SQL Server 2008 Books Online (August 2008)

Moving System Databases ✎ Send Feedback

⬀ See Also

⊟ Collapse All

Implementing Databases > Modifying a Database > Moving Database Files >

This topic describes how to move system databases in SQL Server. Moving system databases may be useful in the following situations:

- Failure recovery. For example, the database is in suspect mode or has shut down because of a hardware failure.
- Planned relocation.
- Relocation for scheduled disk maintenance.

The following procedures apply to moving database files within the same instance of SQL Server. To move a database to another instance of SQL Server or to another server, use the backup and restore or detach and attach operations.

The procedures in this topic require the logical name of the database files. To obtain the name, query the **name** column in the sys.master_files catalog view.

> **Important:**
> If you move a system database and later rebuild the **master** database, you must move the system database again because the rebuild operation installs all system databases to their default location. For more information about rebuilding the **master** database, see "Rebuilding System Databases, Rebuilding the Registry" in How to: Install SQL Server 2008 from the Command Prompt.

⊟ Planned Relocation and Scheduled Disk Maintenance Procedure

To move a system database data or log file as part of a planned relocation or scheduled maintenance operation, follow these steps. This procedure applies to all system databases except the **master** and **Resource** databases.

1. For each file to be moved, run the following statement.

🗐 Copy Code
```
ALTER DATABASE database_name MODIFY FILE ( NAME = logical_name , FILENAME = 'new_path\os_file_name' )
```
2. Stop the instance of SQL Server or shut down the system to perform maintenance. For more information, see Stopping Services.

3. Move the file or files to the new location.

4. Restart the instance of SQL Server or the server. For more information, see Starting and Restarting Services.

5. Verify the file change by running the following query.

🗐 Copy Code
```
SELECT name, physical_name AS CurrentLocation, state_desc
FROM sys.master_files
WHERE database_id = DB_ID(N'<database_name>');
```

If the **msdb** database is moved and the instance of SQL Server is configured for Database Mail, complete these additional steps.

1. Verify that Service Broker is enabled for the **msdb** database by running the following query.

🗐 Copy Code
```
SELECT is_broker_enabled
FROM sys.databases
WHERE name = N'msdb';
```
For more information about enabling Service Broker, see ALTER DATABASE (Transact-SQL).

2. Verify that Database Mail is working by sending a test mail. For more information, see Troubleshooting Database Mail.

⊟ Failure Recovery Procedure

If a file must be moved because of a hardware failure, follow these steps to relocate the file to a new location. This procedure applies to all system databases except the **master** and **Resource** databases.

> **Important:**
> If the database cannot be started, that is it is in suspect mode or in an unrecovered state, only members of the **sysadmin** fixed role can move the file.

1. Stop the instance of SQL Server if it is started.

2. Start the instance of SQL Server in master-only recovery mode by entering one of the following commands at the command prompt. The parameters specified in these commands are case sensitive. The commands fail when the parameters are not specified as shown.

   - For the default (MSSQLSERVER) instance, run the following command:

🗐 Copy Code
```
NET START MSSQLSERVER /f /T3608
```
   - For a named instance, run the following command:

🗐 Copy Code
```
NET START MSSQL$instancename /f /T3608
```
For more information, see How to: Start an Instance of SQL Server (net Commands).

3. For each file to be moved, use **sqlcmd** commands or SQL Server Management Studio to run the following statement.

🗐 Copy Code
```
ALTER DATABASE database_name MODIFY FILE( NAME = logical_name , FILENAME = 'new_path\os_file_name' )
```
For more information about using the **sqlcmd** utility, see Using the sqlcmd Utility.

4. Exit the **sqlcmd** utility or SQL Server Management Studio.

5. Stop the instance of SQL Server. For example, run NET STOP MSSQLSERVER.

6. Move the file or files to the new location.

7. Restart the instance of SQL Server. For example, run NET START MSSQLSERVER.

8. Verify the file change by running the following query.

```
SELECT name, physical_name AS CurrentLocation, state_desc
FROM sys.master_files
WHERE database_id = DB_ID(N'<database_name>');
```

Moving the master Database

To move the **master** database, follow these steps.

1. From the **Start** menu, point to **All Programs**, point to **Microsoft SQL Server**, point to **Configuration Tools**, and then click **SQL Server Configuration Manager**.

2. In the **SQL Server Services** node, right-click the instance of SQL Server (for example, **SQL Server (MSSQLSERVER)**) and choose **Properties**.

3. In the **SQL Server (***instance_name***) Properties** dialog box, click the **Advanced** tab.

4. Edit the **Startup Parameters** values to point to the planned location for the **master** database data and log files, and click **OK**. Moving the error log file is optional.

   The parameter value for the data file must follow the –d parameter and the value for the log file must follow the –l parameter. The following example shows the parameter values for the default location of the **master** data and log files.

```
-dC:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\master.mdf;-eC:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL
```
   If the planned relocation for the **master** data and log files is E:\SQLData, the parameter values would be changed as follows:

```
-dE:\SQLData\master.mdf;-eC:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\LOG\ERRORLOG;-lE:\SQLData\mastlog.ldf
```
5. Stop the instance of SQL Server by right-clicking the instance name and choosing **Stop**.

6. Move the master.mdf and mastlog.ldf files to the new location.

7. Restart the instance of SQL Server.

8. Verify the file change for the **master** database by running the following query.

```
SELECT name, physical_name AS CurrentLocation, state_desc
FROM sys.master_files
WHERE database_id = DB_ID('master');
GO
```

Moving the Resource Database

In SQL Server 2008, the default location of the **Resource** database is *<drive>*:\Program Files\Microsoft SQL Server\MSSQL10.*<instance_name>*\Binn\. The database can be moved; however, we recommend against moving it for two reasons:

● Applying SQL Server service packs and hotfixes restores the database to the \Binn location.

● Moving the **Resource** database in a failover cluster environment to a nonclustered location will cause failover cluster failure.

To move the **Resource** database, follow these steps.

1. Stop the instance of SQL Server if it is started.

2. Start the instance of SQL Server in master-only recovery mode by entering one of the following commands at the command prompt. The parameters specified in these commands are case sensitive. The commands fail when the parameters are not specified as shown.

   ● For the default (MSSQLSERVER) instance, run the following command.

```
NET START MSSQLSERVER /f /T3608
```
   ● For a named instance, run the following command.

```
NET START MSSQL$instancename /f /T3608
```
   For more information, see How to: Start an Instance of SQL Server (net Commands).

3. Using **sqlcmd** commands or SQL Server Management Studio, run the following statements. Change the FILENAME path to match the new location of the data file. Do not change the name of the database or the file names.

```
ALTER DATABASE mssqlsystemresource
    MODIFY FILE (NAME=data, FILENAME= 'new_path_of_master\mssqlsystemresource.mdf');
GO
ALTER DATABASE mssqlsystemresource
    MODIFY FILE (NAME=log, FILENAME= 'new_path_of_master\mssqlsystemresource.ldf');
GO
```
4. Move the mssqlsystemresource.mdf and mssqlsystemresource.ldf files to the new location.

5. Set the **Resource** database to read-only by running the following statement.

```
ALTER DATABASE mssqlsystemresource SET READ_ONLY;
```
6. Exit the **sqlcmd** utility or SQL Server Management Studio.

7. Stop the instance of SQL Server.

8. Restart the instance of SQL Server.

Examples

A. Moving the tempdb database

The following example moves the `tempdb` data and log files to a new location as part of a planned relocation.

> **Note:**
> Because **tempdb** is re-created each time the instance of SQL Server is started, you do not have to physically move the data and log files. The files are created in the new location when the service is restarted in step 3. Until the service is restarted, **tempdb** continues to use the data and log files in existing location.

1. Determine the logical file names of the `tempdb` database and their current location on the disk.

   📋 Copy Code

   ```
   SELECT name, physical_name AS CurrentLocation
   FROM sys.master_files
   WHERE database_id = DB_ID(N'tempdb');
   GO
   ```

2. Change the location of each file by using `ALTER DATABASE`.

   📋 Copy Code

   ```
   USE master;
   GO
   ALTER DATABASE tempdb
   MODIFY FILE (NAME = tempdev, FILENAME = 'E:\SQLData\tempdb.mdf');
   GO
   ALTER DATABASE tempdb
   MODIFY FILE (NAME = templog, FILENAME = 'F:\SQLLog\templog.ldf');
   GO
   ```

3. Stop and restart the instance of SQL Server.

4. Verify the file change.

   📋 Copy Code

   ```
   SELECT name, physical_name AS CurrentLocation, state_desc
   FROM sys.master_files
   WHERE database_id = DB_ID(N'tempdb');
   ```

5. Delete the `tempdb.mdf` and `templog.ldf` files from the original location.

⊟ See Also

Concepts
Resource Database
tempdb Database
master Database
msdb Database
model Database
Moving User Databases

Other Resources
Moving Database Files
Stopping Services
Starting and Restarting Services
ALTER DATABASE (Transact-SQL)

Help and Information
Getting SQL Server 2008 Assistance

Documentation Feedback

Microsoft values your feedback. To rate this topic and send feedback about this topic to the documentation team, click a rating, and then click **Send Feedback**. For assistance with support issues, refer to the technical support information included with the product.

Poor   1   2   3   4   5   Outstanding
       ○   ○   ○   ○   ○

To e-mail your feedback to Microsoft, click here: [Send Feedback]