Back Up a Transaction Log (SQL Server)

02/01/2017 • 7 minutes to read • Contributors

In this article

Limitations and restrictions
Recommendations
Permissions
Back up using SSMS
Back up using T-SQL
Using PowerShell
Related tasks
More information

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

This topic describes how to back up a transaction log in SQL Server 2017 by using SQL Server Management Studio, Transact-SQL, or PowerShell.

Limitations and restrictions

- The BACKUP statement is not allowed in an explicit or implicit transaction. An explicit transaction is one in which you explicitly define both the start and end of the transaction.

Recommendations

- If a database uses either the full or bulk-logged recovery model, you must back up the transaction log regularly enough to protect your data, and to prevent the transaction log from filling. This truncates the log and supports restoring the database to a specific point in time.
- By default, every successful backup operation adds an entry in the SQL Server error log and in the system event log. If you back up the log frequently, these success messages accumulate

FEEDBACK

quickly, resulting in huge error logs, making finding other messages difficult. In such cases you can suppress these log entries by using trace flag 3226, if none of your scripts depend on those entries. For more information, see Trace Flags (Transact-SQL).

Permissions

Check for the correct permissions before you begin!

The BACKUP DATABASE and BACKUP LOG permissions needed are granted by default to members of the **sysadmin** fixed server role, and the **db_owner** and **db_backupoperator** fixed database roles.

Ownership and permission problems on the backup device's physical file can interfere with a backup operation. SQL Server must be able to read and write to the device; the account under which the SQL Server service runs must have write permissions. However, sp_addumpdevice, which adds an entry for a backup device in the system tables, does not check file access permissions. Permissions problems on the backup device's physical file may not become obvious to you until you attemt to access the physical resource when you try to backup or restore. So again, check permissions before you begin!

Back up using SSMS

1. After connecting to the appropriate instance of the SQL Server Database Engine, in Object Explorer, click the server name to expand the server tree.

2. Expand **Databases**, and, depending on the database, either select a user database or expand **System Databases** and select a system database.

3. Right-click the database, point to **Tasks**, and then click **Back Up**. The **Back Up Database** dialog box appears.

4. In the **Database** list box, verify the database name. You can optionally select a different database from the list.

5. Verify that the recovery model is either **FULL** or **BULK_LOGGED**.

6. In the **Backup type** list box, select **Transaction Log**.

7. Optionally, you can select **Copy Only Backup** to create a copy-only backup. A *copy-only backup* is a SQL Server backup that is independent of the sequence of conventional SQL Server backups. For more information, see Copy-Only Backups (SQL Server).

   > ** NOTE!** When the **Differential** option is selected, you cannot create a copy-only backup.

8. Either accept the default backup set name suggested in the **Name** text box, or enter a different name for the backup set.

9. Optionally, in the **Description** text box, enter a description of the backup set.

10. Specify when the backup set will expire:

    - To have the backup set expire after a specific number of days, click **After** (the default option), and enter the number of days after set creation that the set will expire. This value can be from 0 to 99999 days; a value of 0 days means that the backup set will never expire.

      The default value is set in the **Default backup media retention (in days)** option of the **Server Properties** dialog box (**Database Settings** page). To access this dialog box, right-click the server name in Object Explorer and select properties; then select the **Database Settings** page.

    - To have the backup set expire on a specific date, click **On**, and enter the date on which the set will expire.

11. Choose the type of backup destination by clicking **Disk**, **URL** or **Tape**. To select the paths of up to 64 disk or tape drives containing a single media set, click **Add**. The selected paths are displayed in the **Backup to** list box.

    To remove a backup destination, select it and click **Remove**. To view the contents of a backup destination, select it and click **Contents**.

12. To view or select the advanced options, click **Options** in the **Select a page** pane.

13. Select an **Overwrite Media** option, by clicking one of the following:

    - **Back up to the existing media set**

      For this option, click either **Append to the existing backup set** or **Overwrite all existing backup sets**. For more information, see Media Sets, Media Families, and Backup Sets (SQL Server).

      Optionally, select **Check media set name and backup set expiration** to cause the backup operation to verify the date and time at which the media set and backup set expire.

      Optionally, enter a name in the **Media set name** text box. If no name is specified, a media set with a blank name is created. If you specify a media set name, the media (tape or disk) is checked to see whether the actual name matches the name you enter here.

      If you leave the media name blank and check the box to check it against the media, success will equal the media name on the media also being blank.

    - **Back up to a new media set, and erase all existing backup sets**

For this option, enter a name in the **New media set name** text box, and, optionally, describe the media set in the **New media set description** text box. For more information, see Media Sets, Media Families, and Backup Sets (SQL Server).

14. In the **Reliability** section, optionally, check:

    - **Verify backup when finished**.
    - **Perform checksum before writing to media**, and, optionally, **Continue on checksum error**. For information on checksums, see Possible Media Errors During Backup and Restore (SQL Server).

15. In the **Transaction log** section:

    - For routine log backups, keep the default selection, **Truncate the transaction log by removing inactive entries**.
    - To back up the tail of the log (that is, the active log), check **Back up the tail of the log, and leave database in the restoring state**.

      A tail-log backup is taken after a failure to back up the tail of the log in order to prevent work loss. Back up the active log (a tail-log backup) both after a failure, before beginning to restore the database, or when failing over to a secondary database. Selecting this option is equivalent to specifying the NORECOVERY option in the BACKUP LOG statement of Transact-SQL. For more information about tail-log backups, see Tail-Log Backups (SQL Server).

16. If you are backing up to a tape drive (as specified in the **Destination** section of the **General** page), the **Unload the tape after backup** option is active. Clicking this option activates the **Rewind the tape before unloading** option.

17. SQL Server 2008 Enterprise and later supports backup compression. By default, whether a backup is compressed depends on the value of the **backup-compression default** server configuration option. However, regardless of the current server-level default, you can compress a backup by checking **Compress backup**, and you can prevent compression by checking **Do not compress backup**.

    To view the current backup compression default

    - View or Configure the backup compression default Server Configuration Option

    Encryption

    To encrypt the backup file check the **Encrypt backup** check box. Select an encryption algorithm to use for encrypting the backup file and provide a Certificate or Asymmetric key. The available algorithms for encryption are:

- AES 128
- AES 192
- AES 256
- Triple DES

Back up using T-SQL

1. Execute the BACKUP LOG statement to back up the transaction log, specifying the following:
    - The name of the database to which the transaction log that you want to back up belongs.
    - The backup device where the transaction log backup is written.

Example (Transact-SQL)

> **IMPORTANT!!!** This example uses the **AdventureWorks2012** database, which uses the simple recovery model. To permit log backups, before taking a full database backup, the database was set to use the full recovery model. For more information, see [View or Change the Recovery Model of a Database (SQL Server)](#).

This example creates a transaction log backup for the **AdventureWorks2012** database to the previously created named backup device, `MyAdvWorks_FullRM_log1`.

SQL Copy

```sql
BACKUP LOG AdventureWorks2012
   TO MyAdvWorks_FullRM_log1;
GO
```

Using PowerShell

1. Use the **Backup-SqlDatabase** cmdlet and specify **Log** for the value of the **-BackupAction** parameter.

    The following example creates a log backup of the `MyDB` database to the default backup location of the server instance `Computer\Instance`.

    Copy

```
--Enter this command at the PowerShell command prompt, C:\PS>
Backup-SqlDatabase -ServerInstance Computer\Instance -Database MyDB -BackupAction
```

To set up and use the SQL Server PowerShell provider

- SQL Server PowerShell Provider

Related tasks

- Restore a Transaction Log Backup (SQL Server)
- Restore a SQL Server Database to a Point in Time (Full Recovery Model)
- Troubleshoot a Full Transaction Log (SQL Server Error 9002)

More information

BACKUP (Transact-SQL)
Apply Transaction Log Backups (SQL Server)
Maintenance Plans
Full File Backups (SQL Server)

> **Note**
>
> The feedback system for this content will be changing soon. Old comments will not be carried over. If content within a comment thread is important to you, please save a copy. For more information on the upcoming change, **we invite you to read our blog post**.

0 Comments

Sign in

15 people listening

+ Follow

Post comment as...