Table of Contents

SQL Server Configuration Manager Help

SQL Server Services

Analysis Server <server> (SQL Server Configuration Manager)
Analysis Server Properties (Log On Tab)
Analysis Server Properties (Service Tab)
Analysis Server Properties (Advanced Tab)

Notification Services (SQL Server Configuration Manager)
NS$<service name> Properties (Log On Tab)
NS$<service name> Properties (Service Tab)
NS$<service name> Properties (Advanced Tab)

Report Server <server> (SQL Server Configuration Manager)
Report Server Properties (Log On Tab)
Report Server Properties (Service Tab)
Report Server Properties (Advanced Tab)

SQL Server Agent <server> (SQL Server Configuration Manager)
SQL Server Agent Properties (Log On Tab)
SQL Server Agent Properties (Advanced Tab)
SQL Server Agent Properties (Service Tab)

SQL Server Browser (SQL Server Configuration Manager)
SQL Server Browser Properties (Log On Tab)
SQL Server Browser Properties (Service Tab)
SQL Server Browser Properties (Advanced Tab)

SQL Server <server> (SQL Server Configuration Manager)
SQL Server Properties (Log On Tab)
SQL Server Properties (Service Tab)
SQL Server Properties (FILESTREAM Tab)
SQL Server Properties (Always On High Availability Tab)
SQL Server Properties (Startup Parameters Tab)
SQL Server Properties (Advanced Tab)
SQL Server Integration Services 10.0 (SQL Server Configuration Manager)
SQL Server Integration Services Properties (Log On Tab)
SQL Server Integration Services Properties (Service Tab)
SQL Server Integration Services Properties (Advanced Tab)
SQL Full-text Filter Daemon Launcher (SQL Server Configuration Manager)
SQL Full-text Filter Daemon Launcher (Log On Tab)
SQL Full-text Filter Daemon Launcher (Service Tab)
SQL Full-text Filter Daemon Launcher (Advanced)
Unknown Service (Log On Tab)

SQL Server Network Configuration

Protocols for <server> (SQL Server Configuration Manager)
Protocols for MSSQLSERVER Properties (Flags Tab)
Protocols for MSSQLSERVER Properties (Certificate Tab)
Protocols for MSSQLSERVER Properties (Advanced Tab)
Shared Memory Properties
Named Pipes Properties
TCP/IP Properties (Protocols Tab)
TCP/IP Properties (IP Addresses Tab)

SQL Native Client 11.0 Configuration

SQL Server Native Client Configuration Properties (Flags Tab)
Client Protocols (SQL Server Configuration Manager)
Client Protocols Properties (Order Tab)
Client Protocols - Shared Memory Properties (Protocol Tab)
Client Protocols - TCP IP Properties (Protocol Tab)
Client Protocols - Named Pipes Properties (Protocol Tab)

Aliases (SQL Server Configuration Manager)
New Alias (Alias Tab)
<Alias> Properties (Alias Tab)
Creating a Valid Connection String Using Shared Memory Protocol
Creating a Valid Connection String Using TCP IP

SQL Server Browser Service
Connecting Using IPv6
Monitoring the Error Logs

Viewing the SQL Server Error Log

Viewing the Windows Application Log
Use SQL Server Configuration Manager to configure SQL Server services and configure network connectivity. To create or manage database objects, configure security, and write Transact-SQL queries, use SQL Server Management Studio. For more information about SQL Server Management Studio, see SQL Server Books Online.

This section contains the F1 Help topics for the dialogs in SQL Server Configuration Manager.

**NOTE**

SQL Server Configuration Manager cannot configure versions of SQL Server earlier than Microsoft SQL Server 2005.

### Services

SQL Server Configuration Manager manages services that are related to SQL Server. Although many of these tasks can be accomplished using the Microsoft Windows Services dialog, it is important to note that SQL Server Configuration Manager performs additional operations on the services it manages, such as applying the correct permissions when the service account is changed. Using the normal Windows Services dialog to configure any of the SQL Server services might cause the service to malfunction.

Use SQL Server Configuration Manager for the following tasks for services:

- Start, stop, and pause services
- Configure services to start automatically or manually, disable the services, or change other service settings
- Change the passwords for the accounts used by the SQL Server services
- Start SQL Server using trace flags (command line parameters)
- View the properties of services

### SQL Server Network Configuration

Use SQL Server Configuration Manager for the following tasks related to the SQL Server services on this computer:

- Enable or disable a SQL Server network protocol
- Configure a SQL Server network protocol

**NOTE**

For a short tutorial about how to configure protocols and connect to the SQL Server Database Engine, see Tutorial: Getting Started with the Database Engine.

### SQL Server Native Client Configuration

SQL Server clients connect to SQL Server by using the SQL Server Native Client network library. Use SQL Server Configuration Manager for the following tasks related to client applications on this computer:

- For SQL Server client applications on this computer, specify the protocol order, when connecting to
To open SQL Server Configuration Manager

- On the Start menu, point to All Programs, point to Microsoft SQL Server (version), point to Configuration Tools, and then click SQL Server Configuration Manager.

To access SQL Server Configuration Manager Using Windows 8

Because SQL Server Configuration Manager is a snap-in for the Microsoft Management Console program and not a stand-alone program, SQL Server Configuration Manager not does not appear as an application when running Windows 8. To open SQL Server Configuration Manager, in the Search charm, under Apps, type SQLServerManager12.msc (for SQL Server 2014) or SQLServerManager11.msc for (SQL Server 2012), and then press Enter.

See Also

- SQL Server Services
- SQL Server Network Configuration
- SQL Native Client 11.0 Configuration
- Choosing a Network Protocol

instances of SQL Server.

- Configure client connection protocols.

- For SQL Server client applications, create aliases for instances of SQL Server, so that clients can connect using a custom connection string.

For more information about each of these tasks, see F1 help for each task.
This section contains F1 Help topics for the SQL Server Services dialog boxes in Microsoft SQL Server Configuration Manager.

**NOTE**
If you open SQL Server Configuration Manager on a node of a cluster that does not host the resources, the services show as stopped.

**NOTE**
If the nodes of the SQL Server Services do not populate with the SQL Server services available on the computer, you may not have sufficient permissions to access Windows Management Instrumentation (WMI). To configure permissions on WMI, search Books Online for the topic, "How to: Configure WMI to Show Server Status in SQL Server Tools."

**See Also**

SQL Server Configuration Manager Help
This is the service that supports Microsoft Analysis Services. To view the properties of the service, right-click the service in the details pane, and then click Properties.

For information about Analysis Services, see Books Online.
Use the Log On tab of the Analysis Server Properties dialog box to specify the account used by the SSAS service, and to start and stop the service.

**NOTE**
When changing the Account Name used by a service on a clustered instance, the new account must either be a member of the domain group specified during setup for the service being changed, or you must have permission to add members to that group. If you do not have permission to modify group membership, contact the domain administrator.

**Options**

**Local System account**
Specify a local system account, which does not require a password. However, the local system account may restrict the service from interacting with other servers, depending on the privileges granted to the account.

**This account**
Specify a local or domain user account that uses Microsoft Windows Authentication. Microsoft recommends using a domain user account with minimal rights for services. For information about selecting an account, search Books Online for the topic Setting Up Windows Service Accounts.

**Account Name**
Specify the local or domain user account name.

**Password**
Type the password of the account.

**Confirm password**
Type the password of the account again.

**Start**
Start the service.

**Stop**
Stop the service.

**Pause**
Pause the service.

**Resume**
Resume a paused service.
This service is the Microsoft Analysis Services. This service must be running for SSAS to work properly. The property values in light gray cannot be changed using this application.

**Options**

**Binary Path**
Displays the location of the program files used by this service.

**Error Control**
1 indicates `SERVICE_ERROR_NORMAL`. If the service fails to start during computer start up, the startup program logs the error and displays a pop-up message box but continues the startup operation. This value cannot be changed.

**Exit Code**
When an error occurs, the error number appears in this box. Use this number to troubleshoot failures by searching for the number in the Microsoft Knowledge Base or provide the number to your technical support staff.

**Host Name**
Displays the name of the computer or cluster running SSAS.

**Name**
Indicates the display name of the service.

**Process ID**
Displays the number used by Microsoft Windows to keep track of this program’s processes.

**SQL Service Type**
Displays the type of service provided to calling processes. Microsoft SQL Server installs several services.

**Start Mode**
Set this service to the following choices:

- **Manual**: This service does not automatically start when the computer starts. You must start the service using SQL Server Configuration Manager, or some other tool.
- **Automatic**: This service attempts to start when this computer starts.
- **Disabled**: This service cannot be started.

**State**
Indicates whether this service is running, stopped, or disabled.
This service is the Microsoft Analysis Services. If custom properties are defined, they appear on this tab, with their values.

**Options**

**Clustered**
Indicates whether this service is installed as a resource of a clustered server.

**Customer Feedback Reporting**
Indicates whether Service Quality Monitoring has been enabled on this service. For more information on Customer Feedback Reporting, search Books Online for the topic, “Error and Usage Report Settings.”

**Dump Directory**
Displays the location where memory dumps are placed in case of an error.

**Error Reporting**
When set to **Yes**, the Dr. Watson program forwards information to either Microsoft or your error server if a serious failure occurs. For more information on Error Reporting, search Books Online for the topic, “Error and Usage Report Settings.”

**Instance ID**
Indicates the instance that uses this service.
This is the service that supports Notification Services. To view the properties of the service, right-click the service in the details pane, and then click Properties.

For information about Notification Services, see Books Online.
Use the **Log On** tab of the **Notification Services Properties** dialog box to specify the account used by the Notification Services service, and to start and stop the service.

**Options**

**Local System account**
Specify a local system account, which does not require a password. However, the local system account may restrict the service from interacting with other servers, depending on the privileges granted to the account.

**This account**
Specify a local or domain user account that uses Microsoft Windows Authentication. Microsoft recommends using a domain user account with minimal rights for services. For information about selecting an account, search Books Online for the topic, "Setting Up Windows Service Accounts."

**Account Name**
Specify the local or domain user account name.

**Password**
Type the password of the account.

**Confirm password**
Type the password of the account again.

**Start**
Start the service.

**Stop**
Stop the service.

**Pause**
Pause the service.

**Resume**
Resume a paused service.
This service is the Microsoft Notification Services service. The property values in light gray cannot be changed using this application.

Options

**Binary Path**
Displays the location of the program files used by this service.

**Error Control**
1 indicates `SERVICE_ERROR_NORMAL`. If the service fails to start during computer start up, the startup program logs the error and displays a pop-up message box but continues the startup operation. This value cannot be changed.

**Exit Code**
When an error occurs, the error number appears in this box. Use this number to troubleshoot failures by searching for the number in the Microsoft Knowledge Base or provide the number to your technical support staff.

**Host Name**
Displays the name of the computer or cluster running the full text search.

**Name**
Indicates the display name of the service.

**Process ID**
Displays the Microsoft Windows process ID.

**SQL Service Type**
Type of service provided to calling processes. SQL Server installs several services.

**Start Mode**
Set this service to the following choices:

- **Manual**: This service does not automatically start when the computer starts. You must start the service using SQL Server Configuration Manager, or some other tool.
- **Automatic**: This service attempts to start when this computer starts.
- **Disabled**: This service cannot be started.

**State**
Indicates whether this service is running, stopped, or disabled.
Notification Services is implemented as a service named in the format NS$. If custom properties are defined, they appear on this tab, with their values. This tab may have no properties listed.
This is the service that supports Microsoft Reporting Services. To view the properties of the service, right-click the service in the details pane, and then click **Properties**.

For information about Reporting Services, see Books Online.
Use the **Log On** tab of the **Report Server Properties** dialog box to specify the account used by the Report Server service, and to start and stop the service.

**Options**

**Local System account**
Specify a local system account, which does not require a password. However, the local system account may restrict the service from interacting with other servers, depending on the privileges granted to the account.

**This account**
Specify a local or domain user account that uses Microsoft Windows Authentication. Microsoft recommends using a domain user account with minimal rights for services. For information about selecting an account, search Books Online for the topic Setting Up Windows Service Accounts.

**Account Name**
Specify the local or domain user account name.

**Password**
Type the password of the account.

**Confirm password**
Type the password of the account again.

**Start**
Start the service.

**Stop**
Stop the service.

**Pause**
Pause the service.

**Resume**
Resume a paused service.
This service is the Microsoft SQL Server Report Server service. The property values in light gray cannot be changed by using this application.

Options

**Binary Path**
Displays the location of the program files used by this service.

**Error Control**
1 indicates "SERVICE_ERROR_NORMAL". If the service fails to start during computer start up, the startup program logs the error and displays a pop-up message box but continues the startup operation. This value cannot be changed.

**Exit Code**
When an error occurs, the error number appears in this box. Use this number to troubleshoot failures by searching for the number in the Microsoft Knowledge Base or provide the number to your technical support staff.

**Host Name**
Displays the name of the computer or cluster running the full text search.

**Name**
Indicates the display name of the service.

**Process ID**
Displays the Microsoft Windows process ID.

**SQL Service Type**
Type of service provided to calling processes. SQL Server installs several services.

**Start Mode**
Set this service to the following choices:

- Manual: This service does not automatically start when the computer starts. You must start the service using SQL Server Configuration Manager, or some other tool.
- Automatic: This service attempts to start when this computer starts.
- Disabled: This service cannot be started.

**State**
Indicates whether this service is running, stopped, or disabled.

See Also

[SQL Server Services](#)
This service is the Microsoft Reporting Services. If custom properties are defined, they appear on this tab, with their values.

Options

**Customer Feedback Reporting**
Indicates whether Service Quality Monitoring has been enabled on this service. For more information on Customer Feedback Reporting, search Books Online for the topic, "Error and Usage Report Settings."

**Dump Directory**
Displays the location where memory dumps are placed in case of an error.

**Error Reporting**
When set to **Yes**, the Dr. Watson program forwards information to either Microsoft or your error server if a serious failure occurs. For more information on Error Reporting, search Books Online for the topic, "Error and Usage Report Settings."

**Instance ID**
Indicates the instance that uses this service.
This is the service that supports SQL Server Agent. To view the properties of the service, right-click the service in the details pane, and then click Properties.

SQL Server Agent is a Microsoft Windows service that executes scheduled administrative tasks called jobs. For more information about SQL Server Agent, see "About SQL Server Agent" in SQL Server Books Online.
Use the Log On tab of the SQL Server Agent Properties dialog box to specify the account used by the SQL Server Agent service, and to start and stop the service. Changing the password of an account takes effect immediately without restarting the service.

**NOTE**

When changing the account name used by a service on a clustered instance, the new account must be a member of the domain group specified during setup for the service being changed, or you must have permission to add members to that group. If you do not have permission to modify group membership, contact the domain administrator.

**Options**

**Local System account**
Specify a local system account, which does not require a password. However, the local system account may restrict the service from interacting with other servers, depending on the privileges granted to the account.

**This account**
Specify a local or domain user account that uses Windows Authentication. Microsoft recommends using a domain user account with minimal rights for services. For information about selecting an account, search Books Online for "Setting Up Windows Service Accounts."

**Account Name**
Specify the local or domain user account name.

**Password**
Type the password of the account.

**Confirm password**
Type the password of the account again.

**Start**
Start the service.

**Stop**
Stop the service.

**Pause**
Pause the service.

**Resume**
Resume a paused service.
Five properties appear on the Advanced tab by default. If custom properties are defined, they also appear on this tab, with their values.

Options

**Clustered**
Indicates if this service is installed as a resource of a clustered server.

**Customer Feedback Reporting**
Indicates whether Service Quality Monitoring has been enabled on this service. For more information on Customer Feedback Reporting, search Books Online for the topic, "Error and Usage Report Settings."

**Dump Directory**
Displays the location where memory dumps are placed in case of an error.

**Error Reporting**
When set to Yes, the Dr. Watson program forwards information to either Microsoft or your error server if a serious failure occurs. For more information on Error Reporting, search Books Online for the topic, "Error and Usage Report Settings."

**Instance ID**
Indicates the instance that uses this service.
This service is the Microsoft SQL Server Agent service. The property values in light gray cannot be changed using this application.

**Options**

**Binary Path**
Displays the location of the program files used by this service.

**Error Control**
1 indicates `SERVICE_ERROR_NORMAL`. If the service fails to start during computer start up, the startup program logs the error and displays a pop-up message box but continues the startup operation. This value cannot be changed.

**Exit Code**
When an error occurs, the error number appears in this box. Use this number to troubleshoot failures by searching for the number in the Microsoft Knowledge Base or provide the number to your technical support staff.

**Host Name**
Displays the name of the computer or cluster running SQL Server Agent.

**Name**
Indicates the display name of the service.

**Process ID**
Displays the Microsoft Windows process ID.

**SQL Service Type**
Displays the type of service provided to calling processes. SQL Server installs several services.

**Start Mode**
Set this service to the following choices:

- Manual: This service does not automatically start when the computer starts. You must start the service using SQL Server Configuration Manager, or some other tool.

- Automatic: This service attempts to start when this computer starts.

- Disabled: This service cannot be started.

**State**
Indicates whether this service is running, stopped, or disabled. "..." indicates a state change is pending.
This is the service that supports SQL Server Browser. To view the properties of the service, right-click the service in the details pane, and then click Properties.

For a description of the SQL Server Browser service and how it works, see SQL Server Browser Service.
The SQL Server Browser program runs as a service on the server. SQL Server Browser listens for incoming requests for Microsoft SQL Server resources and provides information about SQL Server instances installed on the computer.

SQL Server Browser listens on a UDP port and accepts unauthenticated requests using SQL Server Resolution Protocol (SSRP).

Changing the password of an account takes effect immediately without restarting the service.

## Options

### Local System account
Run the SQL Server Browser service in the security context of the Local System account. When possible, use a low-permission account instead.

### This account
Specify a local or domain user account that uses Windows Authentication. We recommend using a domain user account with minimal rights for services. For information about selecting an account, see "Setting Up Windows Service Accounts" in SQL Server Books Online.

### Browse
Browse for a user or built-in security principal.

---

**IMPORTANT**

Use a low-permission account. For information about the permissions required for the SQL Server Browser Service, see the Security section of SQL Server Browser Service.

---

### Password
Enter the password of the security principal.

### Confirm password
 Confirm the password of the security principal.

### Service status
Indicates whether this service is running, stopped, or disabled. "..." indicates a state change is pending.

### Start
Start the SQL Server Browser service.

### Stop
Stop the SQL Server Browser service.

### Pause
Pause the SQL Server Browser service.

### Resume
Resume a paused SQL Server Browser service.
See Also

SQL Server Browser Service
The SQL Server Browser program runs as a service on the server. SQL Server Browser listens for incoming requests for Microsoft SQL Server resources and provides information about SQL Server instances installed on the computer.

Use the **Service** tab on the **SQL Server Browser Properties** dialog box to view the following options. All properties except **Start Mode** are read-only.

**Options**

**Binary Path**
Displays the location of the program files used by this service.

**Error Control**
1 indicates `SERVICE_ERROR_NORMAL`. If the service fails to start during computer start up, the startup program logs the error and displays a pop-up message box but continues the startup operation. This value cannot be changed.

**Exit Code**
When an error occurs, the error number appears in this box. Use this number to troubleshoot failures by searching for the number in the Microsoft Knowledge Base or provide the number to your technical support staff.

**Host Name**
Displays the name of the computer or cluster running the SQL Server Browser service.

**Name**
Indicates the display name of the service.

**Process ID**
Displays the Windows process ID.

**Service Type**
Displays the type of service provided to calling processes. SQL Server installs several services.

**Start Mode**
Set this service to the following choices:

- **Manual**: This service does not automatically start when the computer starts. You must start the service using SQL Server Configuration Manager, or some other tool.
- **Automatic**: This service attempts to start when this computer starts.
- **Disabled**: This service cannot be started.

**State**
Indicates whether this service is running, stopped, or disabled. "..." indicates a state change is pending.

**See Also**

[SQL Server Browser Service](#)
The SQL Server Browser program runs as a service on the server. SQL Server Browser listens for incoming requests for Microsoft SQL Server resources and provides information about SQL Server instances installed on the computer.

**Options**

**Clustered**
Indicates if this service is installed as a resource of a clustered server.

**Customer Feedback Reporting**
Indicates whether Service Quality Monitoring has been enabled on this service. For more information on Customer Feedback Reporting, search Books Online for the topic, "Error and Usage Report Settings."

**Dump Directory**
The location where memory dumps are placed in case of an error.

**Error Reporting**
When set to Yes, the Dr. Watson program forwards information to either Microsoft or your error server if a serious failure occurs. For more information on Error Reporting, search Books Online for the topic, "Error and Usage Report Settings."

**Instance ID**
Indicates the SQL Server instance that used this SQL Server Agent instance. The default instance is MSSQL10_50.MSSQLSERVER.

**See Also**
SQL Server Browser Service
This is the service that supports Microsoft SQL Server Database Engine. To view the properties of the service, right-click the service in the details pane, and then click **Properties**.

For information about the SQL Server Database Engine, see Books Online.
Use the **Log On** tab of the **SQL Server Properties** dialog box to specify the account used by the SQL Server service, to change the password of an account, and to start and stop the service. Changing the password of an account takes effect immediately.

### NOTE

When changing the account name used by a service on a clustered instance, the new account must be a member of the domain group specified during setup for the service being changed, or you must have permission to add members to that group. If you do not have permission to modify group membership, contact the domain administrator.

For more information about selecting an account to run the service, see “Setting Up Windows Service Accounts” in SQL Server Books Online.

### Options

#### Built-in account

**Local System**

- Specify the local system account. This account does not require a password. However, the local system account may prevent the service from interacting with other servers, depending on the privileges granted to the account.

**Local Service**

- Specify the local service account. This account does not require a password. However, the local service account may prevent the service from interacting with other servers, depending on the privileges granted to the account.

**Network Service**

- **We recommend that you do not use the Network Service account for the SQL Server or the SQL Server Agent services. Local User or Domain User accounts are more appropriate for these services.**

#### This account

Specify a local or domain user account that uses Windows Authentication. We recommend that you use a domain user account that has minimal rights for services.

**Account Name**

Specify the local or domain user account name.

**Password**

Type the password of the account.

**Confirm password**

Type the password of the account again.

**Start**

Start the service.

**Stop**

Stop the service.
**Pause**
Pause the service.

**Resume**
Resume a paused service.

**IMPORTANT**
By default, only members of the local administrators group can start, stop, pause, resume or restart a service. To grant non-administrators the ability to manage services, see [How to grant users rights to manage services in Windows Server 2003](#). (The process is similar on other versions of Windows.)

**NOTE**
When starting SQL Server, a WMI error containing the phrase "not implemented [0x80004001]" may indicate that SQL Server is not installed on the target computer.
Use the Service tab on the MSSQLSERVER Properties dialog box to view or specify the following options.

**Options**

**Binary Path**
Lists the location of the program files used by this service.

**Error Control**
1 indicates `SERVICE_ERROR_NORMAL`. If the service fails to start during computer start up, the startup program logs the error and displays a pop-up message box but continues the startup operation. This value cannot be changed.

**Exit Code**
When an error occurs, the error number appears in this box. Use this number to troubleshoot failures by searching for the number in the Microsoft Knowledge Base or provide the number to your technical support staff.

**Host Name**
Displays the name of the computer or cluster running the Microsoft SQL Server service.

**Name**
Indicates the display name of the service.

**Process ID**
Displays the Windows process ID.

**Service Type**
Displays the type of service provided to calling processes. SQL Server installs several services.

**Start Mode**
Set this service to the following choices:

- **Manual**: This service does not automatically start when the computer starts. You must start the service using SQL Server Configuration Manager, or some other tool.
- **Automatic**: This service attempts to start when this computer starts.
- **Disabled**: This service cannot be started.

**State**
Indicates whether this service is running, stopped, or disabled. "..." indicates a state change is pending.
Use this page to enable FILESTREAM for this installation of SQL Server 2017.

**UIElement List**

**Enable FILESTREAM for Transact-SQL access**
Select to enable FILESTREAM for Transact-SQL access. This control must be checked before the other control options will be available.

**Enable FILESTREAM for file I/O streaming access**
Select to enable Win32 streaming access for FILESTREAM.

**Windows share name**
Use this control to enter the name of the Windows share in which the FILESTREAM data will be stored.

**Allow remote clients to have streaming access to FILESTREAM data**
Select this control to allow remote clients to access this FILESTREAM data on this server.
Use the **Always On High Availability** tab of the **SQL Server Properties** dialog box in SQL Server Configuration Manager to enable or disable the Always On Availability Groups feature in SQL Server 2017. Enabling Always On Availability Groups is a prerequisite for an instance of SQL Server to use availability groups as a high availability and disaster recovery solution.

## Prerequisites

To be enabled for Always On Availability Groups, a server instance must meet the following prerequisites:

- The server instance must reside on a Windows Server Failover Clustering (WSFC) node.
- To be in the same availability group, instances must be in the same WSFC cluster. An availability group cannot span WSFC clusters.
- The server instance must be running an edition of SQL Server that supports Always On availability groups.
- Enable Always On Availability Groups for only one server instance at a time. After enabling Always On Availability Groups, wait until the SQL Server service has restarted before you enable the next server instance.

### NOTE

For information about feature support and for information about additional prerequisites, restrictions, and recommendations for Always On availability groups, see SQL Server 2017 Books Online.

## Dialog Options

### Windows failover cluster name

Displays the name of the WSFC cluster in which the local computer is a node.

### Enable Always On Availability Groups

Use this check box to enable or disable Always On Availability Groups on this instance of SQL Server, as follows:

- If this check box is empty, Always On Availability Groups is currently disabled. To enable Always On Availability Groups, select this check box, click **OK**, and manually restart the SQL Server service.
- If this check box is already selected, Always On Availability Groups is currently enabled. To disable Always On Availability Groups, uncheck the check box and click **OK**. This causes the server instance to restart.

### TIP

After disabling Always On Availability Groups, you should remove any local availability replicas from the server instance. If you remove the last replica of a given availability group, you should also remove the group.

## UIElement List
For more information about follow up after you disable Always On Availability Groups and for information about how to create and configure availability groups, see SQL Server 2017 Books Online.
Use this dialog box to add or remove startup parameters for the Database Engine. Startup parameters can have a large effect on the Database Engine performance. Before adding or changing startup parameters, see the topic "Using the SQL Server Service Startup Options" in SQL Server Books Online.

Options

Specify a startup parameter
To add a parameter, type the parameter, and then click Add.

To modify one of the required parameters, select the parameter in the Existing parameters box, change the values in the Specify a startup parameter box, and then click Update.

Existing parameters
To remove a parameter, select a parameter, and then click Remove.

Parameter Format

Do not enter a separator between parameters. SQL Server Configuration Manager automatically adds the separator. SQL Server Configuration Manager enforces the following parameter requirements.

- Leading and trailing spaces are trimmed from any startup parameter.
- All startup parameters start with a – (dash) and the second value is a letter.

Required Parameters

The following parameters are required. They can be changed but not removed.

- -d is the path of the master.mdf file (the master database data file).
- -l is the path of the master.ldf file (the master database log file).
- -e is the path of the SQL Server error log files.

Caution
If the file path parameters are incorrect SQL Server might not start.

For more information about how to move the master database, see the topic "Moving System Databases" in SQL Server Books Online.

Optional Parameters

All of the supported startup parameters are described in the topic "Using the SQL Server Service Startup Options," in SQL Server Books Online. A startup parameter of -Ttrace# indicates that an instance of SQL Server should be started with a specified trace flag (trace#) in effect. Trace flags are used to start the server with nonstandard behavior. For more information about trace flags, see the topic "Trace Flags ( Transact-SQL)" in SQL Server Books Online.

Caution
You might see additional undocumented startup parameters and trace flags described on the Internet. Undocumented startup parameters and trace flags are created to address uncommon problems or to force certain
The following list describes some common optional parameters.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>SHORT DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>-m</td>
<td>Starts an instance of SQL Server in single-user mode.</td>
</tr>
<tr>
<td>-T1204</td>
<td>Returns the resources and types of locks participating in a deadlock and also the current command affected.</td>
</tr>
<tr>
<td>-T1224</td>
<td>Disables lock escalation based on the number of locks.</td>
</tr>
<tr>
<td>-T3608</td>
<td>Prevents SQL Server from automatically starting and recovering any database except the master database.</td>
</tr>
<tr>
<td>-T7806</td>
<td>Enables a dedicated administrator connection (DAC) on SQL Server Express.</td>
</tr>
</tbody>
</table>

**Caution**

Some optional parameters can change server behavior and may affect performance.

**Permissions**

Use of this page is restricted to users who can change the related entries in the registry. This includes the following users.

- Members of the local administrators group.
- The domain account that is used by SQL Server, if the Database Engine is configured to run under a domain account.

**Books Online References**

For additional information about SQL Server startup parameters, see "How to: Configure Server Startup Options (SQL Server Configuration Manager)" in SQL Server Books Online.
The following properties appear on the Advanced tab by default. If custom properties are defined, they also appear on this tab, with their values.

**Options**

**Clustered**
Indicates if this service is installed as a resource of a clustered server.

**Customer Feedback Reporting**
Indicates whether Service Quality Monitoring has been enabled on this service. For more information on Customer Feedback Reporting, search Books Online for the topic, "Error and Usage Report Settings."

**Data Path**
Displays the path to the SQL Server binaries for this installation of SQL Server.

**Dump Directory**
Displays the location where memory dumps are placed in case of an error.

**Error Reporting**
When set to Yes, the Dr. Watson program forwards information to either Microsoft or your error server if a serious failure occurs. For more information on Error Reporting, search Books Online for the topic, "Error and Usage Report Settings." To change this value, in SQL Server Management Studio Object Explorer, right-click your server, click **Properties**, and then click the **Misc. Server Settings** page. The options are presented in the **Information Reporting** area.

**File Version**
Displays the version of the SQL Server executable.

**Install Path**
Displays the path to the SQL Server binaries for this installation of SQL Server.

**Instance ID**
Indicates the SQL Server instance that used this service.

**Language**
Displays the default language for server messages.

**Registry Root**
Displays the location of the registry keys used by this application.

**Service Pack Level**
Displays the service pack level of this instance of SQL Server.

**SKU Name**
Displays the product stock keeping unit (SKU), sometimes called the product edition.

**Startup Parameters**
Lists any startup parameters that are used by this instance of SQL Server. Parameters are separated by semi-colons. The default parameters include the paths to the data file for the master database (master.mdf), the log file for the master database (mastlog.ldf), and the error log file. For the syntax of startup parameters, search Books Online for the topic Using the SQL Server Service Startup Options.
Stock Keeping Unit
Displays the product stock keeping unit (SKU) number.

Version
Displays the version number of this instance of SQL Server.

Virtual Server Name
Virtual Server Name when SQL Server is installed on a clustered server.
This is the service that supports SSIS. To view the properties of the service, right-click the service in the details pane, and then click Properties.

For information about SSIS, see Books Online.
Use the Log On tab of the Integration Services Properties dialog box to specify the account used by the Integration Services service, and to start and stop the service.

Options

**Local System account**
Specify a local system account, which does not require a password. However, the local system account may restrict the service from interacting with other servers, depending on the privileges granted to the account.

**This account**
Specify a local or domain user account that uses Microsoft Windows Authentication. Microsoft recommends using a domain user account with minimal rights for services. For information about selecting an account, search Books Online for the topic, “Setting Up Windows Service Accounts.”

**Account Name**
Specify the local or domain user account name.

**Password**
Type the password of the account.

**Confirm password**
Type the password of the account again.

**Start**
Start the service.

**Stop**
Stop the service.

**Pause**
Pause the service.

**Resume**
Resume a paused service.
Use the Service tab on the Integration Services Properties dialog box to view or specify the following options.

Options

**Binary Path**
Displays the location of the program files used by this service.

**Error Control**
1 indicates SERVICE_ERROR_NORMAL. If the service fails to start during computer start up, the startup program logs the error and displays a pop-up message box but continues the startup operation. This value cannot be changed.

**Exit Code**
The Microsoft Windows error code defining any problems encountered in starting or stopping the service. This property is set to ERROR_SERVICE_SPECIFIC_ERROR (1066) when the error is unique to the service represented by this class, and information about the error is available in the ServiceSpecificExitCode property. The service sets this value to NO_ERROR (0) when running, and again upon normal termination.

**Host Name**
Displays the name of the computer or cluster running the SSIS service.

**Name**
Indicates the display name of the service.

**Process ID**
Displays the Windows process ID.

**SQL Service Type**
Displays the type of service provided to calling processes. Microsoft SQL Server installs several services.

**Start Mode**
Set this service to the following choices:

- Manual: This service does not automatically start when the computer starts. You must start the service using SQL Server Configuration Manager, or some other tool.
- Automatic: This service attempts to start when this computer starts.
- Disabled: This service cannot be started.

**State**
Indicates whether this service is running, stopped, or disabled. "..." indicates a state change is pending.
Microsoft Integration Services is implemented as the Integration Services service. If custom properties are defined, they appear on this tab, with their values. This tab may have no properties listed.
Beginning in SQL Server 2008, the SQL Full-text Filter Daemon Launcher (FDHOST Launcher) service is used by SQL Server full-text search to start the filter daemon host process, which handles full-text search filtering and word breaking. This service must be running to use full-text search. The FDHOST Launcher service is an instance-aware service that is associated with a specific instance of SQL Server. The FDHOST Launcher service propagates the service account information to each filter daemon host process started. For information about the filter daemon host processes, see "Full-Text Search Architecture" in SQL Server Books Online.
Beginning in SQL Server 2008, the SQL Full-text Filter Daemon Launcher (FDHOST Launcher) service is used by SQL Server full-text search. This service must be running if you use full-text search. For information about the filter daemon host processes, see “Full-Text Search Architecture” in SQL Server Books Online.

Use the Log On tab of the SQL Full-text Filter Daemon Launcher Properties dialog box to specify the account used by the SQL Server full-text service, to change the password of an account, and to start and stop the service. Changing the password of an account takes affect after restarting the service.

NOTE
When changing the account name used by a service on a clustered instance, either the new account must be a member of the domain group specified during setup for the service, or you must have permission to add members to that group. If you do not have permission to modify group membership, contact the domain administrator.

For more information about selecting an account to run the service, see “Setting Up Windows Service Accounts” in SQL Server Books Online.

Options

**Built-in account**

**Local System**
Specify the local system account. This account does not require a password. However, the local system account may prevent the service from interacting with other servers, depending on the privileges granted to the account.

**Local Service**
Specify the local service account. This account does not require a password. However, the local service account may prevent the service from interacting with other servers, depending on the privileges granted to the account.

**Network Service**
We recommend that you do not use the Network Service account for the SQL Server services or the SQL Server Agent services. Local User or Domain User accounts are more appropriate for these services.

**This account**
Specify a local or domain user account that uses Windows Authentication. We recommend that you use a domain user account that has minimal rights for services.

**Account Name**
Specify the local or domain user account name.

**Password**
Type the password of the account.

**Confirm password**
Type the password of the account again.

**Start**
Start the service.

**Stop**
Stop the service.
Pause
Pause the service. Not available for this service.

Resume
Resume a paused service.
Beginning in SQL Server 2008, the SQL Full-text Filter Daemon Launcher (FDHOST Launcher) service is used by SQL Server full-text. This service must be running if you use full-text search. For information about the filter daemon host processes, see "Full-Text Search Architecture" in SQL Server Books Online.

Use the Service tab on the SQL Full-text Filter Daemon Launcher Properties dialog box to view or specify the following options.

Options

**Binary Path**
Lists the location of the program files used by this service.

**Error Control**
1 indicates SERVICE\_ERROR\_NORMAL. If the service fails to start during computer start up, the startup program logs the error and displays a pop-up message box but continues the startup operation. This value cannot be changed.

**Exit Code**
When an error occurs, the error number appears in this box. Use this number to troubleshoot failures by searching for the number in the Microsoft Knowledge Base or provide the number to your technical support staff.

**Host Name**
Displays the name of the computer or cluster running the Microsoft SQL Server service.

**Name**
Indicates the display name of the service.

**Process ID**
Displays the Windows process ID.

**SQL Service Type**
Displays the type of service provided to calling processes. SQL Server installs several services.

**Start Mode**
Set this service to the following choices:

- Manual: This service does not automatically start when the computer starts. You must start the service using SQL Server Configuration Manager, or some other tool.
- Automatic: This service attempts to start when this computer starts.
- Disabled: This service cannot be started.

**State**
Indicates whether this service is running, stopped, or disabled. "..." indicates a state change is pending.
Beginning in SQL Server 2008, the SQL Full-text Filter Daemon Launcher service is used by SQL Server full-text. This service must be running if you use full-text search. For information about the filter daemon host processes, see "Full-Text Search Architecture" in SQL Server Books Online.

No properties appear on the **Advanced** tab by default. If custom properties are defined, they also appear on this tab, with their values.
Microsoft SQL Server Configuration Manager is unable to identify this service.

SQL Server Configuration Manager receives service information from the WMI provider on the computer running the service. Either there was an error while reading the service properties, or the service properties are not complete. To resolve the problem, try closing and reopening SQL Server Configuration Manager, or check the WMI provider on the computer running the service.

The WMI provider is a Windows component. For information on how to check permissions on the WMI provider, see “How to: Configure WMI to Show Server Status in SQL Server Tools” in SQL Server Books Online.

If you believe you are viewing the correct service, use the Log On tab of the Unknown Service Properties dialog box to specify the account used by this service, and to start and stop the service.

**Options**

**Local System account**
Specify a local system account, which does not require a password. However, the local system account may prevent the service from interacting with other servers, depending on the privileges granted to the account.

**This account**
Specify a local or domain user account that uses Windows Authentication. We recommend using a domain user account with minimal rights for services. For information about selecting an account, “Setting Up Windows Service Accounts” in SQL Server Books Online.

**Account Name**
Specify the local or domain user account name.

**Password**
Type the password of the account.

**Confirm password**
Type the password of the account again.

**Start**
Start the service.

**Stop**
Stop the service.

**Pause**
Pause the service.

**Resume**
Resume a paused service.
This section contains the F1 Help topics for the **Server Network Configuration** dialog boxes in SQL Server Configuration Manager.

See Also

[SQL Server Configuration Manager Help](#)
To configure the protocols used by the Database Engine, select **Protocols** for the desired server instance in the console pane, right-click the appropriate protocol in the details pane, and then click **Properties**.
When a certificate is installed on the server, use the **Flags** tab on the **Protocols for MSSQLSERVER Properties** dialog box to view or specify the protocol encryption and hide instance options. Microsoft SQL Server must be restarted to enable or disable the **ForceEncryption** setting.

To encrypt connections, you should provision the SQL Server Database Engine with a certificate. If a certificate is not installed, SQL Server will generate a self-signed certificate when the instance is started. This self-signed certificate can be used instead of a certificate from a trusted certificate authority, but it does not provide authentication or non-repudiation.

**Caution**
Secure Sockets Layer (SSL) connections encrypted using a self-signed certificate do not provide strong security. They are susceptible to man-in-the-middle attacks. You should not rely on SSL using self-signed certificates in a production environment or on servers that are connected to the Internet.

For more information on encryption, see “Encrypting Connections to SQL Server” in SQL Server Books Online.

The login process is always encrypted. When **ForceEncryption** is set to **Yes**, all client/server communication is encrypted, and clients connecting to the Database Engine must be configured to trust the root authority of the server certificate. For more information, see "How to: Enable Encrypted Connections to the Database Engine ( SQL Server Configuration Manager)" in SQL Server Books Online.

**Cluster Servers**
If you want to use encryption with a failover cluster, you must install the server certificate with the fully qualified DNS name of the virtual server on all nodes in the failover cluster. For example, if you have a two-node cluster, with nodes named "test1.<your company>.com" and "test2.<your company>.com" and a virtual server named "virtsql", you need to install a certificate for "virtsql.<your company>.com" on both nodes. You can then check the **ForceEncryption** check box on the **SQL Server Configuration Manager** to configure your failover cluster for encryption.

**Options**

**ForceEncryption**
Force protocol encryption. Encryption is a method for keeping sensitive information confidential by changing data into an unreadable form. Encryption ensures that data remains secure, even if the transmission packets are viewed during the transmission process. To use channel binding, set **Force Encryption** to **On** and configure **Extended Protection** on the **Advanced** tab.

**HideInstance**
Prevent the SQL Server Browser Service from exposing this instance of the Database Engine to client computers that try to locate the instance by using the **Browse** button. In the case of named instances on the server, to connect, client applications must specify the protocol endpoint information. For example, the port number or the named pipe name, such as **tcp:server,5000**. For more information, see **Logging In to SQL Server**.

For more information, see "How to: Enable Encryption Connections to the Database Engine (SQL Server Configuration Manager)" in Books Online.
Use the **Certificate** tab on the **Protocols for MSSQLSERVER Properties** dialog box to select a certificate for Microsoft SQL Server, or to view the properties of a certificate. All fields are blank until a certificate is selected.

Certificates are stored locally for the users on the computer. To load a certificate for use by SQL Server, you must be running SQL Server Configuration Manager under the same user account as the SQL Server service.

**Page Header**

**View**
Provides access to additional details on the certificate. Not available until a certificate is selected in the **Certificate** box. For additional information on certificate details, see your Microsoft Windows documentation.

**Clear**
Removes the selection from the **Certificate** box.

**Certificate**
Name of certificate as determined by security provider. Select a certificate to see the details in the properties grid.

**Options**

**Expiration Date**
The final date for the period in which the certificate is valid.

**Friendly Name**
A friendly or common name for the individual or certification authority to whom the certificate is issued.

**Issued By**
Information regarding the certification authority that issued the certificate.

**Issued To**
Information regarding the recipient of the certificate.
Use the Advanced tab on the Protocols for MSSQLSERVER Properties dialog box to configure Extended Protection for Authentication for the SQL Server Database Engine. Extended Protection is a feature of the network components implemented by the operating system. Extended Protection is available in Windows 7 and Windows Server 2008 R2, and is included in service packs for older operating systems. SQL Server is more secure when connections are made using Extended Protection. Some benefits of Extended Protection require Force Encryption to be selected on the Flags tab.

IMPORTANT
Windows does not enable Extended Protection by default. For information about how to enable Extended Protection in Windows, see the Knowledge Base article, Extended Protection for Authentication.

For more information about how to configure other SQL Server services, and a complete description of Extended Protection, see more recent information on Microsoft.com.

Extended Protection is fully supported by the SQL Server Native Client beginning with SQL Server 2008 R2. Support for Extended Protection for other SQL Server client providers is not currently supported.

Options

Extended Protection
There are three possible values:

- When set to Off, Extended Protection is disabled. The instance of SQL Server will accept connections from any client regardless of whether the client is protected or not. Off is compatible with older and unpatched operating systems, but is less secure. Only use this setting when you know that the client operating systems do not support extended protection.

- When set to Allowed, Extended Protection is required for connections from operating systems that support Extended Protection. Connections from unprotected client applications that are running on protected client operating systems are rejected. Extended Protection is ignored for connections from unprotected operating systems. This setting is more secure than Off, but is not the most secure setting. Use this setting in mixed environments, where some operating systems or applications support Extended Protection and some do not.

- When set to Required, only connections from protected applications on protected operating systems are accepted. This setting is the most secure of the three options but connections from operating systems that do not support Extended Protection will not be able to connect to SQL Server.

Accepted NTLM SPNs
When the instance of SQL Server is identified by more than one NTLM service principal name (SPN), list the SPNs here as a series of strings separated by semicolons. For example, the value MSSQLSvc/HostName1.Contoso.com;MSSQLSvc/HostName2.Contoso.com, indicates that clients attempting to connect to SPNs named MSSQLSvc/HOST1.Contoso.com and MSSQLSvc/HOST2.Contoso.com are allowed. The variable has a maximum length of 2048 characters.
See Also

Extended Protection for Authentication with Reporting Services
Use the Protocol page on the Shared Memory Properties dialog box to enable or disable the shared memory protocol. Shared memory is the simplest protocol to use and has no configurable settings. Because clients using the shared memory protocol can only connect to a Microsoft SQL Server instance running on the same computer, it is not useful for most database activity. Use the shared memory protocol for troubleshooting when you suspect the other protocols are configured incorrectly.

SQL Server must be restarted to enable or disable the protocol.

Options

**Enabled**

Possible values are **Yes** and **No**. The shared memory protocol is enabled by default.

See Also

- Choosing a Network Protocol
- Creating a Valid Connection String Using Shared Memory Protocol
Use the Protocol page on the Named Pipes Properties dialog box to view or change the named pipe that Microsoft SQL Server listens to, when using the Named Pipes protocol.

SQL Server must be restarted to enable or disable the protocol, or change the named pipe.

**Options**

**Enabled**
Possible values are **Yes** and **No**.

**Pipe Name**
Specifies the named pipe on which SQL Server listens. By default, SQL Server listens on: `\\.\pipe\sql\query` for the default instance and `\\.\pipe\MSSQL$<instancename>\sql\query` for a named instance. This field is limited to 2047 characters.

**Creating an Alternate Named Pipe**

To change the named pipe, type the new pipe name in the Pipe Name box and then stop and restart SQL Server. Since `sql\query` is well known as the named pipe used by SQL Server, changing the pipe can help reduce the risk of attack by malicious programs.

**Example**

Type `\\.\pipe\unit\app` to listen on the `unit\app` pipe.

Type `\\.\pipe\acct` to listen on the `acct` pipe.

**See Also**

Enable or Disable a Server Network Protocol
Choosing a Network Protocol
Creating a Valid Connection String Using Named Pipes
Use the **TCP/IP Properties** dialog box to configure the options for the TCP/IP protocol. Click **TCP/IP** in the left pane, to show individual IP address configurations in the details pane.

Microsoft SQL Server must be restarted before the changes take effect.

**Options**

**Enabled**
Possible values are **Yes** and **No**.

**Keep Alive**
Specify the interval (milliseconds) in which keep-alive packets are transmitted to verify that the computer at the remote end of a connection is still available.

**Listen All**
Specify whether SQL Server will listen on all the IP addresses that are bound to network cards on the computer. If set to **No**, configure each IP address separately using the properties dialog box for each IP address. If set to **Yes**, the settings of the **IPAll** properties box will apply to all IP addresses. Default value is **Yes**.

**No Delay**
SQL Server does not implement changes to this property.

**See Also**

- Choosing a Network Protocol
- Creating a Valid Connection String Using TCP IP
Use the **TCP/IP Properties (IP Addresses Tab)** dialog box to configure the TCP/IP protocol options for a specific IP address. Only **TCP Dynamic Ports** and **TCP Port** can be configured for all addresses at once by selecting **IP All**.

Changes take effect when SQL Server is restarted. For information about starting and stopping the SQL Server Browser service, see [Start and Stop the SQL Server Browser Service](#).

**Static vs. Dynamic Ports**

The default instance of SQL Server listens for incoming connections on port 1433. The port can be changed for security reasons or because of a client application requirement. By default, named instances (including SQL Server Express) are configured to listen on dynamic ports. To configure a static port, leave the **TCP Dynamic Ports** box blank and provide an available port number in the **TCP Port** box. For more information about opening ports in the firewall, see Configuring the Windows Firewall to Allow SQL Server Access in Books Online.

**Dynamic Ports**

At startup, when an instance of SQL Server is configured to listen on dynamic ports, it checks with the operating system for an available port, and opens an endpoint for that port. Incoming connections must specify that port number to connect. Since the port number can change each time SQL Server starts, SQL Server provides the SQL Server Browser Service, to monitor the ports, and direct incoming connections to the current port for that instance.

Using dynamic ports complicates connecting SQL Server through a firewall because the port number may change when SQL Server is restarted, requiring changes to the firewall settings. To avoid connection problems through a firewall, configure SQL Server to use a static port.

**Options**

**Active**

Indicates that the IP address is active on the computer. Not available for **IP All**.

**Enabled**

If the **Listen All** property on the **TCP/IP Properties (Protocol Tab)** is set to **No**, this property indicates whether SQL Server is listening on the IP address. If the **Listen All** property on the **TCP/IP Properties (Protocol Tab)** is set to **Yes**, the property is disregarded. Not available for **IP All**.

**IP Address**

View or change the IP address used by this connection. Lists the IP address used by the computer, and the IP loopback address, 127.0.0.1. Not available for **IP All**. The IP address can be in either IPv4 or IPv6 format.

**TCP Dynamic Ports**

Blank, if dynamic ports are not enabled. To use dynamic ports, set to 0.

For **IP All**, displays the port number of the dynamic port used.

**TCP Port**

View or change the port on which SQL Server listens. By default, the default instance of SQL Server listens on port 1433.

The database engine can listen on multiple ports on the same IP address, list the ports, separated by commas, in the format 1433,1500,1501. This field is limited to 2047 characters.
To configure a single IP address to listen on multiple ports, the **Listen All** parameter must also be set to **No**, on the **Protocols Tab** of the **TCP/IP Properties** dialog box. For more information, see “How to: Configure the Database Engine to Listen on Multiple TCP Ports” in SQL Server Books Online.

**Adding or Removing IP Addresses**

SQL Server Configuration Manager displays the IP addresses that were available at the time SQL Server was installed. The available IP addresses can change when network cards are added or removed, when a dynamically assigned IP address expires, when the network structure is reconfigured, or when the computer changes its physical location such as a laptop computer connecting to the network in a different building. To change an IP address, edit the **IP Address** box, and then restart SQL Server.

**Additional topics in Books Online**

Search MSDN for topics such as **Configure a Server to Listen on a Specific TCP Port (SQL Server Configuration Manager)** and **Configure the Database Engine to Listen on Multiple TCP Ports**.

**See Also**

- Choosing a Network Protocol
- Creating a Valid Connection String Using TCP IP
- SQL Server Browser Service
This section contains the F1 Help topics for the **SQL Server Native Client Configuration** dialogs in Microsoft SQL Server Configuration Manager. SQL Server Native Client is the network library that client computers use to connect to SQL Server, starting with Microsoft SQL Server.

The settings configured in SQL Server Native Client Configuration, are used on the computer running the client program. When configured on the computer running SQL Server, they affect only those client programs running on the server.

These settings do not affect clients connecting to previous versions of SQL Server, unless they are using the client tools starting with SQL Server, such as SQL Server Management Studio.

### In this Section

- **SQL Server Native Client Configuration Properties (Flags Tab)**
- **Client Protocols (SQL Server Configuration Manager)**
  - Client Protocols Properties (Order Tab)
  - Client Protocols - Shared Memory Properties (Protocol Tab)
  - Client Protocols - TCP IP Properties (Protocol Tab)
  - Client Protocols - Named Pipes Properties (Protocol Tab)
- **Aliases (SQL Server Configuration Manager)**
  - New Alias (Alias Tab)
  - <Alias> Properties (Alias Tab)
  - Creating a Valid Connection String Using Shared Memory Protocol
  - Creating a Valid Connection String Using TCP IP
  - Creating a Valid Connection String Using Named Pipes
Microsoft SQL Server clients on this machine, communicate with SQL Server servers using the protocols provided in the SQL Server Native Client library file. This page provides configures the client computer to request an encrypted connection using Secure Sockets Layer (SSL). If an encrypted connection cannot be established, the connection will fail.

The login process is always encrypted. The options below apply only to encrypting data. For more information about how SQL Server encrypts communication and for instructions on how to configure the client to trust the root authority of the server certificate, see “Encrypting Connections to SQL Server” and “How to: Enable Encrypted Connections to the Database Engine (SQL Server Configuration Manager)” in SQL Server Books Online.

Options

**Force protocol encryption**
Request a connection using SSL.

**Trust Server Certificate**
When set to No, the client process attempts to validate the server certificate. The client and server must have each have a certificate issues from a public certification authority. If the certificate is not present on the client computer, or if the validation of the certificate fails, the connection is terminated.

When set to Yes, the client does not validate the server certificate, thereby enabling the use of a self-signed certificate.

**Trust Server Certificate** is only available if **Force protocol encryption** is set to Yes.
To configure the order that protocols are used by the Microsoft SQL Server clients on this computer, right-click **ClientProtocols** in the console pane, and then click **Properties**, or right-click the a protocol in the details pane, and then click **Order**.

To configure the properties of the client protocols used by the SQL Server clients on this computer, in the console pane click **ClientProtocols**, in the details pane right-click the desired protocol, and then click **Properties**.
Use the **Order** page on the **Client Protocols Properties** dialog box to view and enable the client protocols.

Click a protocol, and then click **Enable** or **Disable** to move the selected protocol to the **Disabled Protocols** or **Enabled Protocols** list.

Protocols are tried in the order listed, attempting to connect using the top protocol first, and then the second listed protocol, etc. Move protocols up or down the **Enabled Protocols** list, by clicking the up arrow and down arrow buttons. When connecting to Microsoft SQL Server from a client on that computer, the **Shared Memory** protocol will always be tried first, if enabled.

### NOTE

These settings are not used by Microsoft .NET SqlClient. The protocol order for .NET SqlClient is first TCP, and then named pipes, which cannot be changed.

### Options

**Disabled Protocols**
Lists protocols which are installed but are not currently used.

**Enabled Protocols**
Lists protocols which are available for Microsoft SQL Server clients on this computer.

> Enables the currently highlighted protocol in the **Disabled Protocols** box, moving it to the **Enabled Protocols** box.

< Disables the currently highlighted protocol in the **Enabled Protocols** box, moving it to the **Disabled Protocols** box.

**Up arrow**
Moves the highlighted protocol up in the list. This allows you to increase the priority in which the Net-Library will attempt to use the selected protocol for connections.

**Down arrow**
Moves the highlighted protocol down in the list. This allows you to decrease the priority in which the Net-Library will attempt to use the selected protocol for connections.

**Enable Shared Memory Protocol**
Enables the shared memory protocol which is always tried first (if enabled), when connecting to SQL Server from a client on that computer.

### NOTE

If the protocol is specified through a prefix or as part of the connection string, only the specified protocol is attempted.
In Microsoft SQL Server Configuration Manager use the Protocol tab on the Shared Memory Properties dialog box to view or modify shared memory. For more information about connection strings, see Creating a Valid Connection String Using Shared Memory Protocol.

Options

Enabled
Possible values are Yes and No.

See Also

Choosing a Network Protocol
In Microsoft SQL Server Configuration Manager, use the Protocol tab on the TCP/IP Properties dialog box to view or specify the following options. To connect to a different port, type the port number in the Default Port box. For more information about connection strings, see Creating a Valid Connection String Using TCP IP.

Options

**Default Port**
Specifies the port that the TCP/IP Net-library will use to attempt to connect to the target instance of SQL Server. The default value port is 1433.

When connecting to a default instance of Database Engine, the client uses this value. If a default instance has been configured to listen on a different port, change this value to that port number.

When connecting to a named instance of Database Engine, the client will attempt to obtain the port number from the SQL Server Browser Service running on the server computer. If the SQL Server Browser Service is not running, the port number must be provided through this setting, or as part of the connection string.

**Enabled**
Possible values are Yes and No.

**Keep Alive**
This parameter (in milliseconds) controls how often TCP attempts to verify that an idle connection is still intact by sending a KEEPALIVE packet. The default is 30000 milliseconds.

**Keep Alive Interval**
This parameter (in milliseconds) determines the interval separating KEEPALIVE retransmissions until a response is received. The default is 1000 milliseconds.

See Also

Choosing a Network Protocol
New Alias (Alias Tab)
<Alias> Properties (Alias Tab)
In Microsoft SQL Server Configuration Manager use the **Protocol** tab on the **Named Pipes Properties** dialog box to view or modify the description of default pipe. To connect to a different pipe, type the pipe in the **Default Pipe** box. For more information about connection strings, see [Creating a Valid Connection String Using Named Pipes](#).

### Options

**Default Pipe**

Specifies the default pipe the Named Pipes Net-library will use to attempt to connect to the target instance of SQL Server. By default, SQL Server listens on: `\\pipe\sql\query`

To connect to the default pipe, enter `sql\query`

**Enabled**

Possible values are **Yes** and **No**.

### See Also

[Choosing a Network Protocol](#)
An alias is an alternate name that can be used to make a connection. The alias encapsulates the required elements of a connection string, and exposes them with a name chosen by the user. To create an alias for the Microsoft SQL Server clients on this computer, right-click **Aliases** in the console pane, and then click **New Alias**. To configure an existing alias for the SQL Server clients on this computer, click **Aliases** in the console pane, right-click the desired existing alias in the details pane, and then click **Properties**.
An alias is an alternate name that can be used to make a connection. The alias encapsulates the required elements of a connection string, and exposes them with a name chosen by the user. Use the Alias page on the Alias - New dialog box to specify the elements of the connection string for an alias. To change the connection string of an existing alias, see `<Alias> Properties (Alias Tab)`.

All values in the Properties grid do not have to be completed. Valid combinations vary depending on the protocol selected. See the topics listed below for examples of valid combinations.

**Alias Name**
The name (alias) that you want to use to refer to this connection.

**Pipe Name / Port No**
Additional elements of the connection string. The name of this box varies with the selected protocol.

**Protocol**
The protocol used for the connection.

**Server**
The name of the Microsoft SQL Server instance being connected to.

### When to Use an Alias

By default, SQL Server connects to a local instance of SQL Server using the Shared Memory protocol, and to an instance of SQL Server on another computer using either TCP/IP or Named Pipes. Create an alias when you are using TCP/IP or named pipes, and you want to provide a customized connection string, or when you want to use a name other than the server name for the connection.

**Examples**

- SQL Server is not listening on the default TCP/IP port of 1433 so you want to provide a connection string with a different port number.

- SQL Server is not listening on the default named pipe so you want to provide a connection string with a different pipe name.

- An application expects to connect to a database on the server named `ACCT`, but that database has been consolidated as an instance named `ACCT` on a server named `CENTRAL`. The application cannot easily be changed. Create an alias named `ACCT`, with a connection string pointing to `CENTRAL\ACCT`.

### Creating a Valid Connection String

See the following topics for a description and examples of valid combinations of alias properties:

- Creating a Valid Connection String Using Shared Memory Protocol
- Creating a Valid Connection String Using TCP IP
- Creating a Valid Connection String Using Named Pipes
An alias is an alternate name that can be used to make a connection. The alias encapsulates the required elements of a connection string, and exposes them with a name chosen by the user. Use the Alias page on the <Alias name> Properties dialog box to view or specify the elements of connection string of an Alias.

Options

**Alias Name**
The name (alias) that you want to use to refer to this connection.

**Pipe Name / Port No**
Additional elements of the connection string. The name of this box varies with the selected protocol. See the topics listed below for examples.

**Protocol**
The protocol used for the connection.

**Server**
The name of the Microsoft SQL Server instance being connected to.

See Also

- Creating a Valid Connection String Using Shared Memory Protocol
- Creating a Valid Connection String Using TCP IP
- Creating a Valid Connection String Using Named Pipes
Creating a Valid Connection String Using Shared Memory Protocol

3/24/2017 • 1 min to read • Edit Online

Connections to Microsoft SQL Server from a client running on the same computer use the shared memory protocol. Shared memory has no configurable properties. Shared memory is always tried first, and cannot be moved from the top position of the Enabled Protocols list in the Client Protocols Properties list. The Shared Memory protocol can be disabled, which is useful when troubleshooting one of the other protocols.

You cannot create an alias using the shared memory protocol, but if shared memory is enabled, then connecting to the Database Engine by name, creates a shared memory connection. A shared memory connection string uses the format `lpc:<servername>[\instancename]`.

**Connecting to the Local Server**

When connecting to SQL Server running on the same computer as the client, you can use (local) as the server name. This is not encouraged as it leads to ambiguity, however it can be useful when the client is known to be running on the intended computer. For instance, when creating an application for mobile disconnected users, such as a sales force, where SQL Server will run on laptop computers and store project data, a client connecting to (local) would always connect to the SQL Server running on the laptop. The word localhost or a period (.) can be used in place of (local).

**Verifying your Connection Protocol**

The following query will return the protocol used for the current connection.

```sql
SELECT net_transport
FROM sys.dm_exec_connections
WHERE session_id = @@SPID;
```

**Examples:**

The following names will connect to the local computer with the shared memory protocol if it is enabled:

- `<servername>`
- `<servername>\<instancename>`
- `(local)`
- `localhost`

You cannot create an alias for a shared memory connection.

**NOTE**

Specifying an IP Address in the Server box will result in a TCP/IP connection.

See Also
Creating a Valid Connection String Using TCP IP
Creating a Valid Connection String Using Named Pipes
Choosing a Network Protocol
To create a valid connection string using TCP/IP, you must:

- Specify an **Alias Name**.
- For the **Server**, enter either a server name to which you can connect using the **PING** utility, or an IP address to which you can connect using the **PING** utility. For a named instance append the instance name.
- Specify **TCP/IP** for the **Protocol**.
- Optionally, enter a port number for the **Port No**. The default is 1433, which is the port number of the default instance of the Database Engine on a server. To connect to a named instance or a default instance that is not listening on port 1433, you must provide the port number, or start the SQL Server Browser service. For information on configuring the SQL Server Browser service, see [SQL Server Browser Service](#).

At the time of connection, the SQL Server Native Client component reads the server, protocol, and port values from the registry for the specified alias name, and creates a connection string in the format `tcp:<servername>[<instancename>],[port]` or `tcp:<IPAddress>[<instancename>],[port]`.

**NOTE**

The Microsoft Windows Firewall closes port 1433 by default. Because Microsoft SQL Server communicates over port 1433, you must reopen the port if SQL Server is configured to listen for incoming client connections using TCP/IP. For information on configuring a firewall, see “How to: Configure a Firewall for SQL Server Access” in SQL Server Books Online or review your firewall documentation.

SQL Server and SQL Server Native Client fully support both Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6). SQL Server Configuration Manager accepts both IPv4 and IPv6 formats for IP addresses. For information on IPv6, see “Connecting Using IPv6” in SQL Server Books Online.

### Connecting to the Local Server

When connecting to SQL Server running on the same computer as the client, you can use `(local)` as the server name. This is not encouraged as it leads to ambiguity, however it can be useful when the client is known to be running on the intended computer. For instance, when creating an application for mobile disconnected users, such as a sales force, where SQL Server will run on laptop computers and store project data, a client connecting to `(local)` would always connect to the SQL Server running on the laptop. The word `localhost` or a period (`.`) can be used in place of `(local)`.

### Verifying Your Connection Protocol

The following query returns the protocol used for the current connection.

```sql
SELECT net_transport
FROM sys.dm_exec_connections
WHERE session_id = @@SPID;
```

### Examples
### Connecting by server name:

<table>
<thead>
<tr>
<th>Alias Name</th>
<th>&lt;serveralias&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port No</td>
<td>&lt;blank&gt;</td>
</tr>
<tr>
<td>Protocol</td>
<td>TCP/IP</td>
</tr>
<tr>
<td>Server</td>
<td>&lt;servername&gt;</td>
</tr>
</tbody>
</table>

### Connecting by server name to a named instance:

<table>
<thead>
<tr>
<th>Alias Name</th>
<th>&lt;serveralias&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port No</td>
<td>&lt;blank&gt;</td>
</tr>
<tr>
<td>Protocol</td>
<td>TCP/IP</td>
</tr>
<tr>
<td>Server</td>
<td>&lt;servername&gt;&lt;instancename&gt;</td>
</tr>
</tbody>
</table>

### Connecting by server name to a specified port:

<table>
<thead>
<tr>
<th>Alias Name</th>
<th>&lt;serveralias&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port No</td>
<td>&lt;port&gt;</td>
</tr>
<tr>
<td>Protocol</td>
<td>TCP/IP</td>
</tr>
<tr>
<td>Server</td>
<td>&lt;servername&gt;</td>
</tr>
</tbody>
</table>

### Connecting by IP address:

<table>
<thead>
<tr>
<th>Alias Name</th>
<th>&lt;serveralias&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port No</td>
<td>&lt;blank&gt;</td>
</tr>
<tr>
<td>Protocol</td>
<td>TCP/IP</td>
</tr>
<tr>
<td>Server</td>
<td>&lt;IPAddress&gt;</td>
</tr>
</tbody>
</table>

### Connecting by IP address to a named instance:

<table>
<thead>
<tr>
<th>Alias Name</th>
<th>&lt;serveralias&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port No</td>
<td>&lt;blank&gt;</td>
</tr>
<tr>
<td>Protocol</td>
<td>TCP/IP</td>
</tr>
<tr>
<td>Server</td>
<td>&lt;IPAddress&gt;&lt;instancename&gt;</td>
</tr>
</tbody>
</table>

### Connecting by IP address to a specified port:

<table>
<thead>
<tr>
<th>Alias Name</th>
<th>&lt;serveralias&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port No</td>
<td>&lt;port number&gt;</td>
</tr>
<tr>
<td>Protocol</td>
<td>TCP/IP</td>
</tr>
<tr>
<td>Server</td>
<td>&lt;IPAddress&gt;</td>
</tr>
</tbody>
</table>

### Connecting to the local computer using (local):

<table>
<thead>
<tr>
<th>Alias Name</th>
<th>&lt;serveralias&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port No</td>
<td>&lt;blank&gt;</td>
</tr>
<tr>
<td>Protocol</td>
<td>TCP/IP</td>
</tr>
<tr>
<td>Server</td>
<td>(local)</td>
</tr>
</tbody>
</table>

### Connecting to the local computer using localhost:

<table>
<thead>
<tr>
<th>Alias Name</th>
<th>&lt;serveralias&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port No</td>
<td>&lt;blank&gt;</td>
</tr>
<tr>
<td>Protocol</td>
<td>TCP/IP</td>
</tr>
<tr>
<td>Server</td>
<td>localhost</td>
</tr>
</tbody>
</table>
Connecting to a named instance on the local computer:

Alias Name         <serveralias>
Port No            <blank>
Protocol           TCP/IP
Server             localhost

Connecting to the local computer using a period:

Alias Name         <serveralias>
Port No            <blank>
Protocol           TCP/IP
Server             .

Connecting to a named instance on the local computer using a period:

Alias Name         <serveralias>
Port No            <blank>
Protocol           TCP/IP
Server             .\<instancename>

NOTE
For information on specifying the network protocol as a sqlcmd parameter, see "How to: Connect to the Database Engine Using sqlcmd.exe" in SQL Server Books Online.

See Also
Creating a Valid Connection String Using Shared Memory Protocol
Creating a Valid Connection String Using Named Pipes
Choosing a Network Protocol
The SQL Server Browser program runs as a Windows service. SQL Server Browser listens for incoming requests for Microsoft SQL Server resources and provides information about SQL Server instances installed on the computer. SQL Server Browser contributes to the following actions:

- Browsing a list of available servers
- Connecting to the correct server instance
- Connecting to dedicated administrator connection (DAC) endpoints

For each instance of the Database Engine and SSAS, the SQL Server Browser service (sqlbrowser) provides the instance name and the version number. SQL Server Browser is installed with SQL Server.

SQL Server Browser can be configured during setup or by using SQL Server Configuration Manager. By default, the SQL Server Browser service starts automatically:

- When upgrading an installation.
- When installing on a cluster.
- When installing a named instance of the Database Engine including all instances of SQL Server Express.
- When installing a named instance of Analysis Services.

Background

Prior to SQL Server 2000, only one instance of SQL Server could be installed on a computer. SQL Server listened for incoming requests on port 1433, assigned to SQL Server by the official Internet Assigned Numbers Authority (IANA). Only one instance of SQL Server can use a port, so when SQL Server 2000 introduced support for multiple instances of SQL Server, SQL Server Resolution Protocol (SSRP) was developed to listen on UDP port 1434. This listener service responded to client requests with the names of the installed instances, and the ports or named pipes used by the instance. To resolve limitations of the SSRP system, SQL Server 2005 introduced the SQL Server Browser service as a replacement for SSRP.

How SQL Server Browser Works

When an instance of SQL Server starts, if the TCP/IP protocol is enabled for SQL Server, the server is assigned a TCP/IP port. If the named pipes protocol is enabled, SQL Server listens on a specific named pipe. This port, or "pipe," is used by that specific instance to exchange data with client applications. During installation, TCP port 1433 and pipe \sql\query are assigned to the default instance, but those can be changed later by the server administrator using SQL Server Configuration Manager. Because only one instance of SQL Server can use a port or pipe, different port numbers and pipe names are assigned for named instances, including SQL Server Express. By default, when enabled, both named instances and SQL Server Express are configured to use dynamic ports, that is, an available port is assigned when SQL Server starts. If you want, a specific port can be assigned to an instance of SQL Server. When connecting, clients can specify a specific port; but if the port is dynamically assigned, the port number can change anytime SQL Server is restarted, so the correct port number is unknown to the client.

Upon startup, SQL Server Browser starts and claims UDP port 1434. SQL Server Browser reads the registry, identifies all instances of SQL Server on the computer, and notes the ports and named pipes that they use. When
a server has two or more network cards, SQL Server Browser returns the first enabled port it encounters for SQL Server. SQL Server Browser support ipv6 and ipv4.

When SQL Server clients request SQL Server resources, the client network library sends a UDP message to the server using port 1434. SQL Server Browser responds with the TCP/IP port or named pipe of the requested instance. The network library on the client application then completes the connection by sending a request to the server using the port or named pipe of the desired instance.

For information about starting and stopping the SQL Server Browser service, see SQL Server Books Online.

Using SQL Server Browser

If the SQL Server Browser service is not running, you are still able to connect to SQL Server if you provide the correct port number or named pipe. For instance, you can connect to the default instance of SQL Server with TCP/IP if it is running on port 1433.

However, if the SQL Server Browser service is not running, the following connections do not work:

- Any component that tries to connect to a named instance without fully specifying all the parameters (such as the TCP/IP port or named pipe).
- Any component that generates or passes server\instance information that could later be used by other components to reconnect.
- Connecting to a named instance without providing the port number or pipe.
- DAC to a named instance or the default instance if not using TCP/IP port 1433.
- The OLAP redirector service.
- Enumerating servers in SQL Server Management Studio, Enterprise Manager, or Query Analyzer.

If you are using SQL Server in a client-server scenario (for example, when your application is accessing SQL Server across a network), if you stop or disable the SQL Server Browser service, you must assign a specific port number to each instance and write your client application code to always use that port number. This approach has the following problems:

- You must update and maintain client application code to ensure it is connecting to the proper port.
- The port you choose for each instance may be used by another service or application on the server, causing the instance of SQL Server to be unavailable.

Clustering

SQL Server Browser is not a clustered resource and does not support failover from one cluster node to the other. Therefore, in the case of a cluster, SQL Server Browser should be installed and turned on for each node of the cluster. On clusters, SQL Server Browser listens on IP_ANY.

NOTE

When listening on IP_ANY, when you enable listening on specific IPs, the user must configure the same TCP port on each IP, because SQL Server Browser returns the first IP/port pair that it encounters.

Installing, Uninstalling, and Running from the Command Line

By default, the SQL Server Browser program is installed at C:\Program Files (x86)\Microsoft SQL Server\90\Shared\sqlbrowser.exe.
The SQL Server Browser service is uninstalled when the last instance of SQL Server is removed.

SQL Server Browser can be started from the command prompt for troubleshooting, by using the -c switch:

```
<drive>\<path>\sqlbrowser.exe -c
```

### Security

#### Account Privileges

SQL Server Browser listens on a UDP port and accepts unauthenticated requests by using SQL Server Resolution Protocol (SSRP). SQL Server Browser should be run in the security context of a low privileged user to minimize exposure to a malicious attack. The logon account can be changed by using the SQL Server Configuration Manager. The minimum user rights for SQL Server Browser are the following:

- Deny access to this computer from the network
- Deny logon locally
- Deny Log on as a batch job
- Deny Log On Through Terminal Services
- Log on as a service
- Read and write the SQL Server registry keys related to network communication (ports and pipes)

#### Default Account

Setup configures SQL Server Browser to use the account selected for services during setup. Other possible accounts include the following:

- Any domain\local account
- The local service account
- The local system account (not recommended as has unnecessary privileges)

#### Hiding SQL Server

Hidden instances are instances of SQL Server that support only shared memory connections. For SQL Server, set the HideInstance flag to indicate that SQL Server Browser should not respond with information about this server instance.

#### Using a Firewall

To communicate with the SQL Server Browser service on a server behind a firewall, open UDP port 1434, in addition to the TCP port used by SQL Server (e.g., 1433). For information about working with a firewall, see "How to: Configure a Firewall for SQL Server Access" in SQL Server Books Online.

### See Also

Network Protocols and Network Libraries
SQL Server and SQL Server Native Client fully support both Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6). When Windows is configured with IPv6 SQL Server, components automatically recognize the existence of IPv6. No special SQL Server configuration is necessary.

Support includes but is not limited to the following:

- The SQL Server Database Engine and the other server components can listen on both IPv4 and IPv6 addresses at the same time. When both IPv4 and IPv6 are present, you can use SQL Server Configuration Manager to configure the Database Engine to listen only on IPv4 addresses or only on IPv6 addresses.

- When the SQL Server Browser service running on a machine that supports both IPv4 and IPv6 is queried on an IPv4 address, it responds with an IPv4 address and the first IPv4 TCP port in its list. When queried on an IPv6 address, it responds with an IPv6 address and the first IPv6 TCP port in its list. To avoid inconsistency, we recommend that the IPv4 and IPv6 listeners be configured to listen to the same port.

- Tools such as SQL Server Management Studio and SQL Server Configuration Manager accept both IPv4 and IPv6 formats for IP addresses. In most cases, the connection string does not need to be modified if the `<computer_name>\<instance_name>` is specified using server hostname or fully qualified domain name (FQDN). If the server computer has both IPv4 and IPv6, its hostname or FQDN will be resolved into multiple IP addresses, including at least one IPv4 address and multiple IPv6 addresses. SQL Server Native Client attempts to establish connections using these IP addresses in the order received from TCP/IP and uses the first connection that succeeds. Because the order cannot be predicted by SQL Server Native Client, this should be regarded as random order. IPv4 addresses are attempted first if both IPv4 and IPv6 addresses are present. This logic is transparent to the users of ODBC, OLE DB, or ADO.NET.

**NOTE**

If the Database Engine is not listening on IPv4, the attempted IPv4 connection must wait for the time-out period before the IPv6 address is attempted. To avoid this, connect directly to the IPv6 IP address or configure an alias on the client with the IPv6 address.

**See Also**

SQL Server Configuration Manager
SQL Server logs certain system events and user-defined events to the SQL Server error log and the Microsoft Windows application log. Both logs automatically timestamp all recorded events. Use the information in the SQL Server error log to troubleshoot problems related to SQL Server.

The Windows application log provides an overall picture of events that occur on the Windows operating system, as well as events in SQL Server and SQL Server Agent. Use the Windows Event Viewer to view the Windows application log and to filter the information. For example, you can filter events, such as information, warning, error, success audit, and failure audit.

Comparing Error and Application Log Output

You can use both the SQL Server error log and the Windows application log to identify the cause of problems. For example, while monitoring the SQL Server error log, you may encounter error messages that do not contain cause information. By comparing the dates and times for events between these logs, you can narrow the list of probable causes. The SQL Server Management Studio Log File Viewer lets you integrate SQL Server, SQL Server Agent, and the Windows logs into a single list, making it easy to understand related server events and SQL Server events. For more information, see the topic “Log File Viewer” in SQL Server Books Online.

<table>
<thead>
<tr>
<th>TOPIC</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewing the SQL Server Error Log</td>
<td>Contains information about the SQL Server error log and how to view it.</td>
</tr>
<tr>
<td>Viewing the Windows Application Log</td>
<td>Contains information about the Windows application log and how to view it.</td>
</tr>
</tbody>
</table>
Viewing the SQL Server Error Log

3/24/2017 • 1 min to read • Edit Online

View the SQL Server error log to ensure that processes have completed successfully (for example, backup and restore operations, batch commands, or other scripts and processes). This can be helpful to detect any current or potential problem areas, including automatic recovery messages (particularly if an instance of SQL Server has been stopped and restarted), kernel messages, or other server-level error messages.

View the SQL Server error log by using SQL Server Management Studio or any text editor. For more information about how to view the error log, see Open Log File Viewer. By default, the error log is located at `Program Files\Microsoft SQL Server\MSSQL.\MSSQL\LOG\ERRORLOG` and `ERRORLOG.n` files.

A new error log is created each time an instance of SQL Server is started, although the `sp_cycle_errorlog` system stored procedure can be used to cycle the error log files without having to restart the instance of SQL Server. Typically, SQL Server retains backups of the previous six logs and gives the most recent log backup the extension .1, the second most recent the extension .2, and so on. The current error log has no extension.

Be aware that you can also view the SQL Server error log on instances of SQL Server that are offline or cannot start. For more information, see View Offline Log Files.
Viewing the Windows Application Log

When SQL Server is configured to use the Microsoft Windows application log, each SQL Server session writes new events to that log. Unlike the SQL Server error log, a new application log is not created each time you start an instance of SQL Server.

View and manage the Windows application log by using Windows Event Viewer or the Log Viewer in SQL Server Management Studio.

There are three logs that can be viewed with Event Viewer.

<table>
<thead>
<tr>
<th>WINDOWS LOG TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>System log</td>
<td>Records events logged by the Windows operating system components. For example, the failure of a driver or other system component to load during startup is recorded in the system log.</td>
</tr>
</tbody>
</table>
| Security log       | Records security events, such as failed login attempts. This helps track changes to the security system and identify possible breaches to security. For example, attempts to log on to the system may be recorded in the security log, depending on the audit settings in the User Manager.
|                    | Only members of the **sysadmin** fixed server role can view the security log. |
| Application log    | Records events that are logged by applications. For example, a database application might record a file error in the application log. |

For more information about using Event Viewer, managing the application log, and understanding the information it presents, see the Windows documentation.

**To view the Windows application log**

View the Windows Application Log (Windows)