

Switches used in the command line

This is a list of switches you can use with the SQL Compare command line.



- The first data source (/db1, /b1, etc) is the *source*.
- The second data source (/db2, /b2, etc) is the *target*
- The command line syntax of previous versions of SQL Compare is considered deprecated, but is still supported. For example, in SQL Compare 7, the alias for */BackupSet1* was */bs1*. In SQL Compare 8 or later, the alias is now */bks1*. You can continue to use */bs1* in SQL Compare 8 or later, but a message is displayed informing you of the new alias. Deprecated command line syntax will be removed in a future release.

/AbortOnWarnings

Alias: /aow

Specifies that SQL Compare won't run a deployment if there are any serious deployment warnings. If you don't specify this switch, SQL Compare will ignore warnings and run the deployment.

Arguments

None	Don't abort on warnings
Medium	Abort on medium or high warnings
High	Abort on high warnings

The default is None. If you use this switch and there are deployment warnings, exit code 61 is displayed.

For more information on warnings in SQL Compare, see [Deployment Warnings](#).

/activateSerial:<serial number>



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

Attempts to activate SQL Compare.

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

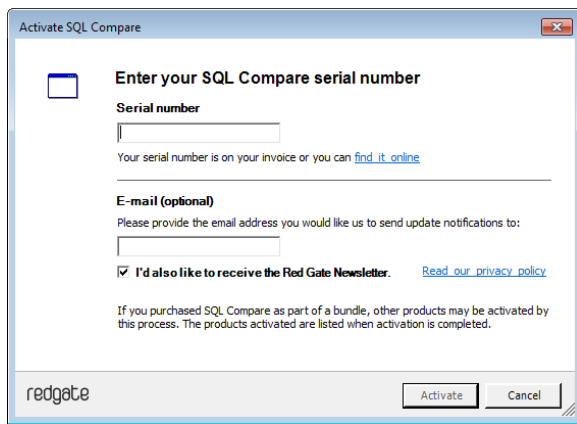
Example

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

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

Contents

- /AbortOnWarnings
- /activateSerial:<serial number>
- /Argfile:<file path>
- /Assertidentical
- /Backup1:<file path1>;<file path2>;...;<file pathN>
- /Backup2:<file path1>;<file path2>;...;<file pathN>
- /BackupPasswords1:<password1>,<password2>,...,<passwordN>
- /BackupPasswords2:<password1>,<password2>,...,<password1N>
- /BackupSet1:<backup set>
- /BackupSet2:<backup set>
- /Database1:<database name>
- /Database2:<database name>
- /deactivateSerial
- /Exclude:<object type>:<regular expression>
- /Filter
- /Force
- /Help
- /HTML
- /IgnoreParserErrors
- /IgnoreSourceCaseSensitivity
- /Include:<object type>:<regular expression>
- /LogLevel:<level>
- /MakeScripts:<folder>
- /MakeSnapshot:<file name>
- /MigrationsFolder:<folder>
- /Options:<option1>,<option2>,<option3>
- /Out:<file path>
- /OutputProject:<file path>
- /OutputWidth:<columns>
- /Password1:<password>
- /Password2:<password>
- /Project:<file path>
- /Quiet
- /Report:<file path>
- /reportAllObjectsWithDifferences
- /ReportType:<report type>
- /Revision1:<revision>
- /Revision2:<revision>
- /ScriptFile:<file path>
- /Scripts1:<folder>
- /Scripts2:<folder>
- /ScriptsFolderXML:<file path>
- /Server1:<server name>
- /Server2:<server name>
- /ShowWarnings
- /Snapshot1:<file name>
- /Snapshot2:<file path>
- /Sourcecontrol1
- /Sourcecontrol2
- /Synchronize
- /SyncScriptEncoding:<script encoding>
- /Tempinstance:<tempinstance>



If you're using the SQL 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](#).

/Argfile:<file path>

Runs a file containing an XML argument specification.

Example

```
sqlcompare /Argfile:XMLFileName.xml
```

For more information, see [Using XML to specify command line arguments](#).

/Assertidentical

When */assertidentical* is specified, SQL Compare will return an exit code of 0 if the objects being compared are identical. If they aren't identical, it will return exit code 79.

/Backup1:<file path1>;<file path2>;...;<file pathN>

Alias: */b1*

Specifies the backup to be used as the source. You must add all of the files making up the backup set you want to compare.

Example

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

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

Example

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

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

/Backup2:<file path1>;<file path2>;...;<file pathN>

Alias: */b2*

Specifies the backup to be used as the target. You must add all of the files making up the backup set you want to compare.

- */TransactionIsolationLevel:*
<transaction isolation level>
- */UserName1:*<username>
- */UserName2:*<username>
- */Verbose*
- */VersionUserName1:*
<username>
- */VersionUserName2:*
<username>
- */VersionPassword1:*
<password>
- */VersionPassword2:*
<password>

Deprecated options

- */AllowIdenticalDatabases*
- */IncludeIdentical:*
<IncludeIdentical>
- */MigrationsFolderXML:*<file path>

Example

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

/BackupPasswords1:<password1>,<password2>,...,<passwordN>

Alias: */bpsw1*

Specifies the password for the source backup.

Example

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

/BackupPasswords2:<password1>,<password2>,...,<password1N>

Alias: */bpsw2*

Specifies the password for the target backup:

Example

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

/BackupSet1:<backup set>

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 source backup set, and use the */BackupSet2* switches to specify the files which make up the target.

Example

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

If the backup set switches aren't specified, SQL Compare uses the latest backup set.

To specify more than one backup file, the file names are separated using semi-colons.

Example

```
sqlcompare /Backup1:D:\BACKUPS\WidgetDev_Full.bak; "D:\BACKUPS\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 password, the passwords are separated using semi-colons.

Example

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

/BackupSet2:<backup set>

Alias: */bks2*

Specifies which backup set to use for the target backup.

Example

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

/Database1:<database name>

Alias: */db1*

Specifies a database to use as the source.

Example

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

/Database2:<database name>

Alias: */db2*

Specifies a database to use as the target.

/deactivateSerial



This switch is case sensitive.

Attempts to deactivate the application. An internet connection is required to deactivate the product.

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



We recommend using filters instead of */Exclude*. For more information, see [Using filters](#).

Arguments:

<ul style="list-style-type: none">• <i>Additional</i>	only those objects that aren't present in the source (eg <i>/db1</i>)
<ul style="list-style-type: none">• <i>Missing</i>	only those objects that aren't present in the target (eg <i>/db2</i>)
<ul style="list-style-type: none">• <i>Different</i>	only those objects that aren't present in both data sources, but are different.
<ul style="list-style-type: none">• <i>Identical</i>	identical objects in the command line output and any generated reports.
<ul style="list-style-type: none">• <i>User specified</i>	objects you specify with a regular expression (eg <i>/include:Table:WidgetPurchases</i>)

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

Example

```
sqlcompare /db1:Customers1 /db2:Customers2 /exclude:table
```

`/exclude:table` specifies that you don't 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 for exclusion use multiple `/exclude` switches. For example, to exclude only tables and views.

Example

```
sqlcompare /db1:WidgetStaging /db2:WidgetProduction /exclude:table:WidgetReferences /exclude:view
```



You can't use `/exclude` with the `/include` and `/project` switches.

For a more detailed example using the `/include` and `/exclude` switches, see [Selecting tables with unrelated names](#).

/Filter

Alias: `/ftr`

Specifies a custom filter to select objects for deployment.

```
sqlcompare /db1:WidgetStaging /db2:WidgetProduction /sync
```

```
  /filter:MarketingViewsOnly.scpf
```

You can set up a filter to include or exclude objects based on their type, name, and owner (schema) name.

This is useful, for example, if you want to create complex selection rules without using regular expressions.



- filters are set up in the graphical user interface
- filters are saved with the extension `.scpf`
- `/filter` can't be used with `/Include` or `/Exclude`
- if you use `/filter` with `/project`, the filter you specify overrides any filter used in the project

For more information, see [Using filters](#).

/Force

Alias: `/f`

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

/Help

Alias: `/?`

Displays the list of switches in the command line with basic descriptions.

If `/help` is used with any switches except `/verbose`, `/html`, `/out`, `/force` or `/outputwidth` then those extra switches will be ignored; the help message will be printed and the process will end with exit code 0.

/HTML

Outputs the help text as HTML. Must be used with the `/help` switch.

/IgnoreParserErrors

If SQL 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 Compare to continue without exiting.

/IgnoreSourceCaseSensitivity

When you are creating a scripts folder using `/makescripts`, SQL Compare automatically detects the case sensitivity of the data source.

Use `/ignoreSourceCaseSensitivity` to disable automatic detection of case sensitivity.

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



We recommend using filters instead of `/include`. For more information, see [Using filters](#).

Arguments:

<ul style="list-style-type: none">• <i>Additional</i>	only those objects that aren't present in the source (eg <code>/db1</code>)
<ul style="list-style-type: none">• <i>Missing</i>	only those objects that aren't present in the target (eg <code>/db2</code>)
<ul style="list-style-type: none">• <i>Different</i>	only those objects that are present in both data sources, but are different.
<ul style="list-style-type: none">• <i>Identical</i>	identical objects in the command line output and any generated reports.
<ul style="list-style-type: none">• <i>User specified</i>	objects you specify with a regular expression (eg <code>/include:Table:WidgetPurchases</code>)
<ul style="list-style-type: none">• <i>StaticData</i>	static data in a source-controlled database or a scripts folder

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:

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

For more detailed information on using the `/include` switch, see [Selecting tables with unrelated names](#).

`/LogLevel:<level>`

Alias: `/log`

Creates a log file with a specified minimum log level.

Log files collect information about the application while you are using it. These files are useful to us if you have encountered a problem. For more information, see [Logging and log files](#).

Arguments:

<ul style="list-style-type: none">• <i>None</i>	Disables logging
<ul style="list-style-type: none">• <i>Error</i>	Reports serious and fatal errors
<ul style="list-style-type: none">• <i>Warning</i>	Reports warning and error messages
<ul style="list-style-type: none">• <i>Verbose</i>	Reports all messages in the log file

The default is None.

Example

```
sqlcompare /db1:WidgetStaging /makeScripts: D:\Scripts Folder /logLevel:Verbose
```



You must use */logLevel* each time you want a log file to be created.

/MakeScripts:<folder>

Alias: */mkscr*

Creates a scripts folder from the data source.

Example

```
sqlcompare /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.

Example

```
sqlcompare /scr1:"C:\Scripts Folders\Widget dev scripts" /scr2:"C:\Scripts Folders\Widget staging scripts" /synchronize
```

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

/MakeSnapshot:<file name>

Alias: */mksnap*

Creates a snapshot from the data source.

Example

```
sqlcompare /db1:WidgetStaging /makeSnapshot:"C:\Widget Snapshots\StagingSnapshot.snp"
```

If the file already exists an error will occur, unless you have also used the */force* switch.

/MigrationsFolder:<folder>

Alias: */mf*

The path to a directory containing migration scripts you want to use.

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

Alias: */o*

Applies the project configuration options used during comparison or deployment:

Example

```
sqlcompare /db1:WidgetStaging /db2:WidgetProduction /options:Default,IgnoreWhiteSpace
```

For a detailed list of these options see [Options used in the command line](#).

/Out:<file path>

Redirects console output to the specified file.

Example

```
sqlcompare /db1:WidgetStaging /db2:WidgetProduction /out:C:\output file
```

/OutputProject:<file path>

Alias: */outpr*

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

Example

```
sqlcompare /db1:WidgetStaging /db2:WidgetProduction /options:Default,IgnoreWhiteSpace /outputProject:"C:\WidgetProject.scp"
```

This also generates a SQL 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 aren't truncated, and that SQL script lines aren't 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:<password>

Alias: */p1*

The password for the source database.

You must also provide a username. If you don't specify a username and password combination, integrated security is used.

Example

```
sqlcompare /db1:WidgetStaging /userName1:User1 /password1:P@ssw0rd /db2:WidgetProduction /userName2:User2 /password2:Pa$$w0rd
```



This switch is only used if the source is a database. If the source is a backup, use */backupPasswords1*

/Password2:<password>

Alias: */p2*

The password for the target database.

/Project:<file path>

Alias: */pr*

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

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

Example

```
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*.

The */project* switch is useful, for example, as you can't specify a [custom filter](#) in the command line, and specifying complex object selections using a regular expression can be unwieldy.

For more information on using projects, and what a project contains, see [Working with projects](#).

/Quiet

Alias: */q*

Quiet mode: no output.

/Report:<file path>

Alias: */r*

Generates a report and writes it to the specified file.

The type of report is defined by the */reportType* switch. If the file already exists an error will occur, unless you have used the */force* switch:

```
sqlcompare /db1:WidgetStaging /db2:WidgetProduction /report:"C:\reports\WidgetReport.html" /reportType:Simple
```

/reportAllObjectsWithDifferences

Alias: */rad*

Includes all objects with differences in the reports, rather than all selected objects.

/ReportType:<report type>

Alias: */rt*

Arguments:

• <i>XML</i>	Simple XML report
• <i>Simple</i>	Simple HTML report
• <i>Interactive</i>	Interactive HTML report
• <i>Excel</i>	Microsoft Excel spreadsheet

This switch defines the file format of the report produced by the */Report* switch. The default is XML.

Example

```
sqlcompare /db1:WidgetStaging /db2:WidgetProduction /report:"C:\reports\WidgetReport.html" /reportType:Simple
```

For more information, see [Exporting the comparison results](#).

/Revision1:<revision>

Alias: */r1*

Specifies the source control revision of the source database. To specify a revision, the database must be linked to SQL Source Control. To specify the latest version, type: *HEAD*



Specifying a revision other than HEAD is only supported with TFS, SVN and Vault. If you're using another source control system, we recommend checking the revision out to a local folder and using the /Scripts1 switch.

The following example compares revision 3 of WidgetStaging with the latest revision of WidgetProduction:

```
sqlcompare /db1:WidgetStaging /revision1:3 /db2:WidgetProduction /revision2:HEAD
```

/Revision2:<revision>

Alias: */r2*

Specifies the source control revision of the target database. To specify a revision, the database must be linked to SQL Source Control.

/ScriptFile:<file path>

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.

Example

```
sqlcompare /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 scripts folder to use as the source.

Example

```
sqlcompare /scripts1:"C:\Scripts Folder\WidgetStagingScript" /db2:WidgetProduction
```

/Scripts2:<folder>

Alias: */scr2*

Specifies the scripts folder to use as the target.

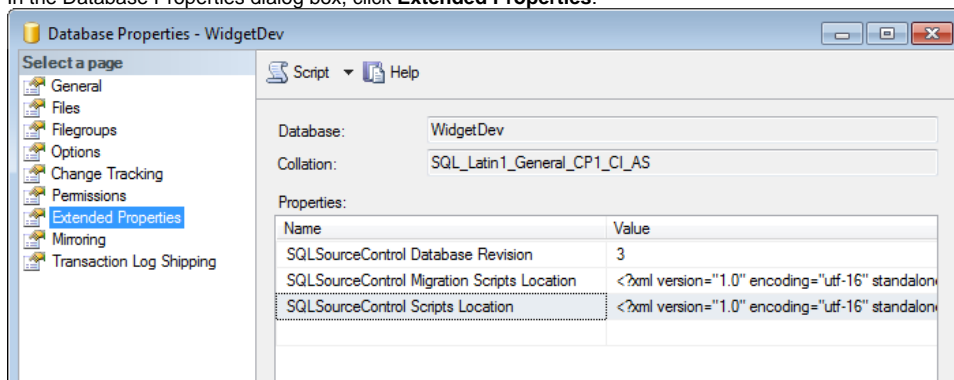
/ScriptsFolderXML:<file path>

Alias: */sfx*

The path to a text file that describes the location of a source control repository.

To create the text file:

1. In the SSMS Object Explorer, right-click a source-controlled database and click **Properties**.
2. In the Database Properties dialog box, click **Extended Properties**:



3. Copy the XML fragment from the *SQLSourceControl Scripts Location* extended property.
4. Create a new text file and paste the XML fragment into it.
5. Save the file.

/Server1:<server name>

Alias: */s1*

Specifies the server on which the source (*/db1:*) database is located. If an explicit path isn't specified, it defaults to *Local*.

Example

```
sqlcompare /server1:Widget_Server\SQL2008 /db1:WidgetStaging /db2:WidgetProduction
```

/Server2:<server name>

Alias: */s2*

Specifies the server on which the target (*/db2:*) database is located. If an explicit path isn't specified, it defaults to *Local*.

/ShowWarnings

Alias: */warn*

Displays any warnings that apply to the deployment.

Example

```
sqlcompare /db1:WidgetStaging /db2:WidgetProduction /showWarnings
```

For more information about warnings in SQL Compare, see [Warnings](#).

/Snapshot1:<file name>

Alias: */sn1*

Specifies the snapshot to use as the source.

Example

```
sqlcompare /snapshot1:"C:\Snapshots\WidgetStagingSnapshot.snp" /db2:WidgetProduction
```

/Snapshot2:<file path>

Alias: */sn2*

Specifies the snapshot to use as the target.

Example

```
sqlcompare /db1:WidgetStaging /snapshot2:"C:\Snapshots\WidgetProductionSnapshot.snp"
```

/Sourcecontrol1

Allows a folder of scripts to be used as the source.

/Sourcecontrol2

Allows a folder of scripts to be used as the target.

/Synchronize

Aliases: */sync* or */synchronise*

Synchronizes (deploys) the databases after comparison.

The target (eg */db2*) is modified; the source (eg */db1*) isn't modified:

Example

```
sqlcompare /db1:WidgetStaging /db2:WidgetProduction /synchronize
```

/SyncScriptEncoding:<script encoding>

Alias: */senc*

Arguments:

<ul style="list-style-type: none">• <i>UTF8</i>	UTF-8 encoding, without preamble
<ul style="list-style-type: none">• <i>UTF8WithPreamble</i>	UTF-8 encoding, with 3-byte preamble
<ul style="list-style-type: none">• <i>Unicode</i>	UTF-16 encoding
<ul style="list-style-type: none">• <i>ASCII</i>	ASCII encoding

Used with */scriptFile*. Specifies the character encoding used when writing the SQL script file. The default is UTF8.

Example

```
sqlcompare /db1:WidgetStaging /db2:WidgetProduction /scriptFile:"C:\Scripts Folder\WidgetSyncScript.sql"  
/syncScriptEncoding:ASCII
```

/Tempinstance:<tempinstance>

Alias: */ti*

Specifies a connection string to a SQL Server instance to be used instead of LocalDB when generating deployment scripts that contain V2 migration scripts.

Example

```
sqlcompare /db1:WidgetStaging /db2:WidgetProduction /Tempinstance:"Data Source=localhost;Integrated Security=SSPI;"
```

For more information about the temporary database, see [Setting the location of the temporary database](#) in the SQL Source Control documentation.

/TransactionIsolationLevel:<transaction isolation level>

Alias: */til*

Specifies the Transaction Isolation Level to set in the SQL script.

/UserName1:<username>

Alias: */u1*

The username for the source database.

If no username is specified, integrated security is used.

Example

```
sqlcompare /db1:WidgetStaging /userName1:User1 /password1:P@ssw0rd /db2:WidgetProduction /userName2:User2 /password2:Pa$$w0rd
```

/UserName2:<username>

Alias: */u2*

The username for the target database.

If no username is specified, integrated security is used.

/Verbose

Alias: */v*

Verbose mode.

/VersionUserName1:<username>

Alias: */vu1*

Specifies the username for the source control server linked to the source database.

Example

```
sqlcompare /db1:WidgetStaging /v1:3 /versionUserName1:User1 /vp1:P@ssw0rd /db2:WidgetProduction /v2:HEAD /versionUserName2:User2 /vp2:Pa$$w0rd
```

If you have a username saved in SQL Source Control, you don't need to specify it in the command line.

/VersionUserName2:<username>

Alias: */vu2*

Specifies the username for the source control server linked to the target database.

/VersionPassword1:<password>

Alias: */vp1*

Specifies the password for the source control server linked to the source database.

Example

```
sqlcompare /db1:WidgetStaging /v1:3 /vu1:User1 /versionpassword1:P@ssw0rd /db2:WidgetProduction /v2:HEAD /vu2:User2 /versionpassword2:Pa$$w0rd
```

If you have a password saved in SQL Source Control, you don't need to specify it in the command line.

/VersionPassword2:<password>

Alias: */vp2*

Specifies the password for the source control server linked to the target database.

Deprecated options

/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* isn't set, and the data sources are identical, SQL Compare returns the error code 63.

/IncludeIdentical:<IncludeIdentical>



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

/MigrationsFolderXML:<file path>



This switch is deprecated. Instead use */migrationsfolder*.

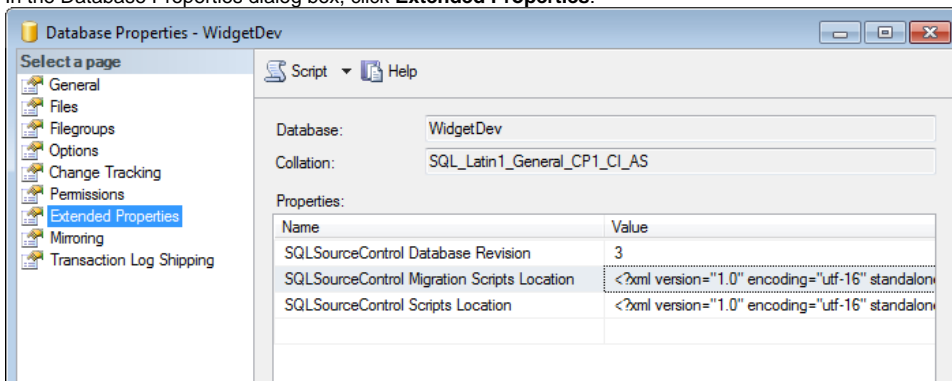
Alias: */mfx*

The path to a text file that describes the location of a migration scripts folder.

To create the text file:

1. In the SSMS Object Explorer, right-click a source-controlled database and click **Properties**.

2. In the Database Properties dialog box, click **Extended Properties**:



3. Copy the XML fragment from the *SQLSourceControl Migration Scripts Location* extended property.
4. Create a new text file and paste the XML fragment into it.
5. Save the file.