

SQL Server Database Stuck in Restoring State

02/14/2024 9:00 am EST

<https://stackoverflow.com/questions/520967/sql-server-database-stuck-in-restoring-state>

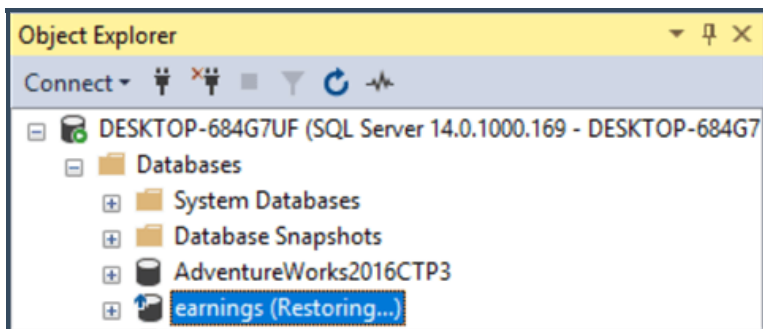
<https://www.mssqltips.com/sqlservertip/5460/sql-server-database-stuck-in-restoring-state/>

SQL Server Database Stuck in Restoring State

By: [Daniel Calbimonte](#) | Updated: 2021-10-28 | [Comments \(8\)](#) | Related: [More > Restore](#)

Problem

My Microsoft SQL Server database is in a restoring state. How does this happen and how can I access my SQL Server database?



Solution

In this article we will show reasons why a SQL Server database is in a restoring state and how you can get access to a database in a restoring state. It is not a very common problem, but when it happens it can be a big headache. In this article, we will see different reasons and possible solutions to solve this.

These steps will work for any version of SQL Server.

Microsoft SQL Server database in RESTORING state after a restore

Usually, the restoring state happens when you are restoring a database. Here we will walk through an example of this.

I will create a full backup file (*.bak file) and transaction log backup file (*.bak file) by running this T-SQL code in SQL Server Management Studio (SSMS).

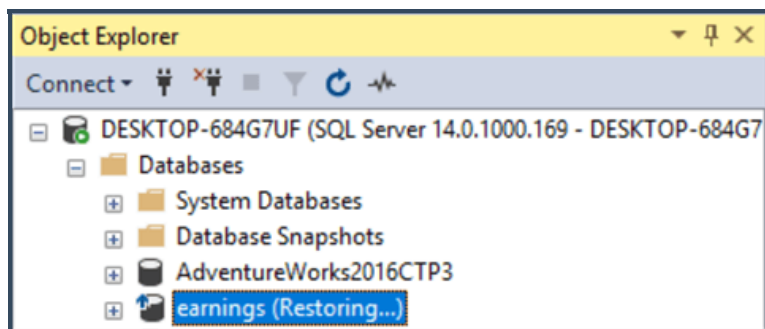
```
BACKUP DATABASE [earnings] TO DISK = N'c:\sql\earnings.bak'  
WITH NOFORMAT, NOINIT, NAME = N'earnings-Full Database Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10  
GO  
  
BACKUP LOG [earnings] TO DISK = N'C:\sql\earnings_LogBackup_2018-06-02_12-42-07.bak'  
WITH NOFORMAT, NOINIT, NAME = N'earnings_LogBackup_2018-06-02_12-42-07', SKIP, NOREWIND, NOUNLOAD, STATS = 1  
0
```

Once we have the SQL Server backups, we will start the restoring process.

In order to restore the full and log backup we need to use the **NORECOVERY** option for the full restore. So, if we just restore the full backup as follows:

```
RESTORE DATABASE [earnings]  
FROM DISK = N'c:\sql\earnings.bak' WITH NORECOVERY, NOUNLOAD, STATS = 10
```

The database will now be in a restoring state. If we forget to restore additional backups, the database will be stuck in this mode.



To finalize the restore and access the database we need to issue the restore command for the log backup as follows:

```
RESTORE LOG [earnings]  
FROM DISK = N'c:\sql\earnings_LogBackup_2018-06-02_12-42-07.bak'
```

SQL Server database in RESTORING state after doing backup log with NORECOVERY

Another reason your database can be in restoring state is when you backup the tail of the log using the **NORECOVERY** option as shown below.

```
BACKUP DATABASE [earnings] TO DISK = N'c:\sql\earnings.bak'  
WITH NOFORMAT, NOINIT, NAME = N'earnings-Full Database Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10  
GO  
  
BACKUP LOG [earnings] TO DISK = N'C:\sql\earnings_LogBackup_2018-06-02_12-42-07.bak'  
WITH NOFORMAT, NOINIT, NAME = N'earnings_LogBackup_2018-06-02_12-42-07', SKIP, NOREWIND, NOUNLOAD,  
NORECOVERY, STATS = 10
```

This will cause the database to change to a restoring state.

To fix this you can restore the database backups as shown above.

Make a SQL Server database in RESTORING state accessible without restoring backups

If the database is stuck in the restoring state and you don't have additional backups to restore, you can recover the database using the following command:

```
RESTORE DATABASE [earnings] WITH RECOVERY
```

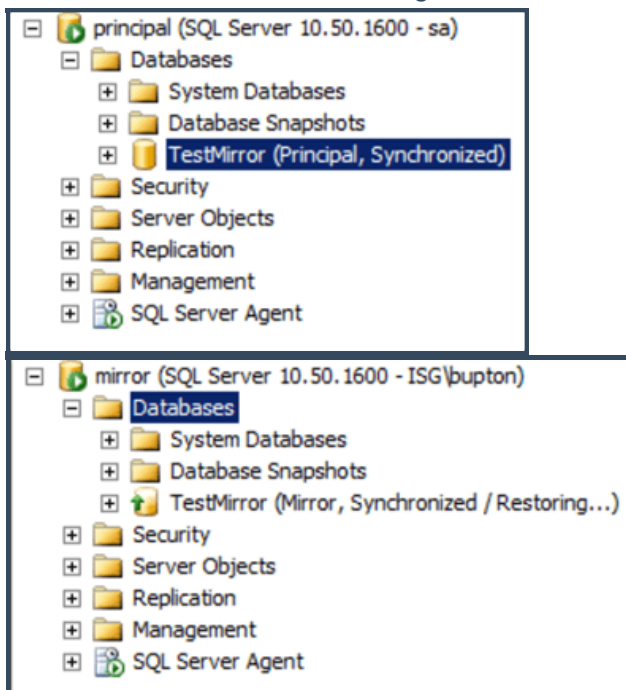
Once you issue this command, the database will be useable, but you won't be able to restore any additional backups for this database without starting all over again with the full backup.

For more details about restoring a database in a restoring state, refer to this article [Recovering a database that is in the restoring state](#).

SQL Server database in RESTORING state for Database Mirroring

Another reason your database is in a restoring state is that it is part of [SQL Server Database Mirroring](#). Database Mirroring is a solution that allows you to have high availability for your database. If there is a database failure on the primary database, the secondary replica database on a different server will take over the database operations. The main database is the Principal Server, the secondary is the Mirror Server and optionally you can have another Mirror Server.

Here is an example. We can see on the left that the Principal server is where the database is accessible. On the right we can see the Mirror that is in a Restoring state.



For more information about Database Mirroring in SQL Server, refer to this link: [Configure SQL Server Database Mirroring Using SSMS](#).

In Database Mirroring, the Mirror Server is in Restoring state until a Failover is done. To access a SQL Server database that is in a restoring state when it is part of Database Mirroring, you can do a manual or automatic failover from the Principal to the Mirror.

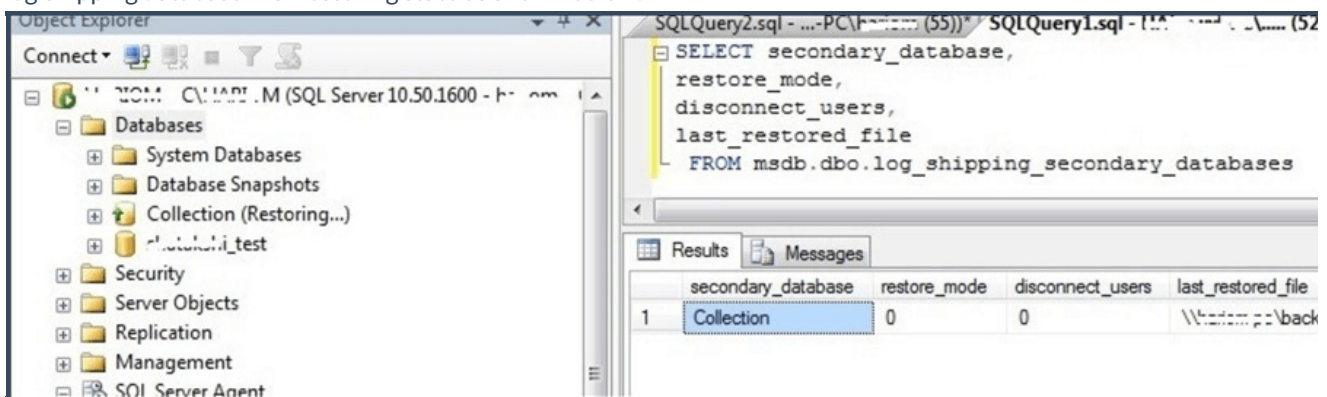
To do an automatic failover, refer to the following link: [Role Switching During a Database Mirroring Session \(SQL Server\)](#).

To break the mirror, you will need to select the database and go to the mirroring page and select the remove mirroring button. The following [article](#) shows how to do it. Once removed, the mirroring database will return to the normal state and you can backup and restore the database as a normal database.

SQL Server database in RESTORING state for Log Shipping

SQL Server Log Shipping allows you to backup the transaction logs and send and restore the backups on a different server in order to have replicas of the database in case the primary server fails.

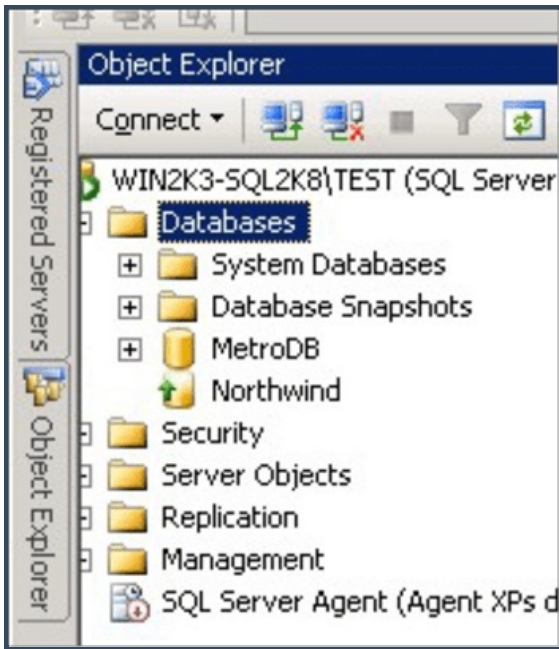
Log Shipping puts the database in a Standby state with recovery or with no recovery. The no recovery mode will show the Log Shipping database in a Restoring state as shown below.



Here is a link to change the state to avoid the restoring state: [Change the restore mode of a secondary SQL Server database in Log Shipping with SSMS](#).

SQL Server database stuck in RESTORING state after restarting the machine

Sometimes the database is in a restoring state after restarting the machine or for some other reason. It usually happens with big databases when a long transaction is in progress and an unexpected server shutdown or restart occurs.



If you have this problem, try this first:

```
RESTORE DATABASE [databasename] WITH RECOVERY
```

If you receive an error that the database is in use, try to set the user to single user mode:

```
USE master;
GO

ALTER DATABASE Database_name
SET SINGLE_USER
WITH ROLLBACK IMMEDIATE; Then try the restore with recovery command again.
```

Then try the restore with recovery command again.

Once restored, you can set to multiple user mode using the following T-SQL command:

```
USE master;
GO

ALTER DATABASE Database_name
SET MULTI_USER;
GO
```

Also, make sure you are on the latest service pack or cumulative update. There are several issues and patches related to recovery problems.

Here are some links with the service packs and cumulate updates:

- [How to obtain the latest service pack for SQL Server 2008](#)
- [How to obtain the latest service pack for SQL Server 2012](#)

- [How to obtain the latest service pack for SQL Server 2014](#)
- [How to obtain the latest service pack for SQL Server 2016](#)
- [Cumulative Update 7 for SQL Server 2017](#)

Also, you will should review the Error Log and the Windows Event Viewer to check for errors. Refer to these links:

- [Identify location of the SQL Server Error Log file](#)
- [Reading the SQL Server log files using TSQL](#)
- [SQL Server Error Logs Tips](#)
- [View the Windows application log \(Windows 10\)](#)

PowerShell to have the SQL Server in status

SQL Server can be run in PowerShell scripts. You can restore your database in PowerShell using the Restore-SqlDatabase cmdlet. By default, the SQL Server module is not installed by default. In order to install the SQL Server module, you may need to run the following command:

```
Install-Module -Name SqlServer
```

Once installed the module, you can run the restore-sqldatabase cmdlet to restore a transaction log with the NORECOVERY option as follows:

```
Restore-SqlDatabase -ServerInstance "myComputerInstance" -Database "Adventureworks2019"  
-BackupFile "c:\backup\adwlog.trn" -RestoreAction Log -NoRecovery
```

The command line will restore the database in a NoRecovery mode. Then the backup will be in restoring status.

When the SQL Server recovery will finish?

In a big database, restoring a database could take a long time. Is there a way to estimate the time that the restoration will take?

Fortunately yes. You can check this information in the error log. We created an exclusive tip for you related to this scenario:

- [Script to find out when SQL Server recovery will finish](#)

Where can I see the recovery status?

There is a system view used to see the recovery status. The name is sys.database_recovery_status. The view will show

information about the database id, the last log backup lsn (log sequence number) and the fork point lsn which is related to the current recovery fork.

Conclusion

In this article, we saw different reasons why a database could be in a Restoring state. Hopefully this will be helpful the next time you are troubleshooting this issue.

I backed up a database:

```
BACKUP DATABASE MyDatabase
TO DISK = 'MyDatabase.bak'
WITH INIT --overwrite existing
```

And then tried to restore it:

```
RESTORE DATABASE MyDatabase
FROM DISK = 'MyDatabase.bak'
WITH REPLACE --force restore over specified database
```

And now the database is stuck in the restoring state.

Some people have theorized that it's because there was no log file in the backup, and it needed to be rolled forward using:

```
RESTORE DATABASE MyDatabase
WITH RECOVERY
```

Except that, of course, fails:

```
Msg 4333, Level 16, State 1, Line 1
The database cannot be recovered because the log was not restored.
Msg 3013, Level 16, State 1, Line 1
RESTORE DATABASE is terminating abnormally.
```

And exactly what you want in a catastrophic situation is a restore that won't work.

The backup contains both a data and log file:

```
RESTORE FILELISTONLY
FROM DISK = 'MyDatabase.bak'
```

```
Logical Name  PhysicalName
=====
MyDatabase    C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\MyDatabase.mdf
MyDatabase_log C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\MyDatabase_log.LDF
```

[sql-serverbackuprestore](#)

I had the exact same issue and all of the solutions failed. Interestingly, I logged onto the SQL server directly and issued the DROP DATABASE db command through SSMS and it worked (earlier I was using SSMS from another machine to issue the commands). I am guessing the other solutions would have worked as well.

- Salman A

Apr 27 '17 at 12:44

[Add a comment](#)

[ActiveOldestScore](#)

784

I had this situation restoring a database to an SQL Server 2005 Standard Edition instance using Symantec Backup Exec 11d. After the restore job completed the database remained in a "Restoring" state. I had no disk space issues-- the database simply didn't come out of the "Restoring" state.

I ran the following query against the SQL Server instance and found that the database immediately became usable:

```
RESTORE DATABASE <database name> WITH RECOVERY
```