SQL SERVER EXPRESS EDITIONS

MAINTENANCE AND PM COORDINATOR SUPPORT DOCUMENT

ALPHA SIMPLICITY SOFTWARE TECHNOLOGIES

Table of Contents

SQL SERVER EXPRESS

Disclaimer: The SQL Server Express editions of our products are not as easy to network as their MS Access counterparts. It is highly recommended that you have knowledgeable people on staff that are highly efficient with SQL Server as we will not be providing much help with this area of the software.

OVERVIEW

The purpose of this document is to try and assist our users that have purchased the SQL Server Express Editions of our products to get them up and functioning correctly with these databases.

How to Install SQL Server 2012 Express

Starting with the .NET versions of Maintenance Coordinator and PM Coordinator, the system now uses SQL Express database engine in place of Microsoft Access databases. These databases are much more robust than their Access counterparts and can support databases up to 10GB in size. Access database could only maintain 2GB.

In general, you should be able to use the default settings as offered in the setup application. The only items that may require special attention is when you're upgrading from an earlier version of SQL Express.

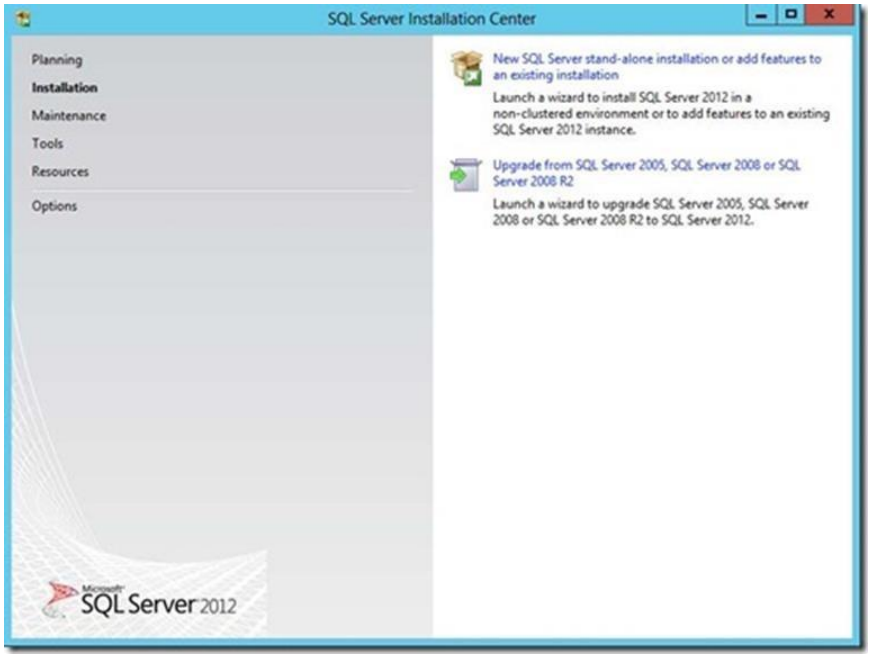
In this document we're going to show you how to install SQL Server 2012 Express Edition on Windows Server 2012. We have included screen shots of each step to make things easier to follow. Note the program can use SQL Server 2012 to 19.

Note: This section as with many sections of this document was written by a 3rd party.

These steps should be similar on other versions of Windows; however, some prerequisites may be required on older versions of Windows.

Step 1.

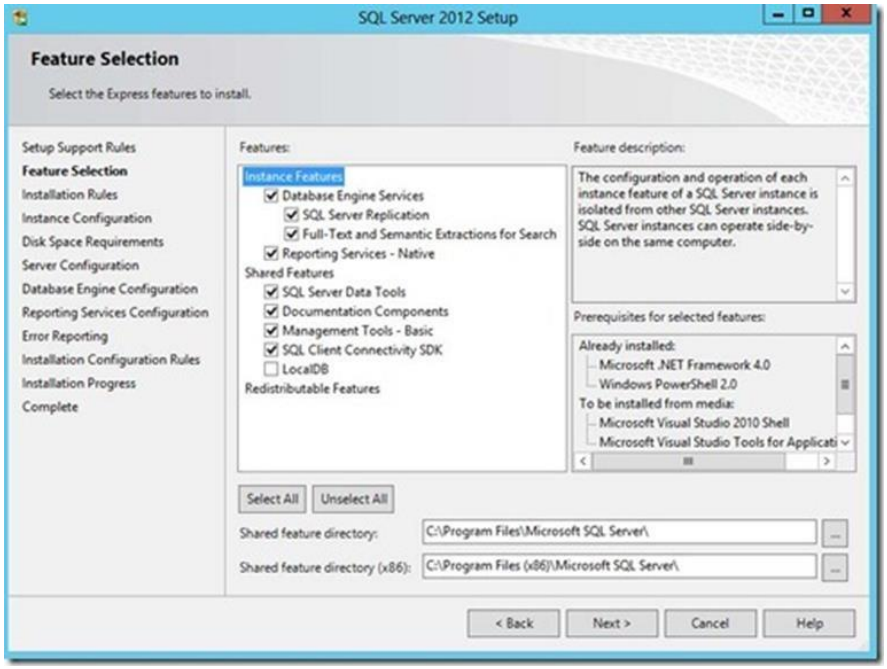
To begin, launch the install program and choose the top option to install a new stand-alone installation.



Step 2.

Read and accept the license agreement and click next. At this point you can choose the features that you want to install. You can also change the install folder if you do not want to install to the default location.

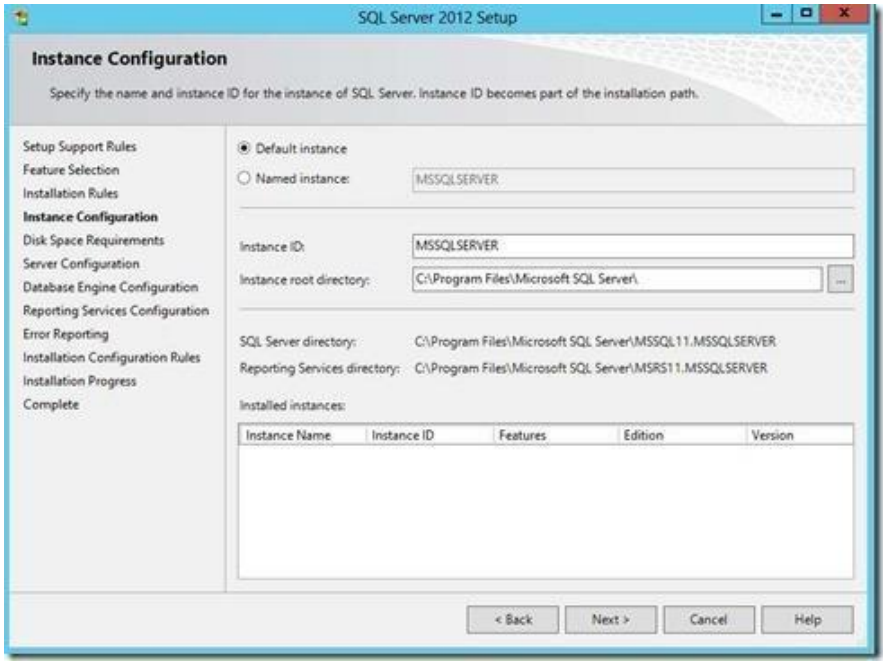
Click **Next** to continue.



Step 3.

Now you can configure the SQL instance. If this is the first instance of SQL Server on your computer, you probably want to change this setting to Default instance. If you already have another instance of SQL Server, you will want to use Named Instance and give it a name.

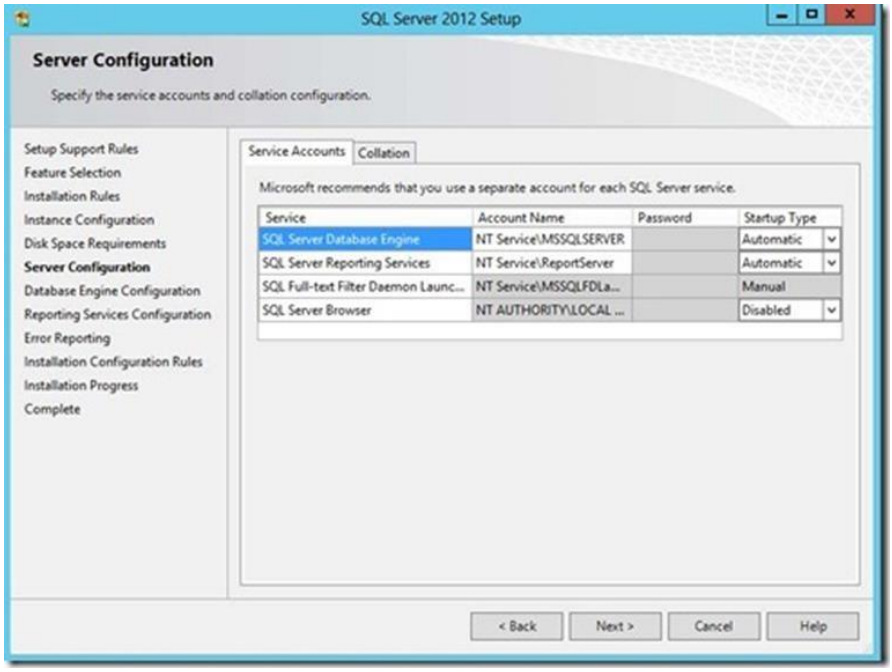
Click **Next** to continue.



Step 4.

The next step is server configuration. From here you can change the Windows services associated with this instance of SQL Server. You can keep the defaults here unless you want to specify different user accounts for the services. You can also change the default collation settings if you are not in the United States.

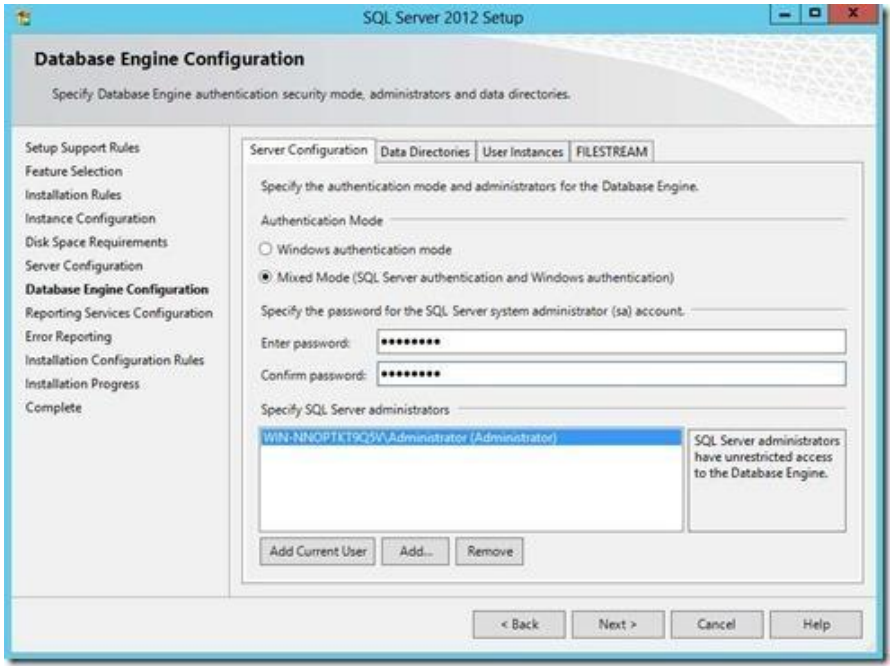
Click **Next** to continue.



NOTE : it is advisable to set the SQL Server Browser to “Automatic”

Step 5.

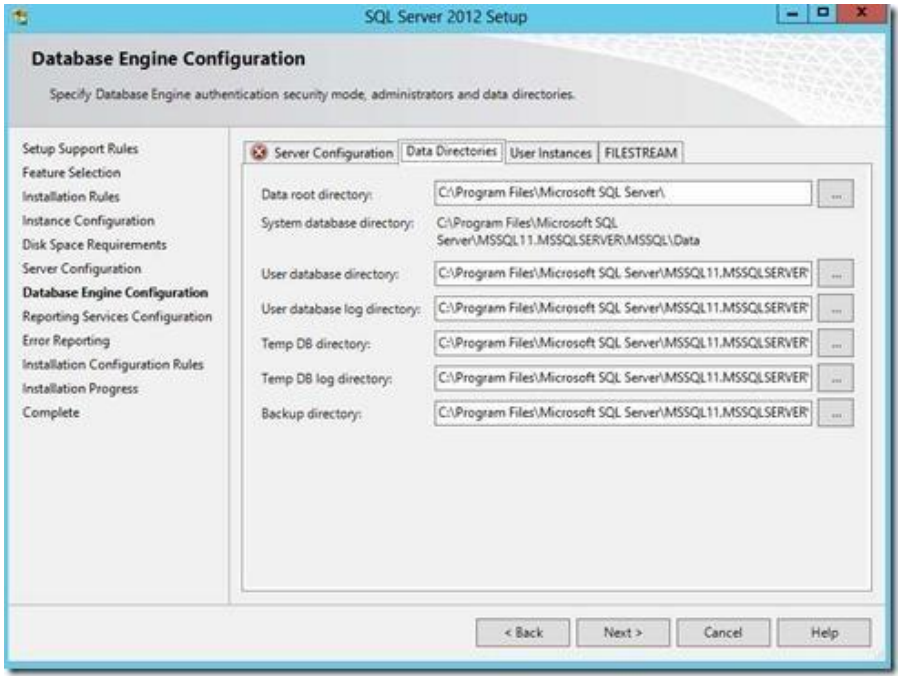
Now we come to the Database Engine configuration. On the Server Configuration tab you can select whether SQL Server will only authenticate using Windows accounts or you can choose Mixed Mode which will allow Windows accounts and SQL accounts. I usually set this to mixed mode and set a password for the “sa” account. You can also add or remove accounts that will be SQL administrators.



Step 6.

On the Data Directories tab you can change the locations of various files that SQL will use. Depending on your server configuration and the load that will be put on SQL, you may want to put the database and log folders on separate drives. For most people, you can leave the default values.

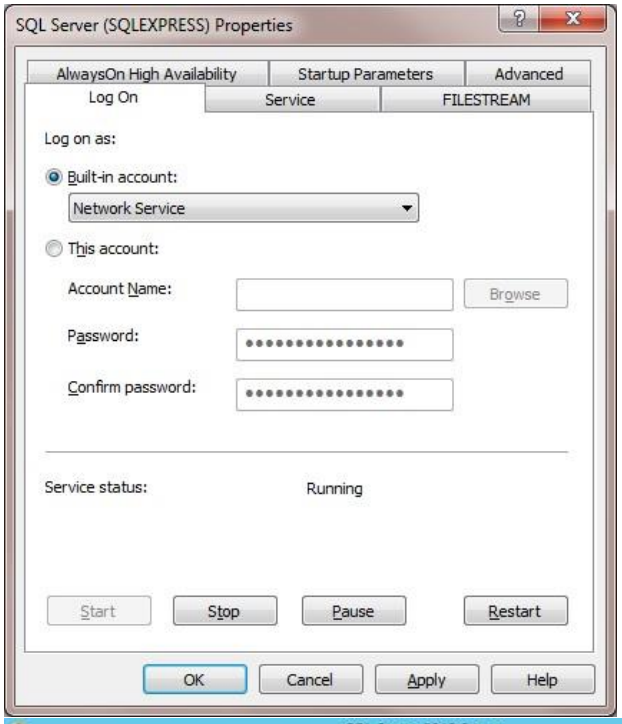
Click **Next** to continue.



Step 7.

The next step configures Reporting Services. Set this to **Install Only** if you don't need reporting services or may need them in the future. You can always go back and configure it later.

Click **Next** to continue.

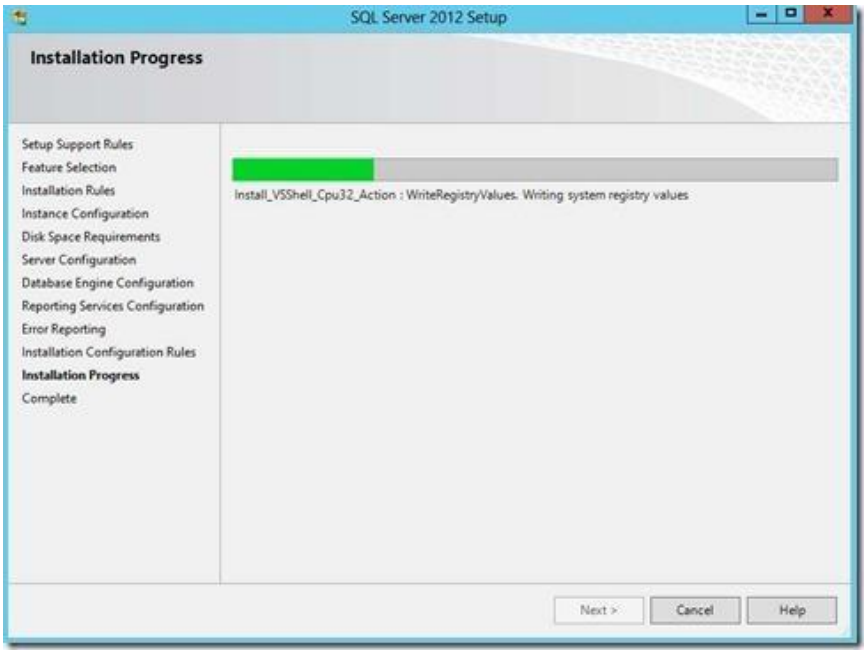


Step 8.

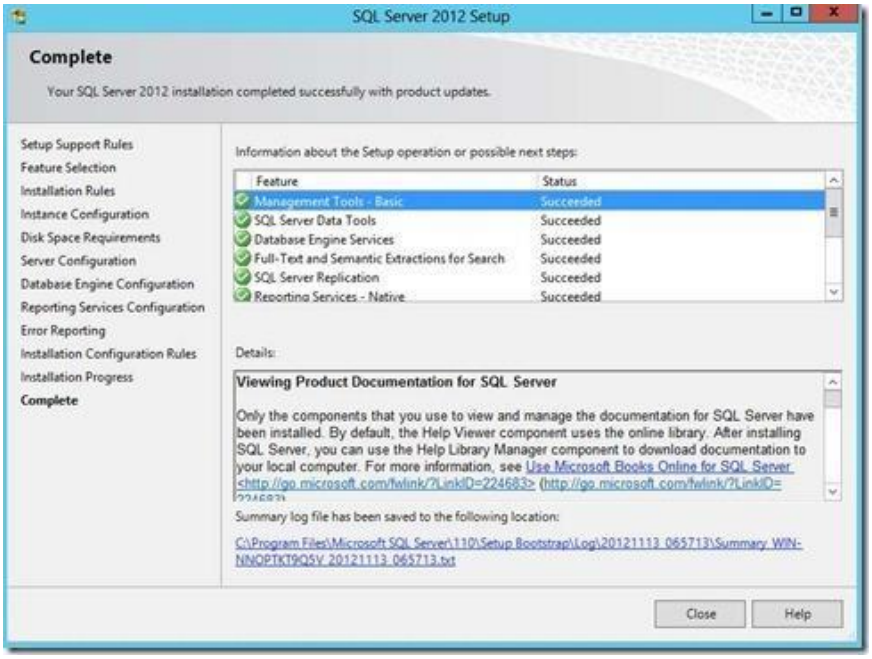
On the Error Reporting step, you can choose whether you would like to send error reports to Microsoft to help them improve future releases of SQL server.

Click **Next** to continue.

At this point, SQL server will install on your computer. This could take a while to complete depending on the computer you are using.

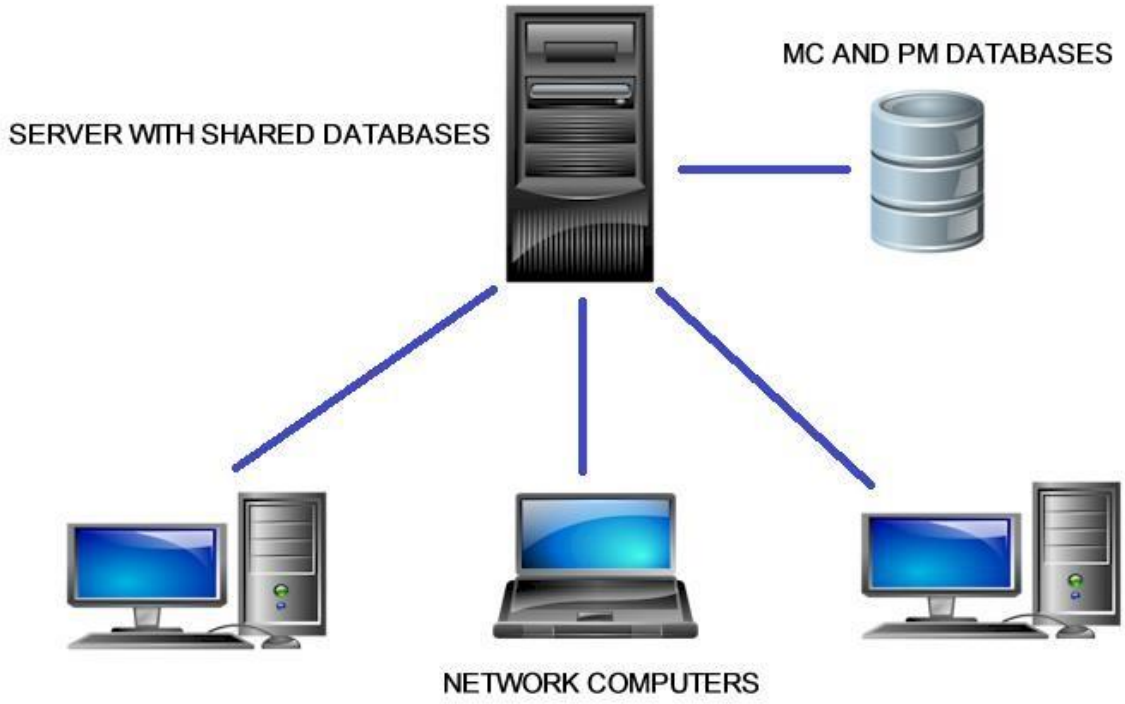


Once the installation has completed, you will receive a screen showing the details of what was completed and if there were any problems.



If everything completed successfully, you can close the window and begin using your new copy of SQL Server 2012.

Installing for Network Use (Basic Configuration)



Please note that a Network/Site License must have been purchased to use the routines outlined in this section. If such a license has not been acquired, then your databases must reside on the local **C:** drive.

To install the program for use on a network, please follow these steps:

- Install the program on each client computer as you would as if it was a single user setup
- Activate the software on each computer it's installed to. **IF NOT ACTIVATED, NETWORK PATHS WILL NOT BE ALLOWED.**
- Copy the databases used by the program to a sharable network drive. These are the files with an .mdf file extension.
- Also copy the database's associated transaction log file. This file will have an extension of .ldf, and shares most of the same name as the database file it's associated with.
- Set up the database connections to use the copied databases.

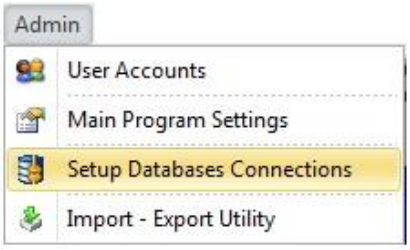
Setting up Database Connections

By default, the databases used by the **Maintenance Coordinator** system are installed to the **C:\Maintenance Coordinator NET** folder. It is highly recommended that you leave a set of these databases in this location. However, there are occasions when you want them to reside in other locations, such as on a network drive or share. When moving the databases, we recommend copying the databases to the new location.

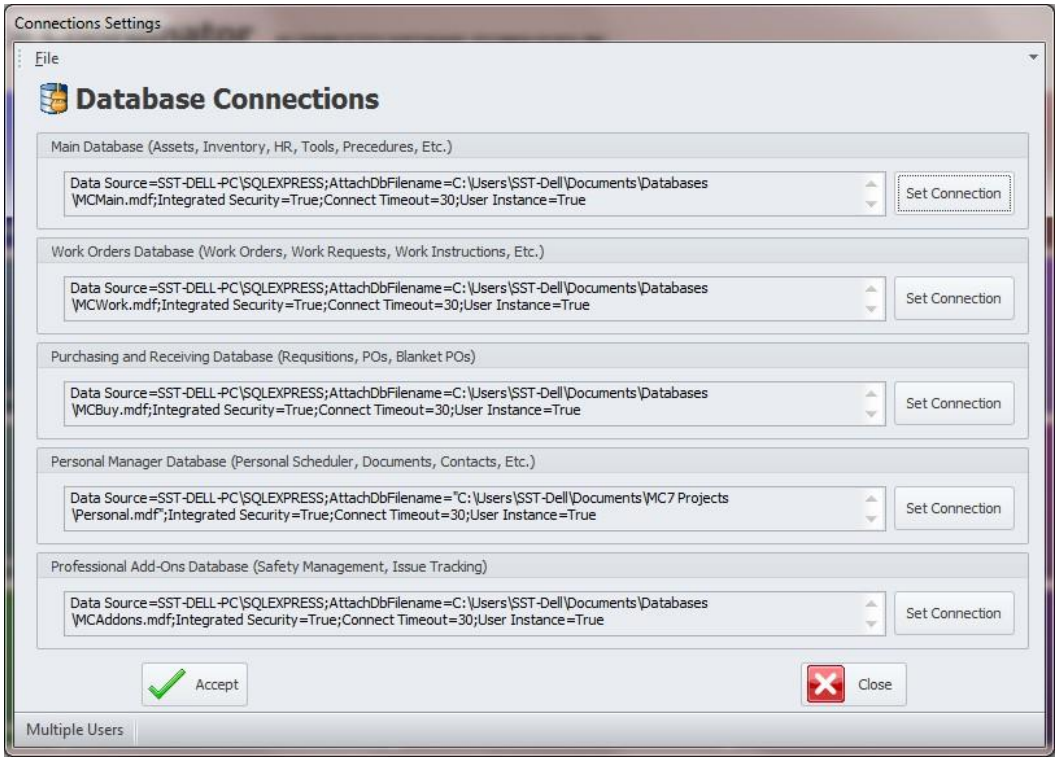
> ***Please note that a Network/Site License must have been purchased to network the databases.***

To setup your database connections:

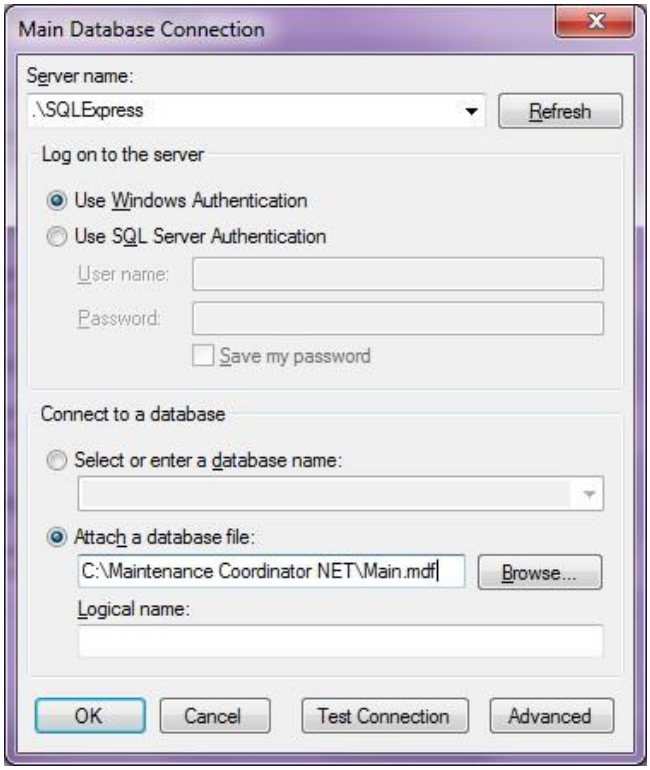
- Log in to the program as a user with administrator rights.
- From the main start menu, open the **Admin** drop down menu and select the **Setup Database Connections** option.



- From the invoked **Connections Settings** dialog, move to the **Main Database** group and click on the **Set Connection** button.



- This will invoke the **SQL Connection** dialog screen as shown in the following illustration:



- Move to the **Sever name** drop down box and select the name of your instance of the SQL server you're using. If the server is not listed, try clicking on the **Refresh** button.
- Move to the **Log on to the server** group and setup your log on authentication information.
    - **Using SQL Express** - If you are using the Express database engines as supplied by us, you'll probability want to check the **Use Windows Authentication** option.
    - **SQL Server** - If you are connecting to full blown SQL Server then complete the **SQL Server Authentication** information.
        - Check the **Use SQL Server Authentication** option.
- Fill in the **User Name** and **Password** fields.
- Next move to the **Connect to a database** group to connect to the actual database.
    - **SQL Server**:
        - Check the **Select or enter a database name** option.
        - Move to the drop down found here and either select or enter the name of the database you are connecting to.
    - **SQL Express**: (Attached database file)
        - Check the **Attach a database file** option.
        - Either use the **Browse** button here to select the database file, or type in the full path to the database (mdf) file in the space provided.

- **Test Connection** – Move to and click on the **Test Connection** button to ensure you have a good connect. If it fails, you may want to click on the '**Advanced'** button to further define you connection properties.
- Click on the **OK** button.
- Repeat the above steps for the remaining databases that require setup.

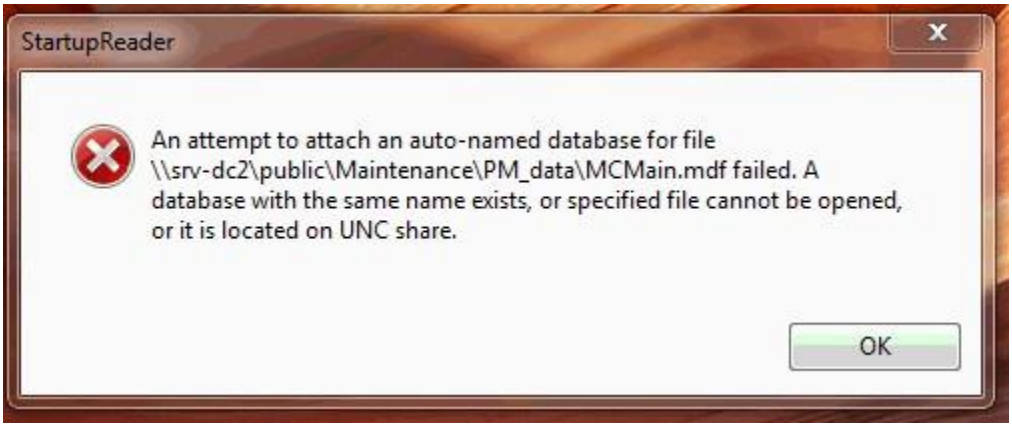
Brief Description of the Databases

- **Main Database** - This database holds the main information or tables dealing with **Maintenance Coordinator**. Some of the included tables are HR (Employees), Inventory and Parts, Contacts and Vendors, Maintenance Documents, Main Programs settings and User Accounts.
- **Work Orders Database** – This database holds everything dealing with work orders and PMs. This also includes work instructions and time tracking information.
- **Purchasing and Receiving Database** – This database maintains information for Purchasing and Receiving support, purchase orders, purchase requisitions and blanket purchase orders.
- **Personal Manager Database** – This database maintains information on personal items. Personal Log, Personal Scheduler, Personal Contacts and Journal information.
- **Professional Add-Ons Database** – This database supports some of the add-on features and their data storage. Currently supported are Issues and Safety Management information.

Connection Errors (Basic Configuration)

A database with the same name exists, or specified file cannot be opened, or it is located on UNC share.

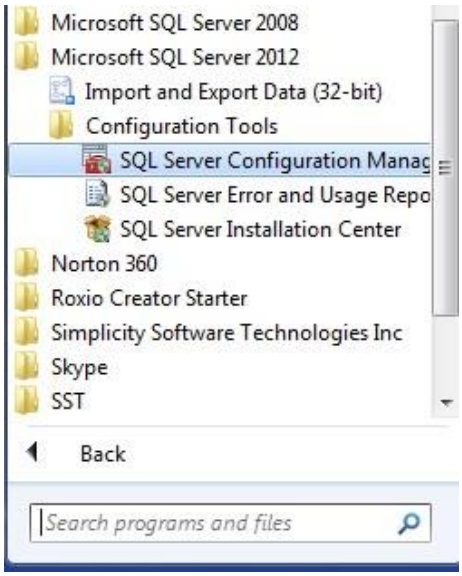
If this error is received, then you may need to setup the server for network use. This procedure is outlined next.



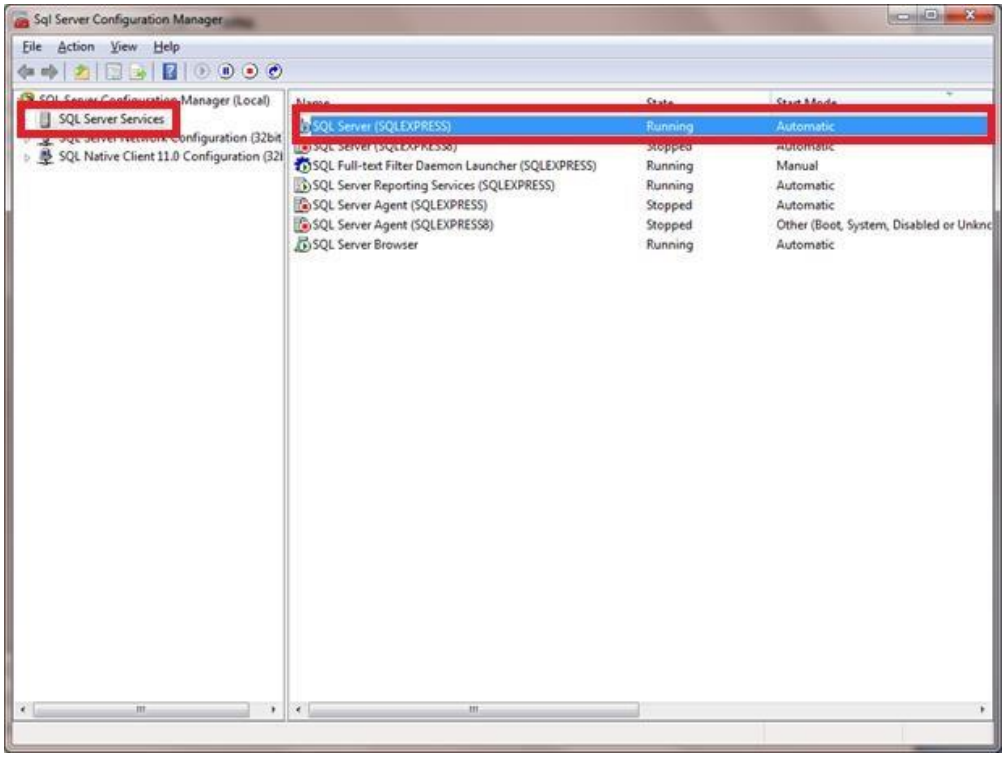
SUGGESTED SOLUTION

Change the default account in SQL Server Configuration Manager. The default Local Service account is used when SQL Server Express is installed and should be changed to use the built-in account '**Network Service**'.

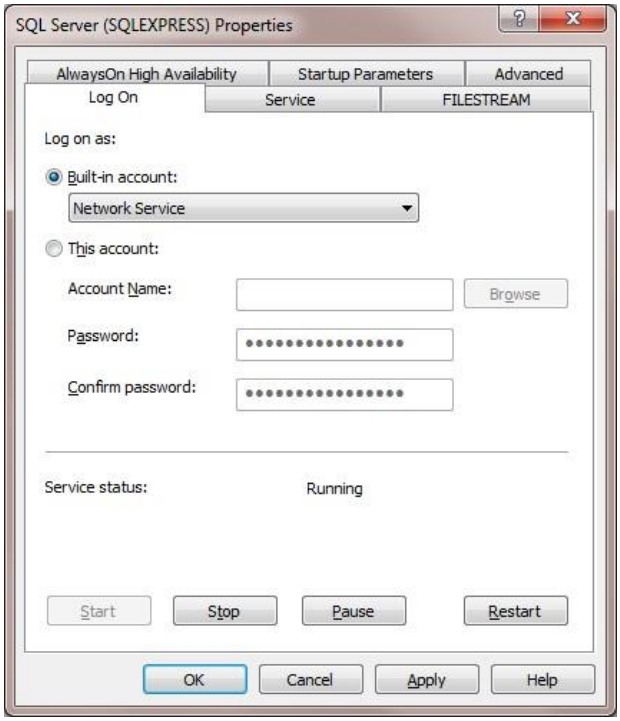
- Click the **Windows Start Menu**.
- Click **All Programs**



- Click **Microsoft SQL Server 2012** option to expand the group.
- Click **Configuration Tools** option.
- Click **SQL Server Configuration Manager**.
- Click **YES** when prompted about making changes to your computer.
- Click the **SQL Server Services** node in the tree on the left of your screen.



- In the main list view, double click on the **SQL Server (SQLEXPRESS)** option. Move to the **Log On** tab and check the **Build-in Account** radio button.



- Move to the Drop-Down list and select the **Network Service** option.
- Click **OK** and let the service restart.

Advanced Topics

From this section forward we will outline more advanced topics, mainly user SQL Server Express databases when a full-blown instance of SQL Server is installed.

How to configure SQL Express 2012 to accept remote connections

When you try to connect to an instance of Microsoft SQL Server 2012 Express from a remote computer, you might receive an error message.

INTRODUCTION

Named SQL instances listen on dynamic ports. This is the function of the SQL Server Browser Service to inform the clients of the actual port. The SQL Browser listens on UDP 1434 and answers all client request with the port number the current instance is using. SQL Server Browser service is required for both TCP and named pipes protocols. SQL Server Browser is used by clients transparently and there is no need for special configuration.

To configure SQL Server 2012 Express to allow remote connections, you must complete these steps:

- Enable remote connections on the instance of SQL Server that you want to connect to from a remote computer.
- Configure SQL server to listen on static port Turn on the SQL Server Browser service.
- Configure the firewall to allow network traffic that is related to SQL Server and to the SQL Server Browser service.

Enable remote connections on the instance of SQL Server that you want to connect to from a remote computer.

1. Open **SQL Management Studio** and right-click server name in the left pane and select **Properties**

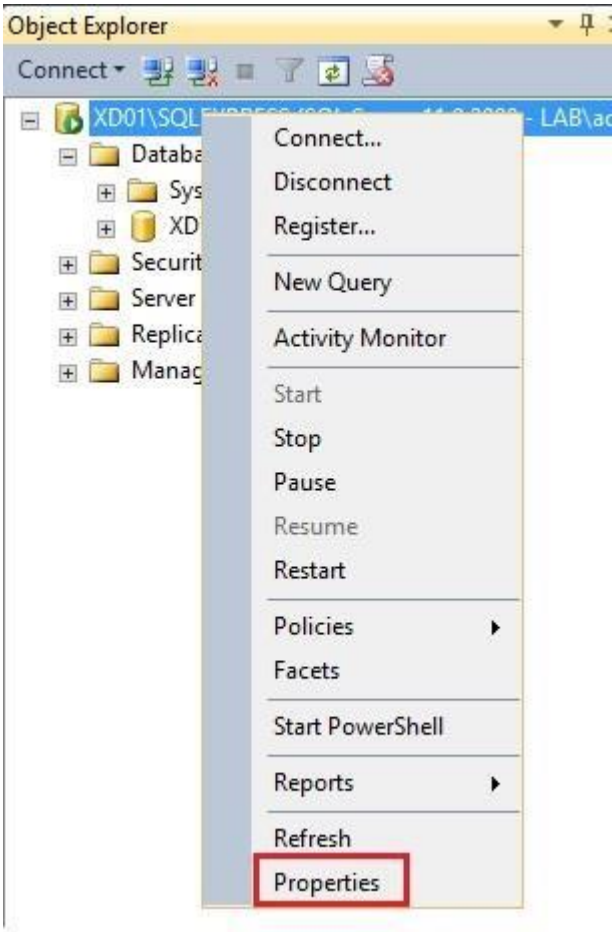


Figure 1

2. Select **Connections** in the left pane and make sure that checkbox **Allow remote connections to this server** is selected as it is shown in Figure 2.

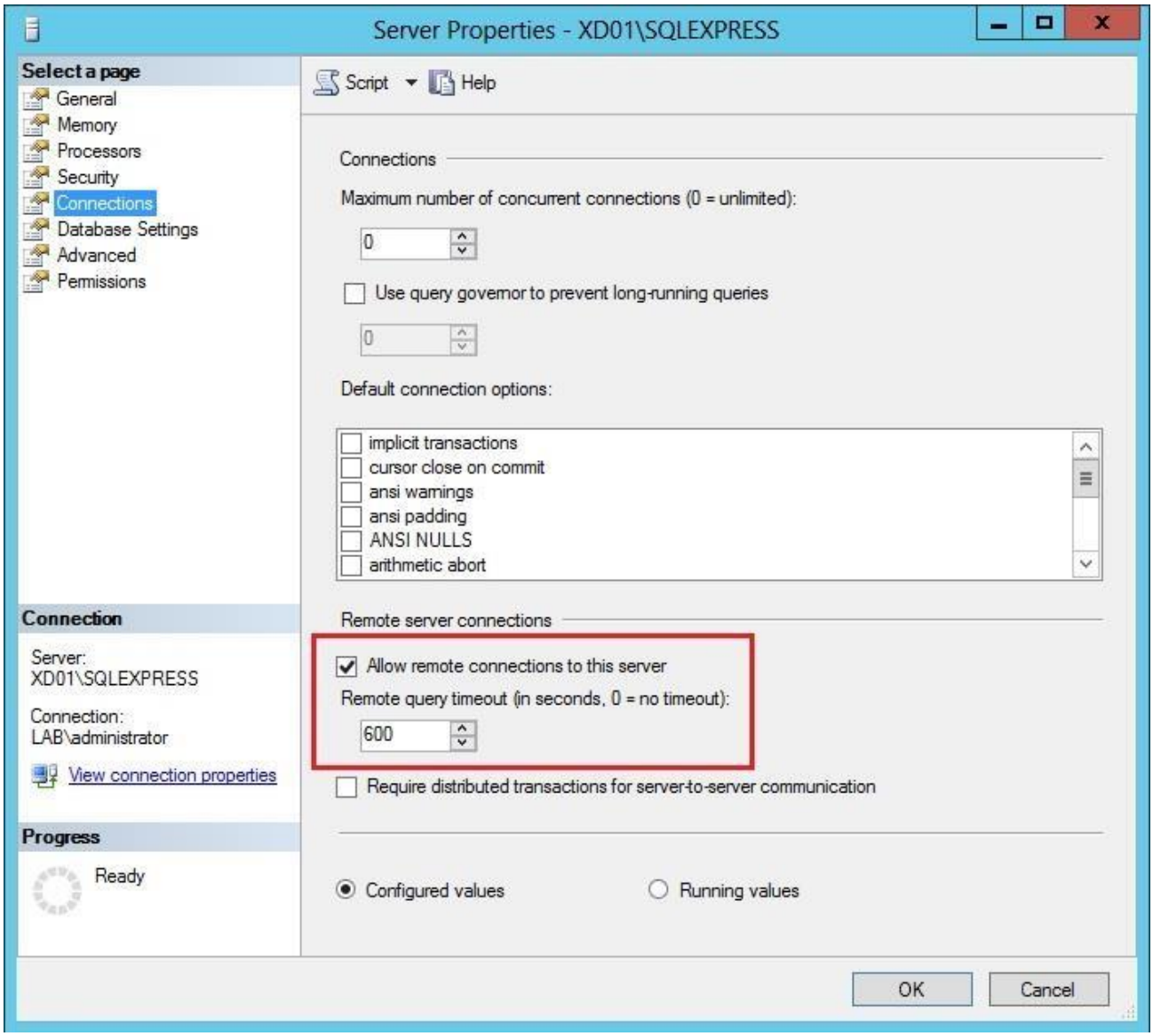


Figure 2

Configure SQL express server to listen on static port.

1. Open **SQL Server Configuration Manager** and click on "**SQL Server Services**" in the left pane.

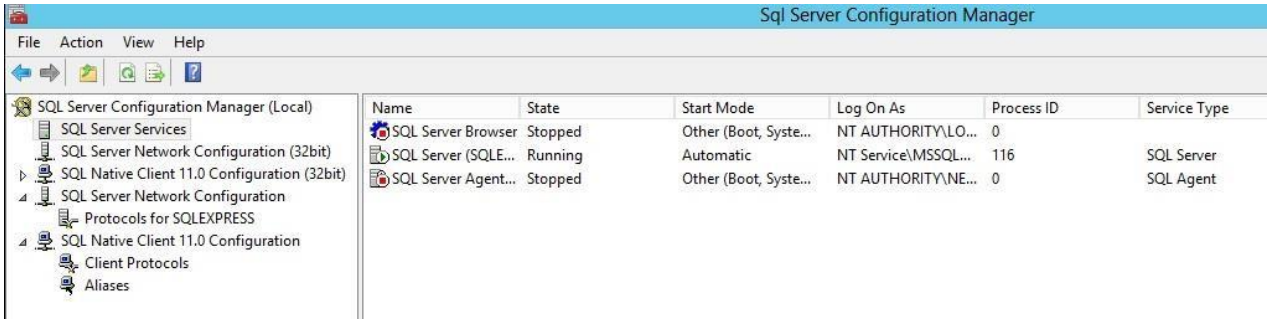


Figure 3

2. In the center pane, is a column that lists the Process ID for each running service. Look for the PID in the row for **SQL Server**. Identify the port that that PID is listening on by typing this into a command prompt:

netstat -ano | find /i "PID-Number-Of-SQL-Server". Based on the details shown in Figure 3 syntax is the following: netstat -ano | find /i "116". The results are shown in Figure 4.

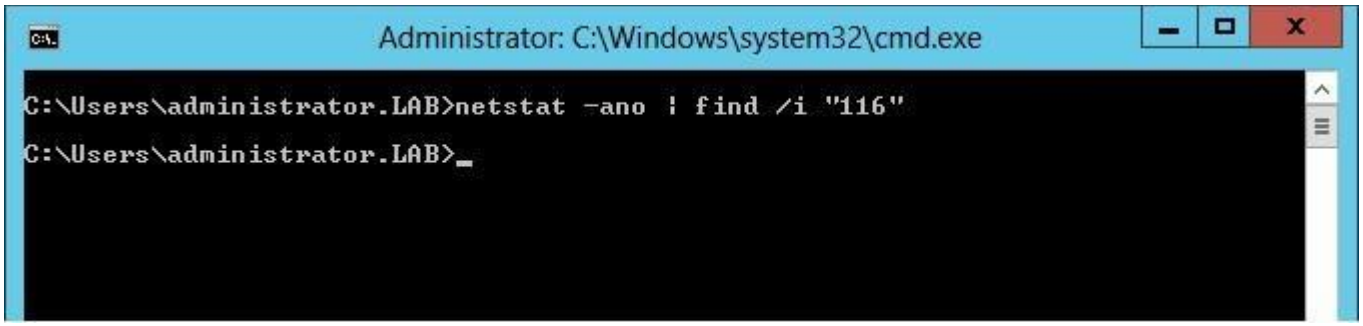


Figure 4

3. There are no results from the command executed in step 3 because TCP/IP protocol is disabled and must be enabled. In **SQL Server Configuration Manager** and click on **SQL Server Network Configuration** in the left pane and right-click **TCP/IP** protocol and select option **Enable**.

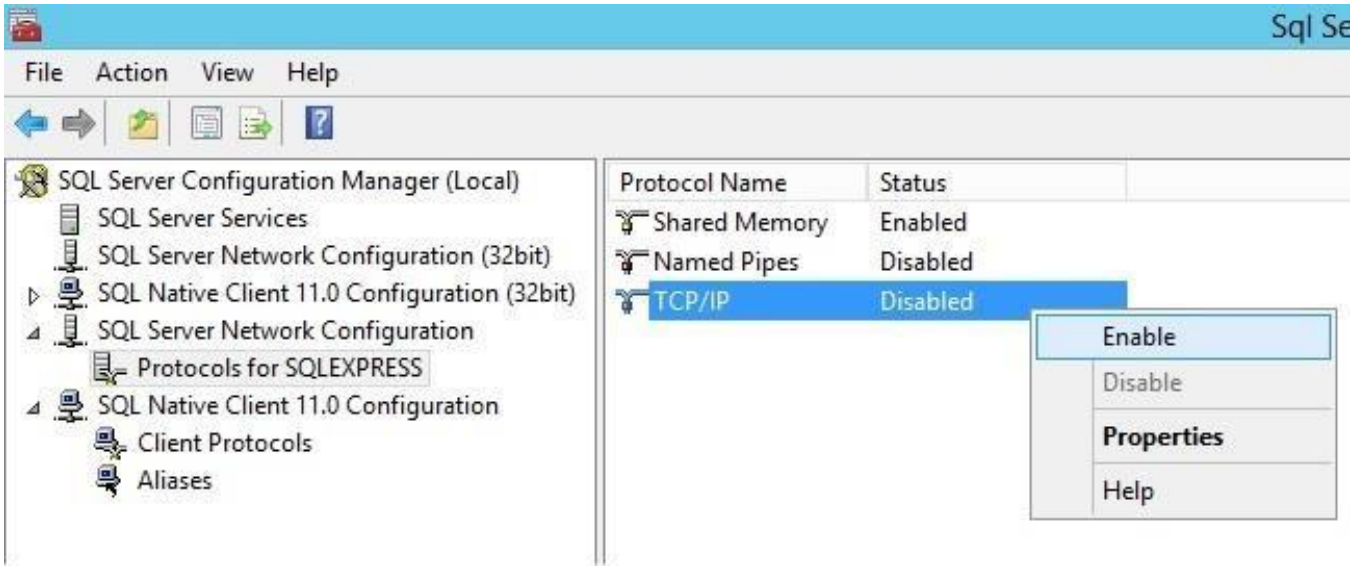


Figure 5

4. Restart SQL Server service and identify the process ID assigned to SQL service.

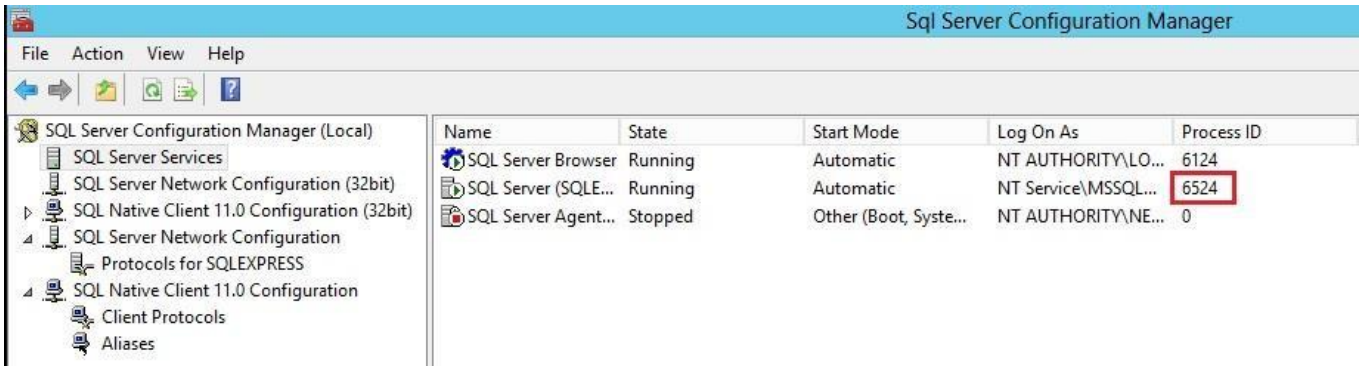


Figure 6

5. In the command prompt execute command: **netstat -ano | find /i "6524"**. The results are shown in Figure 7.

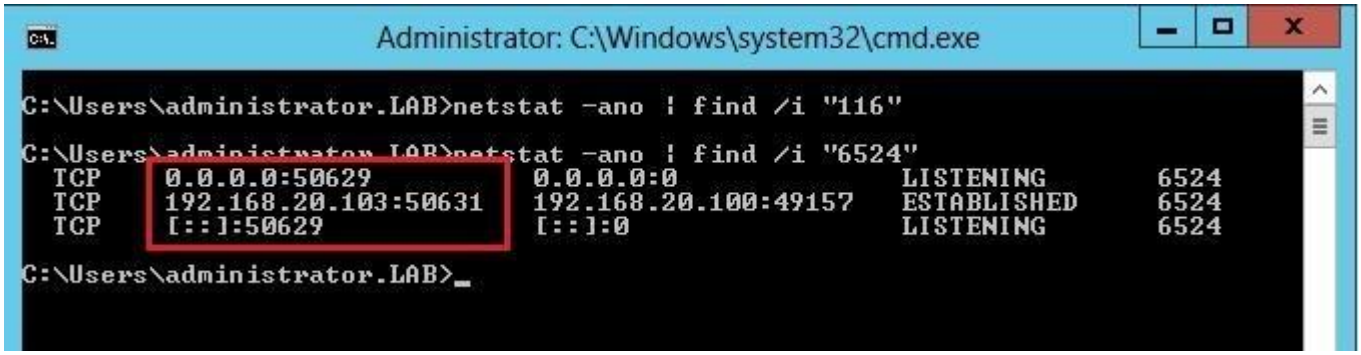


Figure 7

6. In **SQL Server Configuration Manager** and click on **SQL Server Network Configuration** in the left pane and right-click **TCP/IP** protocol and select option **Properties**. Goto **IP Address** tab and scroll-down to **APAll** section. Remove value for **TCP Dynamic Ports** (do not enter Zero 0 !!!) and enter the port 1433 for **TCP Port**.

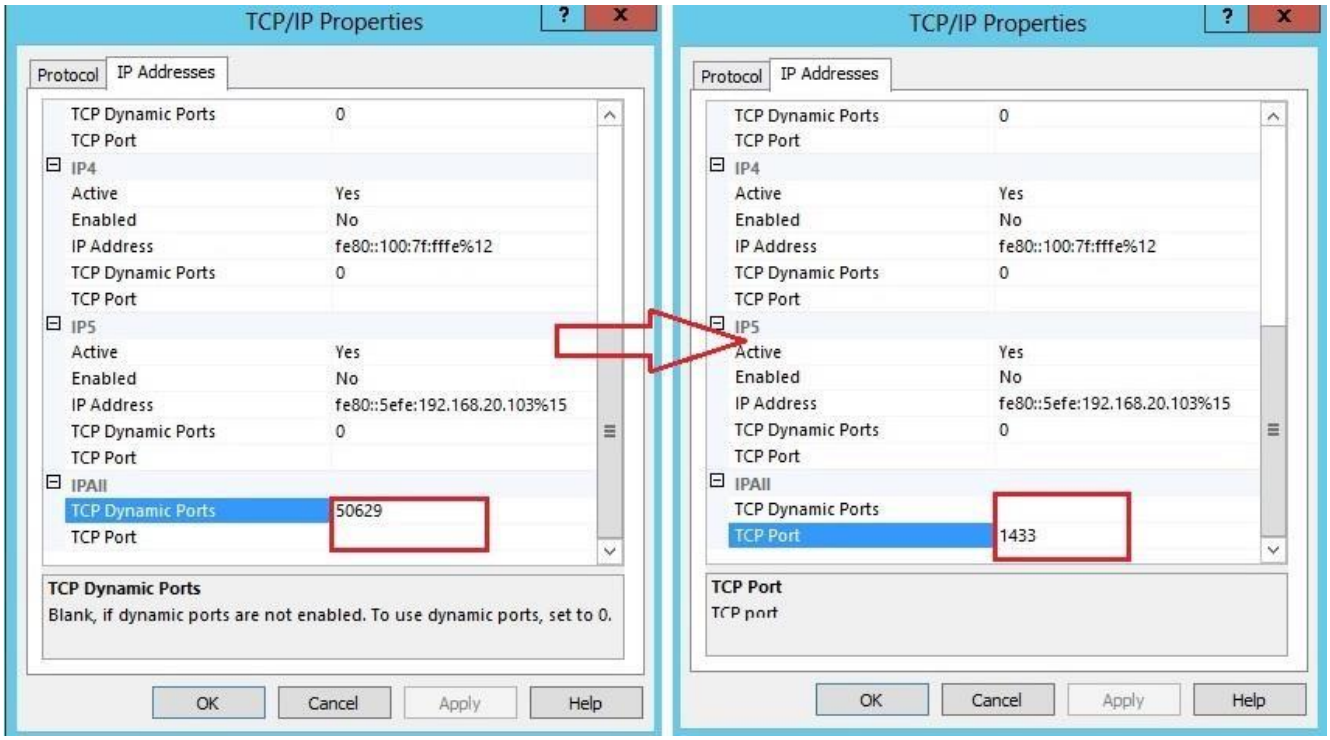


Figure 8

7. Restart SQL Server service, identify new process ID assigned to SQL service and in the command prompt execute command: **netstat -ano | find /i "3948"**. The results are shown in Figure 9.

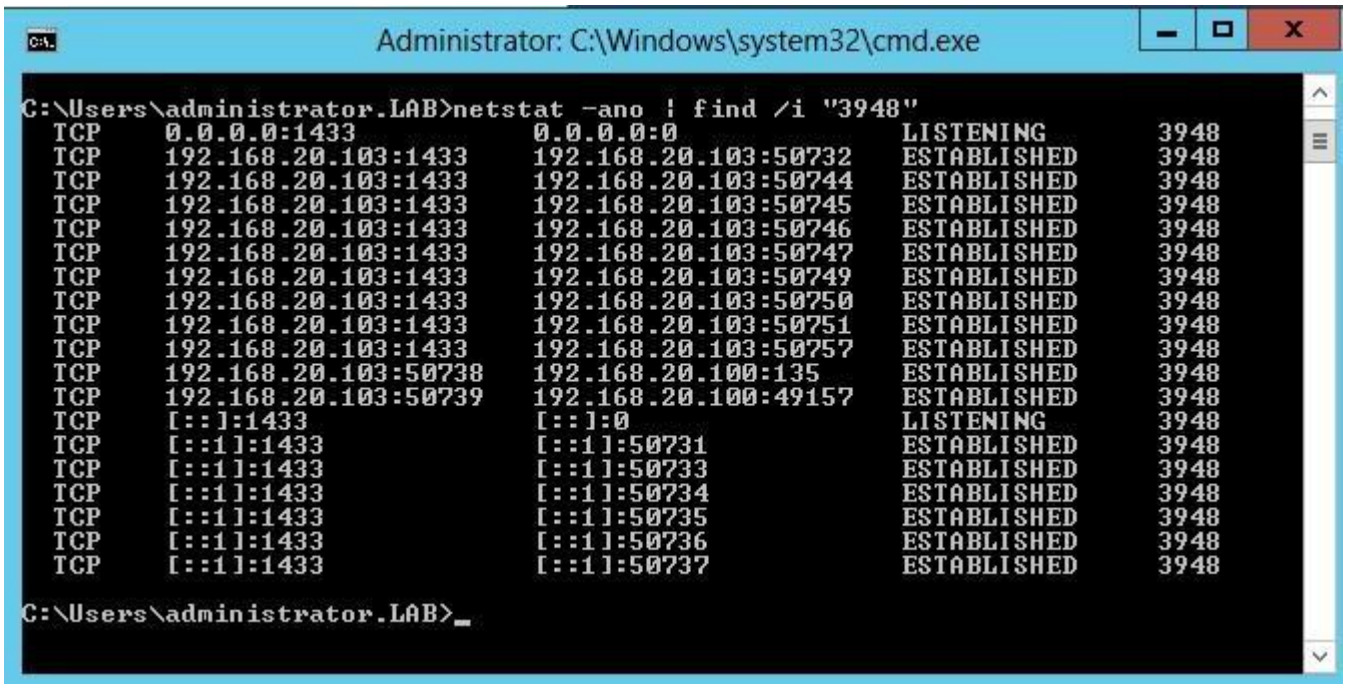


Figure 9

At this stage SQL Express is configured to listen on standard port 1433.

Turn on the SQL Server Browser service.

1. Open **SQL Server Configuration Manager** and click on "**SQL Server Services**" in the left pane, right click **SQL Server Browser service** and select **Properties**.

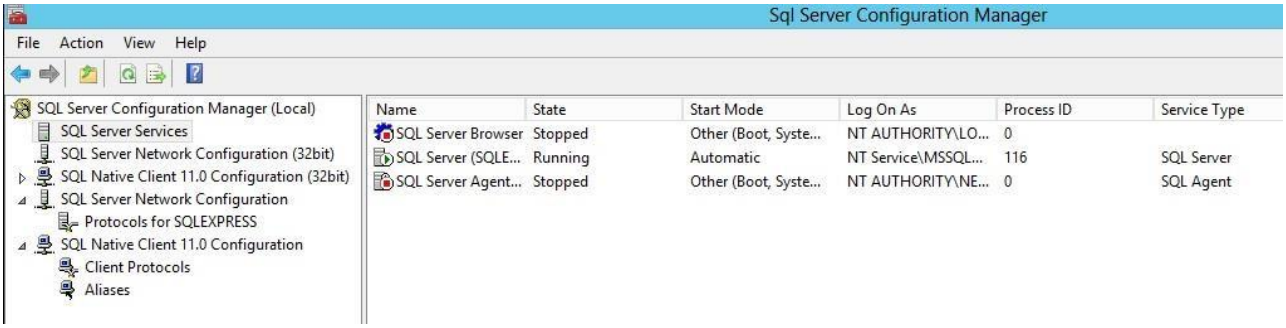


Figure 10

2. Go to **Service** tab and for **Start Mode** option change start type to **Automatic**.

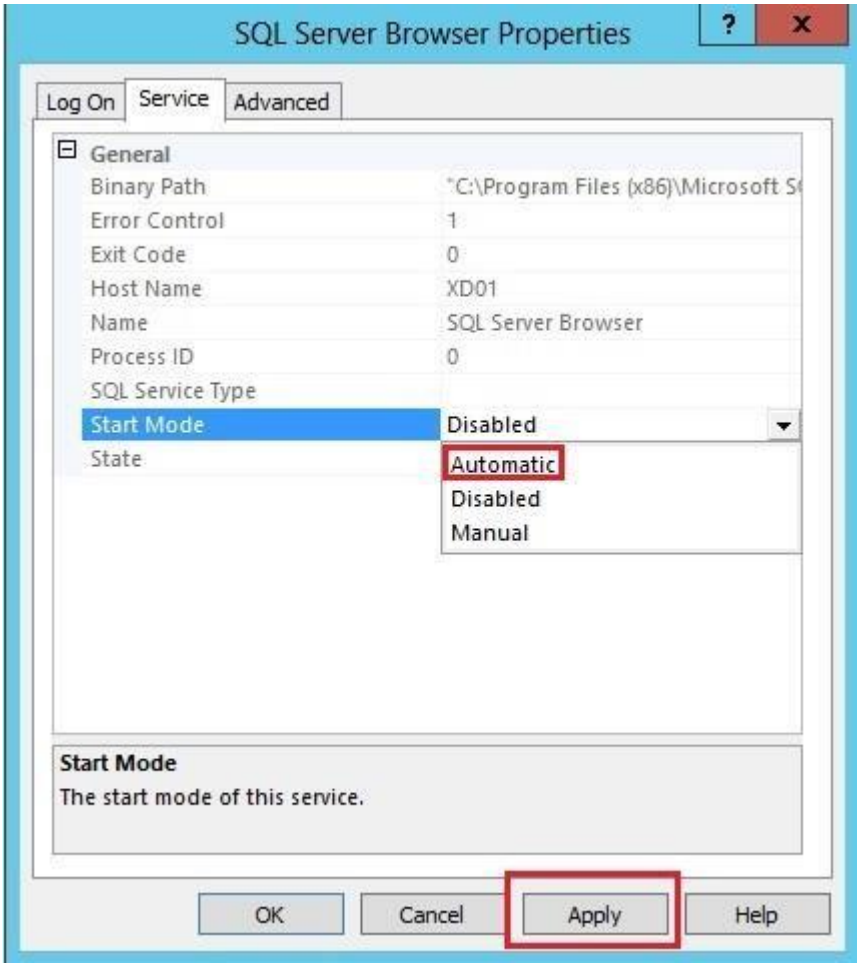


Figure 11

3. Click Start button to start SQL Browser service

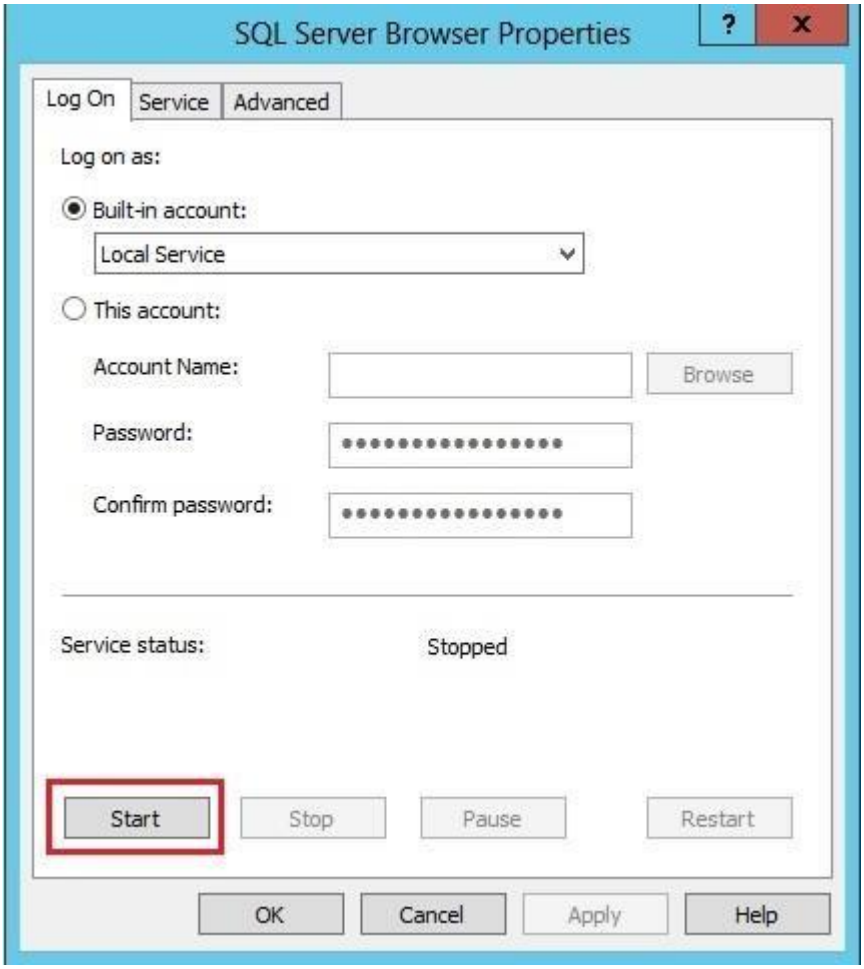


Figure 12

4. Confirm that SQL Server Browser service is up and running as it is shown in Figure 13.

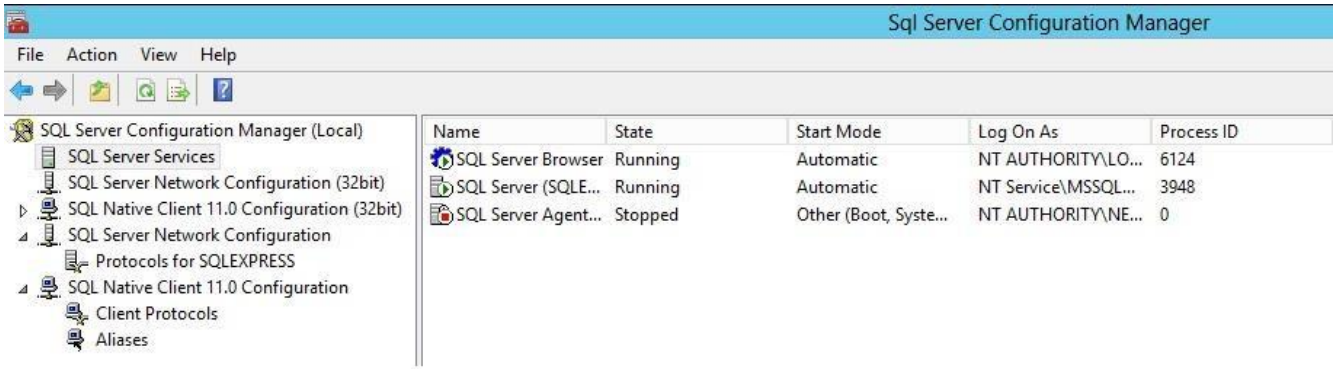


Figure 13

Configure the firewall to allow network traffic that is related to SQL Server and to the SQL Server Browser service.

Four exceptions must be configured in **Windows Firewall** to allow access to **SQL Server:**

1. A port exception for **TCP Port 1433.** In the New Inbound Rule Wizard dialog, use the following information to create a port exception: ○ Select **Port**
   - ○ Select **TCP** and specify port **1433** ○ Allow the connection
   - ○ Choose all three profiles (Domain, Private & Public)
   - ○ Name the rule “SQL – TCP 1433″
2. A port exception for **UDP Port 1434.** Click New Rule again and use the following information to create another port exception:
   - ○ Select **Port**
   - ○ Select **UDP** and specify port **1434** ○ Allow the connection ○ Choose all three profiles (Domain, Private & Public) ○ Name the rule “SQL – UDP 1434
3. A program exception for **sqlservr.exe.** Click New Rule again and use the following information to create a program exception: ○ Select **Program**
   - ○ Click **Browse** to select ‘sqlservr.exe’ at this location: [C:\Program Files\Microsoft SQL Server\MSSQL11.<INSTANCE_NAME>\MSSQL\Binn\sqlservr.exe] where <INSTANCE_NAME> is the name of your SQL instance. ○ Allow the connection ○ Choose all three profiles (Domain, Private & Public) ○ Name the rule SQL – sqlservr.exe
4. A program exception for **sqlbrowser.exe** Click New Rule again and use the following information to create another program exception:

- Select **Program**
- Click **Browse** to select sqlbrowser.exe at this location: [C:\Program Files\Microsoft SQL Server\90\Shared\sqlbrowser.exe].
- Allow the connection ○ Choose all three profiles (Domain, Private & Public) ○ Name the rule SQL – sqlbrowser.exe

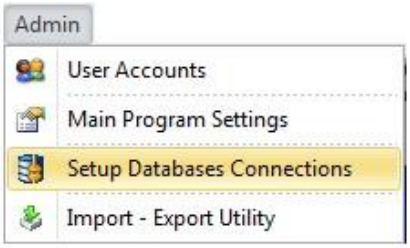
Setting up the Databases

By default, the databases used by the **Maintenance Coordinator** system are installed to the **C:\Maintenance Coordinator NET** folder. It is highly recommended that you leave a set of these databases in this location. However, there are occasions when you want them to reside in other locations, such as on a network drive or share. When moving the databases, we recommend copying the databases to the new location.

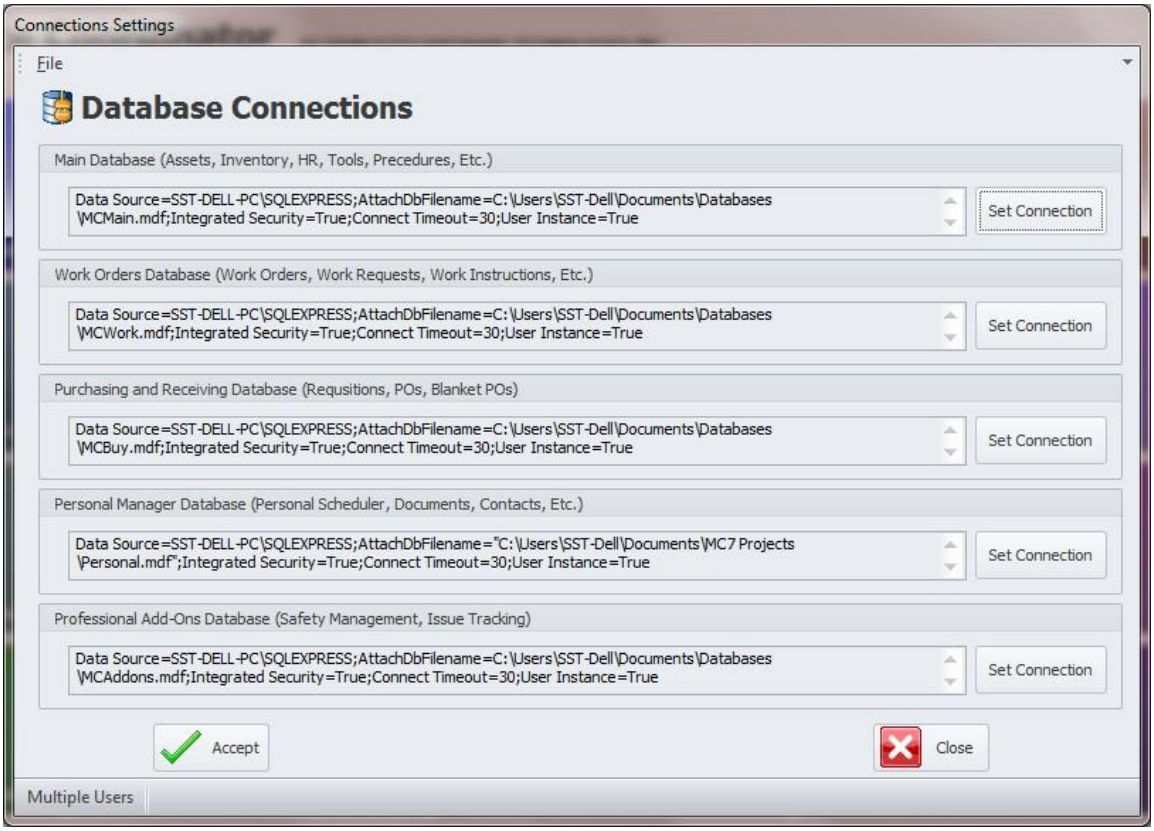
Please note that a Network/Site License must have been purchased to network the databases.

To setup your database connections:

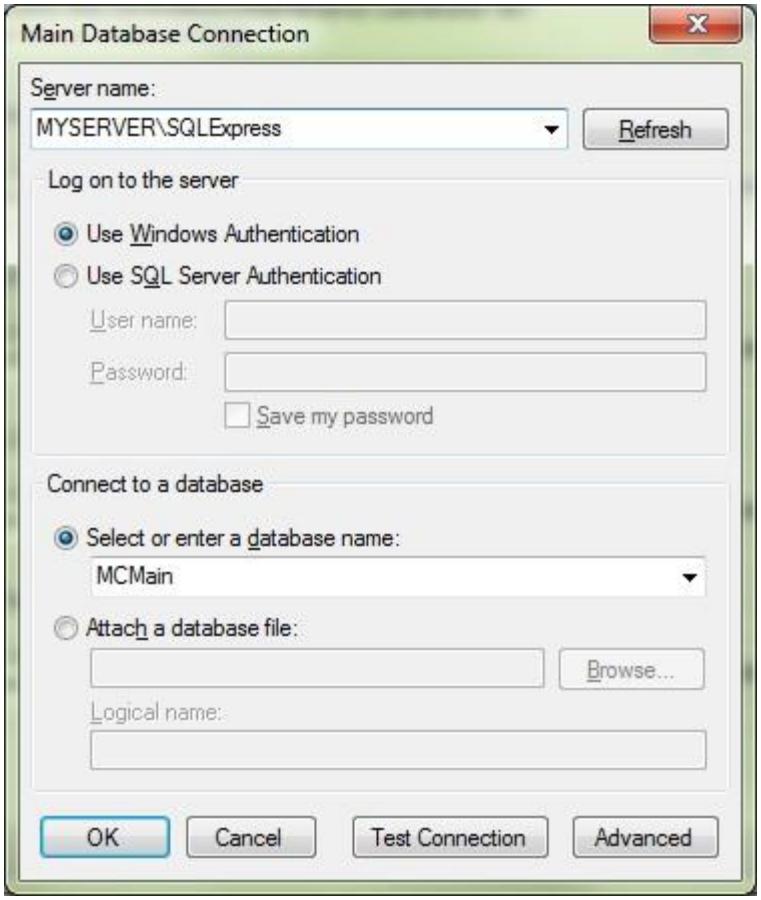
- Log in to the program as a user with administrator rights.
- From the main start menu, open the **Admin** drop down menu and select the **Setup Database Connections** option.



- From the invoked **Connections Settings** dialog, move to the **Main Database** group and click on the **Set Connection** button.

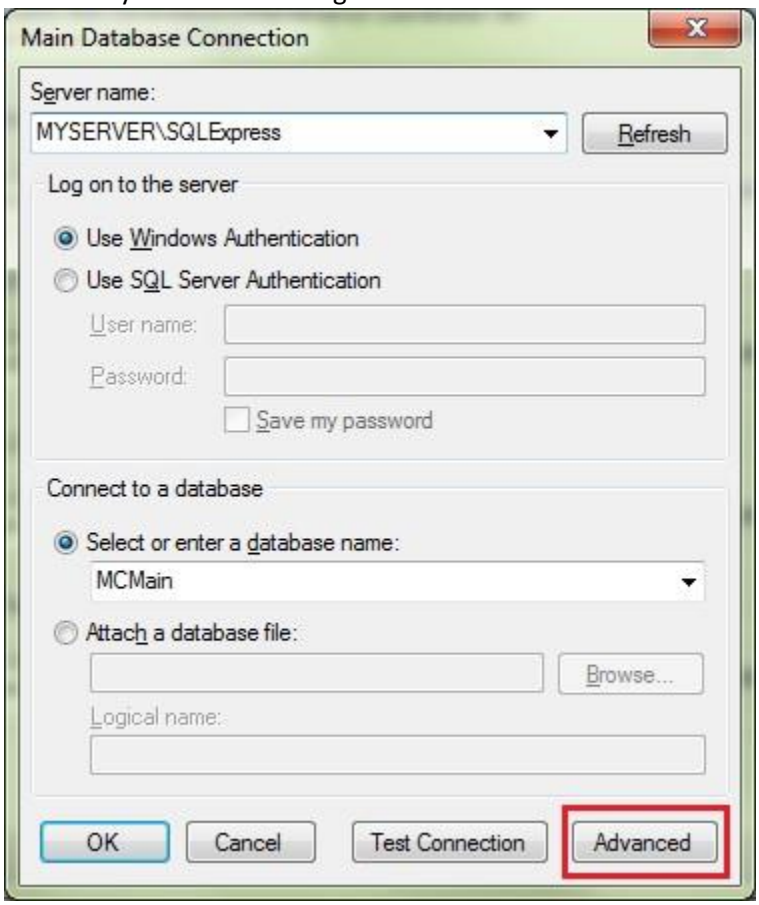


- This will invoke the **SQL Connection** dialog screen as shown in the following illustration:

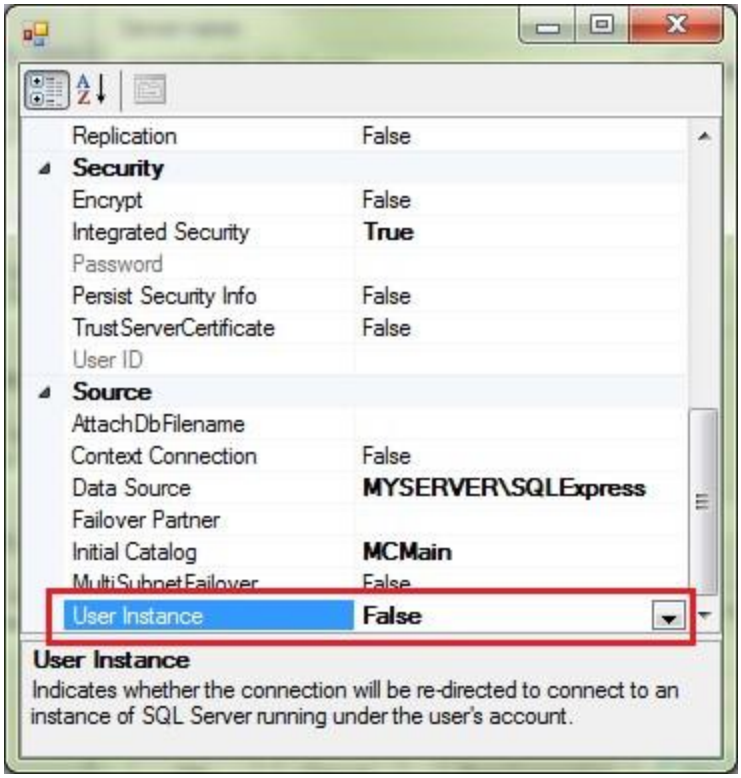


- Move to the **Server name** drop down box and select the name of your instance of the SQL server you're using. If the server is not listed, try clicking on the **Refresh** button or manually type this information in.
- Move to the **Log on to the server** group and setup your log on authentication information.
    - **Using SQL Express** - If you are using the Express database engines as supplied by us, you'll probability want to check the **Use Windows Authentication** option.
    - **SQL Server** - If you are connecting to full blown SQL Server then complete the **SQL Server Authentication** information.
        - Check the **Use SQL Server Authentication** option.
- Fill in the **Username** and **Password** fields.
- Next move to the **Connect to a database** group to connect to the actual database.
    - **SQL Server**:
        - Check the **Select or enter a database name** option.

- Move to the drop down found here and either select or enter the name of the database you are connecting to.



- Move to and click on the **Advanced** button.
- Ensure that **User Instance** is set to **False**.



- **SQL Express**: (Attached database file)
  - Check the **Attach a database file** option.
  - Either use the **Browse** button here to select the database file, or type in the full path to the database (mdf) file in the space provided.

- **Test Connection** – Move to and click on the **Test Connection** button to ensure you have a good connect. If it fails, you may want to click on the **Advanced** button to further define your connection properties.
- Click on the **OK** button.
- Repeat the above steps for the remaining databases that require setup.

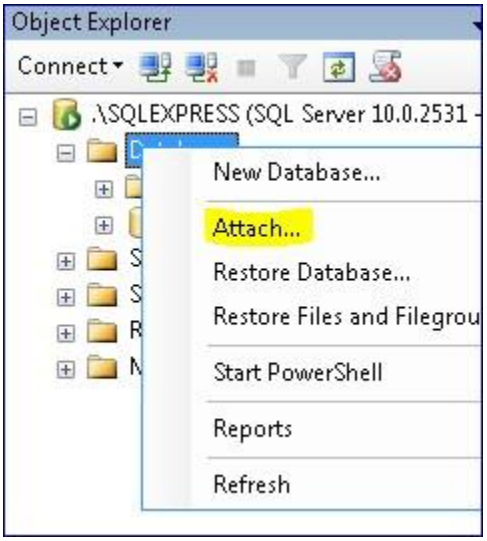
How can I open an .MDF file in SQL Server

(Attach Tutorial & Troubleshooting)

.MDF files are SQL Server database files and .LDF files are the associated log files. But you can't really "open" them. Instead, you have to "attach" to them. Once you attach, you'll see the database in the object explorer of SSMS.

If you're thinking, "Why wasn't the Backup and Restore process used instead of passing around these database files?" It's a valid question, but it doesn't help the reader staring at .MDF and .LDF files with no clue how to access them. However, if you need to move a database around, you should do a search on "backup restore attach detach sql server" (without the quotes) in your favorite search engine. You'll get lots of opinions to help you decide the best approach.

How to Attach in a Perfect World:



Click here to watch a video on this...

There's a couple of different methods. I'll give you the "easy" way through SQL Server Management Studio (SSMS) and then point you to other methods using T-SQL queries.

1. Launch SSMS.
2. Connect to your SQL Server Instance.
3. Right-click on **Databases** in the Object Explorer.
4. Click **Attach**.
5. In the **Attach Databases** window, click the **Add** button.

6. Navigate to the directory containing the .MDF and .LDF files.
7. Select the .MDF file, and press **OK**.
8. Press **OK** again to attach the database.

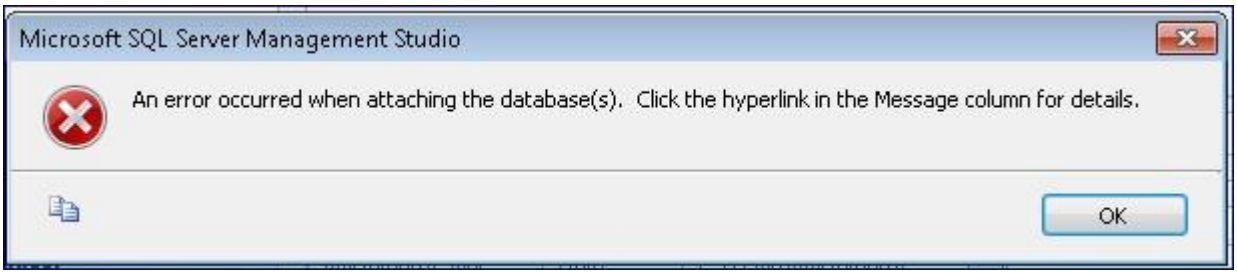
You should see the database appear in the Databases node (press F5 if you don't). You can now explore the data using SSMS. These same steps are documented in the MSDN here. I'm just summarizing to make it easier for you.

There are two ways to accomplish the same thing in a T-SQL script. First, you can use the sp_attach_db stored procedure. This is shown here. Or you can use the CREATE DATABASE command with the FOR ATTACH argument.

So, the World isn't Perfect

If your database is successfully attached, then stop reading and go use it. But if you see any errors or unexpected behaviors, I'll try to share some troubleshooting suggestions.

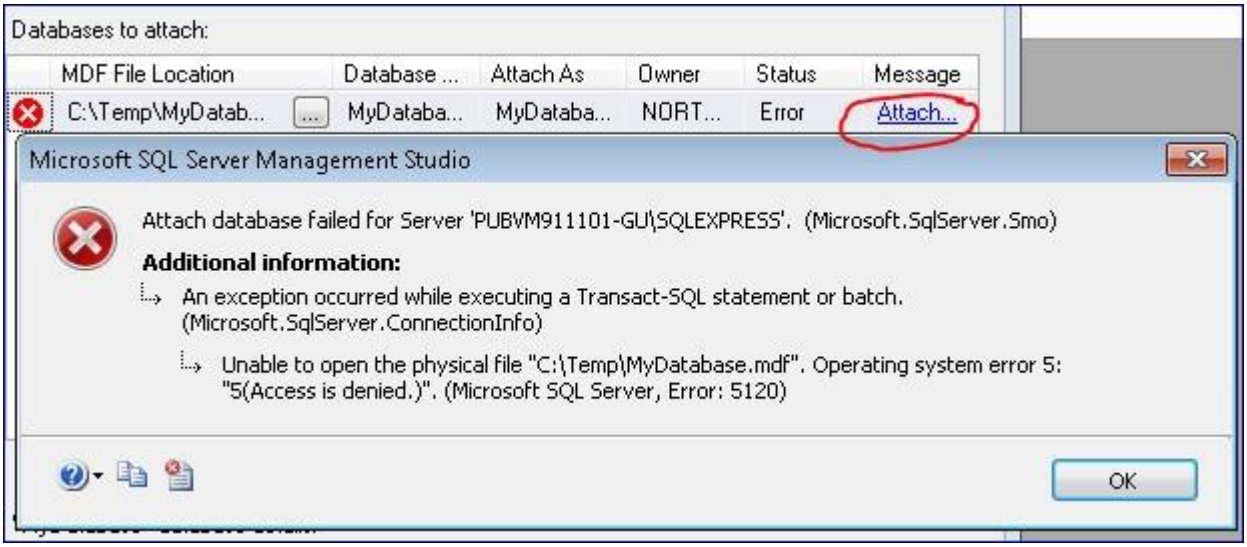
First, you may get a generic error like the following..



An error occurred when attaching the database(s). Click the hyperlink in the Message column for details.

You might be asking, "Where's the hyperlink?" I'll show you in the next screenshot (circled in red), and it will also be our first example.

1. Access Denied:



> Attach database failed for Server ###. (Microsoft.SqlServer.Smo) Unable to open the physical file "####". Operating system error 5: "5(Access is denied.)". (Microsoft SQL Server, Error 5120)

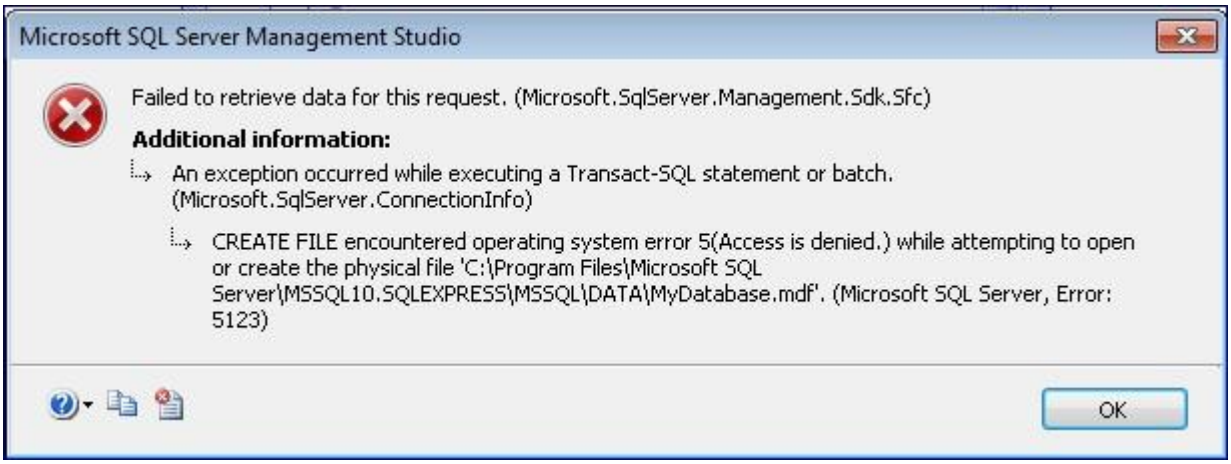
This access denied error could have a few different causes. Basically, you don't have the required access to the .MDF or .LDF files. This can happen if you get the file from someone else. When they detach the database file, the file permissions are changed to give only that user full control. Also, even if you see that the file has full control for the Administrators group, it may not be enough. Remember in Windows 7/Vista, those permissions are often only realized when you're running an application as an Administrator.

So here are several choices to fix it:

1. The easiest solution is to close SSMS and then run it as an Administrator. Perform the attach as an Administrator, and it's likely going to work.
2. Another solution is to explicitly grant full control to the .MDF and .LDF files to your user account. This can be done by right-clicking the files, selecting Properties, and modifying the Security tab.
3. A final solution is to copy the files to the default directory for your other database files. To find out what that is, you can use the sp_helpfile procedure in SSMS. On my machine it is: C:\Program Files\Microsoft SQL Server\MSSQL10_50.SQLEXPRESS\MSSQL\DATA. By copying the files to this directory, they automatically get permissions applied that will allow the attach to succeed.

2. Access Denied Variation 2

There is another variation of the Access Denied message that has a simple solution:



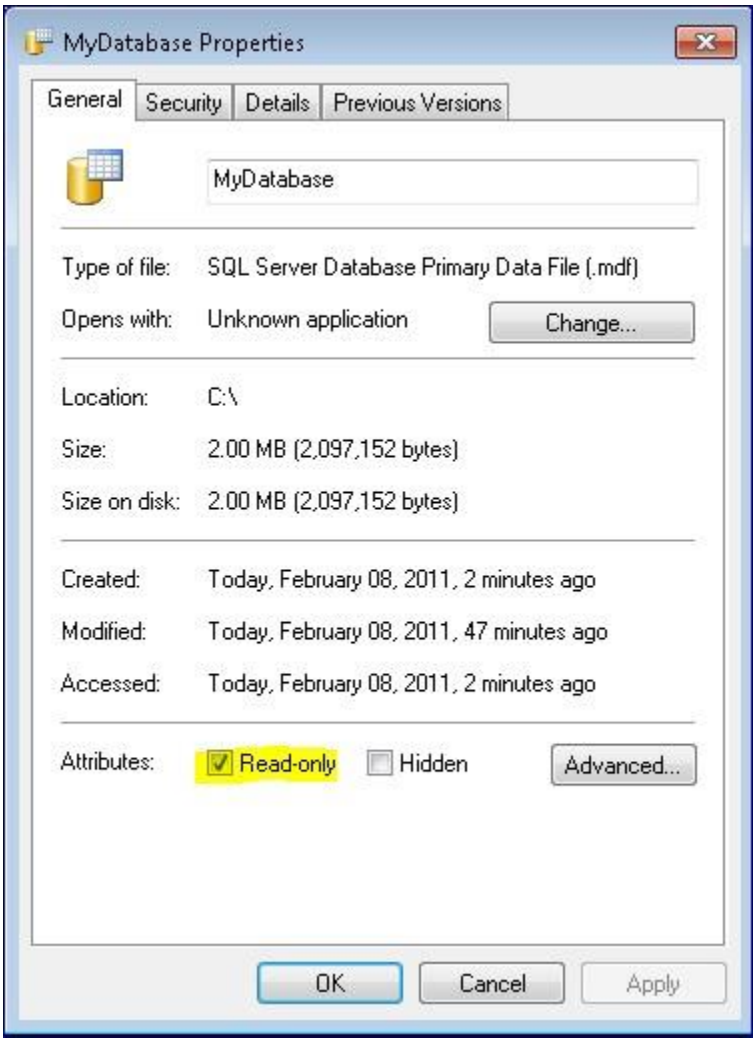
Failed to retrieve data for this request. (Microsoft.SqlServer.Management.Sdk.Sfc) CREATE FILE encountered operating system error 5(Access is denied.) while attempting to open or create the physical file '####'. (Microsoft SQL Server, Error: 5123)

This error means that the file is already opened exclusively by another application. The most likely cause is that this database file is already attached to an instance of SQL Server. Doublecheck your list of databases to see whether it is already in the list. It's also possible for applications to use .MDF files directly with a feature called User Instances. If an application is using a .MDF file in this way, then it would have to close before you could attach the to that database file.

3. Database is Read-only

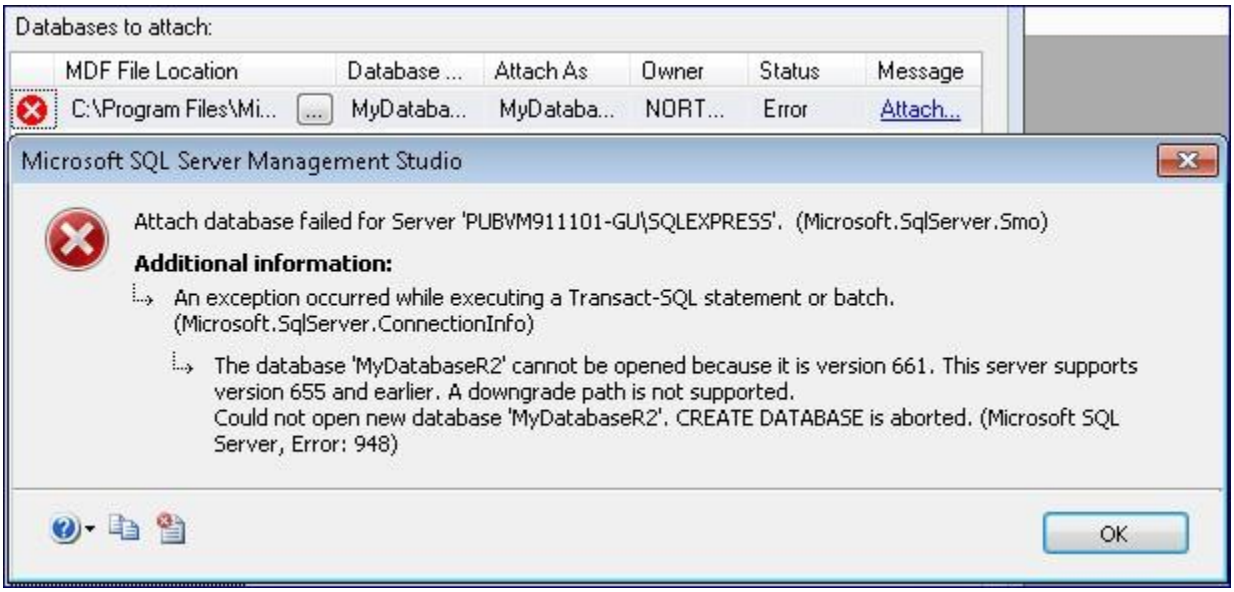
This is not so much an error as an undesired result. When you load the database, it is in a readonly state. You can tell by the gray shading of the databasee icon (not to mention the "ReadOnly" label next to it).

The most probable cause is that the .MDF file you attached to is read-only. Just right-click on the .MDF and .LDF files, select properties, and then uncheck the Read-only check box.



4. Unable to Downgrade

The next error happens when you try to attach a database of a higher version of SQL Server (like SQL Server 2008 R2) to a SQL Server Instance of a lower version (like SQL Server 2005).



Attach database failed for Server '####'. (Microsoft.SqlServer.Smo) An exception occurred while executing a Transact-SQL statement or batch. (Microsoft.SqlServer.ConnectionInfo) The database '####' cannot be opened because it is version 661. This server supports version 655 and earlier. A downgrade path is not supported. Could not open a new database '####'. CREATE DATABASE is aborted. (Microsoft SQL Server, Error: 948)

This has never been supported. You can attach a database of an older version of SQL Server to a newer version, but you can't go the other direction. If you want to verify the version of SQL Server you're running, see this post.

A Parting Word on Detach:

This post is just covering the scenarios of, “How do I open this .MDF file” and “Oh, I have to attach it, why am I getting errors?”. But how were these files obtained in the first place? The answer is that there is a corresponding Detach feature in SQL Server. You can find it by rightclicking on the database in SSMS, selecting tasks, and looking for Detach. Before you do this, though, run the sp_helpfile procedure in a query window in the context of your database. This will tell you *where* the detached .MDF file will be located. After you’ve detached the file, you can attach it to any SQL Server instance. Although this post was about Attach, I think it’s worth mentioning this for completeness.