**FILESTREAM (SQL Server)**

## When to Use FILESTREAM

In SQL Server, BLOBs can be standard **varbinary(max)** data that stores the data in tables, or FILESTREAM **varbinary(max)** objects that store the data in the file system. The size and use of the data determines whether you should use database storage or file system storage. If the following conditions are true, you should consider using FILESTREAM:

* Objects that are being stored are, on average, larger than 1 MB.
* Fast read access is important.
* You are developing applications that use a middle tier for application logic.

For smaller objects, storing **varbinary(max)** BLOBs in the database often provides better streaming performance.

## FILESTREAM Storage

FILESTREAM storage is implemented as a **varbinary(max)** column in which the data is stored as BLOBs in the file system. The sizes of the BLOBs are limited only by the volume size of the file system. The standard **varbinary(max)** limitation of 2-GB file sizes does not apply to BLOBs that are stored in the file system.

To specify that a column should store data on the file system, specify the FILESTREAM attribute on a **varbinary(max)** column. This causes the Database Engine to store all data for that column on the file system, but not in the database file.

FILESTREAM data must be stored in FILESTREAM filegroups. A FILESTREAM filegroup is a special filegroup that contains file system directories instead of the files themselves. These file system directories are called data containers. Data containers are the interface between Database Engine storage and file system storage.

When you use FILESTREAM storage, consider the following:

* When a table contains a FILESTREAM column, each row must have a nonnull unique row ID.
* Multiple data containers can be added to a FILESTREAM filegroup.
* FILESTREAM data containers cannot be nested.
* When you are using failover clustering, the FILESTREAM filegroups must be on shared disk resources.
* FILESTREAM filegroups can be on compressed volumes.

# Enable and configure FILESTREAM

**APPLIES TO:** YesSQL Server NoAzure SQL Database NoAzure Synapse Analytics (SQL DW) NoParallel Data Warehouse

Before you can start to use FILESTREAM, you must enable FILESTREAM on the instance of the SQL Server Database Engine. This topic describes how to enable FILESTREAM by using SQL Server Configuration Manager.

## Enabling FILESTREAM

#### To enable and change FILESTREAM settings

1. On the **Start** menu, point to **All Programs**, point to SQL Server 2019 (15.x), point to **Configuration Tools**, and then click **SQL Server Configuration Manager**.
2. In the list of services, right-click **SQL Server Services**, and then click **Open**.
3. In the **SQL Server Configuration Manager** snap-in, locate the instance of SQL Server on which you want to enable FILESTREAM.
4. Right-click the instance, and then click **Properties**.
5. In the **SQL Server Properties** dialog box, click the **FILESTREAM** tab.
6. Select the **Enable FILESTREAM for Transact-SQL access** check box.
7. If you want to read and write FILESTREAM data from Windows, click **Enable FILESTREAM for file I/O streaming access**. Enter the name of the Windows share in the **Windows Share Name** box.
8. If remote clients must access the FILESTREAM data that is stored on this share, select **Allow remote clients to have streaming access to FILESTREAM data**.
9. Click **Apply**.
10. In SQL Server Management Studio, click **New Query** to display the Query Editor.
11. In Query Editor, enter the following Transact-SQL code:

EXEC sp\_configure filestream\_access\_level, 2

RECONFIGURE

1. Click **Execute**.
2. Restart the SQL Server service.

# Create a FILESTREAM-Enabled Database

**APPLIES TO:** YesSQL Server NoAzure SQL Database NoAzure Synapse Analytics (SQL DW) NoParallel Data Warehouse

This topic shows how to create a database that supports FILESTREAM. Because FILESTREAM uses a special type of filegroup, when you create the database, you must specify the CONTAINS FILESTREAM clause for at least one filegroup.

A FILESTREAM filegroup can contain more than one file. For a code example that demonstrates how to create a FILESTREAM filegroup that contains multiple files, see [CREATE DATABASE (SQL Server Transact-SQL)](https://docs.microsoft.com/en-us/sql/t-sql/statements/create-database-sql-server-transact-sql?view=sql-server-2017).

### To create a FILESTREAM-enabled database

1. In SQL Server Management Studio, click **New Query** to display the Query Editor.
2. Copy the Transact-SQL code from the following example into the Query Editor. This Transact-SQL code creates a FILESTREAM-enabled database called Archive.

CREATE DATABASE AspNetCoreFileUploadFileTable

ON PRIMARY

(Name = AspNetCoreFileUploadFileTable,

FILENAME = 'C:\databases\AspNetCoreFileUploadFileTable\FTDB.mdf'),

FILEGROUP FTFG CONTAINS FILESTREAM

(NAME = AspNetCoreFileUploadFileTableFS,

FILENAME='C:\databases\AspNetCoreFileUploadFileTable\FS')

LOG ON

(Name = AspNetCoreFileUploadFileTableLog,

FILENAME = 'C:\databases\AspNetCoreFileUploadFileTable\FTDBLog.ldf')

WITH FILESTREAM (NON\_TRANSACTED\_ACCESS = FULL,

DIRECTORY\_NAME = N'AspNetCoreFileUploadFileTable');

GO

# Enableling File stream in database

EXEC sp\_configure filestream\_access\_level, 2

# RECONFIGURE

# Link to get more into deep-:

# <https://www.sqlshack.com/sql-server-filetable-the-next-generation-of-sql-filestream/>