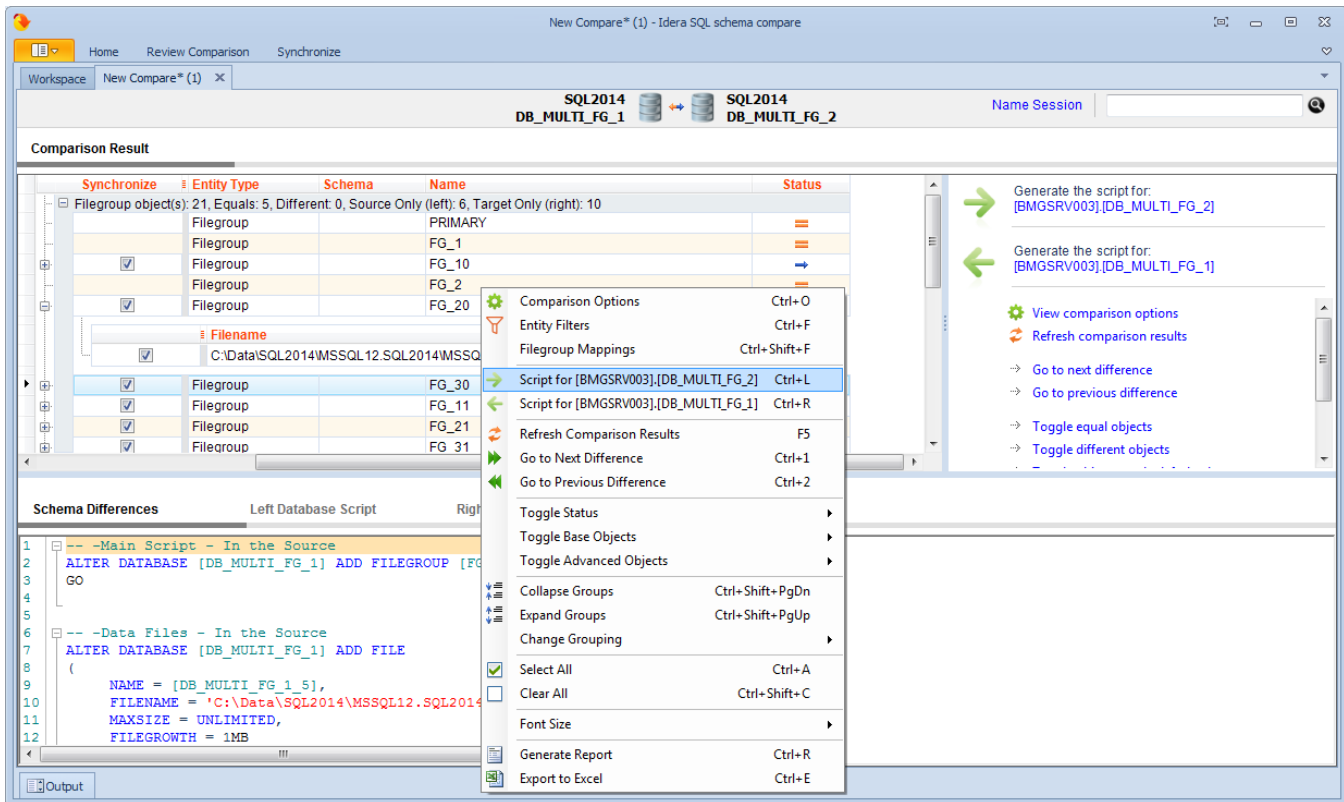


SQL Schema Compare Filegroup Mappings

Filegroup mappings allow you to compare and synchronize filegroups. Mappings are accessed from the comparison grid context menu.



About filegroup mappings

Filegroup mappings provide a way to map, compare and script filegroups for databases that have different filegroup structures. Consider, for example, what happens when you compare the following databases:

Database: DB_1

Contains the filegroup: FG_1

Contains the Employees table created on FG_1 filegroup

Database: DB_2

Contains the filegroup: FG_2

Contains the Employees table created on FG_2 filegroup

The default comparison finds a schema difference in the Employees table triggered by the filegroup clause. The script generated for the DB_2 database attempts to create the table on FG_1 filegroup, while the script generated for DB_1 attempts to create the table on FG_2 filegroup.

While this could work in most of the cases, there are scenarios in which you might want to keep the filegroup unchanged when transferring objects between the DB_1 and the DB_2 databases. You can achieve this by creating a mapping for the filegroups FG_1 and FG_2. When a mapping exist, the comparison and scripting of the Employees table changes as follows:

- FG_1 and FG_2 are considered equivalent, which means that no schema difference is triggered by the filegroup clause of the Employees table.
- If other schema differences require the Employees table to be synchronized, it will be created on the "mapped" filegroup, not the original one.

Working with filegroup mappings

IDERA SQL Schema Compare provides a few mapping rules that you can choose:

- Mapping filegroups by name: it is the default mapping option.
- Ignoring filegroups when comparing schema: indicates that the filegroup clause is ignored and does not trigger a schema difference between objects.
- Custom mappings: allows you to create custom mappings between database filegroups.

Filegroup Mapping Rules

Specify the filegroup mapping rules

Filegroups **Filestream Filegroups**

☐ Map filegroups by name (default)
 ☒ Script the filegroup clause when synchronizing schema

☐ Ignore the filegroup clause when comparing schema

☒ Map the filegroups as follows:

Available filegroups

[BMGSRV003].[DB_MULTI_FG_1]
PRIMARY
FG_30

Map selected filegroups

[BMGSRV003].[DB_MULTI_FG_2]
PRIMARY
FG_31
FG_41
FG_51

Mapped filegroups (click on the link button to delete a mapping)

Left Filegroup		Right Filegroup
FG_1		FG_1
FG_10		FG_11
FG_2		FG_2
FG_20		FG_21

OK Cancel

You can create a filegroup mapping as follows:

- Select a filegroup from the list of filegroups on the left database.
- Select a filegroup from the list of filegroups on the right database.
- Click **Map Selected Filegroup**. A mapping row will appear in the **Mapped Filegroup** grid.

To delete a mapping, select the mapping row in the **Mapped Filegroup** grid, and then click **Link**. The members of the mapped filegroup will appear on the **Available Filegroup** grids.

i If one of the databases contains filestream filegroups, a second tab appears in the mapping form that allows you to map filestream filegroups in the same way you map the regular filegroups.

Scripting filegroups

The mapping form provides an additional option that affects the scripting of the filegroup clause for various database objects. When unchecked, database objects that support the filegroup clause are created in the DEFAULT filegroup or, in some cases depending on the object, the filegroup option is ignored.

Objects affected by filegroup mappings

Database objects affected by the filegroup mapping option include:

- **Tables.** Affects the filegroup, text filegroup, filestream filegroup and the filegroup option of the change data capture
- **Primary Keys.** Affects the filegroup and the filestream filegroup clause
- **Unique Constraints.** Affects the filegroup and the filestream filegroup clause
- **Relational Indexes.** Affects the filegroup and the filestream filegroup clause
- **Spatial Indexes.** Affects the filegroup clause
- **Full-Text Indexes.** Affects the filegroup clause
- **Service Broker Queues.** Affects the filegroup clause
- **Partition Schemes**

The following object are NOT affected by the filegroup scripting options:

- **Database Filegroups**
- **Partition Schemes**

Best practices and restrictions

Mapping the filegroups, when not done properly, could produce unexpected results. These simple rules and restrictions can help you achieve the intended results:

- Filegroup mappings are one-to-one.
- Even though it is not required, we recommend that you map all filegroups. The filegroups that are not mapped explicitly, will be mapped by name (the default rule).
- Filestream filegroups are mapped separately from the regular filegroups. You will notice that if one of the databases that you are comparing contains filestream filegroups, the mapping form includes an additional tab for filestream filegroups. The mapping rules are the same as the ones for regular filegroups.
- The filegroup scripting option, when unchecked, effects database objects differently. Some objects, such as tables, indexes, are scripted with the DEFAULT filegroup clause. Other objects, such as change data capture, ignore the filegroup option entirely. The database filegroups and partition schemes are not effected by the scripting option.
- SQL Server 2005 supports the filegroup for a full-text catalog. According to SQL Server documentation, beginning with SQL 2008, the filegroup clause has no effect on the full-text catalogs, therefore the full-text catalogs do not participate in the filegroup mappings.

Total compare from data to schema with IDERA [SQL Comparison Toolset](#). [Learn more](#) > >

IDERA Website	Products	Purchase	Support	Community	About Us	Resources	Legal
-------------------------------	--------------------------	--------------------------	-------------------------	---------------------------	--------------------------	---------------------------	-----------------------

