Switches used in the command line

This page provides a list of the switches you can use with the SQL Data Compare command line.

Note that:

- the first data source (/db1, /b1, and so on) is the *source*
- the second data source (/db2, /b2, and so on) is the *target*
- there are changes to the command line syntax of SQL Data Compare in version 8 and later
  The names, aliases, and behavior of some switches and options is different to that of earlier versions.
  For more information, see Changes to the command line in SQL Data Compare 8.

/activateSerial:<serial number>

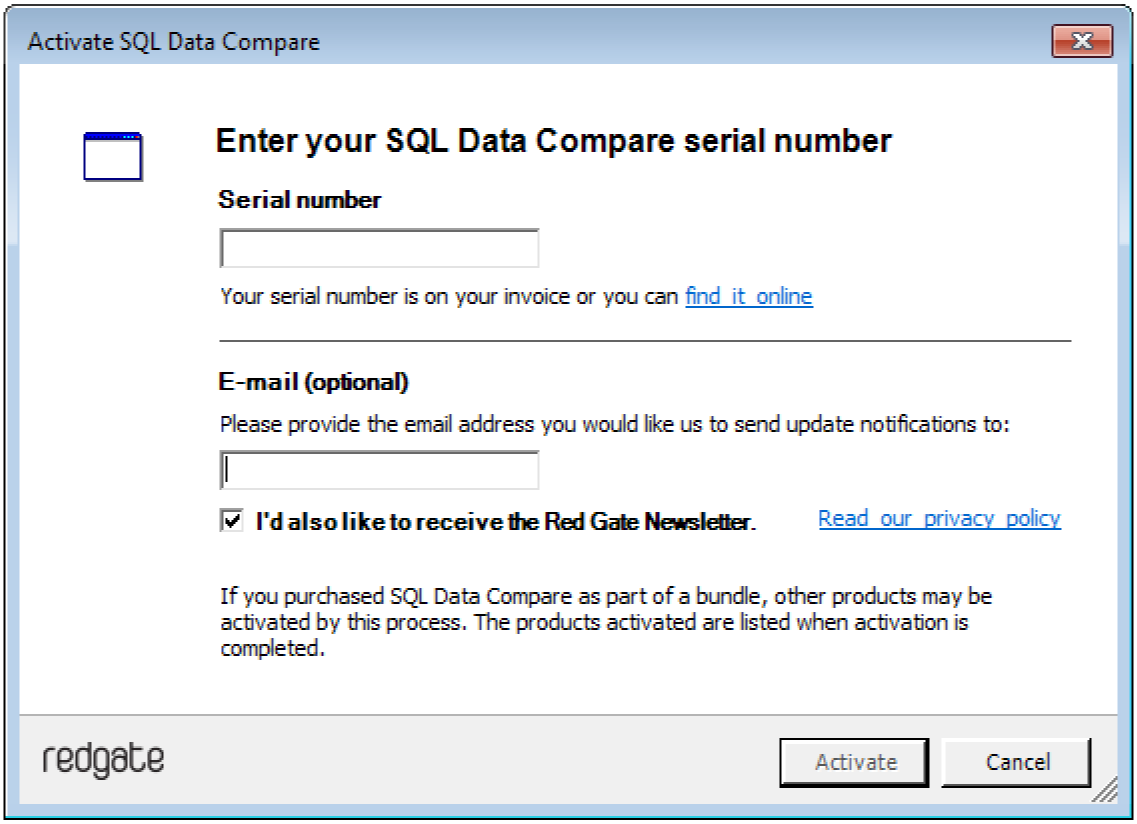
- This switch is case sensitive.
- An internet connection is required to activate SQL Data Compare from the command line.
  For information about how to activate manually without an internet connection, see Activating.

Attempts to activate SQL Data Compare.

You can specify a SQL Data Compare Professional serial number, or a serial number for bundle such as the SQL Developer Bundle:

```
sqldatacompare /activateSerial:123-456-789012-ABCD
```

If you run the switch without specifying a serial number, the **Activate SQL Data Compare** dialog box is displayed:



Activate SQL Data Compare

Enter your SQL Data Compare serial number

Serial number

Your serial number is on your invoice or you can find it online

E-mail (optional)

Please provide the email address you would like us to send update notifications to:

☑ **I'd also like to receive the Red Gate Newsletter.** Read our privacy policy

If you purchased SQL Data Compare as part of a bundle, other products may be activated by this process. The products activated are listed when activation is completed.

If you're using the SQL Data Compare command line on a server, you need a DLM Automation Suite license (previously called SQL Automation Pack).

For information about how to activate the DLM Automation Suite, see Activating.

/AllowIdenticalDatabases

This switch is deprecated. Instead use */Include:Identical*

/Include:Identical suppresses the exit code if the two data sources are identical. If */Include:Identical* is not set, SQL Data Compare returns the error code 63.

/Argfile:<argfile>

Runs a file containing an XML argument specification:

```
sqldatacompare /Argfile:XMLFileName.xml
```

For more information see Using XML to specify command line arguments.

/Backup1:<filename1>;<filename2>;...;<filenameN>

Alias: */b1*

Specifies the backup to be used as the first data source (the *source*). You must add all of the files making up the backup set you want to compare:

```
sqldatacompare /Backup1:D:\BACKUPS\WidgetStaging.bak /db2:WidgetStaging
```

To specify more than one backup file, the file names are separated using semicolons:

```
sqldatacompare /Backup1:D:\BACKUPS\WidgetDev_Full.bak;
     D:\BACKUPS\WidgetDev_Diff.bak /db2:WidgetDev
```

/Backup2:<filename1>;<filename2>;...;<filenameN>

Alias: */b2*

Specifies the backup to be used as the second data source (the *target*). You must add all of the files making up the backup set you want to compare:

```
sqldatacompare /db1:WidgetStaging /Backup2:D:\BACKUPS\WidgetStaging.bak
```

/BackupPasswords1:<Password1>,<Password2>,...,<Password1N>

Alias: */bpsw1*

Specifies the password for the first backup:

```
sqldatacompare /Backup1:D:\BACKUPS\WidgetStaging.bak
     /BackupPasswords1:P@ssw0rd /db2:WidgetProduction
```

/BackupPasswords2:<Password1>,<Password2>,...,<Password1N>

Alias: */bpsw2*

Specifies the password for the second backup:

```
sqldatacompare /db1:WidgetStaging
     /Backup2:D:\BACKUPS\WidgetProduction.bak /BackupPassword2:P@ssw0rd
```

/BackupSet1:<backupset>

Alias: */bks1*

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:

```
sqldatacompare /Backup1:"D:\MSSQL\BACKUP\WidgetDev.bak"
     /BackupSet1:"2008-09-23 Full Backup" /db2:WidgetLive
```

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

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

```
sqldatacompare /Backup1:D:\BACKUPS\WidgetDev_Full.bak;
     "D:\BACKUPS\WidgetDev_Diff.bak" /db2:WidgetDevlopment
```

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 password, the passwords are separated using semicolons.

```
sqldatacompare /Backup1:D:\BACKUPS\WidgetDev.sqb /BackupPassword1:Pa$$w0rd
     /db2:WidgetLive
```

/BackupSet2:<backupset>

Alias: */bks2*

Specifies which backup set to use for the second backup:

```
sqldatacompare /db1:WidgetProduction /BackupSet2:"2008-09-23 Full Backup"
```

/CaseSensitive

This switch is deprecated. Instead use */Options:CaseSensitiveObjectDefinition*

/Columns

This switch is deprecated. Instead use */IncludeColumns* and */ExcludeColumns*

/ComparisonKeys:<table or view name as regular expression>:<index name>

Alias: */ck*

Specifies a unique index to be used to identify rows for comparison.

The name of the table or view is specified using a regular expression - you do not have to specify fully-qualified names. It is recommended that you use a regular expression which matches only one table or view:

```
sqldatacompare /Database1:WidgetStaging /Database2:WidgetProduction
     /IncludeColumns:[WidgetPrices]:Price
     /ComparisonKeys:[WidgetPrices]:PK_WidgetPrices
```

⚠

- */ComparisonKeys* must be used with the */IncludeColumns* switch
- with */ComparisonKeys* you can only specify an index as the comparison key, no other columns can be specified

To specify a comparison key that is is not an index, use the GUI to set up and save a project with the settings you require. You can then use that project from the command line with the */Project* switch.

For more information on using the GUI to set comparison keys, see *Selecting the comparison key*, under Selecting tables and views.

/Database1:<database1>

Alias: */db1*

Specifies a database to use as the source:

```
sqldatacompare /Database1:WidgetStaging /Database2:WidgetProduction
```

/Database2:<database2>

Alias: */db2*

Specifies a database to use as the source:

```
sqldatacompare /Database1:WidgetStaging /Database2:WidgetProduction
```

/Exclude:<object type>:<regular expression>

Arguments:

• *Additional*	only those objects that are not present in the source (eg */db1*)
• *Missing*	only those objects that are not present in the target (eg */db2*)
• *Different*	only those objects that are present in both data sources, but are different.
• *Identical*	identical objects in the command line output and any generated reports.
• *User specified*	objects you specify with a regular expression (eg */Include:Table:WidgetPurchases*)

To specify the list of objects to exclude, use the */exclude* switch:

```
sqldatacompare /db1:Customers1 /db2:Customers2 /Exclude:table
```

`/Exclude:table` specifies that you do not want to compare tables; you only want to compare other objects such as views, stored procedures, and so on.

To specify more than one object or object type type for exclusion use multiple */Exclude* switches. For example, to exclude only tables and views:

```
sqldatacompare /db1:WidgetStaging /db2:WidgetProduction
     /Exclude:table:WidgetReferences /Exclude:view
```

Note that you cannot use */Exclude* with the */Include* and */Project* switches.

For a more detailed example of how to use the */include* and */exclude* switches, see Example: selecting tables with unrelated names.

/ExcludeColumns:<table or view name>:<regular expression>

Alias: */ec*

Specifies which columns within a table or view are excluded from deployment.

Table or view names are matched using a regular expression so that you do not have to specify fully qualified names. However, it does not usually make sense to supply a regular expression that matches more than one table or view. Multiple column names should be specified as a comma-delimited list.

```
sqldatacompare /db1:WidgetStaging /db2:WidgetProduction
     /ExcludeColumns:WidgetReferences:WidgetName
```

Note that */ExcludeColumns* takes precedence over */IncludeColumns*

/Export:<directory>

Alias: */e*

Exports the comparison results to the specified directory as a *.csv* file.

```
sqldatacompare /db1:WidgetStaging /db2:WidgetProduction
     /Export:C:\WidgetResults
```

Note that the file created is always called *Summary.csv*

You specify the directory in which the file is created.

/ExportIdenticalTables

This switch is deprecated. Instead use */Include:Identical*

/Force

Alias: */f*

This forces the overwriting of any output files that already exist. If this switch is not used and a file of the same name already exists, the program will exit with the exit code indicating an IO error.

/IgnoreParserErrors

If SQL Data Compare encounters any high level errors when parsing a scripts folder, it will exit with an error code of 62.

Use */ignoreParserErrors* to force SQL Data Compare to continue without exiting.

/Include:<object type>:<regular expression>

Arguments:

• *Additional*	only those objects that are not present in the source (eg */db1*)
• *Missing*	only those objects that are not present in the target (eg */db2*)
• *Different*	only those objects that are present in both data sources, but are different.
• *Identical*	identical objects in the command line output and any generated reports.
• *User specified*	objects you specify with a regular expression (eg */Include:Table:WidgetPurchases*)

This switch is used to specify the list of objects to include. You can use an */Include* switch for each object that you want to compare. However, this can be unwieldy if there is a long list. Instead, you can use the pipe character (|) to separate the table names:

```
sqldatacompare /db1:Customers1 /db2:Customers2 /Include:table
     /Include:table:\[Product\]^|Customer^|Order^|Invoice
```

/Include:Identical suppresses the exit code if the two data sources are identical. If */Include:Identical* is not set, SQL Data Compare returns the error code 63.

For a more detailed example of how to use the */include* switch, see: Example: selecting tables with unrelated names

/IncludeAdditional

This switch is deprecated. Instead use */Include:Additional*

/IncludeColumns

Alias: */ic*

Specifies which columns within a table or view are included in the deployment.

Table or view names are matched using a regular expression so that you do not have to specify fully qualified names. However, it does not usually make sense to supply a regular expression that matches more than one table or view. Multiple column names should be specified as a comma-delimited list.

```
sqldatacompare /db1:WidgetStaging /db2:WidgetProduction
     /IncludeColumns:WidgetReferences:WidgetName
```

Note that */ExcludeColumns* takes precedence over */IncludeColumns*

/MakeScripts:<folder>

Alias: */mkscr*

Creates a scripts folder from the first (source) data source.

```
sqldatacompare /db1:WidgetStaging
     /MakeScripts:"C:\Scripts Folders\Widget staging scripts"
```

If the folder already exists an error will occur. To merge scripts into an existing scripts folder, compare them with that folder and use the */synchronize* switch:

```
sqldatacompare /scr1:"C:\Scripts Folders\Widget dev scripts"
     /scr2:"C:\Scripts Folders\Widget staging scripts" /Synchronize
```

/options:<option1>,<option2>,<option3>

Alias: */o*

Applies the project configuration options used during comparison or deployment:

```
sqldatacompare /db1:WidgetStaging /db2:WidgetProduction
     /Options:Default,IgnoreWhiteSpace
```

For a detailed list of these options, see Options used in the command line.

/Out:<FileName>

Redirects console output to the specified file:

```
sqldatacompare /db1:WidgetStaging /db2:WidgetProduction /Out:C:\output file
```

/OutputProject:<FileName>

Alias: */outpr*

Writes the settings used for the comparison to the specified SQL Data Compare project file:

```
sqldatacompare /db1:WidgetStaging /db2:WidgetProduction
     /Options:Default,IgnoreWhiteSpace /OutputProject:"C:\WidgetProject.scp"
```

This also generates a SQL Data Compare project file. These files end with a *.scp* extension. If the file already exists an error will occur, unless you have also used the */force* switch.

/OutputWidth:<columns>

Forces the width of console output.

This can be used to ensure that database object names etc are not truncated, and that SQL script lines are not wrapped or broken. This is particularly useful when redirecting output to a file as it allows you to overcome the limitations of the default console width of 80 characters.

/Password1:<password1>

Alias: */p1*

The password for the first database (source).

You must also provide a username. If you do not specify a username and password combination, integrated security is used:

```
sqldatacompare /db1:WidgetStaging /UserName1:User1 /Password1:P@ssw0rd
     /db2:WidgetProduction /UserName2:User2 /Password2:Pa$$w0rd
```

Note that this switch is only used if the source is a database. If the source is a backup, use */BackupPasswords1*

/Password2:<password2>

Alias: */p2*

The password for the second database.

/Project

Alias: */pr*

Uses a SQL Data Compare project (*.scp*) file for the comparison.

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

```
sqldatacompare /Project:"C:\SQLCompare\Projects\Widgets.scp"
```

Note that:

- 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 cannot use the */Include* and */Exclude* switches with */Project*.

/SyncScriptEncoding

Alias: */senc*

Arguments:

• *UTF8*	UTF-8 encoding, without preamble
• *UTF8WithPreamble*	UTF-8 encoding, with 3-byte preamble
• *Unicode*	UTF-16 encoding

- *ASCII*	ASCII encoding

Specifies the character encoding used when writing the SQL script file. The default is UTF8.

For example:

```
sqldatacompare /db1:WidgetStaging /MakeScripts: D:\Scripts Folder
     /SyncScriptEncoding:ASCII
```

/ScriptFile:<scriptfile>

Alias: */sf*

Generates a SQL script to migrate the changes which can be executed at a later time. If the file already exists an error will occur, unless you use the */Force* switch:

```
sqldatacompare /db1:WidgetStaging /db2:WidgetProduction
     /ScriptFile: "C:\Scripts Folder\WidgetSyncScript.sql"
```

/Scriptfile can be used when the target (*/db2*, */scr2*, */sn2*) is a database, a snapshot, or a scripts folder.

If the target is a snapshot or a scripts folder, the generated script modifies a database with the schema represented by that snapshot or scripts folder.

/Scripts1:<folder>

Alias: */scr1*

Specifies the script folder to use as the first data source:

```
sqldatacompare /Scripts1:"C:\Scripts Folder\WidgetStagingScript"
     /db2:WidgetProduction
```

/Scripts2:<folder>

Alias: */scr2*

Specifies the script folder to use as the second data source.

/Server1:<server1>

Alias: */s1*

Specifies the server on which the first (*/db1:*) data source is located. If an explicit path is not specified, it defaults to *Local.*

```
sqldatacompare /Server1:Widget_Server\SQL2008 /db1:WidgetStaging
     /db2:WidgetProduction
```

/Server2:<server2>

Alias: */s2*

This specifies the server on which the second (*/db2:*) data source is located. If an explicit path is not specified, it defaults to *Local.*

/Synchronize

Alias: */sync*

Deploys the data after comparison.

The target (for example, */db2*) is modified; the source (for example, */db1*) is not modified:

```
sqldatacompare /db1:WidgetStaging /db2:WidgetProduction
     /Synchronize
```

/UserName1:<username1>

Alias: */u1*

The username for the first database.

If no username is specified, integrated security is used.

```
sqldatacompare /db1:WidgetStaging /UserName1:User1 /Password1:P@ssw0rd
     /db2:WidgetProduction /UserName2:User2 /Password2:Pa$$w0rd
```

/UserName2:<username2>

Alias: */u2*

The username for the second database.

If no username is specified, integrated security is used.