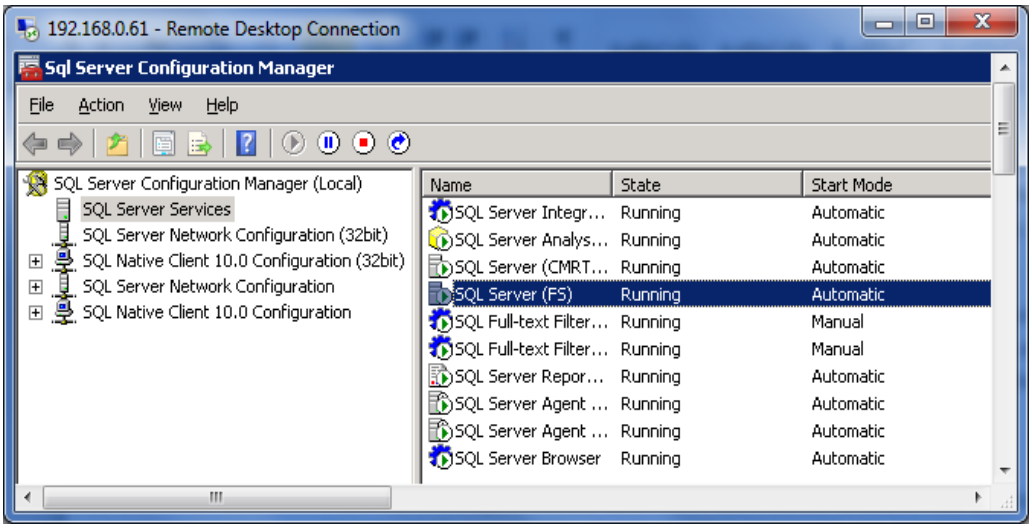
Adding FILESTREAM to an existing table in a database in SQL 2008 R2

FILESTREAM is a great feature of SQL Server. It alleviates some of the problems of storing large unstructured data files such as documents, PDFs, and images, (BLOBs) in a structured database. It is very easy to enable FILESTREAM on a database/table when you are creating them. Most of us, however, get to inherit these things and are then told to "fix it."

When initially presented with the problem, I searched the internet for a solution, and could not find something that is complete, so after a lot of tinkering with different things found online, and some help from a colleague, here are the steps to follow to enable FILESTREAM on an existing database and then convert an existing BLOB table into a FILESTREAM table.

There are three steps needed enable FILESTREAM on the database. You need to turn on the FILESTREAM feature on the instance, turn on the feature on the database, and then let the database know where to store the data on the file system.

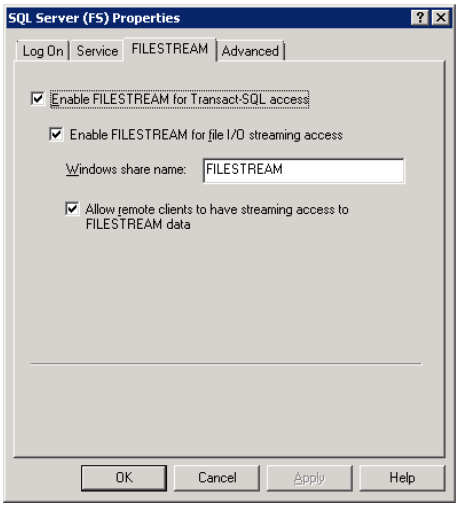
I. Enabling FILESTREAM on an existing instance
   a) Create a file share for FILESTREAM. I usually call mine very aptly "FILESTREAM"
   b) Turning on FILESTREAM
      1) On the Start menu, go to All Programs → Microsoft SQL Server *2008/2008R2/2012*→Configuration Tools→ click SQL Server Configuration Manager.
      2) In the list of services, right-click SQL Server Services→ click Open.
      3) Right click on the SQL Server Service (*instance name)* and click Properties



      4) In the SQL Server Properties dialog box, click the FILESTREAM tab and go through the options
         a. Select the Enable FILESTREAM for Transact-SQL access check box – This turns FILESTREAM on

b. If you want to read and write FILESTREAM data from Windows, select “Enable FILESTREAM for file I/O streaming access.” Enter the name of the Windows shared folder in the Windows Share Name box.

c. If remote clients must access the FILESTREAM data that is stored on this share, select the “Allow remote clients to have streaming access to FILESTREAM data.”



d. Click Apply.

The feature is enabled, but we now need to set the appropriate access levels. To do that, open the query editor in SSMS and type in the following and click Execute when done:

```
EXEC sp_configure filestream_access_level, 2
RECONFIGURE
```

Finally, restart the SQL Server service to enable the changes.

II. Enabling FILESTREAM on an existing database

a) Add the File Group for FILESTREAM to the database and tell the database that this file group contains FILESTREAM objects

```
ALTER DATABASE DBNAME
ADD FILEGROUP FILESTREAMGroupName
 CONTAINS FILESTREAM
GO
```

b) Tell the database which directory to use. This directory cannot exist on the file system, otherwise you will get an error

```
ALTER DATABASE DBNAME
ADD FILE (NAME='FILESTREAMGroupName', FILENAME='Drive:\FolderName')
TO FILEGROUP FILESTREAMGroupName
GO
```

Now that FILESTREAM is enabled on the Instance and the database, we will need to set it up on the table, move existing BLOB data into the appropriate column, and adjust the column names properly. This is a 4 step process: create an identity column, turn on FILESTREAM, create a

FILESTREAM column, move the data over. There may be 2 additional steps of dropping the old column and renaming the new column, depending on the application requirements. So here we go.

III. Enabling FILESTREAM on an existing table

a) Create a ROWGUID column

```
USE DBNAME
ALTER Table tablename
Add columnname uniqueidentifier not null ROWGUIDCOL unique default newid()
GO
```

b) Turn on FILESTREAM

```
USE DBNAME
ALTER Table tablename
SET (filestream_on=FILESTREAMGroupName)
GO
```

c) Add FILESTREAM column to the table

```
USE DBNAME
ALTER Table tablename
Add columnname varbinary(max) FILESTREAM null
GO
```

d) Move data into the new column

```
UPDATE tablename
SET new_columnname=old_columnname
GO
```

e) Drop the old column

```
ALTER Table tablename
DROP column old_columnname
GO
```

f) Rename the new FILESTREAM column to the old column name

```
Use DBNAME
EXEC sp_rename 'tablename.newcolumnname', 'oldcolmnname', 'COLUMN'
GO
```

That's it. We are done. Validate by uploading the unstructured data into the database.