**How to: Back Up a Database (SQL Server Management Studio)**

This topic explains how to create a full database backup.

|  |
| --- |
| **NoteNote** |
| When you specify a back up task by using SQL Server Management Studio, you can generate the corresponding Transact-SQL [BACKUP](http://msdn.microsoft.com/en-us/library/ms186865(SQL.105).aspx) script by clicking the **Script** button and selecting a script destination. |

To back up a database

1. After connecting to the appropriate instance of the Microsoft 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. You can perform a database backup for any recovery model (**FULL**, **BULK\_LOGGED**, or **SIMPLE**).
6. In the **Backup type** list box, select **Full**.

Note that after creating a full database backup, you can create a differential database backup; for more information, see [How to: Create a Differential Database Backup (SQL Server Management Studio)](http://msdn.microsoft.com/en-us/library/ms188248(SQL.105).aspx).

1. 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](http://msdn.microsoft.com/en-us/library/ms191495(SQL.105).aspx).

|  |
| --- |
| **NoteNote** |
| When the **Differential** option is selected, you cannot create a copy-only backup. |

1. For **Backup component**, click **Database**.
2. Either accept the default backup set name suggested in the **Name** text box, or enter a different name for the backup set.
3. Optionally, in the **Description** text box, enter a description of the backup set.
4. Specify when the backup set will expire and can be overwritten without explicitly skipping verification of the expiration data:
   * 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, 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.

For more information about backup expiration dates, see [BACKUP (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms186865(SQL.105).aspx).

1. Choose the type of backup destination by clicking **Disk** 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**.

1. To view or select the advanced options, click **Options** in the **Select a page** pane.
2. 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 [Appending to Existing Backup Sets](http://msdn.microsoft.com/en-us/library/ms191505(SQL.105).aspx) and [Overwriting Backup Sets](http://msdn.microsoft.com/en-us/library/ms191211(SQL.105).aspx).

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.

* + **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 [Creating a New Media Set](http://msdn.microsoft.com/en-us/library/ms188726(SQL.105).aspx).

1. 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 [Detecting and Coping with Media Errors During Backup and Restore](http://msdn.microsoft.com/en-us/library/ms189055(SQL.105).aspx).
2. 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.

|  |
| --- |
| **NoteNote** |
| The options in the **Transaction log** section are inactive unless you are backing up a transaction log (as specified in the **Backup type** section of the **General** page). |

1. SQL Server 2008 Enterprise and later supports [backup compression](http://msdn.microsoft.com/en-us/library/bb964719(SQL.105).aspx). 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**

* + [How to: View or Change the backup compression default Option (SQL Server Management Studio)](http://msdn.microsoft.com/en-us/library/bb933863(SQL.105).aspx)

# How to Backup Database in SQL Server - Create a Full Back up of Database

Feb 20, 2012  
*Page: 1/3*

## Introduction

In this article we will take a look at the steps you need to follow to take a full backup of SQL Server Database. The steps mentioned in this article are applicable for SQL Server 2005, SQL Server 2008, SQL Server 2008 R2 and SQL Server 2012. If you are interested in knowing the steps you need to follow to restore a full database back up in SQL Server then read the following article titled “[**How to Restore Database in SQL Server**](http://www.mytechmantra.com/LearnSQLServer/How-to-Restore-Database-SQL-Server-P1.html)”.

"This article has a total of 3 pages including this page. Click the links below to open all 3 pages in a seperate window to learn How to Back up database in SQL Server. [Page 1](http://www.mytechmantra.com/LearnSQLServer/How-to-Backup-Database-SQL-Server-P1.html), [Page 2](http://www.mytechmantra.com/LearnSQLServer/How-to-Backup-Database-SQL-Server-P2.html), [Page 3](http://www.mytechmantra.com/LearnSQLServer/How-to-Backup-Database-SQL-Server-P3.html)"

### Permissions Required to Take Database Backup in SQL Server

In order to take backups a user must be a member of DBCREATOR Server Role and DB\_OWNER Database Role else you will receive the below mentioned error while performing backups.

### Error Message

Msg 262, Level 14, State 1, Line 1  
BACKUP DATABASE permission denied in database 'DatabaseName'.  
Msg 3013, Level 16, State 1, Line 1  
BACKUP DATABASE is terminating abnormally.

### Sample TSQL Script to Create Login with DBCREATOR Server Role and DB\_OWNER Database Role

CREATE LOGIN BackupRestoreAdmin WITH PASSWORD='$tr0ngP@$$w0rd'  
GO  
CREATE USER BackupRestoreAdmin FOR LOGIN BackupRestoreAdmin  
GO  
EXEC sp\_addsrvrolemember 'BackupRestoreAdmin', 'dbcreator'  
GO  
EXEC sp\_addrolemember 'db\_owner','BackupRestoreAdmin'  
GO

Database Administrator or a user who is a member of **DBCREATOR** Server Role and **DB\_OWNER** Database Role can take a full backup of database using either:-

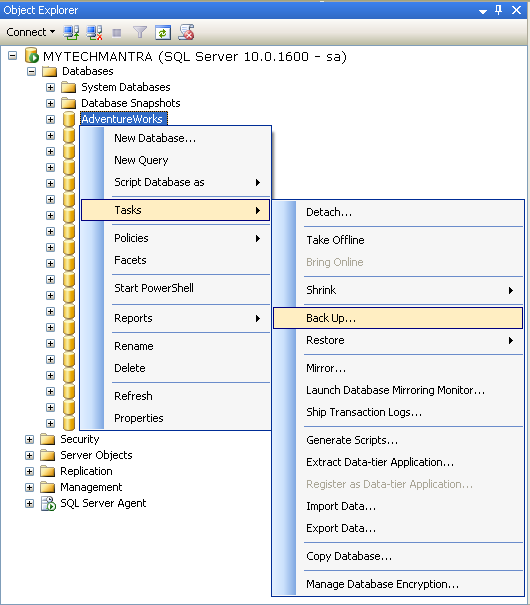
• Using SQL Server Management Studio   
• Using TSQL scripts

Let us take a look at both the above mentioned options to perform a full backup of a database in SQL Server.

### Take Full Backup of a Database Using SQL Server Management Studio

1. Open **SQL Server Management Studio** and connect to the appropriate instance of Microsoft SQL Server Database Engine in **Object Explorer**.

2. Expand **Databases** node and then right click the database which you want to take a full backup and point to **Tasks**, and then click **Back Up…** option as shown in the below snippet to open up **Back Up Database** dialog box.

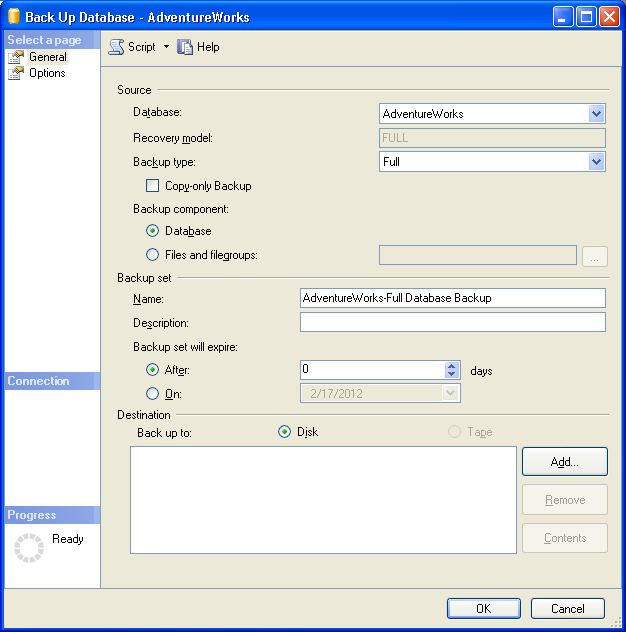


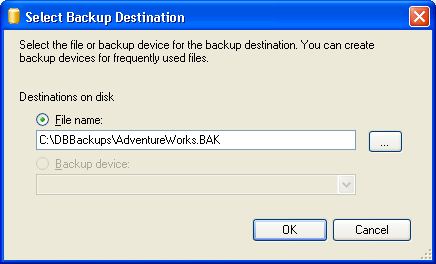
3. In **Back Up Database** Dialog box, in Database List box, verify the database name. Optionally you can even select a different database to backup. You will be able to perform a Full Backup of a database which is **FULL, BULK\_LOGGED** or **Simple** recovery model. To know more about different recovery models in SQL Server read the following article titled [**Understanding SQL Server Recovery Model**](http://www.mytechmantra.com/LearnSQLServer/SQL_Server_Recovery_Model.html).

4. In **Backup type** list box select Full. Once you have taken a full backup of a database then you can perform **Differential** **backup** or **Transactional log** **backups**. However, if you database is in Simple recovery model you will not be able to take a Transactional log backup. This is by design from Microsoft.

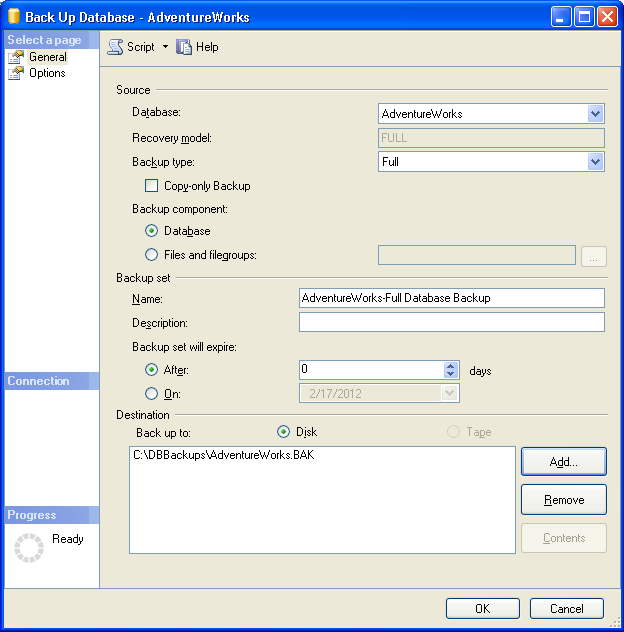
5. If you want to perform a [**Copy-Only**](http://www.mytechmantra.com/LearnSQLServer/COPY-ONLY-Backups-In-SQL-Server-P1.html)backup of a database then choose **Copy-Only Backup** option in Back Up Database Dialog box. Using copy-only backup option you can take a full, differential or transactional log backup which is independent of the sequence of convential SQL Server backups. The copy-only backup option was introduced in SQL Server 2005.

6. In **Backup Component** select **Databases** and in **Backup set** leave the default Name and optionally enter Desciption. Leave the default value as 0 days for **Backup set will expire after** option.

  
7. Click the **Add** button to **select the file or backup device for the backup destination** as shown in the below snippet and click OK to return to Backup Database Dialog box.

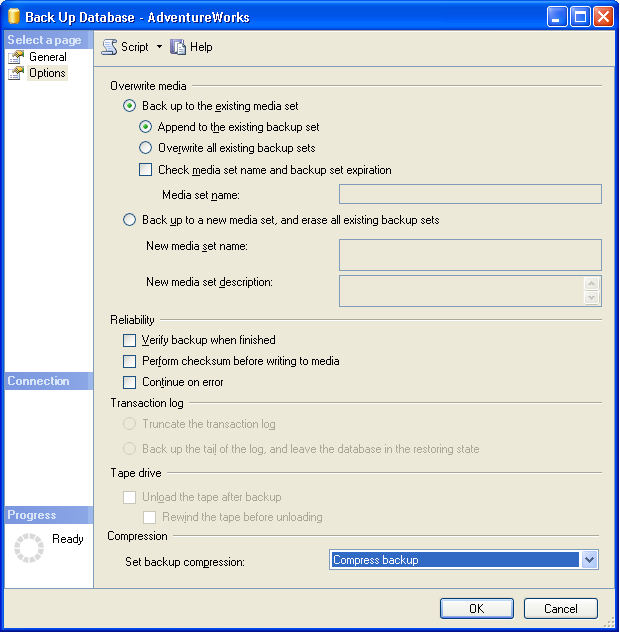


8. Once you have completed the **General settings** you will be able to see a screen similar to below snippet.

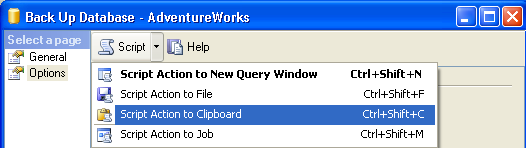


9. Select **Options** in the **Select a page** pane to view the advanced options. In **Overwrite Media** choose **Back up to the existing media set** and **Append to the existing backup set** option as shown in the below snippet.

10. Under **Reliability** section, you can select the checkbox **Verify backup when finished** option. By selecting this option you can make sure the database backup is good.

 11. Under **Compression** section, you can select the Compress backup from the drop down list as shown in the below snippet. Microsoft introduced Database Backup Compression Feature in SQL Server 2008 for Enterprise Edition. To know more about Backup Compression read the article titled[**Implement Backup Compression Feature In SQL Server 2008**](http://www.mytechmantra.com/LearnSQLServer/Backup_Compression_Feature_In_SQL_Server_2008_P1.html).

12. To generate TSQL Script for the database backup click Scripts and choose Script Action to your choice from the different options which are available as shown in the snippet below.



13. Once the database is successfully backed up you will get a popup message similar to the one shown in below snippet.

