Restore Files and Filegroups (SQL Server)

03/14/2017 • 8 minutes to read • Contributors

In this article

Before You Begin
Using SQL Server Management Studio
Using Transact-SQL
See Also

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

This topic describes how to restore files and filegroups in SQL Server 2017 by using SQL Server Management Studio or Transact-SQL.

In This Topic

- **Before you begin:**

  Limitations and Restrictions

- Security

- **To restore files and filegroups, using:**

  SQL Server Management Studio

  Transact-SQL

Before You Begin

Limitations and Restrictions

FEEDBACK

- The system administrator restoring the files and filegroups must be the only person currently using the database to be restored.
- RESTORE is not allowed in an explicit or implicit transaction.
- Under the simple recovery model, the file must belong to a read-only filegroup.
- Under the full or bulk-logged recovery model, before you can restore files, you must back up the active transaction log (known as the tail of the log). For more information, see Back Up a Transaction Log (SQL Server).
- To restore a database that is encrypted, you must have access to the certificate or asymmetric key that was used to encrypt the database. Without the certificate or asymmetric key, the database cannot be restored. As a result, the certificate that is used to encrypt the database encryption key must be retained as long as the backup is needed. For more information, see SQL Server Certificates and Asymmetric Keys.

Security

Permissions

If the database being restored does not exist, the user must have CREATE DATABASE permissions to be able to execute RESTORE. If the database exists, RESTORE permissions default to members of the **sysadmin** and **dbcreator** fixed server roles and the owner (**dbo**) of the database (for the FROM DATABASE_SNAPSHOT option, the database always exists).

RESTORE permissions are given to roles in which membership information is always readily available to the server. Because fixed database role membership can be checked only when the database is accessible and undamaged, which is not always the case when RESTORE is executed, members of the **db_owner** fixed database role do not have RESTORE permissions.

Using SQL Server Management Studio

To restore files and filegroups

1. After you connect 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**. Depending on the database, either select a user database or expand **System Databases**, and then select a system database.
3. Right-click the database, point to **Tasks**, and then click **Restore**.
4. Click **Files and Filegroups**, which opens the **Restore Files and Filegroups** dialog box.

5. On the **General** page, in the **To database** list box, enter the database to restore. You can enter a new database or choose an existing database from the drop-down list. The list includes all databases on the server, excluding the system databases **master** and **tempdb**.

6. To specify the source and location of the backup sets to restore, click one of the following options:

   - **From database**

     Enter a database name in the list box. This list contains only databases that have been backed up according to the **msdb** backup history.

   - **From device**

     Click the browse button. In the **Specify backup devices** dialog box, select one of the listed device types in the **Backup media type** list box. To select one or more devices for the **Backup media** list box, click **Add**.

     After you add the devices you want to the **Backup media** list box, click **OK** to return to the **General** page.

7. In the **Select the backup sets to restore** grid, select the backups to restore. This grid displays the backups available for the specified location. By default, a recovery plan is suggested. To override the suggested recovery plan, you can change the selections in the grid. Any backups that depend on a deselected backup are deselected automatically.

Column head	Values
Restore	The selected check boxes indicate the backup sets to be restored.
Name	The name of the backup set.
File Type	Specifies the type of data in the backup: **Data**, **Log**, or **Filestream Data**. Data that is contained in tables is in **Data** files. Transaction log data is in **Log** files. Binary large object (BLOB) data that is stored on the file system is in **Filestream Data** files.
Type	The type of backup performed: **Full**, **Differential**, or **Transaction Log**.
Server	The name of the Database-Engine instance that performed the backup operation.
File Logical Name	The logical name of the file.

Database	The name of the database involved in the backup operation.
Start Date	The date and time when the backup operation began, presented in the regional setting of the client.
Finish Date	The date and time when the backup operation finished, presented in the regional setting of the client.
Size	The size of the backup set in bytes.
User Name	The name of the user who performed the backup operation.

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

9. In the **Restore options** panel, you can choose any of the following options, if appropriate for your situation.

   Restore as filegroup
   Indicates that an entire filegroup is being restored.

   Overwrite the existing database
   Specifies that the restore operation should overwrite any existing databases and their related files, even if another database or file already exists with the same name.

   Selecting this option is equivalent to using the REPLACE option in a Transact-SQL RESTORE statement.

   Prompt before restoring each backup
   Asks you for confirmation before restoring each backup set.

   This option is particularly useful where you must swap tapes for different media sets, such as when the server has one tape device.

   Restrict access to the restored database
   Makes the restored database available only to the members of **db_owner**, **dbcreator**, or **sysadmin**.

   Selecting this option is synonymous to using the RESTRICTED_USER option in a Transact-SQL RESTORE statement.

10. Optionally, you can restore the database to a new location by specifying a new restore destination for each file in the **Restore database files as** grid.

Column head	Values
Original File Name	The full path of a source backup file.
File Type	Specifies the type of data in the backup: **Data**, **Log**, or **Filestream Data**. Data that is contained in tables is in **Data** files. Transaction log data is in **Log** files. Binary large object (BLOB) data that is stored on the file system is in **Filestream Data** files.
Restore As	The full path of the database file to be restored. To specify a new restore file, click the text box and edit the suggested path and file name. Changing the path or file name in the **Restore As** column is equivalent to using the MOVE option in a Transact-SQL RESTORE statement.

11. The **Recovery state** panel determines the state of the database after the restore operation.

    Leave the database ready for use by rolling back the uncommitted transactions. Additional transaction logs cannot be restored. (RESTORE WITH RECOVERY)
    Recovers the database. This is the default behavior. Choose this option only if you are restoring all of the necessary backups now. This option is equivalent to specifying WITH RECOVERY in a Transact-SQL RESTORE statement.

    Leave the database non-operational, and don't roll back the uncommitted transactions. Additional transaction logs can be restored. (RESTORE WITH NORECOVERY)
    Leaves the database in the restoring state. To recover the database, you will need to perform another restore using the preceding RESTORE WITH RECOVERY option (see above). This option is equivalent to specifying WITH NORECOVERY in a Transact-SQL RESTORE statement.

    If you select this option, the **Preserve replication settings** option is unavailable.

    Leave the database in read-only mode. Roll back the uncommitted transactions, but save the rollback operation in a file so the recovery effects can be undone. (RESTORE WITH STANDBY)
    Leaves the database in a standby state. This option is equivalent to specifying WITH STANDBY in a Transact-SQL RESTORE statement.

    Choosing this option requires that you specify a standby file.

    Rollback undo file
    Specify a standby file name in the **Rollback undo file** text box. This option is required if you leave the database in read-only mode (RESTORE WITH STANDBY).

Using Transact-SQL

To restore files and filegroups

1. Execute the RESTORE DATABASE statement to restore the file and filegroup backup, specifying:
   - The name of the database to restore.
   - The backup device from where the full database backup will be restored.
   - The FILE clause for each file to restore.
   - The FILEGROUP clause for each filegroup to restore.
   - The NORECOVERY clause. If the files have not been modified after the backup was created, specify the RECOVERY clause.
2. If the files have been modified after the file backup was created, execute the RESTORE LOG statement to apply the transaction log backup, specifying:
   - The name of the database to which the transaction log will be applied.
   - The backup device from where the transaction log backup will be restored.
   - The NORECOVERY clause if you have another transaction log backup to apply after the current one; otherwise, specify the RECOVERY clause.

     The transaction log backups, if applied, must cover the time when the files and filegroups were backed up until the end of log (unless ALL database files are restored).

Example (Transact-SQL)

This example restores the files and filegroups for the `MyDatabase` database. To restore the database to the current time, two transaction logs are applied.

SQL Copy

```sql
USE master;
GO
-- Restore the files and filesgroups for MyDatabase.
RESTORE DATABASE MyDatabase
   FILE = 'MyDatabase_data_1',
   FILEGROUP = 'new_customers',
   FILE = 'MyDatabase_data_2',
   FILEGROUP = 'first_qtr_sales'
   FROM MyDatabase_1
   WITH NORECOVERY;
GO
```

```
-- Apply the first transaction log backup.
RESTORE LOG MyDatabase
   FROM MyDatabase_log1
   WITH NORECOVERY;
GO
-- Apply the last transaction log backup.
RESTORE LOG MyDatabase
   FROM MyDatabase_log2
   WITH RECOVERY;
GO
```

See Also

Restore a Database Backup Using SSMS
Back Up Files and Filegroups (SQL Server)
Create a Full Database Backup (SQL Server)
Back Up a Transaction Log (SQL Server)
Restore a Transaction Log Backup (SQL Server)
RESTORE (Transact-SQL)

> **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

0 people listening

+ Follow

Post comment as...