

Simple examples using the command line

This topic provides some simple examples of how to use the command line interface.

For more detailed information on using the command line, see:

- [Example - selecting single tables for comparison](#)
- [Example - selecting tables with unrelated names](#)
- [Switches used in the command line](#)

Comparing and deploying database schema

To compare the database structure of *WidgetStaging* with *WidgetProduction*:

```
sqlcompare /Database1:WidgetStaging /Database2:WidgetProduction
```

To compare the database structure of *WidgetStaging* with *WidgetProduction*, and deploy the databases by updating *WidgetProduction*:

```
sqlcompare /Database1:WidgetStaging /Database2:WidgetProduction /Synchronize
```

Using an existing SQL Compare project from the command line

This is useful, for example, as you can't choose or create a [custom filter](#) from the command line, and specifying complex object selections using a regular expression can be unwieldy.

To use a project you have saved as "*widgets.scp*" from the command line:

```
sqlcompare /project:"C:\SQLCompare\Projects\Widgets.scp"
```



- When you use a project, all objects that were selected for comparison when you saved the project are automatically included.
- When you use the command line, your project option selections are ignored and the defaults are used. Use */options* to specify any additional options you want to use with a command line project. For more information, see: [Options used in the command line](#).
- If you want to include or exclude objects from an existing project, you must modify your selection using the graphical user interface. You can't use the */include* and */exclude* switches with */project*.

Scheduling or automating a comparison or deployment

You can use the Microsoft Windows Scheduled Task wizard to schedule a comparison by creating a script to run the comparison.

The following example compares the structure of *WidgetStaging* and *WidgetProduction*, and outputs the results as the file *log_file.txt*

First create the script:

```
cd "C:\Program Files\Red Gate\SQL Compare 8" sqlcompare /db1:WidgetStaging /db2:WidgetProduction >> C:\log_file.txt
```

Next save the script as a *.bat* file. You specify the *.bat* file as the program to run from within the Scheduled Task wizard by browsing to it.

To schedule a deployment of the two databases, updating the database *WidgetProduction*, you would create the script:

```
cd "C:\Program Files\Red Gate\SQL Compare 8" SQLCompare /db1:WidgetStaging /db2:WidgetProduction /Synchronize
```

In these examples MS-DOS batch scripting is used, a basic scripting language that is supported on all versions of Windows. If preferred, you could use VBScript, JScript, PHP, Perl, Python or any other scripting language of your choice.

Creating a scripts folder

You can create a folder of object creation scripts representing the schema of a data source. To export *WidgetProduction* to a scripts folder:

```
sqlcompare /Database1:WidgetProduction /Makescripts:"C:\WidgetProductionScripts"
```

If the folder does not already exist, it is created. All the subfolders containing different object types in the schema are automatically created beneath the specified folder. If the folder does exist, it must be empty or the export will fail.

For more information on scripts folders, see [Working with scripts folders](#).

Creating a snapshot

You can create a SQL Compare snapshot representing the schema of a data source. To export *WidgetProduction* to a snapshot:

```
sqlcompare /Database1:WidgetProduction /Makesnapshot:"C:\Snapshots\WP_1.snp"
```

For more information on SQL Compare schema snapshots, see: [Working with snapshots](#).

Using a scripts folder or snapshot as a data source

To compare the *WidgetProduction* scripts folder with the *WidgetStaging* database:

```
sqlcompare /Scripts1:"C:\WidgetProductionScripts" /Database2:WidgetStaging
```

To compare the *WidgetStaging* database with the *WidgetProduction* scripts folder and deploy the scripts

```
sqlcompare /Database1:WidgetStaging /Scripts2:"C:\WidgetProductionScripts" /Synchronize /Force
```

The */force* switch specifies that any read-only files in the scripts folder that need to be edited during deployment will be made writable. If you do not include the */force* switch and read-only files need to be modified, the deployment will fail and an error message will be displayed.

To compare two snapshots of *WidgetStaging*:

```
sqlcompare /Snapshot1:"C:\Snapshots\WidgetProd_1.snp" /Snapshot2:"C:\Snapshots\WidgetProd_2.snp"
```

To output the deployment SQL script, for example for auditing purposes, and overwrite the file if it already exists:

```
sqlcompare /database1:WidgetStaging /database2:WidgetProduction /scriptfile:"C:\SQLScripts\Widgets.sql" /force
```

Using a backup as a data source

To compare a backup of *WidgetDev* with *WidgetLive*:

```
sqlcompare /Backup1:"D:\MSSQL\BACKUP\WidgetDev_20080807_143235.sqb" /db2:WidgetLive
```

If you are comparing a backup set that contains multiple files, use the */backupset1* switch to specify the files which make up the first backup set, and use the */backupset2* switches to specify the files which make up the second:

```
sqlcompare /Backup1:"D:\MSSQL\BACKUP\WidgetDev.bak" /Backupset1:"2008-09-23 Full Backup" /db2:WidgetLive
```

If the backup set switches are not specified, SQL Compare uses the latest backup set.

To specify more than one backup file, separate the file names using semicolons.

```
sqlcompare /Backup1:"D:\WidgetDev_Full.bak";"D:\WidgetDev_Diff.bak" /db2:WidgetDevelopment
```

For encrypted backups that have been created using SQL Backup, use the */BackupPasswords1* and */BackupPasswords2* switches to specify the passwords; when there is more than one backup password, separate the passwords using semicolons.

```
sqlcompare /Backup1:"D:\MSSQL\BACKUP\WidgetDev.sqb" /BackupPasswords1:Pa$$w0rd /db2:WidgetLive
```



If you are comparing a differential backup, you must also specify the associated full backup.

For more information, see: [Working with backups](#).