Praveen Madupu - +91 98661 30093
Sr SQL Server DBA, Dubai
praveensqldba12@gmail.com

Moving SQL Server system databases (such as `master`, `model`, `msdb`, and `tempdb`) from one drive to another can be necessary due to space limitations, performance optimizations, or other storage-related reasons.

The process must be done with great care since these databases are critical to the SQL Server instance's functionality. Below is a step-by-step guide to moving these databases.

Steps to Move SQL Server System Databases

1. Back up Databases (if applicable)

Even though the master, model, and msdb databases are system databases, it's a good practice to back up these databases before making any significant changes.

2. Identify the Current File Locations

First, you need to determine where the system databases are currently stored.

Run the following query in SQL Server Management Studio (SSMS):

```
SELECT name, physical_name AS CurrentLocation
FROM sys.master_files
WHERE database_id IN (1, 2, 3, 4);
```


This will return the current file locations for the following system databases:

- `master`: database_id = 1
- `tempdb`: database_id = 2
- `model`: database_id = 3
- `msdb`: database_id = 4

Praveen Madupu - +91 98661 30093
Sr SQL Server DBA, Dubai
praveensqldba12@gmail.com

3. Move the `master` Database

The `master` database is crucial because it stores information about all other databases, logins, server configuration, etc. Moving it requires the SQL Server instance to be restarted in minimal configuration mode.

a. Modify Startup Parameters

You need to modify the startup parameters for SQL Server to point to the new location of the `master` database files.

1. Open **SQL Server Configuration Manager**.
2. In the **SQL Server Services** section, right-click on the **SQL Server (InstanceName)** service and select **Properties**.
3. In the **Startup Parameters** tab, you will see entries like:

-dC:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\master.mdf

-lC:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\mastlog.ldf

These point to the `master` database's `.mdf` (data file) and `.ldf` (log file).

Modify these paths to the new location, such as:

-dD:\SQLData\master.mdf

-lD:\SQLLogs\mastlog.ldf

4. Click **Apply** and then **OK**.

b. Stop the SQL Server Service

Now, stop the SQL Server service from SQL Server Configuration Manager.

c. Move the Files

Go to the current location of the `master.mdf` and `mastlog.ldf` files and copy them to the new location specified in the startup parameters (e.g., `D:\SQLData\master.mdf` and `D:\SQLLogs\mastlog.ldf`).

d. Start the SQL Server Service

After moving the files, restart the SQL Server service.

If the `master` database files are moved correctly, the service should start successfully.

Praveen Madupu - +91 98661 30093
Sr SQL Server DBA, Dubai
praveensqldba12@gmail.com

4. Move the `msdb` and `model` Databases

The process for moving the `msdb` and `model` databases is simpler compared to the `master` database. It involves updating the file locations within SQL Server and restarting the instance.

a. Set the Database Offline

First, take the databases offline:

```
ALTER DATABASE msdb SET OFFLINE;

ALTER DATABASE model SET OFFLINE;
```

b. Move the Files

1. Navigate to the current location of the `msdb.mdf`, `msdblog.ldf`, `model.mdf`, and `modellog.ldf` files.
2. Copy these files to the new location (e.g., `D:\SQLData\msdb.mdf`, `D:\SQLLogs\msdblog.ldf`, etc.).

c. Modify File Locations in SQL Server

Once the files are moved, inform SQL Server of the new file locations by running the following commands:

```
-- For the model database
ALTER DATABASE model MODIFY FILE (NAME = modeldev, FILENAME = 'D:\SQLData\model.mdf');

ALTER DATABASE model MODIFY FILE (NAME = modellog, FILENAME = 'D:\SQLLogs\modellog.ldf');

-- For the msdb database
ALTER DATABASE msdb MODIFY FILE (NAME = MSDBData, FILENAME = 'D:\SQLData\msdb.mdf');

ALTER DATABASE msdb MODIFY FILE (NAME = MSDBLog, FILENAME = 'D:\SQLLogs\msdblog.ldf');
```


d. Bring the Databases Back Online

Now, bring the databases back online:

```
ALTER DATABASE msdb SET ONLINE;

ALTER DATABASE model SET ONLINE;
```

Praveen Madupu - +91 98661 30093
Sr SQL Server DBA, Dubai
praveensqldba12@gmail.com

5. Move the `tempdb` Database

The `tempdb` database is recreated every time SQL Server restarts, so you need to modify the file locations and restart the SQL Server instance.

a. Modify File Locations in SQL Server

Run the following commands to modify the location of the `tempdb` data and log files:

```
USE master;

GO

ALTER DATABASE tempdb MODIFY FILE (NAME = tempdev, FILENAME = 'D:\SQLData\tempdb.mdf');

ALTER DATABASE tempdb MODIFY FILE (NAME = templog, FILENAME = 'D:\SQLLogs\templog.ldf');

GO
```

b. Restart SQL Server

Stop and restart the SQL Server service to apply the changes. Upon restart, SQL Server will create new `tempdb` files in the new location.

6. Verify the Changes

After restarting SQL Server, verify that all system databases have been moved to the new locations by running the following query:

```
SELECT name, physical_name AS CurrentLocation

FROM sys.master_files

WHERE database_id IN (1, 2, 3, 4);
```

Ensure that the paths listed match the new file locations.

7. Update Backup/Restore Scripts and Maintenance Plans

If you have any backup scripts, restore scripts, or maintenance plans that reference the old file paths, update them to reflect the new paths.

Praveen Madupu - +91 98661 30093
Sr SQL Server DBA, Dubai
praveensqldba12@gmail.com

Summary

1. **Back up your system databases** (master, model, msdb) before making changes.
2. **Move `master`** by updating the SQL Server service startup parameters, moving the files, and restarting the service.
3. **Move `msdb` and `model`** by setting them offline, moving the files, and then bringing them back online after modifying the file paths.
4. **Move `tempdb`** by updating the file paths in SQL Server and restarting the service.
5. **Verify the new locations** after all changes have been made.

This process ensures a smooth relocation of your SQL Server system databases.