UDP port, so click the *New Rule...* choice from the list on the right and repeat this process but with the following changes:

UDP Port 1434

Rule name: Open Inbound UDP Ports for SQL Server

Does this rule apply to TCP or UDP?

☐ TCP

☒ UDP

Does this rule apply to all local ports or specific local ports?

☐ All local ports

☒ Specific local ports:

Example: 80, 443, 5000-5010

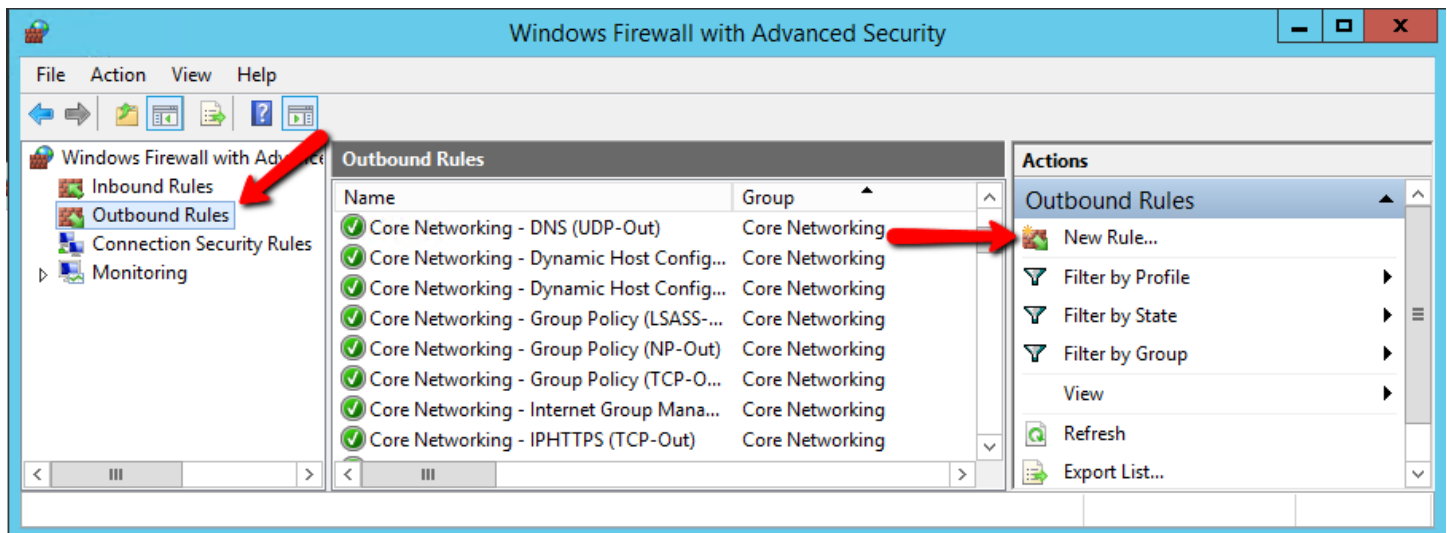
Name:

Description (optional):

Adding Outbound Rules

Creating the outbound rules is an extremely similar process as that used to create the inbound rules.

Select the *Outbound Rules* option on the left, then click the *New Rule...* option on the right.



On the next dialog select *Port* and click the *Next* button.

What type of rule would you like to create?

☐ **Program**
Rule that controls connections for a program.

☒ **Port**
Rule that controls connections for a TCP or UDP port.

☐ **Predefined:**
BranchCache - Content Retrieval (Uses HTTP)
Rule that controls connections for a Windows experience.

☐ **Custom**
Custom rule.

On the next dialog, select *TCP* and *Specific remote ports*, then provide the values: 135, 1433, 1434, 2383, 4022

Does this rule apply to TCP or UDP?

☒ **TCP**
☐ **UDP**

Does this rule apply to all remote ports or specific remote ports?

☐ **All remote ports**
☒ **Specific remote ports:** 135, 1433, 1434, 2383, 4022
Example: 80, 443, 5000-5010

Then click the Next button.

On the next dialog, change the choice to *Allow the connection* then click the *Next* button:

What action should be taken when a connection matches the specified conditions?

☒ **Allow the connection**
This includes connections that are protected with IPsec as well as those are not.

☐ **Allow the connection if it is secure**
This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.
[Customize...](#)

☐ **Block the connection**

On the next dialog, select *Domain* and *Private*, then click the *Next* button:

When does this rule apply?

☒ **Domain**
Applies when a computer is connected to its corporate domain.

☒ **Private**
Applies when a computer is connected to a private network location, such as a home or work place.

☐ **Public**
Applies when a computer is connected to a public network location.

Provide a name for the new rule, such as:

Open Outbound TCP Ports for SQL Server

Name:

Description (optional):

Then click the *Finish* button. The new rule should appear on the list of rules.

We need to add another outbound rule to open the UDP port, so click the *New Rule...* choice from the list on the right and repeat this process but with the following changes:

UDP Port 1434

Rule name: Open Outbound UDP Ports for SQL Server

Does this rule apply to TCP or UDP?

☐ TCP

☒ UDP

Does this rule apply to all local ports or specific local ports?

☐ All local ports

☒ Specific local ports:

Example: 80, 443, 5000-5010

Name:

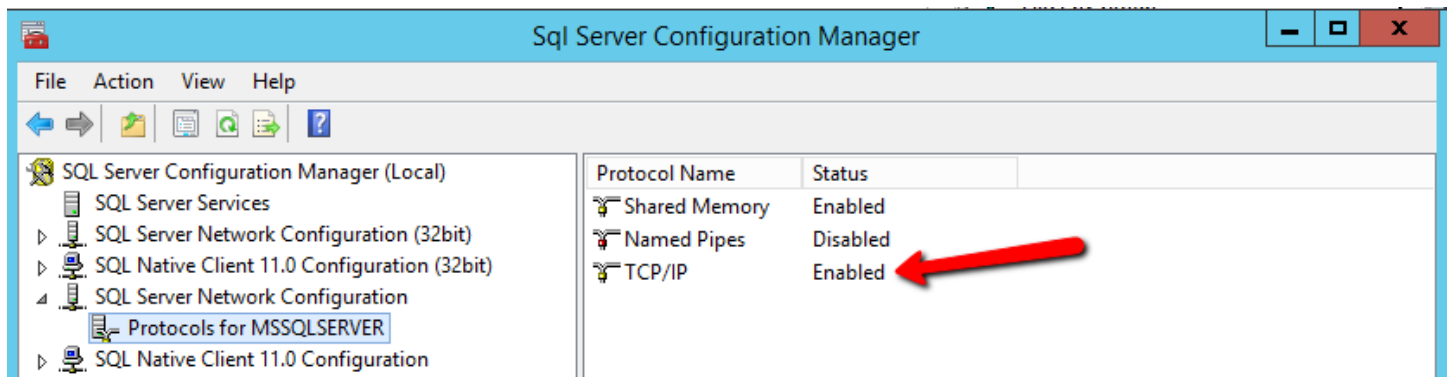
Description (optional):

Enabling TCP/IP Protocol in SQL Server

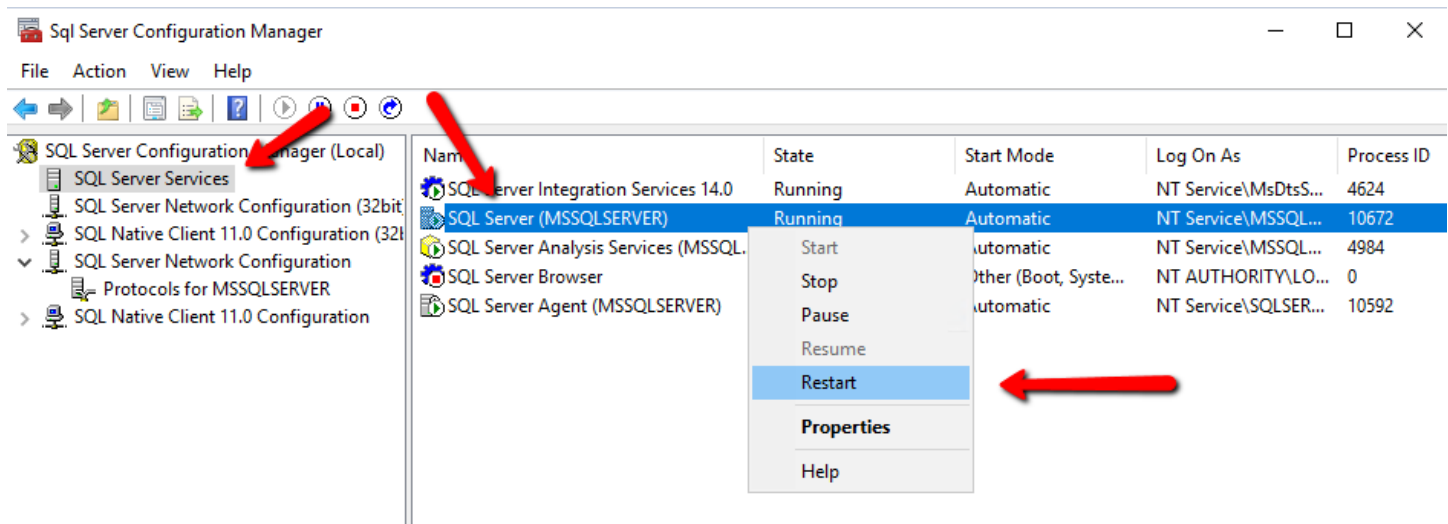
In some installations of SQL Server, connecting to the Database Engine from another computer is not enabled unless an administrator uses Configuration Manager to enable it. To enable connections from another computer:

Open SQL Server Configuration Manager.

Using Configuration Manager, in the left pane expand **SQL Server Network Configuration**, and then select the instance of SQL Server that you want to connect to. The right-pane lists the connection protocols available. Shared Memory is normally enabled. It can only be used from the same computer, so most installations leave Shared Memory enabled. To connect to SQL Server from another computer you will normally use TCP/IP. If TCP/IP is not enabled, right-click **TCP/IP**, and then click **Enable**.



If you changed the enabled setting for any protocol you must restart the Database Engine. In the left pane select **SQL Server Services**. In the right-pane, right-click the instance of the Database Engine, and then click **Restart**.



Installing SQL Server Management Studio

Starting with SQL Server 2016, SQL Server Management Tools are no longer installed from the main installer for SQL server itself.

These tools are needed to manipulate the entire SQL Database system, such as for creating databases, changing databases, defining security, performing database backups and restores, scheduling database tasks etc.

These tools will be needed to configure and permission databases provided by CTC. While it is not necessary to install them on the server, it can be very convenient to have them also available on the server. **They can be installed on any workstation**, which would be desirable for administering the databases without having to login to the server.

As of this writing, Microsoft makes these tools available from this URL:

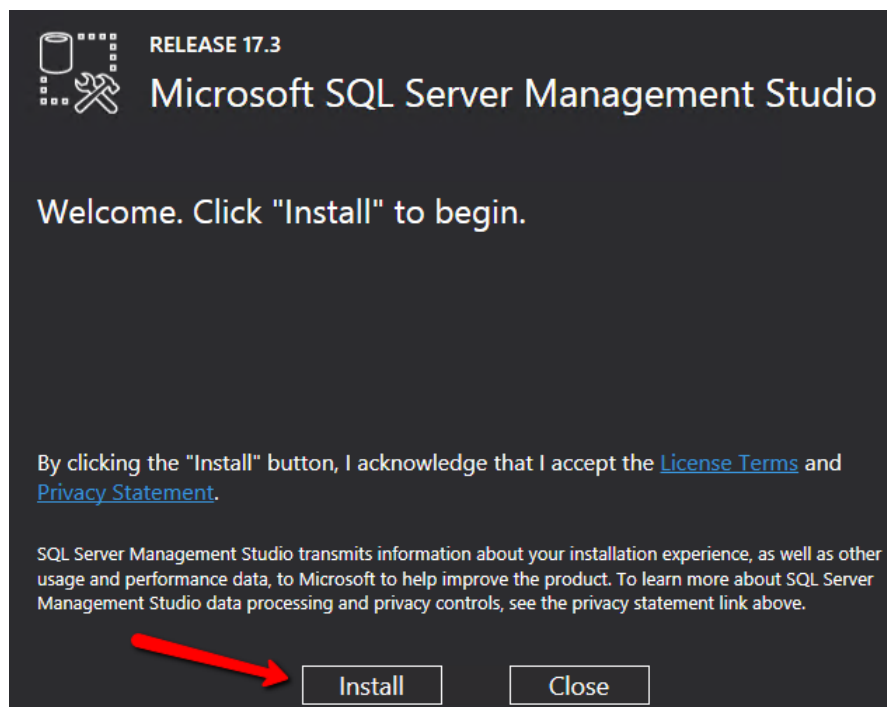
<https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms>

IMPORTANT: Installing these tools may require the computer to be rebooted once the installation completes.

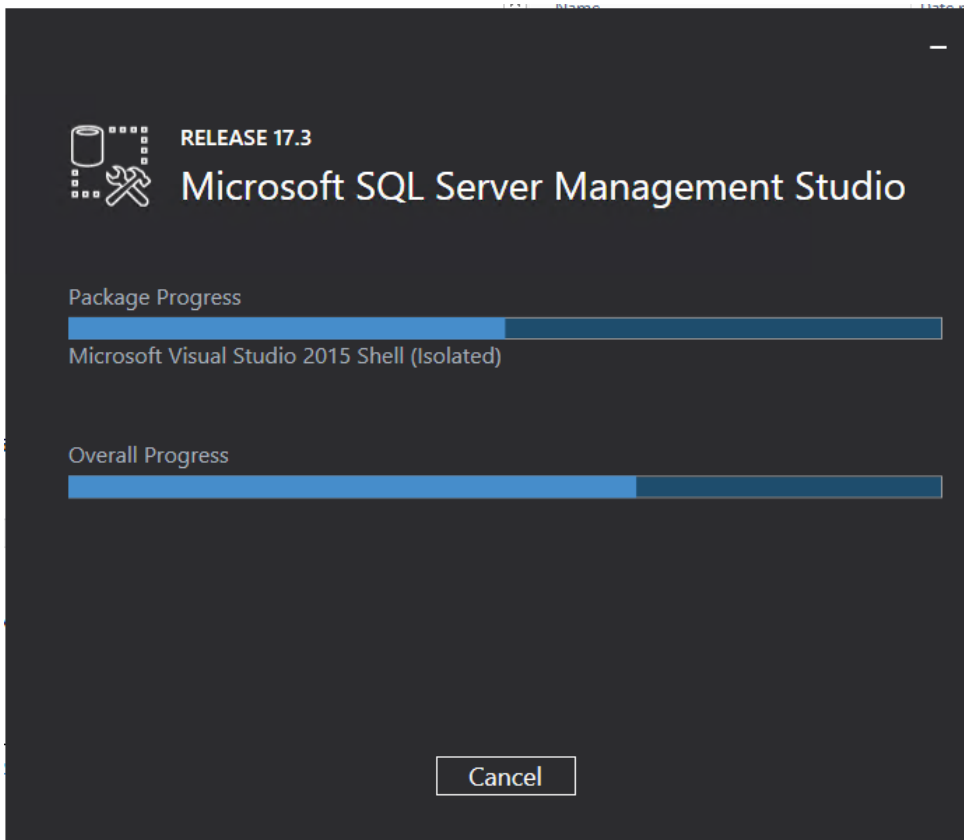
In this example we will be installing these tools as available for Microsoft SQL Server 2016 on the database server itself, which is using the Windows Server 2012 operating system.

Download and start the installer provided by Microsoft.

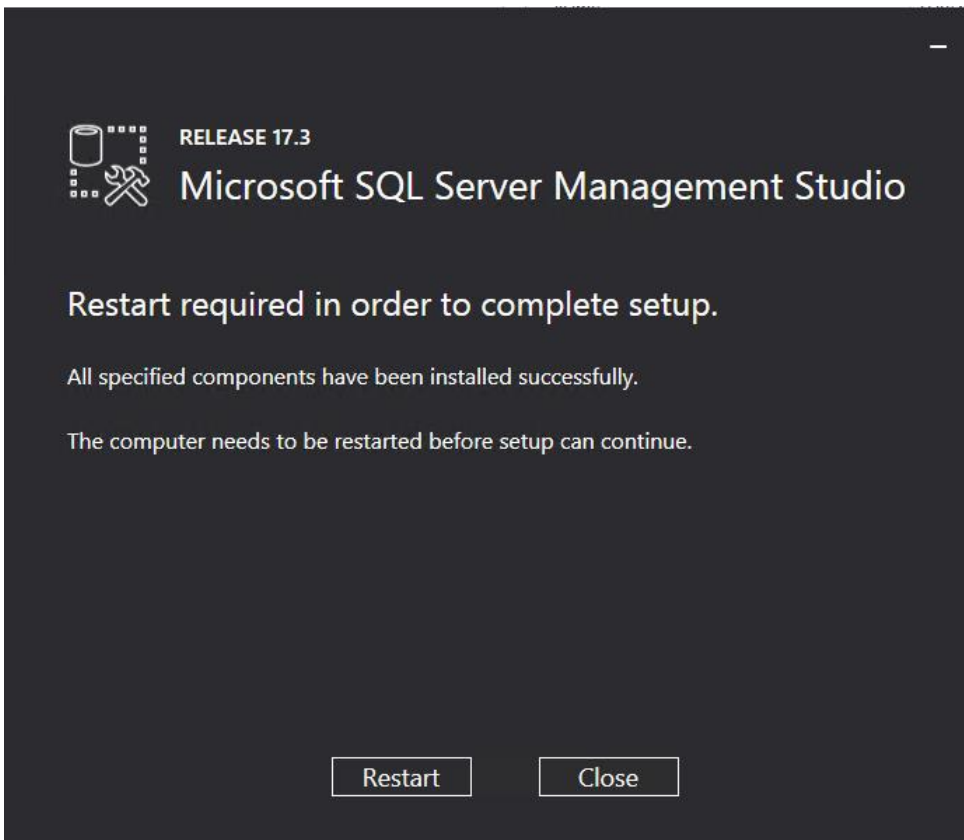
Click the Install button to begin:



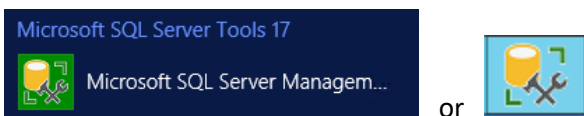
The next dialog will display the installation progress.



When complete, a server restart may be required:

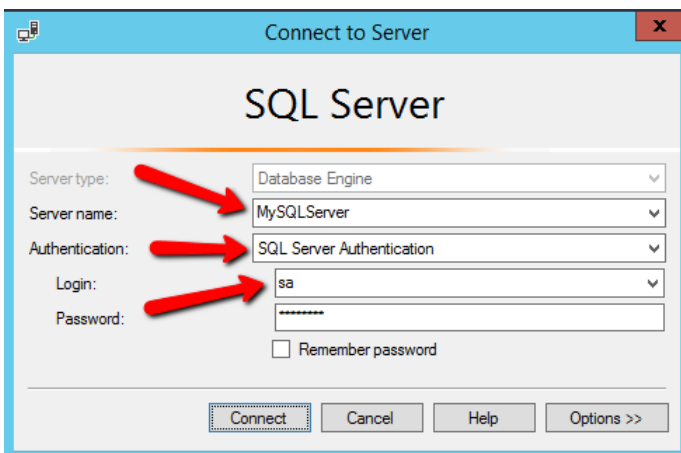


Once the installation is complete, an icon to launch SQL Server Management Studio will be available.



Once it launches you'll be prompted to connect to a SQL server instance. If you are logged into the same computer that has SQL server installed, a server name of **localhost** will work.

In this example we enabled SQL Server Authentication when installing SQL Server, so we could login with the "sa" administrative account, like this:



The *Authentication* choice could be changed to Windows Authentication if whoever is currently logged in and running SQL Server Management Studio (on any computer) has permissions to access the database.

The “Server name” in the example above is just the computer name where SQL Server has been installed. This is because we installed the full version of SQL Server (**not the free SQL Server Express edition**), and we installed the default instance name. This provides the simplest connection, which is just the computer name of the computer on which SQL Server was installed.

IMPORTANT: The *Server name* field is used to specify both the name of the computer *and the SQL instance name (if required)* to which to connect.

For example, if you installed the free SQL Server Express version and are using SQL Server Management Studio on the server computer itself, the “Server name” could be:

localhost\SQLEXPRESS

If you’re connecting to SQL Server running on different computer that has a SQL Server Express instance installed, the correct “Server name” could be:

<ComputerName>\SQLEXPRESS

If you’re connecting to SQL Server running on a different computer and you provided your own instance name during the installation, the correct “Server name” could be:

<ComputerName>\<InstanceName>

Knowing the correct value to use will be critical when defining a connection string to the database.

SQL Server Management Studio itself can therefore be used to test the settings needed to connect to the correct database instance, and once the correct settings are known they can then be used when defining the connection string to use in other CTC software.

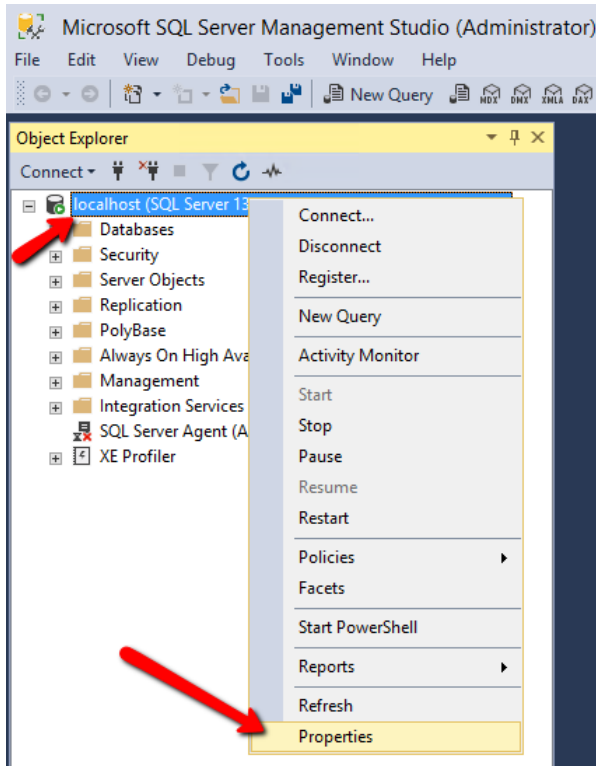
Additional Notes and Suggestions

Turning on Backup Compression

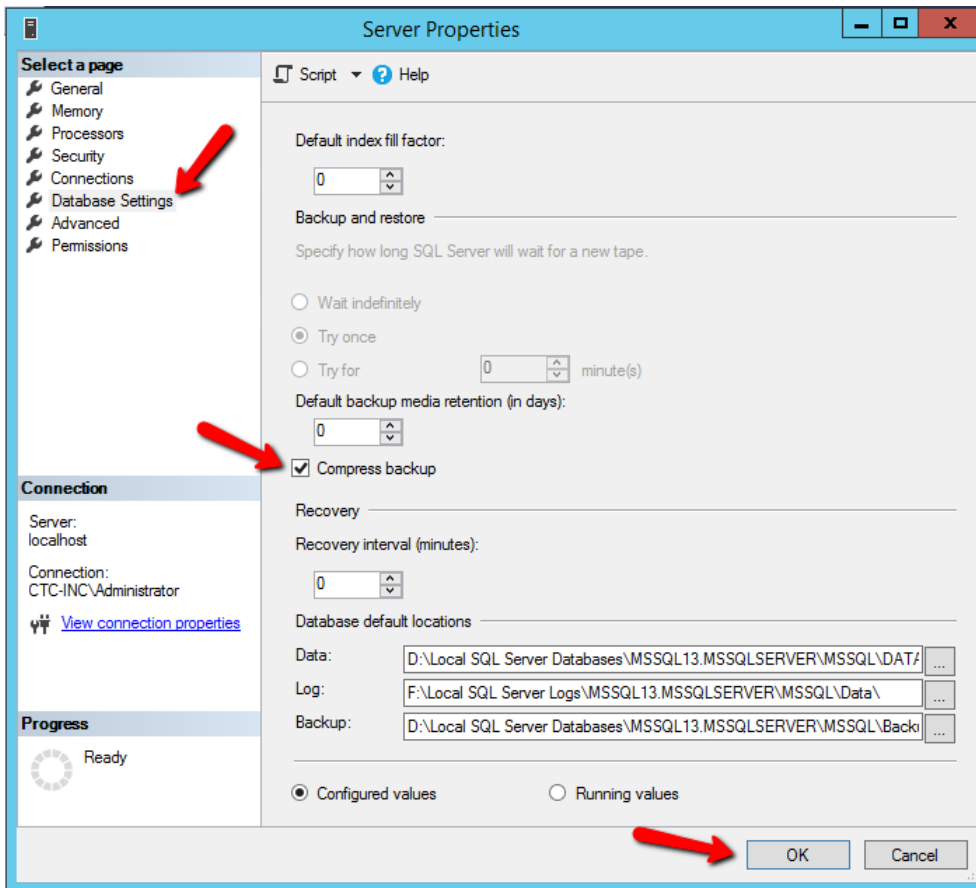
Full versions of SQL Server allow defining that when database backups are made to have them be compressed files.

To check or change the default value for creating backups with compression:

Right-click on the server name and select Properties:



On the next dialog, select the *Database Settings* item in the list on the left and under *Backup and Restore* check the “Compress backup” checkbox, then click the OK button in the lower right corner:

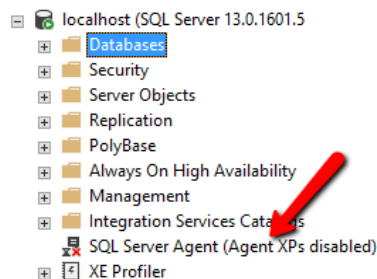


Fixing SQL Server Agent

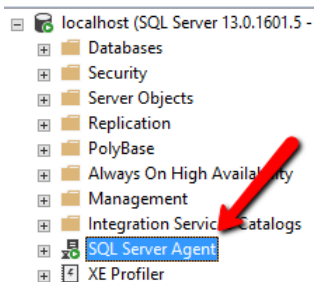
SQL Server Agent is a service that was installed which allows doing things like scheduling activities in SQL server, such as periodic backups.

If you need to schedule jobs on SQL server, it is important that this service runs correctly.

When connecting to the database with SQL Server Management Studio an issue with SQL Server Agent may appear:



If you right-click on *SQL Server Agent* and choose the **Start** option, it may start normally:



However, if it doesn't start, this article can help resolve the issue:

<https://www.mssqltips.com/sqlservertip/2729/how-to-start-sql-server-agent-when-agent-xps-show-disabled/>

Scheduling Automatic Backups

If you would like to schedule automatic backups of your databases, CTC recommends the free tools that are available from this web site:

<https://ola.hallengren.com/>