

Sync-DatabaseSchema

Sync-DatabaseSchema

Updates a target database schema to match a source database schema.

Syntax

```
Sync-DatabaseSchema -Source <Object> -Target <DatabaseConnection> [-FilterPath <string>] [-IgnoreStaticData] [-IgnoreAdditional] [-SQLCompareOptions <string>] [-SQLDataCompareOptions <string>] [-TransactionIsolationLevel <TransactionIsolationLevel>] [-QueryBatchTimeout <int>] [-AbortOnWarningLevel <WarningSeverity>] [-IgnoreParserErrors] [-SqlCmdVariables <hashtable>] [<CommonParameters>]

Sync-DatabaseSchema -Source <Object> -Target <DatabaseConnection> [-FilterPath <string>] [-IgnoreStaticData] [-IgnoreAdditional] [-SQLCompareOptions <string>] [-SQLDataCompareOptions <string>] [-TransactionIsolationLevel <TransactionIsolationLevel>] [-QueryBatchTimeout <int>] [-Force] [-IgnoreParserErrors] [-SqlCmdVariables <hashtable>] [<CommonParameters>]
```

Description

The Sync-DatabaseSchema cmdlet updates a target database to match a source database schema.

The source schema can be a database connection, a build artifact, or a SQL Source Control project. The target schema must be a database connection.

Parameters

-Source <System.Object>

The source database schema used to update the target database. This can be:

- a Database Connection object created by the New-DatabaseConnection cmdlet
- a database connection string
- a path for a NuGet package or .zip file. This must contain a scripts folder located at db\state
- a Database Build Artifact object produced by the New-DatabaseBuildArtifact cmdlet
- a path for a scripts folder, created by SQL Compare or from your SQL Source Control database repository

Aliases	None
Required?	true
Position?	named
Default Value	None
Accept Pipeline Input	false
Accept Wildcard Characters	false

-Target <RedGate.Versioning.Automation.Compare.SchemaSources.DatabaseConnection>

The target database to be updated. This can be:

- a Database Connection object created by the New-DatabaseConnection cmdlet
- a database connection string

Aliases	None
Required?	true
Position?	named
Default Value	None
Accept Pipeline Input	false
Accept Wildcard Characters	false

-FilterPath <System.String>

The path to a .scpf filter file.

Use this parameter to specify a filter file to be used when performing the sync operation. This will override any Filter.scpf file in the source.

Aliases	None
Required?	false
Position?	named
Default Value	None
Accept Pipeline Input	false
Accept Wildcard Characters	false

-IgnoreStaticData <System.Management.Automation.SwitchParameter>

When SQL Change Automation performs a sync operation and the source is a NuGet package or a SQL Source Control project, by default it will include static data flagged by SQL Source Control.

If you specify this parameter, static data will be ignored.

Aliases	None
Required?	false
Position?	named
Default Value	False
Accept Pipeline Input	false
Accept Wildcard Characters	false

-IgnoreAdditional <System.Management.Automation.SwitchParameter>

When SQL Change Automation performs a sync operation, by default it will drop all additional objects in the target database.

If you specify this parameter, additional objects will be ignored.

Enabling this prevents new and unexpected objects in the target being dropped - note that the checks and tests applied to this database may not have taken those objects into account and may therefore not be valid.

This parameter can't be used simultaneously with filter files.

Aliases	None
Required?	false
Position?	named
Default Value	False
Accept Pipeline Input	false
Accept Wildcard Characters	false

-SQLCompareOptions <System.String>

Specifies the SQL Compare options to use when performing the sync operation. SQL Change Automation applies a default set of options (listed below). To include additional options, specify a comma-delimited list of the option names (eg 'IgnoreComments, ObjectExistenceChecks'). To turn off a default option, precede the option name with a minus sign (eg '-ForceColumnOrder').

This parameter will be ignored if the value specified is \$null or empty.

By default, the following Compare options are used:

- ConsiderNextFilegroupInPartitionSchemes
- DecryptPost2KEncryptedObjects
- DoNotOutputCommentHeader
- ForceColumnOrder
- IgnoreCertificatesAndCryptoKeys

- IgnoreDatabaseAndServerName
- IgnoreTSQLT
- IgnoreUsersPermissionsAndRoleMemberships
- IgnoreUserProperties
- IgnoreWhiteSpace
- IgnoreWithElementOrder
- IncludeDependencies
- ThrowOnFileParseFailed
- UseCompatibilityLevel

For more information about SQL Compare options, see <http://www.red-gate.com/sca/ps/help/compareoptions>.

Aliases	None
Required?	false
Position?	named
Default Value	None
Accept Pipeline Input	false
Accept Wildcard Characters	false

-SQLDataCompareOptions <System.String>

Specifies the SQL Data Compare options to use when creating the script for validation. To include additional options, specify a comma-delimited list of the option names (eg 'DisableAndReenableDDLTriggers, CompressTemporaryFiles').

Aliases	None
Required?	false
Position?	named
Default Value	None
Accept Pipeline Input	false
Accept Wildcard Characters	false

-TransactionIsolationLevel <RedGate.Versioning.Automation.Shared.Domain.TransactionIsolationLevel>

Use this parameter to specify the isolation level for the transactions during the sync operation. Permitted values are: Serializable, Snapshot, RepeatableRead, ReadCommitted and ReadUncommitted. The default level is Serializable.

See <http://msdn.microsoft.com/en-gb/library/ms173763.aspx> for more details on transaction isolation levels.

Possible values: Serializable, Snapshot, RepeatableRead, ReadCommitted, ReadUncommitted

Aliases	None
Required?	false
Position?	named
Default Value	Serializable
Accept Pipeline Input	false
Accept Wildcard Characters	false

-QueryBatchTimeout <System.Int32>

The execution timeout, in seconds, for each batch of queries during the sync operation. The default value is 30 seconds. A value of zero indicates that no execution timeout will be enforced.

Aliases	None
---------	------

Required?	false
Position?	named
Default Value	0
Accept Pipeline Input	false
Accept Wildcard Characters	false

-AbortOnWarningLevel <RedGate.Versioning.Automation.Shared.Domain.WarningSeverity>

Use this parameter to set the minimum warning level that will cause the sync operation to abort.

Valid warning severity levels are:

- High
- Medium
- Low
- Information
- None (do not abort for any warnings)

The default setting is 'High'.

This parameter can't be used in addition to the Force parameter.

Possible values: Information, Low, Medium, High, None

Aliases	None
Required?	false
Position?	named
Default Value	High
Accept Pipeline Input	false
Accept Wildcard Characters	false

-Force <System.Management.Automation.SwitchParameter>

Use this parameter to prevent the sync operation from aborting regardless of any warnings.

An alternative way of doing this is to set the AbortOnWarningLevel to 'None'.

This parameter can't be used in addition to the AbortOnWarningLevel parameter.

Aliases	None
Required?	false
Position?	named
Default Value	False
Accept Pipeline Input	false
Accept Wildcard Characters	false

-IgnoreParserErrors <System.Management.Automation.SwitchParameter>

Use this parameter to force the SQL Compare engine to ignore any parser errors.

Aliases	None
Required?	false
Position?	named
Default Value	False
Accept Pipeline Input	false
Accept Wildcard Characters	false

-SqlCmdVariables <System.Collections.Hashtable>

Aliases	None
Required?	false
Position?	named
Default Value	None
Accept Pipeline Input	false
Accept Wildcard Characters	false

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see <http://technet.microsoft.com/en-us/library/hh847884.aspx>.

Inputs

The input type is the type of the objects that you can pipe to the cmdlet.

- **None.**
You cannot pipe input to this cmdlet.

Return values

The output type is the type of the objects that the cmdlet emits.

- **RedGate.Versioning.Automation.Compare.Domain.DatabaseSyncResult**

Examples

----- EXAMPLE 1 -----

```
$project = "C:\Work\scripts"
$staging = New-DatabaseConnection -ServerInstance "staging01\sql2014" -Database "Staging"
Sync-DatabaseSchema -Source $project -Target $staging
```

This example show how to update a database to match a SQL Source Control project.

----- EXAMPLE 2 -----

```
$buildArtifact = Import-DatabaseBuildArtifact "C:\Work\buildArtifacts\DatabaseBuildArtifact.1.0.0.nupkg"
$production = New-DatabaseConnection -ServerInstance "prod01\sql2014" -Database "Production" -Username
"AutomationUser" -Password "P@ssw0rd"
Sync-DatabaseSchema -Source $buildArtifact -Target $production
```

This example show how to update a test database to match a SQL Change Automation build artifact.

----- EXAMPLE 3 -----

```
$validatedProject = Invoke-DatabaseBuild "C:\Work\scripts"
$test = New-DatabaseConnection -ServerInstance "test01\sql2014" -Database "Test" -Username "AutomationUser" -
>Password "P@ssw0rd"
Sync-DatabaseSchema -Source $validatedProject -Target $test
```

This example shows how to update a test database to match a validated SQL Source Control project.

----- EXAMPLE 4 -----

```
$project = "C:\Work\scripts"
$test = New-DatabaseConnection -ServerInstance "test01\sql2014" -Database "Test" -Username "AutomationUser" -
Password "P@ssw0rd"

Sync-DatabaseSchema -Source $project -Target $test -AbortOnWarningLevel Medium
```

This example shows how to abort the update if there are any warnings at medium level or above.

Set the -AbortOnWarningLevel parameter to 'None' or use the Force parameter if you want the cmdlet to ignore all warnings.

----- EXAMPLE 5 -----

```
$project = "C:\Work\scripts"
$test = New-DatabaseConnection -ServerInstance "test01\sql2014" -Database "Test" -Username "AutomationUser" -
Password "P@ssw0rd"

$syncResult = Sync-DatabaseSchema -Source $project -Target $test
$syncResult.UpdateSql | Out-File -FilePath "C:\Export\update.sql"
```

This example show how to write the updateScript that was run to a file.

----- EXAMPLE 6 -----

```
$project = "C:\Work\scripts"
$test = New-DatabaseConnection -ServerInstance "test01\sql2014" -Database "Test" -Username "AutomationUser" -
Password "P@ssw0rd"

$options = "IgnoreComments, ObjectExistenceChecks, -ForceColumnOrder"

Sync-DatabaseSchema -Source $project -Target $test -SQLCompareOptions $options
```

This example shows how to specify SQL Compare options to use when performing the sync operation.

In this example, the \$options variable specifies that the IgnoreComments and ObjectExistenceChecks options should be included in addition to the default set of SQL Compare options. The minus sign before the ForceColumnOrder parameter indicates that this default option will be turned off.