

How to Troubleshoot Connecting to the SQL Server Database Engine



This is an exhaustive list of troubleshooting techniques to use when you cannot connect to the SQL Server Database Engine. These steps are not in the order of the most likely problems which you probably already tried. These steps are in order of the most basic problems to more complex problems. These steps assume that you are connecting to SQL Server from another computer by using the TCP/IP protocol, which is the most common situation. These steps are written for SQL Server 2008 R2 with a client running Windows 7, however the steps generally apply to other versions of SQL Server and other operating systems with only slight modifications.

These instructions are particularly useful when troubleshooting the "**Connect to Server**" error, which can be Error Number: 11001 (or 53), Severity: 20, State: 0

"A network-related or instance-specific error occurred while establishing a connection to SQL Server. The server was not found or was not accessible. Verify that the instance name is correct and that SQL Server is configured to allow remote connections. "

"(provider: Named Pipes Provider, error: 40 - Could not open a connection to SQL Server) (Microsoft SQL Server, Error: 53)" or "(provider: TCP Provider, error: 0 - No such host is known.) (Microsoft SQL Server, Error: 11001)"

This error usually means that the SQL Server computer can't be found or that the TCP port number is either not known, or is not the correct port number, or is blocked by a firewall.

Not included

This topic does not include information about SSPI errors. For SSPI errors, see [How to troubleshoot the "Cannot generate SSPI context" error message](#) 

This topic does not include information about Kerberos errors. For help, see [Microsoft Kerberos Configuration Manager for SQL Server](#) 

This topic does not include information about SQL Azure Connectivity. For more information, see [SQL Azure Connectivity Troubleshooting Guide](#) 

Table of Contents

- [Not included](#)
- [Gathering Information about the Instance of SQL Server](#)
- [Enable Protocols](#)
- [Testing TCP/IP Connectivity](#)
- [Testing a Local Connection](#)
- [Opening a Port in the Firewall](#)
- [Testing the Connection](#)
- [See Also](#)

Gathering Information about the Instance of SQL Server

1. Confirm the instance of the SQL Server Database Engine is installed and running.
 1. Logon to the computer hosting the instance of SQL Server.
2. On the Start menu, point to **All Programs**, point to **Microsoft SQL Server 2008 R2**, point to **Configuration Tools**, and then click **SQL Server Configuration Manager**.

3. Using Configuration Manager, in the left pane select **SQL Server Services**. In the right-pane confirm that the instance of the Database Engine is present and running. The instance named **MSSQLSERVER** is a default (unnamed) instance. There can only be one default instance. Other (named) instances will have their names listed between the parentheses. SQL Server Express uses the name **SQLEXPRESS** as the instance name unless someone named it something else during setup. Make a note of the name of the instance that you are trying to connect to. Also, confirm that the instance is running, by looking for the green arrow. If the instance has a red square, right-click the instance and then click **Start**. It should turn green.
 4. If you are attempting to connect to a named instance, make sure the SQL Server Browser service is running.
2. Get the IP Address of the computer.
 1. On the Start menu, click **Run**. In the **Run** window type **cmd**, and then click **OK**.
 2. In the command prompt window, type **ipconfig** and then press enter. Make a note of the **IPv4** Address and the **IPv6** Address. (SQL Server can connect using the older IP version 4 protocol or the newer IP version 6 protocol. Your network could allow either or both. Most people start by troubleshooting the **IPv4** address. It's shorter and easier to type.)
 3. Get the TCP port number used by SQL Server. In most cases you are connecting to the Database Engine from another computer using the TCP protocol.
 1. Using SQL Server Management Studio on the computer running SQL Server, connect to the instance of SQL Server. In Object Explorer, expand **Management**, expand **SQL Server Logs**, and then double-click the current log.
 2. In the Log Viewer, click the **Filter** button on the toolbar. In the **Message contains text** box, type **server is listening on**, click **Apply filter**, and then click **OK**.
 3. A message similar to **Server is listening on ['any' <ipv4> 1433]** should be listed. This message indicates that this instance of SQL Server is listening on all the computers IP Addresses (for IP version 4) and is listening to TCP port 1433. (TCP port 1433 is usually the port used by the Database Engine. Only one instance of SQL Server can use a port, so if there is more than one instance of SQL Server installed, some instances must use other port numbers.)
Note: IP address 127.0.0.1 is probably listed. It is called the loopback adapter address and can only be connected to from processes on the same computer. It can be useful for troubleshooting, but you can't use it to connect from another computer.

Enable Protocols

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

1. On the Start menu, point to **All Programs**, point to **Microsoft SQL Server 2008 R2**, point to **Configuration Tools**, and then click **SQL Server Configuration Manager**.
2. Using Configuration Manager, in the left pane expand **SQL Server Network Configuration** (or **SQL Server Network Configuration (32bit)**), and then select the instance of SQL Server that you want to connect to. The right-pane lists the connection protocols available. Shared Memory is normally enabled. It can only be used from the same computer, so most installations leave Shared Memory enabled. To connect to SQL Server from another computer you will normally use TCP/IP. If TCP/IP is not enabled, right-click **TCP/IP**, and then click **Enable**.
3. If you changed the enabled setting for any protocol you must restart the Database Engine. In the left pane select **SQL Server Services**. In the right-pane, right-click the instance of the Database Engine, and then click **Restart**.

Testing TCP/IP Connectivity

Connecting to SQL Server by using TCP/IP requires that Windows can establish the connection.

1. On the Start menu, click **Run**. In the **Run** window type **cmd**, and then click **OK**.
2. In the command prompt window, type ping and then the IP Address of the computer that is running SQL Server. For example, **ping 192.168.1.101** using an IPv4 address, or **ping fe80::d51d:5ab5:6f09:8f48%11** using an IPv6 address. (You must replace the numbers after ping with the ip addresses on your computer.)
3. If your network is properly configured you will receive a response such as **Reply from <IP address>**. If you receive an error such as "**Destination host unreachable**." or "**Request timed out**." then TCP/IP is not correctly configured. (Check that the IP address was correct and was correctly typed.) Errors at this point could indicate a problem with the client computer, the server computer, or something about the network such as a router. For more information, see [How to Troubleshoot Basic TCP/IP Problems](#).

- Next, if the ping test succeeded using the IP address, test that the computer name can be resolved to the TCP/IP address. On the client computer, in the command prompt window, type **ping** and then the computer name of the computer that is running SQL Server. For example, **ping newofficepc**.
- If you receive an error such as "**Destination host unreachable**," or "**Request timed out**," you might have old (stale) name resolution information cached on the client computer. Type **ipconfig /flushdns** to clear the DNS (Dynamic Name Resolution) cache. Then ping the computer by name again. With the DNS cache empty, the client computer will check for the newest information about the IP address for the server computer.
- If your network is properly configured you will receive a response such as **Reply from <IP address>**. If you can successfully ping the server computer by IP address but receive an error such as "**Destination host unreachable**," or "**Request timed out**," when pinging by computer name, then name resolution is not correctly configured. (For more information, see [How to Troubleshoot Basic TCP/IP Problems](#).) Successful name resolution is not required to connect to SQL Server, but if the computer name cannot be resolved, then connections must be made specifying the IP address. This is not ideal, but name resolution can be fixed later.

Testing a Local Connection

Before troubleshooting a connection problem from another computer, first test your ability to connect from a client application on the computer that is running SQL Server. This procedure uses SQL Server Management Studio. Management Studio might not have been installed when you installed the Database Engine. You can install Management Studio from the SQL Server CD by running setup and selecting the Management Tools option. If you are running SQL Server Express, you can download the free SQL Server Management Studio Express from <http://www.microsoft.com/downloads/en/details.aspx?FamilyID=08e52ac2-1d62-45f6-9a4a-4b76a8564a2b>. (If Management Studio is not available you can test the connection using the **sqlcmd.exe** utility which is installed with the Database Engine.)

1. Logon to the computer where SQL Server is installed, using a login that has permission to access SQL Server. (SQL Server 2008 installation requires at least one login to be specified as a SQL Server Administrator. If you do not know an administrator, see [Troubleshooting: Connecting to SQL Server When System Administrators Are Locked Out](#).)

2. On the Start menu, point to **All Programs**, point to **Microsoft SQL Server 2008 R2**, and then click **SQL Server Management Studio**.

3. In the **Connect to Server** dialog box, in the **Server type** box, select **Database Engine**. In the **Authentication** box, select **Windows Authentication**. In the **Server name** box, type one of the following:

| Connecting to: | Type: | Example: |
|------------------|---------------------------------|-----------------|
| Default instance | The computer name | ACCNT27 |
| Named Instance | The computer name\instance name | ACCNT27\PAYROLL |

Note: When connecting to a SQL Server from a client application on the same computer, the shared memory protocol is used. Shared memory is a type of local named pipe, so sometimes errors regarding pipes are encountered.

If you receive an error at this point, you will have to resolve it before proceeding. There are many possible things that could be a problem. Your login might not be authorized to connect. Your default database might be missing.

Note: Some error messages passed to the client intentionally do not give enough information to troubleshoot the problem. This is a security feature to avoid providing an attacker with information about SQL Server. To view the complete information about the error, look in the SQL Server error log. The details are provided there. If you are receiving error 18456 "Login failed for user", Books Online topic <http://msdn.microsoft.com/en-us/library/cc645917.aspx> contains additional information about error codes. And Aaron Bertrand's blog has a very extensive list of error codes at http://www2.sqlblog.com/blogs/aaron_bertrand/archive/2011/01/14/sql-server-v-next-denali-additional-states-for-error-18456.aspx.

4. If you can connect using shared memory, test connecting using TCP. You can force a TCP connection by specifying **tcp:** before the name. For example:

| Connecting to: | Type: | Example: |
|------------------|--------------------------------------|---------------------|
| Default instance | tcp: The computer name | tcp:ACCNT27 |
| Named Instance | tcp: The computer name/instance name | tcp:ACCNT27\PAYROLL |

If you can connect with shared memory but not TCP, then you must fix the TCP problem. The most likely issue is that TCP is not enabled. To enable TCP, See the **Enable Protocols** steps above.

5. If your goal is to connect with an account other than an administrator account, once you can connect as an administrator, try the connection again using the Windows Authentication login or the SQL Server Authentication login that the client application will be using.

Opening a Port in the Firewall

Beginning with Windows XP Service Pack 2, the Windows firewall is turned on and will block connections from another computer. To connect using TCP/IP from another computer, on the SQL Server computer you must configure the firewall to allow connections to the TCP port used by the Database Engine. If you are connecting to a named instance or a port other than TCP port 1433, you must also open the UDP port 1434 for the SQL Server Browser service. For step by step instruction on opening a port in the Windows firewall, see [How to: Configure a Windows Firewall for Database Engine Access](#).

Testing the Connection

Once you can connect using TCP on the same computer, it's time to try connecting from the client computer. You could theoretically use any client application, but to avoid additional complexity, install the SQL Server Management tools on the client and make the attempt using SQL Server Management Studio.

1. On the client computer, using SQL Server Management Studio, attempt to connect using the IP Address and the TCP port number in the format IP address comma port number. For example, **192.168.1.101,1433** If this doesn't work, then you probably have one of the following problems:

- Ping of the IP address doesn't work, indicating a general TCP configuration problem. Go back to the section **Testing TCP/IP Connectivity**.
- SQL Server is not listening on the TCP protocol. Go back to the section **Enable Protocols**.
- SQL Server is listening on a port other than the port you specified. Go back to the section **Gathering Information about the Instance of SQL Server**.
- The SQL Server TCP port is being blocked by the firewall. Go back to the section **Opening a Port in the Firewall**.

2. Once you can connect using the IP address and port number, attempt to connect using the IP address without a port number. For a default instance, just use the IP address. For a named instance, use the IP address and the instance name in the format IP address backslash instance name, for example **192.168.1.101\PAYROLL** If this doesn't work, then you probably have one of the following problems:

- If you are connecting to the default instance, it might be listening on a port other than TCP port 1433, and the client isn't attempting to connect to the correct port number.
- If you are connecting to a named instance, the port number is not being returned to the client.

Both of these problems are related to the SQL Server Browser service, which provides the port number to the client. The solutions are:

- Start the SQL Server Browser service. Go back to the section **Gathering Information about the Instance of SQL Server**, section 1.b.
- The SQL Server Browser service is being blocked by the firewall. Open UDP port 1434 in the firewall. Go back to the section **Opening a Port in the Firewall**.
- The UDP port 1434 information is being blocked by a router. UDP communication (datagrams) are not designed to pass through routers. This keeps the network from getting filled with low priority traffic. You might be able to configure your router to forward UDP traffic, or you can decide to always provide the port number when you connect.
- If the client computer is using Window 7 or Windows Server 2008, (or a more recent operating system,) the UDP traffic might be dropped by the client operating system because the response from the server is returned from a different IP address than was queried. This is a security feature blocking "loose source mapping." For more information, see the **Multiple Server IP Addresses** section of the Books Online topic [Troubleshooting: Timeout Expired](#). You might be able to configure the client to use the correct IP address, or you can decide to always provide the port number when you connect.

3. Once you can connect using the IP address (or IP address and instance name), attempt to connect using the computer name (or computer name and instance name). Put **tcp:** in front of the computer name to force a TCP/IP connection. For example, for a default instance use something like **tcp:ACCNT27** For a named instance use something like **tcp:ACCNT27\PAYROLL** If you could connect using the IP address but not using the computer name, then you have a name resolution problem. Go back to the section **Testing TCP/IP Connectivity**, section 4.

4. Once you can connect using the computer name forcing TCP, attempt connecting using the computer name but not forcing TCP. For example, for a default instance use just the computer name such as **ACCNT27** For a named instance use the computer name and instance name like **ACCNT27\PAYROLL** If you could connect using while forcing TCP, but not without forcing TCP, then the client is probably using another protocol (such as named pipes).

a. On the client computer, using SQL Server Configuration Manager, in the left-pane expand **SQL Native Client 10.0 Configuration**, and then select **Client Protocols**.

b. On the right-pane, Make sure TCP/IP is enabled. If TCP/IP is disabled, right-click **TCP/IP** and then click **Enable**.

c. Make sure that the protocol order for TCP/IP is a smaller number than the named pipes or VIA protocols. Generally you should leave Shared Memory as order 1 and TCP/IP as order 2. Shared memory is only used when the client and SQL Server are running on the same computer. All enabled protocols are tried in order until one succeeds, except that shared memory is skipped when the connection is not to the same computer.

See Also

Another important place to find a huge amount of SQL Server General & Database Engine related articles is the TechNet Wiki itself. The best entry point is [SQL Server General & Database Engine Resources on the TechNet Wiki](#)