

| impersonated | | |
|--------------|-----|--|
| Detach | Yes | Only the account performing the operation. Additional accounts can be added by an operating system administrator, if they are needed after the database is detached. |
| Detach | No | The SQL Server (MSSQLSERVER) service account and members of the local Windows Administrators group. |
| Attach | Yes | The SQL Server (MSSQLSERVER) service account and members of the local Windows Administrators group. |
| Attach | No | The SQL Server (MSSQLSERVER) service account. |

For more information on file system permissions granted to the per-service SIDs for the SQL Server service, see [Configure File System Permissions for Database Engine Access](#).

⊗ Caution

We recommend that you do not attach or restore databases from unknown or untrusted sources. Such databases could contain malicious code that might execute unintended Transact-SQL code or cause errors by modifying the schema or the physical database structure. Before you use a database from an unknown or untrusted source, run **DBCC CHECKDB** on the database on a nonproduction server and also examine the code, such as stored procedures or other user-defined code, in the database. For more information about attaching databases and information about changes that are made to metadata when you attach a database, see [Database Detach and Attach \(SQL Server\)](#).

Permissions

Requires `CREATE DATABASE`, `CREATE ANY DATABASE`, or `ALTER ANY DATABASE` permission.

Use SQL Server Management Studio (SSMS)

Before moving a database

If you are moving a database, before you detach it from its existing SQL Server instance, use the **Database properties** page to review the files associated with the database and their current locations.

1. In SQL Server Management Studio Object Explorer, connect to the instance of the SQL Server Database Engine and then expand the instance.
2. Expand **Databases**, and select the name of the user database you want to detach.
3. Right-click the database name, select **Properties**. Select the **Files** page and review the entries in the **Database files:** table.

Be sure to account for all files associated with the database before you detach, move, and attach. Then, proceed with the detach, file copy, and attach database steps in the next section. For more information, see [Detach a Database](#).

Attach a database

1. In SQL Server Management Studio **Object Explorer**, connect to an instance of SQL Server Database Engine, and then select to expand that instance view in SSMS.
2. Right-click **Databases** and select **Attach**.
3. In the **Attach Databases** dialog box, to specify the database to be attached, select **Add**. In the **Locate Database Files** dialog box, select the location where the database resides and expand the directory tree to find and select the .mdf file of the database; for example:

```
C:\Program Files\Microsoft SQL  
Server\MSSQL13.MSSQLSERVER\MSSQL\DATA\AdventureWorks2012_Data.mdf
```

Important

Trying to select a database that is already attached generates an error.

Databases to attach

Displays information about the selected databases.

<no column header>

Displays an icon indicating the status of the attach operation. The possible icons are described in the **Status** description.

MDF File Location

Displays the path and file name of the selected MDF file.

Displays the path and file name of the selected MDF file.

Database Name

Displays the name of the database.

Attach As

Optionally, specifies a different name for the database to attach as.

Owner

Provides a drop-down list of possible database owners from which you can optionally select a different owner.

Status

Displays the status of the database according to the following table:

| Icon | Status text | Description |
|---|-------------|---|
| (No icon) | (No text) | Attach operation has not been started or might be pending for this object. This is the default when the dialog is opened. |
| Green, right-pointing triangle | In progress | Attach operation has been started but it is not complete. |
| Green check mark | Success | The object has been attached successfully. |
| Red circle containing a white cross | Error | Attach operation encountered an error and did not complete successfully. |
| Circle containing two black quadrants (on left and right) and two white quadrants (on top and bottom) | Stopped | Attach operation was not completed successfully because you stopped the operation. |
| Circle containing a curved arrow pointing counter-clockwise | Rolled Back | Attach operation was successful but it has been rolled back due to an error during attachment of another object. |

Message

Displays either a blank message or a "File not found" hyperlink.

Add

Find the necessary main database files. When you select an .mdf file, applicable information is automatically filled in the respective fields of the **Databases to attach** grid.

Remove

Removes the selected file from the **Databases to attach** grid.

" <database_name> " database details

Displays the names of the files to be attached. To verify or change the pathname of a file, select the **Browse** button (...).

ⓘ Note

If a file does not exist, the **Message** column displays "Not found." If a log file is not found, it exists in another directory or has been deleted. You need to either update the file path in the **database details** grid to point to the correct location or remove

the log file from the grid. If an .ndf data file is not found, you need to update its path in the grid to point to the correct location.

Original File Name

Displays the name of the attached file belonging to the database.

File Type

Indicates the type of file, **Data** or **Log**.

Current File Path

Displays the path to the selected database file. The path can be edited manually.

Message

Displays either a blank message or a "**File not found**" hyperlink.

Use Transact-SQL

Before moving a database

If you are moving a database, before it is detached from its existing SQL Server instance, use the `sys.database_files` system catalog view to review the files associated with the

database and their current locations. For more information, see [sys.database_files](#) (Transact-SQL).

1. In SQL Server Management Studio, select **New Query** to open the Query Editor.
2. Copy the following Transact-SQL script into the Query Editor, and then select **Execute**. This script displays the location of the physical database files. Be sure to account for all files when you move the database via detach/attach.

SQL

```
USE [database_name]
GO
SELECT type_desc, name, physical_name from sys.database_files;
```

Be sure to account for all files associated with the database before you detach, move, and attach. Then, proceed with the detach, file copy, and attach database steps in the next section. For more information, see [Detach a Database](#).

To attach a database

1. Connect to the Database Engine.
2. From the Standard bar, select **New Query**.
3. Use the [CREATE DATABASE](#) statement with the FOR ATTACH clause.

Copy and paste the following example into the query window and select **Execute**. This example attaches the all files of the AdventureWorks2019 database and renames the database to MyAdventureWorks .

SQL

```
CREATE DATABASE MyAdventureWorks
ON (FILENAME = 'C:\MySQLServer\AdventureWorks_Data.mdf'),
(FILENAME = 'C:\MySQLServer\AdventureWorks_Log.ldf')
FOR ATTACH;
```

Your database might have additional data files (commonly, .mdf or .ndf), and require additional files to include in the CREATE DATABASE ... FOR ATTACH statement.

Additionally, any filegroups for FILESTREAM data must also be included in the statement. For more information on attaching a FILESTREAM-enabled database, see [Move a FILESTREAM Enabled Database](#).

Note

Alternatively, you can use the `sp_attach_db` or `sp_attach_single_file_db` stored procedure. However, these procedures will be removed in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature. We recommend that you use `CREATE DATABASE ... FOR ATTACH` instead.

After upgrading a SQL Server database

Database compatibility level

After you upgrade a database by using the attach method, the database becomes available. The database will be automatically upgraded to the internal version level of the new instance. If the database has full-text indexes, the upgrade process either imports, resets, or rebuilds them, depending on the setting of the **Full-Text Upgrade Option** server property. If the upgrade option is set to **Import** or **Rebuild**, the full-text indexes are unavailable during the upgrade. Depending on the amount of data being indexed, importing can take several hours, and rebuilding can take up to 10 times longer. Note also that when the upgrade option is set to **Import**, if a full-text catalog is not available, the associated full-text indexes are rebuilt.

After the upgrade, the database compatibility level remains at the compatibility level before the upgrade, unless the previous compatibility level is not supported on the new version. In this case, the upgraded database compatibility level is set to the lowest supported compatibility level. For example, if you attach a database that was compatibility level 90 before attaching it to an instance of SQL Server 2019 (15.x), after the upgrade the compatibility level is set to 100, which is the lowest supported compatibility level in SQL Server 2019 (15.x). For more information, see [ALTER DATABASE Compatibility Level \(Transact-SQL\)](#).

Change Data Capture (CDC)

If you are attaching a database from an instance of SQL Server 2014 (12.x) or earlier, which had Change Data Capture (CDC) enabled, you must execute the following command to upgrade the Change Data Capture (CDC) metadata:

```
SQL
```

```
USE <database name>  
EXEC sys.sp_cdc_vupgrade
```

For more information, see [Error when you attach a CDC-enabled database to an instance of SQL Server 2016 or SQL Server 2017 on Windows](#).

See also

- [Database Detach and Attach \(SQL Server\)](#)
- [Detach a Database](#)
- [CREATE DATABASE \(SQL Server Transact-SQL\)](#)
- [Configure File System Permissions for Database Engine Access](#)

Next steps

- [Manage metadata when making a database available on another server](#)
- [ALTER DATABASE Compatibility Level \(Transact-SQL\)](#)

