Rebuild System Databases

06/06/2016 • 9 minutes to read • Contributors

In this article

Before You Begin
Rebuild System Databases
Post-Rebuild Tasks
Rebuild the resource Database
Create a New msdb Database
Troubleshoot Rebuild Errors
See Also

THIS TOPIC APPLIES TO: SQL Server Azure SQL Database Azure SQL Data Warehouse Parallel Data Warehouse

System databases must be rebuilt to fix corruption problems in the master, model, msdb, or resource system databases or to modify the default server-level collation. This topic provides step-by-step instructions to rebuild system databases in SQL Server 2017.

In This Topic

- **Before you begin:**

  Limitations and Restrictions

  Prerequisites

- **Procedures:**

  Rebuild System Databases

  Rebuild the resource Database

  Create a New msdb Database

FEEDBACK

- **Follow Up:**

  Troubleshoot Rebuild Errors

Before You Begin

Limitations and Restrictions

When the master, model, msdb, and tempdb system databases are rebuilt, the databases are dropped and re-created in their original location. If a new collation is specified in the rebuild statement, the system databases are created using that collation setting. Any user modifications to these databases are lost. For example, you may have user-defined objects in the master database, scheduled jobs in msdb, or changes to the default database settings in the model database.

Prerequisites

Perform the following tasks before you rebuild the system databases to ensure that you can restore the system databases to their current settings.

1. Record all server-wide configuration values.

   Copy

   ```
   SELECT * FROM sys.configurations;
   ```

2. Record all service packs and hotfixes applied to the instance of SQL Server and the current collation. You must reapply these updates after rebuilding the system databases.

   Copy

   ```
   SELECT
   SERVERPROPERTY('ProductVersion ') AS ProductVersion,
   SERVERPROPERTY('ProductLevel') AS ProductLevel,
   SERVERPROPERTY('ResourceVersion') AS ResourceVersion,
   SERVERPROPERTY('ResourceLastUpdateDateTime') AS ResourceLastUpdateDateTime,
   SERVERPROPERTY('Collation') AS Collation;
   ```

3. Record the current location of all data and log files for the system databases. Rebuilding the system databases installs all system databases to their original location. If you have moved system database data or log files to a different location, you must move the files again.

   Copy

   ```
   SELECT name, physical_name AS current_file_location
   FROM sys.master_files
   ```

```
WHERE database_id IN (DB_ID('master'), DB_ID('model'), DB_ID('msdb'), DB_ID('tempd
```

4. Locate the current backup of the master, model, and msdb databases.

5. If the instance of SQL Server is configured as a replication Distributor, locate the current backup of the distribution database.

6. Ensure you have appropriate permissions to rebuild the system databases. To perform this operation, you must be a member of the **sysadmin** fixed server role. For more information, see Server-Level Roles.

7. Verify that copies of the master, model, msdb data and log template files exist on the local server. The default location for the template files is C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\Binn\Templates. These files are used during the rebuild process and must be present for Setup to succeed. If they are missing, run the Repair feature of Setup, or manually copy the files from your installation media. To locate the files on the installation media, navigate to the appropriate platform directory (x86 or x64) and then navigate to setup\sql_engine_core_inst_msi\Pfiles\SqlServr\MSSQL.X\MSSQL\Binn\Templates.

Rebuild System Databases

The following procedure rebuilds the master, model, msdb, and tempdb system databases. You cannot specify the system databases to be rebuilt. For clustered instances, this procedure must be performed on the active node and the SQL Server resource in the corresponding cluster application group must be taken offline before performing the procedure.

This procedure does not rebuild the resource database. See the section, "Rebuild the resource Database Procedure" later in this topic.

To rebuild system databases for an instance of SQL Server:

1. Insert the SQL Server 2017 installation media into the disk drive, or, from a command prompt, change directories to the location of the setup.exe file on the local server. The default location on the server is C:\Program Files\Microsoft SQL Server\130\Setup Bootstrap\SQLServer2016.

2. From a command prompt window, enter the following command. Square brackets are used to indicate optional parameters. Do not enter the brackets. When using a Windows operating system that has User Account Control (UAC) enabled, running Setup requires elevated privileges. The command prompt must be run as Administrator.

   Setup /QUIET /ACTION=REBUILDDATABASE /INSTANCENAME=InstanceName /SQLSYSADMINACCOUNTS=accounts [/SAPWD= StrongPassword] [/SQLCOLLATION=CollationName]

Parameter name	Description
/QUIET or /Q	Specifies that Setup run without any user interface.
/ACTION=REBUILDDATABASE	Specifies that Setup re-create the system databases.
/INSTANCENAME=*InstanceName*	Is the name of the instance of SQL Server. For the default instance, enter MSSQLSERVER.
/SQLSYSADMINACCOUNTS=*accounts*	Specifies the Windows groups or individual accounts to add to the **sysadmin** fixed server role. When specifying more than one account, separate the accounts with a blank space. For example, enter **BUILTIN\Administrators MyDomain\MyUser**. When you are specifying an account that contains a blank space within the account name, enclose the account in double quotation marks. For example, enter **NT AUTHORITY\SYSTEM**.
[/SAPWD=*StrongPassword*]	Specifies the password for the SQL Server **sa** account. This parameter is required if the instance uses Mixed Authentication (SQL Server and Windows Authentication) mode. **** Security Note ****The **sa** account is a well-known SQL Server account and it is often targeted by malicious users. It is very important that you use a strong password for the **sa** login. Do not specify this parameter for Windows Authentication mode.
[/SQLCOLLATION=*CollationName*]	Specifies a new server-level collation. This parameter is optional. When not specified, the current collation of the server is used. **** Important ****Changing the server-level collation does not change the collation of existing user databases. All newly created user databases will use the new collation by default. For more information, see Set or Change the Server Collation.
[/SQLTEMPDBFILECOUNT=NumberOfFiles]	Specifies the number of tempdb data files. This value can be increased up to 8 or the number of cores, whichever is higher.

	Default value: 8 or the number of cores, whichever is lower.
[/SQLTEMPDBFILESIZE=FileSizeInMB]	Specifies the initial size of each tempdb data file in MB. Setup allows the size up to 1024 MB. Default value: 8
[/SQLTEMPDBFILEGROWTH=FileSizeInMB]	Specifies the file growth increment of each tempdb data file in MB. A value of 0 indicates that automatic growth is off and no additional space is allowed. Setup allows the size up to 1024 MB. Default value: 64
[/SQLTEMPDBLOGFILESIZE=FileSizeInMB]	Specifies the initial size of the tempdb log file in MB. Setup allows the size up to 1024 MB. Default value: 8. Allowed range: Min = 8, max = 1024.
[/SQLTEMPDBLOGFILEGROWTH=FileSizeInMB]	Specifies the file growth increment of the tempdb log file in MB. A value of 0 indicates that automatic growth is off and no additional space is allowed. Setup allows the size up to 1024 MB. Default value: 64 Allowed range: Min = 8, max = 1024.
[/SQLTEMPDBDIR=Directories]	Specifies the directories for tempdb data files. When specifying more than one directory, separate the directories with a blank space. If multiple directories are specified the tempdb data files will be spread across the directories in a round-robin fashion. Default value: System Data Directory
[/SQLTEMPDBLOGDIR=Directory]	Specifies the directory for the tempdb log file. Default value: System Data Directory

3. When Setup has completed rebuilding the system databases, it returns to the command prompt with no messages. Examine the Summary.txt log file to verify that the process completed successfully. This file is located at C:\Program Files\Microsoft SQL Server\130\Setup Bootstrap\Logs.

4. RebuildDatabase scenario deletes system databases and installs them again in clean state. Because the setting of tempdb file count does not persist, the value of number of tempdb files is not known during setup. Therefore, RebuildDatabase scenario does not know the count of tempdb files to be re-added. You can provide the value of the number of tempdb files again with the SQLTEMPDBFILECOUNT parameter. If the parameter is not provided, RebuildDatabase will add a default number of tempdb files, which is as many tempdb files as the CPU count or 8, whichever is lower.

Post-Rebuild Tasks

After rebuilding the database you may need to perform the following additional tasks:

- Restore your most recent full backups of the master, model, and msdb databases. For more information, see Back Up and Restore of System Databases (SQL Server).

  > **Important**
  >
  > If you have changed the server collation, do not restore the system databases. Doing so will replace the new collation with the previous collation setting.

  If a backup is not available or if the restored backup is not current, re-create any missing entries. For example, re-create all missing entries for your user databases, backup devices, SQL Server logins, end points, and so on. The best way to re-create entries is to run the original scripts that created them.

> **Important**
>
> We recommend that you secure your scripts to prevent their being altered by unauthorized by individuals.

- If the instance of SQL Server is configured as a replication Distributor, you must restore the distribution database. For more information, see Back Up and Restore Replicated Databases.
- Move the system databases to the locations you recorded previously. For more information, see Move System Databases.
- Verify the server-wide configuration values match the values you recorded previously.

Rebuild the resource Database

The following procedure rebuilds the resource system database. When you rebuild the resource database, all service packs and hot fixes are lost, and therefore must be reapplied.

To rebuild the resource system database:

1. Launch the SQL Server 2017 Setup program (setup.exe) from the distribution media.
2. In the left navigation area, click **Maintenance**, and then click **Repair**.
3. Setup support rule and file routines run to ensure that your system has prerequisites installed and that the computer passes Setup validation rules. Click **OK** or **Install** to continue.
4. On the Select Instance page, select the instance to repair, and then click **Next**.
5. The repair rules will run to validate the operation. To continue, click **Next**.
6. From the **Ready to Repair** page, click **Repair**. The Complete page indicates that the operation is finished.

Create a New msdb Database

If the **msdb** database is damaged and you do not have a backup of the **msdb** database, you can create a new **msdb** by using the **instmsdb** script.

> ⊗ **Warning**
>
> Rebuilding the **msdb** database using the **instmsdb** script will eliminate all the information stored in **msdb** such as jobs, alert, operators, maintenance plans, backup history, Policy-Based Management settings, Database Mail, Performance Data Warehouse, etc.

1. Stop all services connecting to the Database Engine, including SQL Server Agent, SSRS, SSIS, and all applications using SQL Server as data store.
2. Start SQL Server from the command line using the command: `NET START MSSQLSERVER /T3608`

   For more information, see [Start, Stop, Pause, Resume, Restart the Database Engine, SQL Server Agent, or SQL Server Browser Service](#).
3. In another command line window, detach the **msdb** database by executing the following command, replacing *<servername>* with the instance of SQL Server:
   ```
   SQLCMD -E -S<servername> -dmaster -Q"EXEC sp_detach_db msdb"
   ```
4. Using the Windows Explorer, rename the **msdb** database files. By default these are in the DATA sub-folder for the SQL Server instance.
5. Using SQL Server Configuration Manager, stop and restart the Database Engine service normally.