

SQL Comparison Toolset[®]

Version 7.1.5

I D E R A

Table of Contents

1. Release notes	5
1.1 New features and fixed issues	6
1.2 Previous features and fixed issues	8
1.3 Known issues	25
1.4 Recommended IDERA Solutions	27
2. Welcome to SQL Comparison Toolset	28
2.1 What is IDERA SQL Comparison Toolset?	29
2.2 How SQL Comparison Toolset Helps	30
2.3 Document conventions	31
2.4 About IDERA	32
2.4.1 IDERA products	33
2.4.2 Contact IDERA	34
2.4.3 Legal notice	35
3. Getting started	39
3.1 Requirements	40
3.2 Supported platforms	41
3.3 Activating a license	42
3.4 Updates and support	44
4. IDERA SQL Schema Compare	45
4.1 SQL Schema Compare Application window	46
4.2 SQL Schema Compare Application Settings	47
4.3 SQL Schema Compare Workspace	50
4.3.1 SQL Schema Compare Add Databases	52
4.3.2 SQL Schema Compare Add Snapshots	54
4.3.3 SQL Schema Compare Remove Databases	56
4.4 SQL Schema Compare Comparing Schemas	57
4.4.1 SQL Schema Compare Object Types	58
4.4.2 SQL Schema Compare Options	60
4.4.2.1 SQL Schema Compare Comparison Options	61
4.4.2.2 SQL Schema Compare Scripting Options	63
4.4.2.3 SQL Schema Compare Execution Options	64
4.4.3 SQL Schema Compare Comparison Results	65
4.4.3.1 SQL Schema Compare Equal but Different	67
4.4.3.2 SQL Schema Compare Refreshing Results	68
4.5 SQL Schema Compare Working with Filegroups	69
4.5.1 SQL Schema Compare Comparing Filegroups	70
4.5.2 SQL Schema Compare Filegroup Mappings	72
4.6 SQL Schema Compare Entity Filters	76
4.6.1 SQL Schema Compare Entity Filters Overview	77
4.6.2 SQL Schema Compare Exclusion Rules	79
4.6.3 SQL Schema Compare Working with Entity Filters	81
4.7 SQL Schema Compare Synchronizing Schemas	83
4.7.1 SQL Schema Compare Selecting Objects	84
4.7.2 SQL Schema Compare Synchronization Script	86
4.7.3 SQL Schema Compare Working with the Script	88

4.7.4 SQL Schema Compare Executing Synchronization Script	90
4.8 SQL Schema Compare Comparison Sessions	92
4.9 SQL Schema Compare Tools and Utilities	93
4.9.1 SQL Schema Compare IDERA Snapshot Converter	94
4.9.2 SQL Schema Compare IDERA Snapshot Utility	95
5. IDERA SQL Schema Compare command line	97
5.1 SQL Schema Compare Command Line Arguments	98
5.2 SQL Schema Compare XML Config File	100
5.3 SQL Schema Compare Comparing Databases	103
5.4 SQL Schema Compare Creating Snapshots	104
5.5 Customizing SQL Schema Compare	105
5.5.1 SQL Schema Compare Excluding Objects	106
5.5.1.1 SQL Schema Compare Excluding Objects by Type	107
5.5.1.2 SQL Schema Compare Excluding Objects by Name	110
5.5.2 SQL Schema Compare Changing Comparison Options	112
5.5.3 SQL Schema Compare Entity Filters customization	115
5.5.4 SQL Schema Compare Schema Filters	120
5.5.5 SQL Schema Compare Changing Filegroup Settings	122
5.5.6 SQL Schema Compare Changing Object Mappings	123
5.6 SQL Schema Compare Error Codes	124
6. IDERA SQL Data Compare	125
6.1 SQL Data Compare Application window	126
6.2 SQL Data Compare Workspace	127
6.2.1 SQL Data Compare Add Databases	129
6.2.2 SQL Data Compare Remove Databases	131
6.2.3 SQL Data Compare Application Settings	132
6.3 SQL Data Compare Comparing	134
6.3.1 SQL Data Compare Comparing Databases	135
6.3.2 SQL Data Compare Comparing Selected Tables	136
6.3.3 SQL Data Compare Comparison Options	138
6.3.4 SQL Data Compare Scripting Options	139
6.3.5 SQL Data Compare Preparing the Comparison	141
6.3.5.1 SQL Data Compare Mapped Objects	142
6.3.5.2 SQL Data Compare Mapping Rules	144
6.3.5.3 SQL Data Compare Include/Exclude Objects	146
6.3.5.4 SQL Data Compare Comparison Keys	148
6.3.5.5 SQL Data Compare Where Clause	150
6.3.6 SQL Data Compare Comparing Data	151
6.4 SQL Data Compare Synchronizing	154
6.4.1 SQL Data Compare Generating Synchronization Script	155
6.4.2 SQL Data Compare Working with the Script	157
6.4.3 SQL Data Compare Executing Synchronization Script	159
6.5 SQL Data Compare Comparison Sessions	160
7. IDERA SQL Data Compare command line	161
7.1 SQL Data Compare Command Line Arguments Overview	162
7.2 SQL Data Compare XML Configuration File Overview	164
7.3 Customizing SQL Data Compare	166
7.3.1 SQL Data Compare Modifying Comparison Options	167

7.3.2 SQL Data Compare Changing Mapping Type	168
7.3.3 SQL Data Compare Excluding Database Objects	169
7.3.4 SQL Data Compare Excluding Columns	170
7.3.5 SQL Data Compare Defining Custom Keys	171
7.3.6 SQL Data Compare Defining Mapping Rules	172
7.3.7 SQL Data Compare Specifying WHERE Clauses	173
7.4 SQL Data Compare XML Configuration File Reference	174
7.5 SQL Data Compare Error Codes	180

Release notes

IDERA SQL Comparison Toolset is a bundle of two products, IDERA SQL Schema Compare and IDERA SQL Data Compare. While SQL Schema Compare allows you to compare SQL Server database schemas and propagate schema changes from one environment to another, SQL Data Compare lets you compare and synchronize tables across servers, databases, and versions when performing data migrations, copying databases, or auditing data.

To get a quick glimpse into the newest features, fixed issues, and known issues in this release of IDERA SQL Comparison Toolset, review the following sections of the Release Notes:

- [New features and fixed issues](#)
- [Review issues fixed by this release](#)
- [Review previous features and fixed issues](#)
- [See known issues](#)
- [See list of recommended IDERA Solutions](#)

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

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

New features and fixed issues

IDERA SQL Comparison Toolset provides the following new features and fixed issues.

7.1.5 New features

There are no new features in this release.

7.1.5 Fixed issues

Resolves an issue causing an error before the comparison completes

This release resolves an issue causing some IDERA SQL Schema compare users to see an error message stating that an item with the same key is already added to the database. This error appeared before the comparison was complete and related to database assemblies didn't map properly due to multiple files or multiple file having the same name but a different path or extension.

Corrects an issues caused by conflicting data types

IDERA SQL Comparison Toolset 7.1.5 corrects a minor scripting issue with the user-defined types based on SQL Server native types **varchar(max)**, **nvarchar(max)**, and **varbinary(max)**.

7.1 New features

Supports spatial index AUTO_GRID tessellation schemes

IDERA SQL Comparison Toolset 7.1 supports the following spatial index auto-tessellation schemes:

- **GEOMETRY_AUTO_GRID** Supported on SQL Server 2012, SQL Server 2014, SQL Server 2016, and Azure v12.
- **GEOGRAPHY_AUTO_GRID** Supported on SQL Server 2012, SQL Server 2014, SQL Server 2016, and Azure v12.

Workspace displays a warning if JavaScript is disabled

The IDERA SQL Comparison Toolset Workspace now displays a warning message if JavaScript in Internet Explorer is disabled. JavaScript is not critical, but is needed for some of the Workspace functionalities. A master option in the app.config can hide this warning. The toolset requires JavaScript to be enabled for IE only.

7.1 Fixed issues

Improves data export to Microsoft Excel

IDERA SQL Data Compare includes the following resolutions regarding the export functionality:

- **Export to CSV File.** The exported CSV file now is fully compliant with .csv specifications. This includes the correct data separators (depending on the data type), the proper handling of the new-lines embedded in the column values, etc.
- **Export to Text File.** With the new option to export to text, the user can select field and row separators and to include or exclude column name.

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

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

Previous features and fixed issues

This build includes many new features and fixed issues, including the following updates from previous releases.

7.0 New features

Installation and configuration updates

IDERA SQL Comparison Toolset 7.0 contains the following installation and configuration updates:

- The shortcuts for all IDERA SQL Comparison Toolset executables now support the option to **Run as administrator**.
- Installation no longer supports Microsoft .NET 2.0.
- The Snapshot Converter utility is no longer included in IDERA SQL Comparison Toolset. This utility was used to provide snapshot conversion from IDERA SQL Comparison Toolset 2.5 and therefore is no longer necessary.
- The IDERA SQL Comparison Toolset version, stored in the registry key **HKLM\SOFTWARE\Idera\SQLtoolbox\SQL comparison toolset**, now reflects version 7.0.0.0.

SQL Schema Compare Microsoft SQL Server 2016 support

IDERA SQL Comparison Toolset 7.0 offers support for the following Microsoft SQL Server 2016 features:

Security policies

IDERA SQL Schema Compare supports the new securities policy on a regular disc table with no restrictions. IDERA SQL Schema Compare handles all synchronization scenarios and can:

- create a policy
- drop a policy.
- alter the policy by adding, dropping, or altering the policy predicates
- alter the policy state
- alter the policy “not for replication” attribute.

In addition to the regular tables, IDERA SQL Schema Compare also supports policies on other database objects. Note that each type of policy does contain some restrictions and other subtleties imposed by SQL Server, such as:

- Security policies on memory tables. The predicate in this case can be created only with a natively-compiled function. Being natively-compiled makes the function a non-transactional object. In the context of schema, it means that during the database synchronization, the function is executed outside of the main transaction similar to memory tables. The security policy itself remains transactional.
- Security policies on views. This type of policy supports only the FILTER predicate and not the BLOCK.
- Azure v11 does not support security policies.
- Azure v12 supports security policies.

IDERA SQL Schema Compare 7.0 includes the new comparison option **Compare security**

policies. This option determines whether the policies are compared and synchronized.

Column master keys

IDERA SQL Schema Compare supports the new column master key (CMK). The CMK comparison includes the following limitations:

- The CMK definition contains a component referred to as the “key-store-provider.” This is a software product installed in the client machine that provides access to the “store” containing the CMK. ***If the provider does not exist in the client machine***, the CMK synchronization fails. Because the provider is a resource outside of the database, IDERA SQL Schema Compare cannot check whether it exists. It will however generate a warning so that the user is aware of the issue.
- Another component of the CMK definition is the “key-path”. This is the key, in the key store, that provides the CMK encryption functionality. ***If the key is not found in the client machine***, the CMK in the database fails. Because this key is a resource outside of the database, IDERA SQL Schema Compare cannot verify whether it exists. It will however generate a warning so that the user is aware of the issue.
- IDERA SQL Schema Compare cannot synchronize changes between two CMKs that have columns bound to them. This type of change requires a table rebuild, which cannot occur when columns are encrypted.

It is important to mention that Azure v12 supports CMK, but uses a provider that is different from the on-premise SQL Server. IDERA SQL Schema Compare compares the CMK on Azure and generates a warning if the provider is not supported by the target database. Note that the warning may not be completely accurate because the list of providers is not fully known and goes beyond the scope of the database. The only Azure provider known at this time is the Azure Key Vault provider.

IDERA SQL Schema Compare 7.0 includes the following new options for the CMK:

- **Compare column master keys.** Determines whether the CMK are compared.
- **Compare column master key provider.** Indicates whether the key-store-provider of the CMK is compared.
- **Compare column master key pat.** Indicates whether the key-path component of the CMK is compared.

Column encryption keys

IDERA SQL Schema Compare supports the new column encryption keys (CEK), which work with the CMK to provide column encryption in SQL Server 2016. CEKs are also supported on Azure v12.

The new option **Compare column encryption keys** determines whether the CEKs are compared and synchronized.

Natively-compiled functions

Natively-compiled functions are SQL functions created using the `natively-compiled` attribute. They are used by memory-object features, such as the security policies on memory tables or

default/check constrains on these tables. IDERA SQL Schema Compare supports the natively-compiled functions. Note that these functions are non-transactional, so they are executed outside the main schema transaction when databases are synchronized.

There are no new options for natively-compiled functions. The options for standard functions control the natively-compiled functions as well.

Azure v12 does not support natively-compiled functions.

Natively-compiled triggers

Natively-compiled triggers are SQL triggers created using the natively-compiled attribute. They are used primarily by the memory tables. IDERA SQL Schema Compare supports the natively-compiled triggers. Similar to other natively-compiled objects, these triggers are non-transactional.

There are no new options for natively-compiled triggers.

Azure v12 does not support natively-compiled triggers.

Columns

IDERA SQL Schema Compare 7.0 supports the following SQL Server 2016 column enhancements.

- **Encryption.** IDERA SQL Schema Compare supports the column encryption specified via the clause: `ENCRYPTED WITH (...)` with some restrictions. For example, a schema change that requires a table rebuild is not supported when the table contains encrypted columns. In this case, IDERA SQL Schema Compare marks the table “non-comparable.” This type of change is not supported because data transfer, which is part of the rebuilt script, is not possible when the table is encrypted.

IDERA SQL Schema Compare 7.0 includes two new options for the column encryption feature:

- **Compare column encryption.** Indicates whether the properties used by the column encryption is compared.
- **Script column encryption.** Indicates whether the encryption is scripted.

Note that encrypted columns are supported on Azure v12.

- **Dynamic Data Mask.** IDERA SQL Schema Compare 7.0 supports the column dynamic data mask specified via the clause: `MASKED WITH (FUNCTION = email()|default()|partial()...)` clause. This feature provides data obfuscation for a column.

IDERA SQL Schema Compare 7.0 includes two new options for the dynamic data mask:

- **Compare column mask.** Indicates whether the mask is compared.
- **Script column mask.** Determines whether the mask is scripted.

Note that dynamic data mask is supported on Azure v12.

- **System-versioning system-time columns.** System-time columns are type `DateTime2` columns created with special attributes that can be used by the new system-versioning feature in SQL Server 2016.

IDERA SQL Schema Compare supports the system-time columns, including the “hidden” attribute and the new `sys-versioning` table clause.

System-versioned tables

System-versioned tables, sometimes referred to as “temporal” tables, are fully supported. Because sys-versioning is a major feature in SQL Server 2016, with many parts and objects, it is hard to fully document it. IDERA SQL Schema Compare:

- supports system-time columns created with the attribute `GENERATED ALWAYS AS ROW START|ROW END`.
- supports the `HIDDEN` attribute for the system-columns.
- maps the system-time columns by the “period-type” instead of name. The period type is the `ROW START|ROW END` attribute.
- because the system-time columns have the same type and do not contain user data, IDERA SQL Schema Compare always marks them as equal. They do not trigger a table difference.
- adds the system-time columns even when the target table is not system-versioned. This is done so that the user can enable the system-versioning via the `ALTER TABLE` statement.
- supports the new table embedded clause: `PERIOD FOR SYSTEM_TIME ([ValidFrom],[ValidTo])`.
- supports the new table option: `SYSTEM_VERSIONING = ON (HISTORY_TABLE = <table>, DATA_CONSISTENCY_CHECK = ON)`.
- supports the system-versioning on partition tables. The primary table, the history table or both can be partitioned.
- supports system-versioning on memory tables.

When it comes to scripting, system-versioned tables are notoriously more complicated than the regular tables. Most schema changes do not succeed when the sys-versioning is enabled. When necessary, IDERA SQL Schema Compare disables sys-versioning, synchronizes the table, and enables it back. This action triggers multiple warnings to make sure that the user is aware when the sys-versioning changes.

IDERA SQL Schema Compare 7.0 adds the following new options for the sys-versioned tables:

- **Compare system versioning.** Determines whether the sys-versioning is compared.
- **Ignore system-versioning history table name.** Indicates whether the name of the history table should be ignored. Because the History table is hidden from the user, its name is not relevant.
- **Script system versioning.** Determines whether the sys-versioning is scripted.
- **Drop system-versioning history table.** Indicates whether the history table is dropped when the system-versioning is disabled or the primary table is dropped.

IDERA SQL Schema Compare supports the system-versioning on Azure v12.

In-Memory tables

SQL Server 2016 introduces many enhancements related to memory tables. IDERA SQL Schema Compare supports the following:

- New unique constraints on a memory table.
- New check constraint on a memory table.
- New foreign keys on a memory table. The foreign key references another memory table.
- New `AFTER` triggers on a memory table. Triggers are natively-compiled.
- New system-versioning on memory tables.

- New memory table `IDENTITY`. SQL Server 2016 limits this to `IDENTITY(1,1)`.

Extended properties are supported on all objects. It is important to note that all new objects, including the memory tables, are non-transactional. IDERA SQL Schema Compare executes them outside the main transaction when databases are synchronized.

The comparison options for memory tables existed in the previous version of IDERA SQL Schema Compare. Version 7.0 adds a new **Compare hash index bucket count** option, which determines whether the `BUCKET_COUNT` property of a hash index is compared. A “hash index” can be added to a memory table type as well.

Memory tables have the following limitations:

- **Memory tables are non-transactional.** This is a SQL Server restriction. However, other database objects that normally are transactional, when associated with the memory tables, become non-transactional as well. For example, the database schema is a transactional object. When the schema is associated with a memory-table, it changes to non-transactional. The same is true for extended properties, permissions, unique constraints, primary keys, etc.
- **SQL Server 2016 supports memory table indexes on n(var)char columns of any collation.** SQL Server 2014 is more restrictive and requires a `_BIN` collation. This seemingly small change could cause synchronization issues between a SQL Server 2016 and a SQL Server 2014.

Azure v12 does not support memory tables.

Stretch tables

IDERA SQL Schema Compare supports the stretched tables or the remote data archiving feature. RDA is supported even on system-versioned tables (the history in this case can be configured as a stretched table).

Schema changes that require a table rebuild are not supported when the table is stretched. Most of the changes on a stretched table require additional script that disables stretching. IDERA SQL Schema Compare generates warnings, so that the user is aware when the table stretching changes.

Version 7.0 adds the following new options specifically for the stretched tables:

- **Compare remote data archiving.** Determines whether the RDA feature is compared.
- **Script remote data archiving.** Determines whether the RDA is scripted.
- **Script history remote data archiving.** Determines whether the RDA of a history table is scripted. This applies to the sys-versioned tables only.
- **Abandon remote data when disabling archiving.** Indicates whether the remote data is discarded when the RDA is disabled.

Azure does not support stretched tables. The remote data of a stretched table is on Azure, but the primary stretched table is supported only on-premise databases.

Stretched database

IDERA SQL Schema Compare cannot enable a database for stretching. It does however generate a template that can be used by the user to manually, outside the toolset, enable the

database stretching. The following T-SQL snippet, commented out, represents the template:

```

/*
--
-- The following template can be used to enable remote data
archiving in the database
--

--1. run sp_configure and turn the 'remote data archive' option on
EXEC sp_configure 'remote data archive' , '1';
GO
RECONFIGURE;
GO

--2. create the database master key
CREATE MASTER KEY ENCRYPTION BY PASSWORD = '<password>';
GO

--3. create a database credential to use for communication between
the on-premise database and the remote Azure database
CREATE DATABASE SCOPED CREDENTIAL <credential> WITH IDENTITY =
'<identity>' , SECRET = '<password>';
GO

--4. turn on the remote data archiving in the database
ALTER DATABASE <database>
SET REMOTE_DATA_ARCHIVE = ON
(
SERVER = '<azure-server>.database.windows.net' ,
CREDENTIAL = <credential>
);
GO
*/

```

It is worth mentioning that IDERA SQL Schema Compare does not automatically enable the database stretching for two reasons:

1. It cannot create the database master key, since the key password is not available in the database catalog.
2. It cannot create the database-scoped credential used for stretching. The credential identity and the secret, which are an Azure account and a password, are not in the database catalog.

External tables

IDERA SQL Schema Compare 7.0 does not support the external tables.

User-defined table types

IDERA SQL Schema Compare 7.0 brings the following changes to the user-defined table types:

- In addition to memory table types, indexes are now supported on regular table types.
- Objects associated with a memory table type, including the table type, are

non-transactional.

- IDERA SQL Schema Compare 7.0 fixes an issue with extended properties on columns. In some cases, they failed to properly synchronize.

Columnstore indexes

IDERA SQL Schema Compare 7.0 supports the following new features on columnstore indexes:

- The new columnstore filter predicate, specified via the WHERE clause.
- The new columnstore COMPRESSION_DELAY option.

Note that columnstore indexes are supported on a premium tier of Azure v12. As a result, IDERA SQL Schema Compare allows and supports them on all versions of Azure.

New permissions

SQL Server 2016 contains the following new permissions supported by IDERA SQL Schema Compare:

- ALTER ANY COLUMN ENCRYPTION KEY (ALCK)
- ALTER ANY COLUMN MASTER KEY (ALCM)
- ALTER ANY DATABASE SCOPED CONFIGURATION (ALDC)
- ALTER ANY EXTERNAL DATA SOURCE (AEDS)
- ALTER ANY EXTERNAL FILE FORMAT (AEFF)
- ALTER ANY SECURITY POLICY (ALSP)
- VIEW ANY COLUMN ENCRYPTION KEY DEFINITION (VWCK)
- VIEW ANY COLUMN MASTER KEY DEFINITION (VWCM)
- ALTER ANY MASK (AAMK)
- UNMASK (UMSK)
- EXECUTE ANY EXTERNAL SCRIPT (EAES)

Azure v12 supports the same permissions.

New comparison options

Below is a summary of the new options supported by IDERA SQL Schema Compare 7.0:

- **Compare security policies.** Indicates whether the security policies are compared.
- **Compare column master keys.** Indicates whether the CMK are compared.
- **Compare column master key provider.** Whether the provider of a CMK is compared.
- **Compare column master key path.** Whether the key-path property of a CMK is compared.
- **Compare column encryption keys.** Indicates whether the CEK are compared.
- **Compare column encryption.** Indicates whether the properties related to column encryption, such as the encryption key, the encryption algorithm, are compared.
- **Compare column mask.** Indicates whether the column dynamic data mask is compared.
- **Compare system versioning.** Whether the sys-versioning of the regular or memory tables is compared.
- **Ignore system versioning history table name.** Whether the name of the history table associated with the sys-versioning is ignored.
- **Compare hash index bucket count.** Whether the BUCKET_COUNT property of an hash index is compared.

- **Compare remote data archiving.** Whether the remote data archiving feature (or stretching) is compared.
- **Script column encryption.** Whether the column encryption is scripted.
- **Script column mask.** Whether the column dynamic data mask is scripted.
- **Script system versioning.** Whether the sys-versioning is scripted.
- **Drop system versioning history.** Indicates whether the history table should be dropped when the primary table is dropped or when the sys-versioning is disabled.
- **Script remote data archiving.** Whether the remote data archiving (or stretching) is scripted.
- **Script history remote data archiving.** Whether the remote data archiving (or stretching) is scripted specifically for the history of a system-versioned table.
- **Abandon remote data when disabling archiving.** Whether the Azure remote data should be discarded when the stretching is disabled.

Other changes

IDERA SQL Schema Compare 7.0 contains other changes including:

1. The SQL Server Database Selection popup now includes the entity filters. It allows the user to set the filters before comparing databases. Filters are fully saved and retained in the workspace session.

? Unknown Attachment

2. The IDERA SQL Schema Compare command line <Database> element supports the following new properties:
 - a. **ConnectionString.** The entire database connection script.
 - b. **PacketSize.** The SQL connection packet-size.
 - c. **Pooled.** Indicates whether the SQL connection should be pooled.
 - d. **OtherProperties.** Other connection properties.
3. The command line displays operational message on the screen and logs them at the same time; used to be either or in the previous version.
4. The command line /quiet option does not display messages on the screen. Messages are logged regardless of this option.
5. The command line script and log file are not created if databases are equal; used to be empty in the previous version.
6. The command line uses by default UTF-8 encoding for all files.

SQL Schema Compare Command Line wizard

IDERA SQL Schema Compare 7.0 contains the new Command Line wizard, which allows the user to create a command line xml file step-by-step, via a simple and intuitive interface.

? Unknown Attachment

In addition to the Command Line wizard, IDERA SQL Schema Compare includes a new option in the schema compare UI. It allows the user to export a comparison session and create a command line directly from the UI.

? Unknown Attachment

Data Compare updates

New features and enhancements

IDERA SQL Data Compare 7.0 supports the following new features:

- Compares the data on system-versioned tables, both disc and memory tables.
- Can disable, via an option, the system-versioning before synchronizing the data.
- Compares the stretched tables.
- Can set, via an option, the stretch table query scope. This can be set to either LOCAL_ONLY or LOCAL_AND_REMOTE (default).
- Supports the new memory table foreign keys and natively-compiled triggers. The objects are non-transactional.

IDERA SQL Data Compare 7.0 changes the transactional isolation level from “SERIALIZED” to “READ COMMITTED” in order to supported the script that disables the foreign keys on memory tables.

IDERA SQL Data Compare does not support encrypted tables.

New Compare options

IDERA SQL Data Compare 7.0 contains the following new options:

- **Compare system versioned tables.** Indicates whether the sys-versioned tables (or temporal tables) are compared.
- **Disable system versioning.** Whether the sys-versioning is disabled before synchronizing the data.
- **Compare stretched tables.** Whether the stretched tables are compared.
- **Stretched table query scope.** Whether the query against a stretched table selects only the local data (LOCAL_ONLY) or both local and remote data (LOCAL_AND_REMOTE).

Data Compare Command Line Wizard

IDERA SQL Data Compare 7.0 brings the new command line wizard. The wizard is similar to schema and creates an xml file specified for the data compare command line. In addition the wizard, the user can create the config file directly from the data compare UI.

? Unknown Attachment

7.0 Fixed Issues

Schema Compare

Version 7 fixes the following issues:

- It fixes an issue with the SPARSE column. The NULL|NOT NULL constraints should be specified after the SPARSE attribute.
- It fixes an issue that occurred when the synchronization script contains extended characters, such as Chinese letters. Schema now uses UTF-8 encoding for the script and the log files.

- It fixes an issue with the DATA_COMPRESSION attribute on a filestream table. If the filestream table contains an embedded primary key, data compression must be specified either under the primary key or the table, but not both.

5.0 New features

Microsoft Azure SQL Database v11 and v12 Preview support

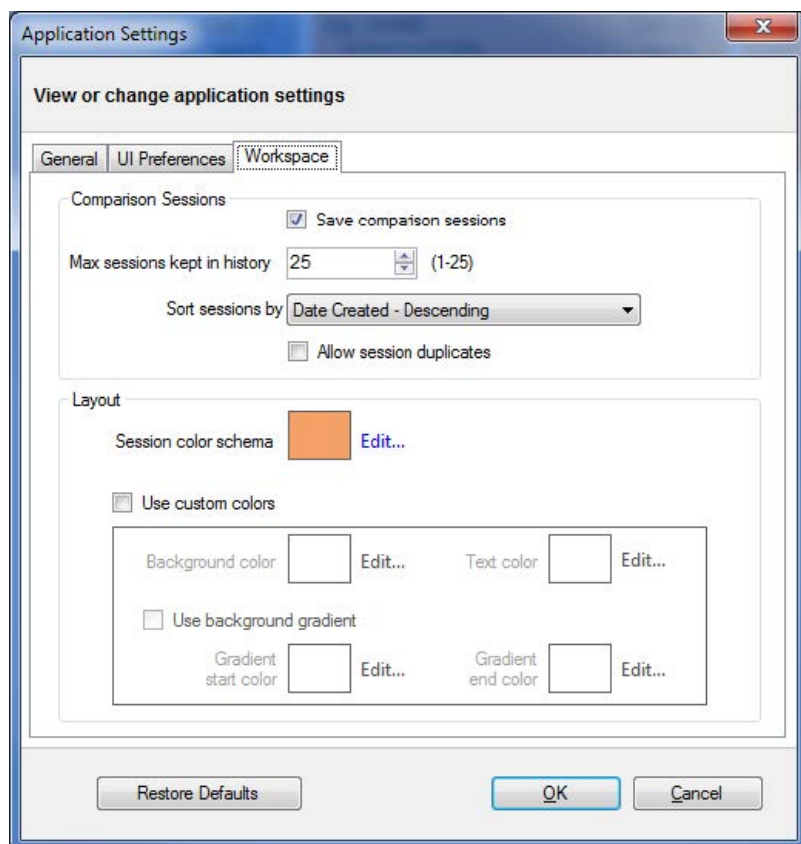
IDERA SQL Comparison Toolset 5.0 offers full support for Microsoft Azure SQL Database v11 and the preview of v12. Comparing cloud-based or on-premise databases uses the same process within IDERA SQL Comparison Toolset. The toolset handles the platform differences automatically and produces any warnings for you to address when necessary.

SQL Data Compare also supports the Azure data restriction that a table or view must have a clustered index. The SQL Data Compare user interface includes a warning and a new popup window that displays a list of any tables or views excluded because of this restriction.

While IDERA SQL Comparison Toolset now supports Microsoft Azure SQL Database v12, only the Preview release was available during development of the IDERA SQL Comparison Toolset. Note that Microsoft may make changes before the final product release.

Multiple session support

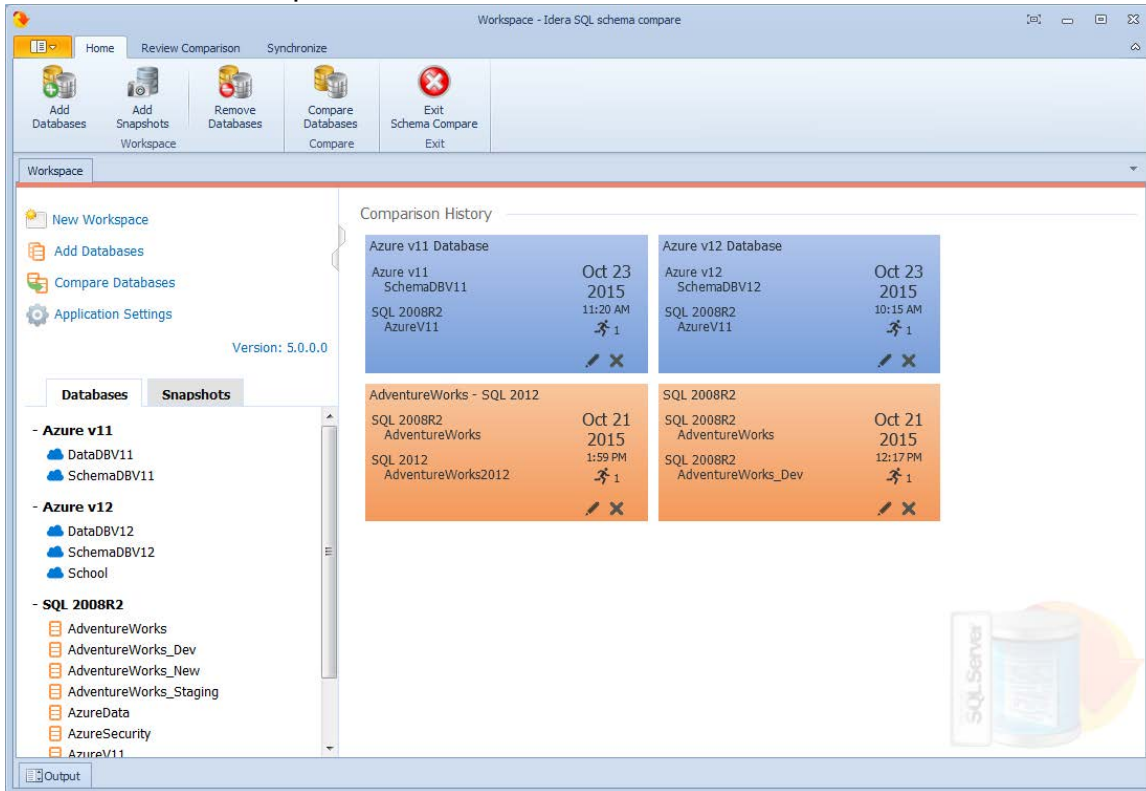
IDERA SQL Comparison Toolset now allows you to have multiple sessions for the same SQL database. You can control this feature on the Workspace tab of the Application Settings window, as show in the following image.



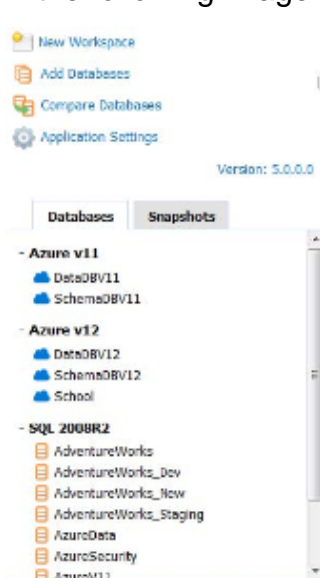
SQL Schema Compare Workspace tab user interface improvement

The SQL Schema Compare Workspace tab offers user interface improvements including:

- The Workspace tab now displays the session name for you to easily locate a session, even when there are multiple sessions for the same databases, as shown in the following image.



- The Databases tab of the Workspace now includes a cloud icon next to Microsoft Azure SQL Databases for you to distinguish between cloud and on-premise databases, as shown in the following image.

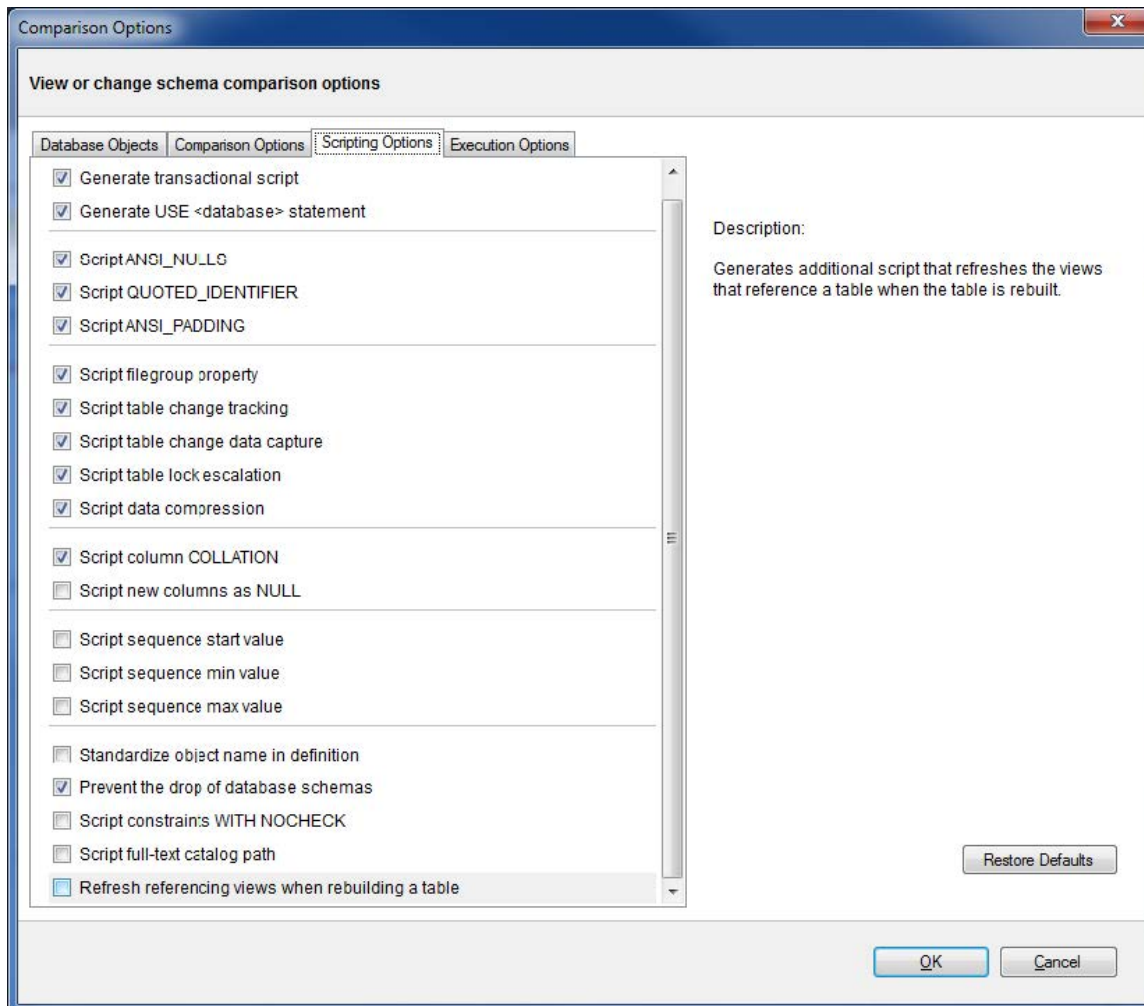


- A new landing page appears in SQL Schema Compare only when the Workspace is empty. The page makes it easier for a first-time user to navigate the toolset.

SQL Schema Compare refresh option

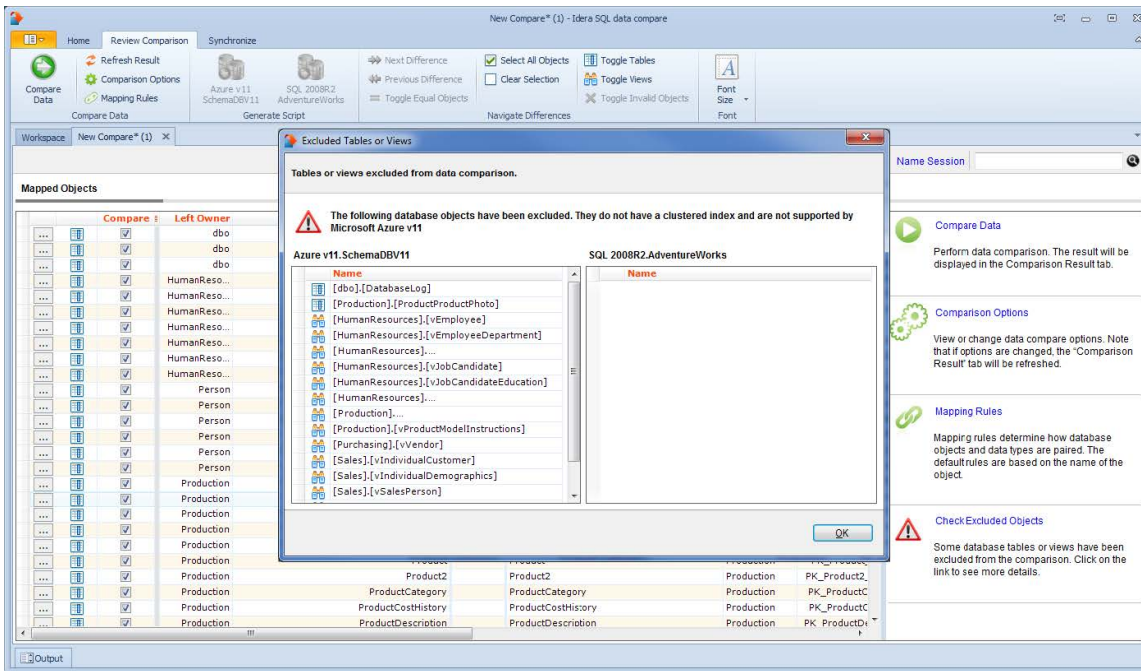
SQL Schema Compare introduces a new comparison option named **Refresh referencing views**

when rebuilding a table. The option is located on the Scripting Options tab of the Comparison Options window, and it is disabled by default. When enabled, the table-rebuild portion of the schema synchronization script contains additional script that refreshes the views that reference the table that is being rebuilt. You can also use this option in the command line.



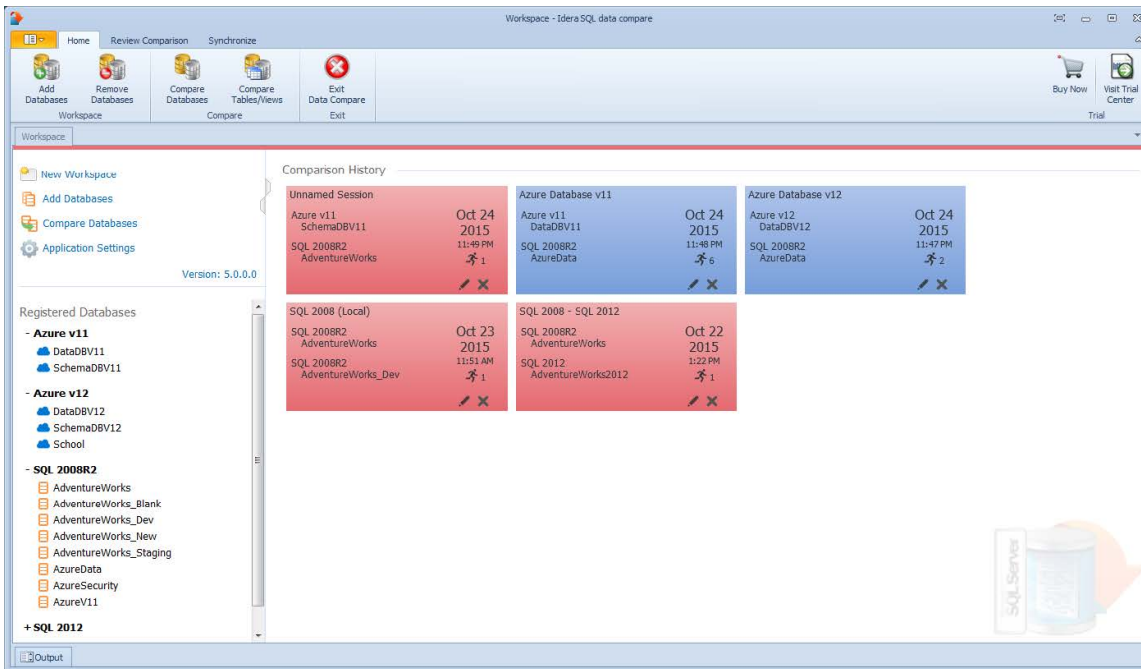
SQL Data Compare user interface improvements for Microsoft Azure SQL Database v11

SQL Data Compare includes a new exclusion window that displays a list of database objects, such as table or views, excluded from data comparison due to the Microsoft Azure SQL Database v11 restriction related to missing clustered indexes. You can access this window from a link that appears during the pairing step of the data comparison. Note that the link is visible only if the restriction has affected tables or views in the database.



IDERA purchase and Trial Center support in trial versions

Two new buttons are included in the trial version of both the SQL Schema Compare and SQL Data Compare user interfaces. These buttons allow you to visit the purchasing page to upgrade a trial version of IDERA SQL Comparison Toolset to a full version, or allow you to access the Trial Center page for more information about using a trial version of this product. Note that these buttons appear only when you are using a trial version of IDERA SQL Comparison Toolset.



5.0 Fixed Issues

- Fixes a “false” difference triggered by an unspecified TEXTIMAGE filegroup of a table. This prevents unnecessary table rebuilds.

- Performs an additional check before scripting the where clause of a database statistic. In some cases, SQL Server flags the stats as having a where clause, when they actually do not have one.
- Updates the spatial index script to handle unspecified grid levels. Schema comparison also ignores unspecified levels to prevent “false” differences between indexes.
- Adds the data compression option to the unique constraint’s scripting. The previous version did not consider data compression for unique constraints.
- Fixes an issue related to the collation of a column with a user-defined data type. Since collation for this type of columns can’t be scripted but could still trigger a difference, the collation clause is ignored during column comparison.
- Fixes an issue with assembly files. Now only the file name of an assembly is compared instead of including the path and the extension.
- During the SQL Server discovery, the previous version of IDERA SQL Comparison Toolset generated an error when no SQL Servers were found in the network. This issue effected the Add Database window in both SQL Schema Compare and SQL Data Compare.

4.1 New features

Windows .NET 4.0 support

- IDERA SQL Comparison Toolset 4.1 supports all Microsoft Windows operating systems using .NET 4.0.

SQL Schema Compare

- **Enhanced user interface.** SQL Schema Compare includes a new ribbon control that improves the usability of the tool.
- **Entity filter replace schema filters.** Entity filters replace schema filters used to filter data. In the user interface, you can access these filters from the context menu of the comparison grid. The command line interface includes three additional sample xml files that provide additional information about how to use the entity filters. For more information about using these filters, see the online help accessible from within IDERA SQL Comparison Toolset.

4.1 Fixed Issues

- This release removes a table dependency that negatively affected performance for some users with large databases.
- This release fixes a table scripting issue with the TEXTIMAGE_ON clause. SQL Server requires the TEXTIMAGE_ON clause when a table contains large binary fields, such as image, text, and ntext. This issue caused the SQL Server system catalog to become out-of-date and fail to report the clause properly. IDERA SQL Comparison Toolset 4.1 contains additional checks and scripts the TEXTIMAGE_ON only when it is required.
- SQL Comparison Toolset 4.1 includes a fix related to dependencies on SQL Server 2005. This issue occurred on some SQL Server 2008 databases when the compatibility level was set to SQL Server 2005.

4.0 New features

SQL Server 2014 Support

IDERA SQL Comparison Toolset 4.0 supports the use of SQL Server 2014 with the following

notes:

SQL Schema Compare

- **Memory-Optimized tables.** SQL Server 2014 supports a new memory table and other objects associated with it, such as:
 - a new Hash index
 - a new Range index
 - a new database filegroup that contains the memory tables

The Comparison Options window includes a new option to support memory-optimized tables. The **Compare memory-optimized tables** check box is located on the Comparison Options tab under the **Table Options** section.

- **BUCKET_COUNT property.** The new BUCKET_COUNT property for hash indexes is supported.
- **ColumnStore indexes .** SQL Server 2014 supports a new type of index called a "columnstore," which is fully supported by SQL Schema Compare.
- **XML indexes .** SQL Server 2014 introduces two new xml indexes and associated enhancements, such as:
 - a new primary selective xml index
 - a new secondary selective xml index
 - new xml index namespaces
 - new xml index paths
- **INCREMENTAL = ON | OFF statistic .** The new statistic property, INCREMENTAL = ON | OFF is supported.
- **ALTER ANY DATABASE EVENT SESSION permission .** The new permission type, ALTER ANY DATABASE EVENT SESSION is supported.
- **Non-supported permission types .** SQL Server 2014 introduces three new permission types that have server-scope, which are ignored by SQL Schema Compare as it can synchronize only those permissions that have database-scope. These new permissions include:
 - CONNECT ANY DATABASE
 - IMPERSONATE ANY LOGIN
 - SELECT ALL USER SECURABLES
- **Natively-compiled stored procedures .** SQL Schema Compare 4.0 fully supports natively-compiled stored procedures in SQL Server 2014.

SQL Data Compare

- **Memory-Optimized tables .** SQL Server 2014 supports memory tables except for custom keys. This issue is a limitation in SQL Server 2014 itself, which does not allow updates to the primary key columns of a memory table.
- **Hash indexes .** Hash indexes are fully supported in SQL Data Compare, which also considers these indexes when choosing the unique comparison key.
- **ColumnStore indexes .** ColumnStore indexes are supported by SQL Data Compare with the following exception. SQL Server 2014 does not allow data changes on a table that contains a non-clustered columnstore index. To synchronize this index, it is disabled before data synchronization, and then enabled after the sync is complete. Note that a clustered columnstore is updatable and does not need disabling before synchronization.

Filegroup support in SQL Schema Compare

SQL Schema Compare includes new options for working with filegroups and the filtergroup clause of database objects. The **Compare Database Filegroups** check box on the Schema Objects tab of the Comparison Options window indicates whether you want to compare and synchronize the database filegroups including the data files.

Filegroup data files also are supported including allowing the user to change the physical file mapped to a data file directly in the comparison grid. Note the following limitations:

- You can add a filegroup only to the database. Altering or dropping is not supported.
- You can add data files only to new filegroups. Dropping or event-adding data files to existing filegroups is not supported.

Filegroup mappings allow you to change the way filegroups are mapped, compared, scripted, and synchronized. IDERA SQL Comparison Toolset now offers the following options for mapping filegroups:

- **Mapping by name** . This is the default setting.
- **Ignoring filegroups entirely** . This option ensures that any changes in the filegroups do not trigger a schema difference by ignoring the filegroup clause of tables, indexes, and other database objects. The **Compare Database Filegroups** option determines whether filegroups at a database level, including data files, are compared and synchronized.
- **Custom mapping** . This option allows you to map filegroups as needed.

Note that filestream filegroups are mapped separately from regular filegroups, although the rules for mapping are the same.

Scripting the filegroup clause is a new option in the Filegroup Mapping Rules window that determines how the filegroup clause is scripted for some database objects. It affects only those objects that support the filegroup clause in tables, indexes, etc. and does not affect filegroups and partition schemes. When the **Script the filegroup clause when synchronizing schema** check box is clear, the filegroup clause is scripted using the DEFAULT filegroup or ignored in some cases.

New comparison options in SQL Data Compare

SQL Data Compare includes two new options for comparing data. The **Compare and synchronize memory tables** option in the Data Compare Options window allows you to compare and synchronize your memory tables. The **Compare and synchronize columnstore tables** option allows you to compare and synchronize your columnstore tables.

New SQL Comparison Toolset workspace

IDERA SQL Comparison Toolset 4.0 features a new, easier-to-use interface for navigating through session history.

- The navigation tree now offers two tabs to separate live databases from database snapshots that are added to the workspace.
- Workspace links are grouped at the top of the window.
- You can adjust the width of the tree panel for ease-of use.
- The Application Settings window now allows you to customize the look and feel of the workspace including changing the colors of sessions to group similar sessions together.

- The Customize Session window includes new sorting options allow you to organize and manage your session history, edit session data such as name, description, and count. The **Execution Count** field in the Customize Session window indicates the number of times a session is launched.

SQL Comparison Toolset removal from the SQL Toolbox framework

The IDERA SQL Comparison Toolset is now separated from the SQL Toolbox, meaning that it is no longer necessary to install the IDERA SQL Toolbox framework prior to installing IDERA SQL Comparison Toolset.

New CLI options

This release includes two new options for the command line interface. The **CompareMemoryTables** option indicates whether you want to compare and synchronize memory-optimized tables. The **CompareColumnStoreTables** option indicates whether you want to compare and synchronize columnstore tables.

4.0 Fixed Issues

SQL Schema Compare

- This release fixes an issue related to missing dependencies associated with database synonyms.
- An issue causing a "System.NullReferenceException" error appearing during a comparison is resolved. this issue resulted from missing data compression in the table that contains the indexes.

SQL Data Compare

- This release fixes an issue related to extended properties on foreign keys.

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

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

Known issues

IDERA strives to ensure our products provide quality solutions for your SQL Server needs. The following known issues are described in this section. ***If you need further assistance with any issue***, please contact [IDERA Support](#).

Known issues in 7.1

Removes Azure v12 support for assemblies and CLR objects

IDERA SQL Comparison Toolset 7.1 removes Azure v12 support for assemblies and CLR objects including procedures, functions, triggers, types etc.

Two different databases appear equal to some users

IDERA SQL Comparison Toolset includes a number of options that can cause the following issues:

- IDERA SQL Schema Compare index comparison options may “hide” differences between two different indexes and make them appear equal. Currently, IDERA SQL Schema Compare supports seven index options available by clicking **Comparison Options > Index Options**. If an option is unchecked, the property affected by it is not compared.
- When check constraints have the same name on different tables, they may not be identified as different when using IDERA SQL Schema Compare. For example, the check constraints would look as follows:

```
LOC_TX.g_1835_ck
LOC_CA.g_1835_ck
```

Note the following operation of IDERA SQL Schema Compare:

1. SQL Server supports two types of check constraints: column-constraints and table-constraints. A column constraint is applied to one column only while a table constraint affects multiple columns.
2. Table-constraints have a table-scope, i.e. the constraints on the left database are compared with constraints of the same table on the right database.
3. Column constraints have a table and column scope, so constraints on the left database are compared with the constraints of the same table, under the same column, on the right database.

Using the example, the constraints of the LOC_TX table on the left database are compared with those of the LOC_TX table on the right database. Constraints of LOC_TX and LOC_CA are not mixed together.

IDERA SQL Schema Compare can ignore the name of a check constraint if the option **Ignore check constraint name** is checked. This option affects only column constraints and it is located under **Comparison Options > Constraint Options**.

Comparison grid appears to not work properly if a specific option is checked

To some users, it appears as if the **Select All / Clear All** feature in the comparison grid does not work properly. An object still can appear in the synchronization script even after being unchecked in the comparison grid. This behavior is by design and has to do with the dependencies in the database. To ensure that an object is excluded permanently, the user must uncheck the comparison option **Include dependent objects**.

Apparent error during comparison

Some users may see an error occurring during object pairing before the schema comparison. This issue usually results from a database created with case-sensitive collation and containing objects with the same name but different case, such as **Employee** and **employee**. Note that while this error is triggered, IDERA SQL Schema Compare does continue to compare the databases. However, from the set of objects with the same name, only one of them is compared and the remainder is ignored.

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

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

Recommended IDERA Solutions

IDERA strives to ensure our products provide quality solutions for your database needs. The following IDERA Solutions were recently added to the knowledge base at our [Customer Service Portal](#).

Title
"The trial period has expired," message is displayed when first installing SQL Comparison Toolset on Windows Vista with UAC enabled.
Synchronization script produces errors because objects are created out of order.
SQL Data Compare does not compare all tables and some are listed with the comment "Non-comparable; missing unique key."
Synchronization script references objects which do not exist in the source database.
How can I schedule a comparison operation?

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

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

Welcome to SQL Comparison Toolset

The IDERA SQL Comparison Toolset provides many different tools to diagnose, administer, and analyze SQL Server instances across your enterprise.

When you need to quickly identify and synchronize data and object changes, you can use IDERA SQL Comparison Toolset to:

- Perform SQL Data Compare and SQL Schema Compare tasks and synchronization
- Capture schema snapshots to preserve change history
- Synchronize SQL data automatically via command-line
- Customize your environment using workspaces and saved comparison options

This wiki contains useful information that will help you get started and take full advantage of the best SQL Server database comparison and synchronization tools in the market. The products covered in this wiki include:

- [IDERA SQL Schema Compare](#)
- [IDERA SQL Schema Compare Command Line](#)
- [IDERA SQL Data Compare](#)
- [IDERA SQL Data Compare Command Line](#)

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

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

What is IDERA SQL Comparison Toolset?

Data schema and data comparison is sometimes an extremely time consuming and confusing task when completed manually, and the process can be prone to error. Rather than spending hours digging through and scrutinizing data and the overall database structure, then manually generating scripts, wouldn't it be much easier if there was a tool to automate this process?

IDERA SQL Comparison Toolset provides an easy, automated way to compare schema and data, generate synchronization scripts, and deploy those scripts to synchronize databases. Whether migrating changes from test to production or repairing unwanted changes, IDERA SQL Comparison Toolset saves you time and reduce the headaches of performing manual database comparisons mistakenly applied and unwanted changes.

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

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

How SQL Comparison Toolset Helps

When you need to quickly identify and synchronize data and object changes, you can use IDERA SQL Comparison Toolset to:

- Perform SQL Data Compare and SQL Schema Compare tasks and synchronization
- Capture schema snapshots to preserve change history
- Synchronize SQL data automatically via command-line
- Customize your environment using workspaces and saved comparison options

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

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

Document conventions

IDERA documentation uses consistent conventions to help you identify items throughout the printed online library.

Convention	Specifying
Bold	Window items
<i>Italics</i>	Book and CD titles Variable names New terms
Fixed Font	File and directory names Commands and code examples Text typed by you
Straight brackets, as in [value]	Optional command parameters
Curly braces, as in {value}	Required command parameters
Logical OR, as in value 1 value 2	Exclusively command parameters where only one of the options can be specified

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

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

About IDERA

IDERA is a leading provider of application and server management solutions. We have a wide variety of performance management products for Microsoft SQL Server, and award-winning server backup solutions for both managed service providers and enterprise customers. IDERA products install in minutes and start solving server problems immediately, giving administrators more time, reduced overhead and expenses, and increased server performance and reliability. We are a Microsoft Gold Certified partner, headquartered in Houston, Texas, with offices in Asia Pacific, Australia, New Zealand, Europe, Africa, and Latin America. So we're everywhere your IT needs are.

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

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

IDERA products

Our tools are engineered to scale from managing a single server to enterprise deployments with thousands of servers. IDERA products combine ease of use with a design that installs in minutes, configures in hours, and deploys worldwide in days. To learn more about IDERA products, visit the IDERA Web site at www.idera.com.

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

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

Contact IDERA

Please contact IDERA with your questions and comments. We look forward to hearing from you. For support around the world, please contact us or your local partner.

For a complete list of our partners, please visit our IDERA website.

Sales	713.523.4433 1.877.GO.IDERA (464.3372) (only in the United States and Canada)
Sales Email	sales@idera.com
Support	713.533.5144 1.877.GO.IDERA (464.3372) (only in the United States and Canada) www.idera.com/support
Website	www.idera.com

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

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

Legal notice

IDERA, Inc. ("Idera") makes information and products available on this web site, subject to the following terms and conditions. By accessing this web site, you agree to these terms and conditions. Idera reserves the right to change these terms and conditions, and the products, services, prices, and programs mentioned in this web site at any time, at its sole discretion, without notice. Idera reserves the right to seek all remedies available by law and in equity for any violation of these terms and conditions. THIS WEB SITE COULD INCLUDE TECHNICAL OR OTHER INACCURACIES. CHANGES ARE PERIODICALLY MADE TO THE INFORMATION HEREIN. HOWEVER, IDERA MAKES NO COMMITMENT TO UPDATE MATERIALS ON THIS WEB SITE.

Trademark

IntelliCompress, PowerShellPlus, SharePoint enterprise manager, SharePoint security manager, SharePoint diagnostic manager, SharePoint backup, SharePoint performance monitor, SQL Elements, SQL check, SQL change manager, SQL comparison toolset, SQL compliance manager, SQLcompliance, SQLcm, SQL defrag manager, SQL diagnostic manager, SQLdm, SQL mobile manager, SQLpermissions, SQLsafe, SQLsafe Freeware Edition, SQLsafe Lite, SQL schema manager, SQLsecure, SQLsmarts, SQLstats, SQLtool, SQL toolbox, SQL virtual database, SQLvdb, virtual database, Idera, BBS Technologies and the Idera logo are trademarks or registered trademarks of Idera, Inc., or its subsidiaries in the United States and other jurisdictions. All other company and product names may be trademarks or registered trademarks of their respective companies. Elements of this web site are protected by trade dress or other laws and may not be imitated or reproduced in whole or in part.

Copyright

The information on this web site is protected by copyright. Except as specifically permitted, no portion of this web site may be distributed or reproduced by any means, or in any form, without Idera's prior written consent.

Use of the Software

The software and accompanying documentation available to download from this web site are the copyrighted work of Idera. Use of the software is governed by the terms of the License Agreement, which accompanies such software. If no license accompanies the download, the terms of the license, which accompanied the original product being updated, will govern. You will not be able to use, download, or install any software unless you agree to the terms of such License Agreement.

Use of web site information

Except as otherwise indicated on this web site, you may view, print, copy, and distribute documents on this web site subject to the following terms and conditions:

1. The document may be used solely for informational, personal, non-commercial purposes;
2. Any copy of the document or portion thereof must include all copyright and proprietary notices in the same form and manner as on the original;
3. The document may not be modified in any way; and
4. Idera reserves the right to revoke such authorization at any time, and any such use shall be

discontinued immediately upon notice from Idera.

Documents specified above do not include logos, graphics, sounds or images on this web site or layout or design of this web site, which may be reproduced or distributed only when expressly permitted by Idera.

Warranties and Disclaimers; Liability Limitations

EXCEPT AS EXPRESSLY PROVIDED OTHERWISE IN AN AGREEMENT BETWEEN YOU AND IDERA, ALL INFORMATION AND SOFTWARE ON THIS WEB SITE ARE PROVIDED "AS IS" WITHOUT WARRANTY OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

IDERA ASSUMES NO RESPONSIBILITY FOR ERRORS OR OMISSIONS IN THE INFORMATION OR SOFTWARE OR OTHER DOCUMENTS WHICH ARE REFERENCED BY OR LINKED TO THIS WEB SITE.

IN NO EVENT SHALL IDERA BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER (INCLUDING WITHOUT LIMITATION, THOSE RESULTING FROM: (1) RELIANCE ON THE MATERIALS PRESENTED, (2) COSTS OF REPLACEMENT GOODS, (3) LOSS OF USE, DATA OR PROFITS, (4) DELAYS OR BUSINESS INTERRUPTIONS, (5) AND ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF INFORMATION) WHETHER OR NOT IDERA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

Submissions

With the exception of credit card numbers for the purchase of products and services, Idera does not want to receive confidential or proprietary information through its web site.

Any information sent to Idera, with the exception of credit card numbers, will be deemed NOT CONFIDENTIAL. You grant Idera an unrestricted, irrevocable license to display, use, modify, perform, reproduce, transmit, and distribute any information you send Idera, for any and all commercial and non-commercial purposes.

You also agree that Idera is free to use any ideas, concepts, or techniques that you send Idera for any purpose, including, but not limited to, developing, manufacturing, and marketing products that incorporate such ideas, concepts, or techniques.

Idera may, but is not obligated to, review or monitor areas on its web site where users may transmit or post communications, including bulletin boards, chat rooms, and user forums. Idera is not responsible for the accuracy of any information, data, opinions, advice, or statements transmitted or posted on bulletin boards, chat rooms, and user forums.

You are prohibited from posting or transmitting to or from this web site any libelous, obscene, defamatory, pornographic, or other materials that would violate any laws. However, if such communications do occur, Idera will have no liability related to the content of any such communications.

Some companies, names, and data in this document are used for illustration purposes and may not represent real companies, individuals, or data.

Governing Law and Jurisdiction

You agree that all matters relating to your access to, or use of, this web site shall be governed by the laws of the state of Texas. You agree and hereby submit to the exclusive personal jurisdiction and venue of the state courts of Texas located in Harris County, Texas, and the United States District Court for the Southern District of Texas, with respect to such matters.

Idera makes no representation that information on this web site are appropriate or available for use in all countries, and prohibits accessing materials from territories where contents are illegal. Those who access this site do so on their own initiative and are responsible for compliance with all applicable laws.

Export Control Laws

Certain Idera products, including software, documentation, services, and related technical data, available on the Idera and other web sites are subject to export controls administered by the United States (including, but not limited to, the U.S. Department of Commerce Export Administration Regulations ("EAR")) and other countries including, controls for re-export under European Union, the Singapore Strategic Goods Control Act, and the import regulations of other countries. Diversion contrary to U.S. or other applicable law of any Idera product or service is prohibited. Export, re-export or import of products and services may require action on your behalf prior to purchase and it is your responsibility to comply with all applicable international, national, state, regional and local laws, and regulations, including any import and use restrictions. Idera products and services are currently prohibited for export or re-export to Cuba, Iran, North Korea, Sudan, Syria, or to any country then subject to U.S. trade sanctions. Idera products and services are prohibited for export or re-export to any person or entity named on the U.S. Department of Commerce Denied Persons, Entities and Unverified Lists, the U.S. Department of State's Debarred List, or the U.S. Department of Treasury's lists of Specially Designated Nationals, Specially Designated Narcotics Traffickers, or Specially Designated Terrorists. Idera products and services are prohibited from use with chemical or biological weapons, sensitive nuclear end-users, or missiles, drones or space launch vehicles capable of delivering such weapons. By downloading or using any product from this web site, or purchasing any service, you are acknowledging that you have read and understood this notice and agree to comply with all applicable export control laws. You are also representing that you are not under the control of, located in, or a resident or national of any prohibited country, and are not a prohibited person or entity. This notice is not intended to be a comprehensive summary of the export laws that govern the products and services. It is your responsibility to consult with a legal advisor to ensure compliance with applicable laws.

United States Government Rights

All Idera products and publications are commercial in nature. The software, publications, and software documentation available on this web site are "Commercial Items," as that term is defined in 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are defined in 48 C.F.R. §252.227-7014(a)(5) and 48 C.F.R. §252.227-7014(a)(1), and used in 48 C.F.R. §12.212 and 48 C.F.R. 227.7202, as applicable. Pursuant to 48 C.F.R. §12.212, 48 C.F.R. §252.227-7015, 48

C.F.R. §227.7202 through 227.7202-4, 48 C.F.R. §52.227-19, and other relevant sections of the Code of Federal Regulations, as applicable, Idera's publications, commercial computer software, and commercial computer software documentation are distributed and licensed to United States Government end users with only those rights as granted to all other end users, according to the terms and conditions contained in the license agreements that accompany the products and software documentation, and the terms and conditions herein.

© 2003-2017 Idera, Inc., all rights reserved.

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

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

Getting started

When you download the IDERA SQL Comparison Toolset from our web site, you receive an .msi file that you can run to install the IDERA SQL Comparison Toolset package. The installation set up adds the IDERA file group under the All Programs menu as well as places shortcuts of the IDERA SQL Schema Compare and IDERA SQL Data Compare on the desktop.

For two weeks from the installation date the products are fully functional and have no restrictions. After the trial period expires, you must acquire a license. You can purchase a license from our website at IDERA.com.

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

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

Requirements

Minimum requirements for running IDERA SQL Comparison Toolset are:

- .NET Framework 4, available from the [Microsoft Developer Tools Download Center](#)
- at least 50MB of free space on the hard drive where the application will be installed
- Windows XP or later / higher versions of Windows operating systems
- minimum of 512MB of memory; 1GB or higher recommended
- minimum resolution of 1024x768 suggested for the GUI
- SQL Server 2005, 2008, 2008 R2, 2012, 2014, 2016

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

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

Supported platforms

IDERA SQL Schema Compare supports all editions of the following SQL Server versions:

- SQL Server 2016
- SQL Server 2014
- SQL Server 2012
- SQL Server 2008/2008R2
- SQL Server 2005

IDERA SQL Data Compare supports all editions of the following SQL Server versions:

- SQL Server 2016
- SQL Server 2014
- SQL Server 2012
- SQL Server 2008/2008R2
- SQL Server 2005
- SQL Server 2000

IDERA SQL Schema and Data Compare supports the following Azure versions:

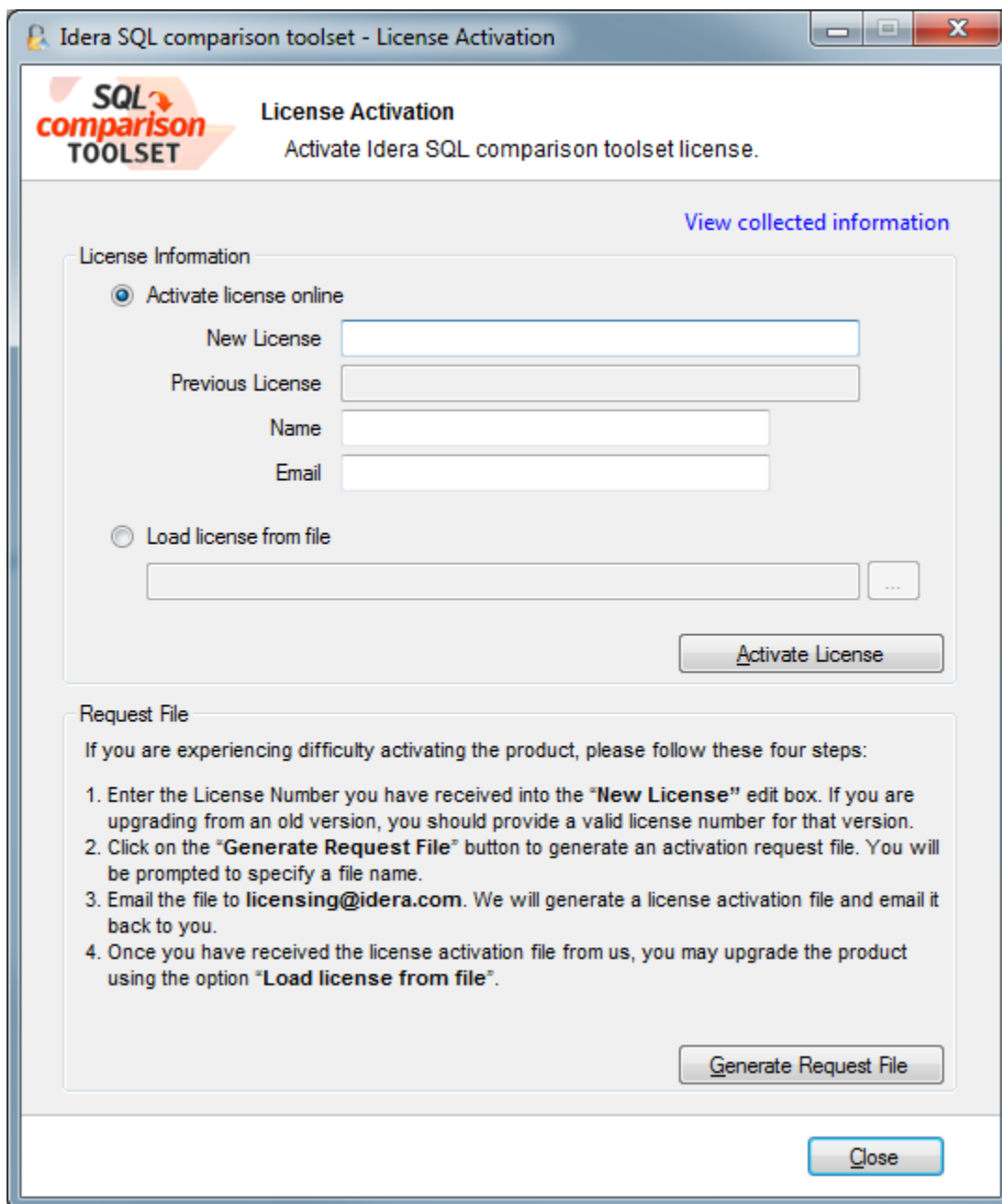
- Azure v12
- Azure v11

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

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

Activating a license

If you have acquired a license, you can quickly activate. Go to **Start > All Programs > IDERA > SQL Comparison Toolset > License Activation**. In the License Activation window, type or paste the license number you received, and then click **Activate License**. The license activate utility will connect to a web service running on our web servers that will validate the license you provided. Within a few seconds you should see a confirmation message indicating that the product was successfully upgraded.



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

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

Updates and support

We update our products periodically but will only notify our customers via email when we publish an update that contains a critical fix or that is considered a major upgrade. Therefore, to ensure that you are always running the latest version of our products, we recommend that you check our site, www.idera.com, regularly for product updates.

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

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

IDERA SQL Schema Compare

Welcome to IDERA SQL Schema Compare, the tool of choice for thousands of SQL Server DBAs and developers around the globe. Identifying schema differences between two SQL Server databases and synchronizing those schemas is easy, fast, and safe with IDERA SQL Schema Compare.

This release IDERA SQL Schema Compare is built on top of a new, robust, efficient and scalable comparison framework. It supports all versions of on-premise SQL Server from SQL Server 2005 to SQL Server 2016, as well as Azure v11 and v12.

Please direct any questions, comments and suggestions about the IDERA SQL Schema Compare to support@idera.com.

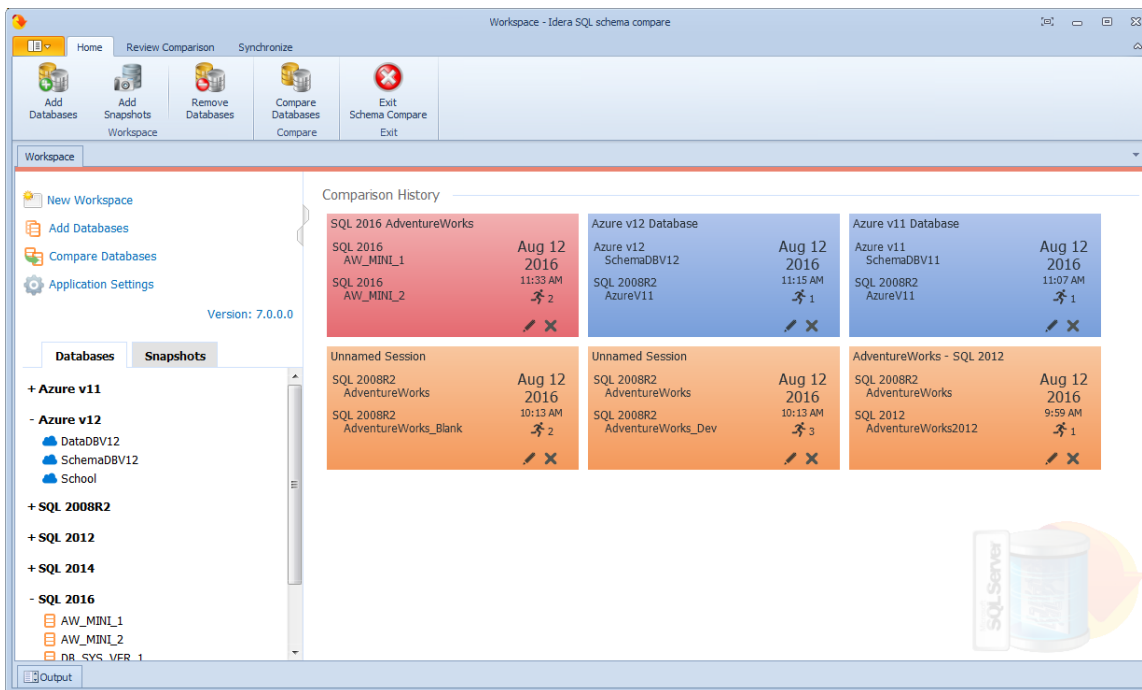
Total compare from data to schema with IDERA [SQL Comparison Toolset](#). [Learn more >>](#)

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

SQL Schema Compare Application window

IDERA SQL Schema Compare divides the main application window into three areas:

1. **The ribbon.** The context sensitive ribbon provides all the command buttons and icons you need to do your job quickly. The ribbon is organized in three distinct tabs each of which corresponding to the main steps of the database schema comparison and synchronization process and containing the relevant command buttons.
2. **Main panel.** The main panel contains a permanent tab called the **Workspace** which you cannot close, and a tab for each active comparison session.
3. **Output window.** The output window provides a running log of the tasks being performed in IDERA SQL Schema Compare.



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

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

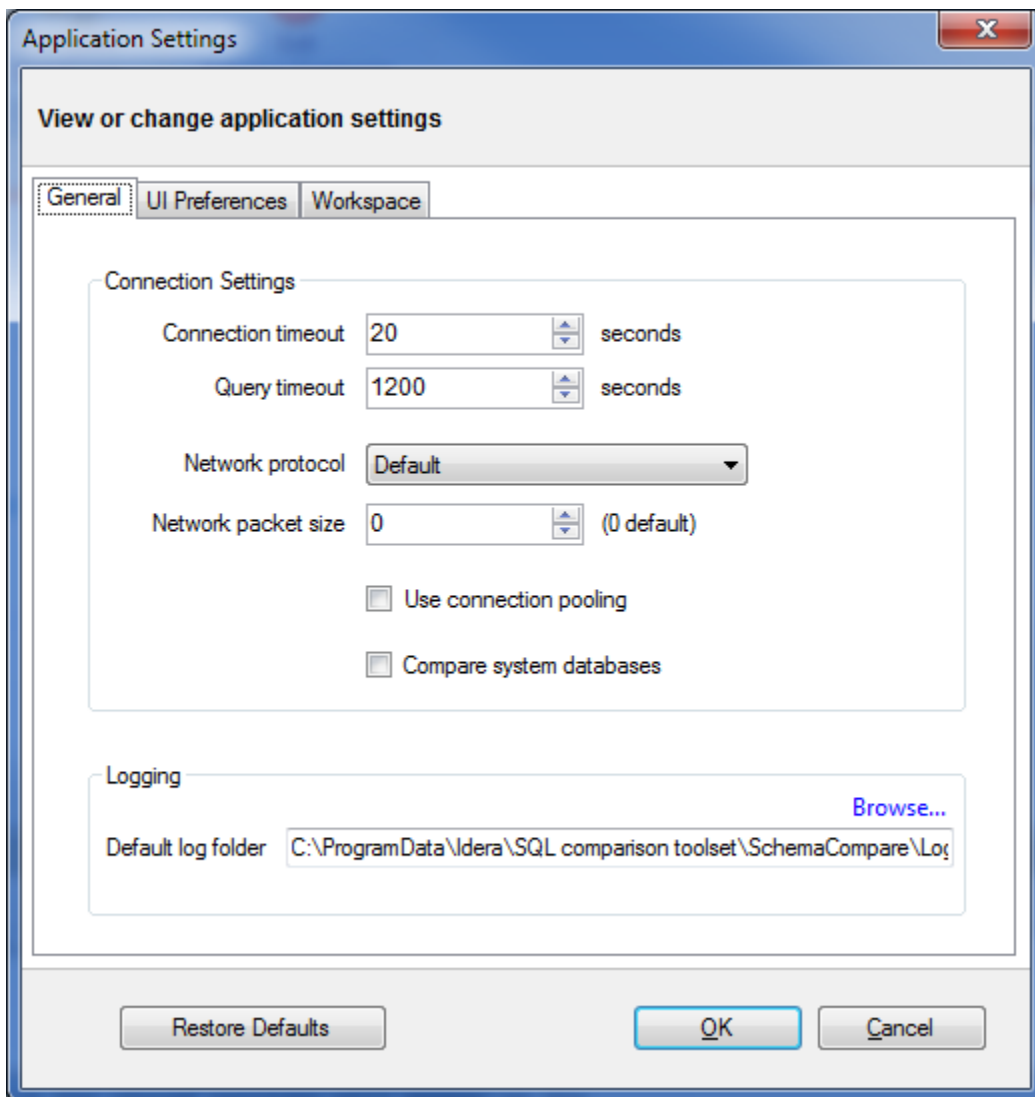
SQL Schema Compare Application Settings

Application settings allow you to tweak the behavior of IDERA SQL Schema Compare. It can be accessed from the drop-down menu off of the top-left corner product icon or from the link at the bottom-left corner on the Workspace tab.

Application level settings are grouped into the General, UI Preferences, and Workspace tabs.

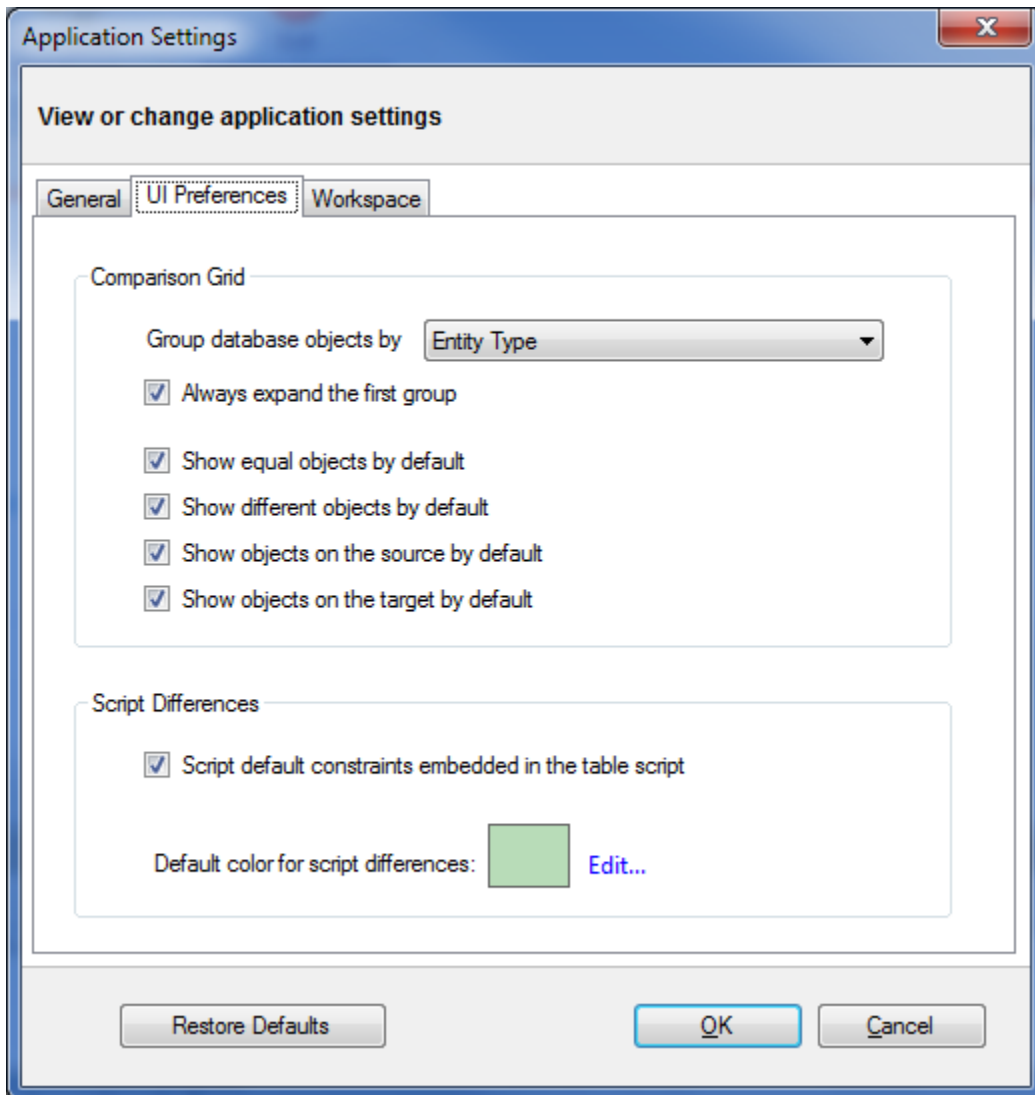
General

- **Connection Settings.** Generally, there is no need to change the connection settings, however, in certain scenarios you might need to adjust those values. From the application settings window you can set the **Connection Timeout**, **Query Timeout**, **Network Protocol**, and **Network Packet Size**.
- **Log Setting.** Indicates the location where the IDERA SQL Schema Compare log files are saved.



UI Preferences

- **Comparison Grid.** Allow you to adjust the behavior of the comparison results grid based on your preferences.
- **Script Differences.** Allow you to change the highlight color for the script differences as well as determine how the default constraints are scripted. The way the default constraints are scripted has no bearing on the comparison results or on the synchronization script, the only difference is on where you see those constraints scripted. By embedding the default constraints in the table script it is easier to see why two columns might be marked as different. If the default constraints are scripted separately then you might see two seemingly identical columns marked as different and you will need to scroll down below the table script to see what is different.



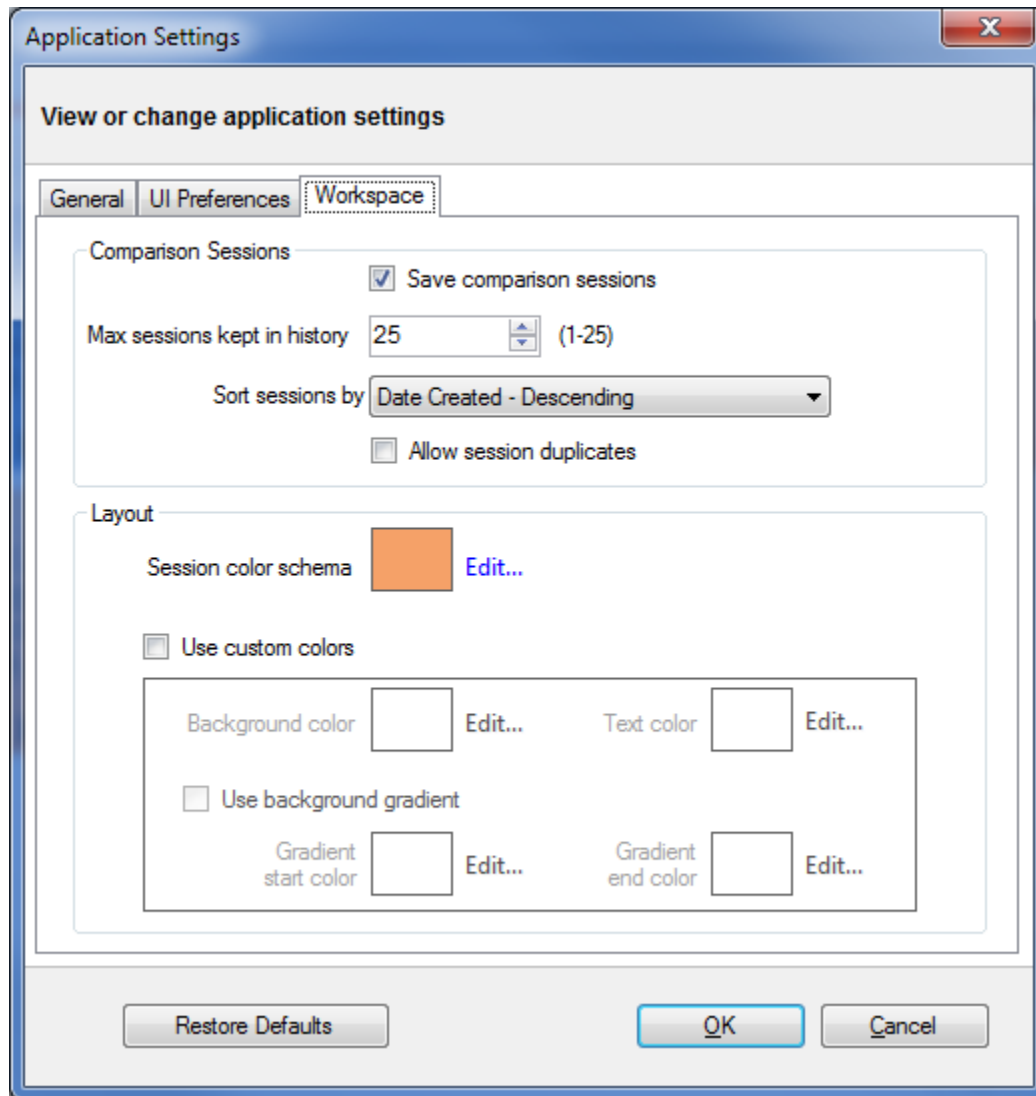
Workspace

Contains settings related to the workspace and comparison sessions.

- **Sessions**
 - **Save sessions.** By default is checked. It instructs the IDERA SQL Schema Compare to save comparison sessions for future use. A saved session contains all the necessary information so that you can rerun the comparison with one click. Saved

sessions appear on the main panel of the Workspace tab.

- **Max sessions kept in history.** By default IDERA SQL Schema Compare keeps the last 25 sessions. Although there is no noticeable performance difference on application launch related to the number of sessions stored you may choose to set this parameter to a lower number. Allowable values are 1 to 25.
- **Sort sessions by:** change the order in which the stored sessions appear in the Workspace tab.
- **Layout**
 - Allows you to custom session coloring. You can select from built-in color schemas or pick custom colors.



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

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

SQL Schema Compare Workspace

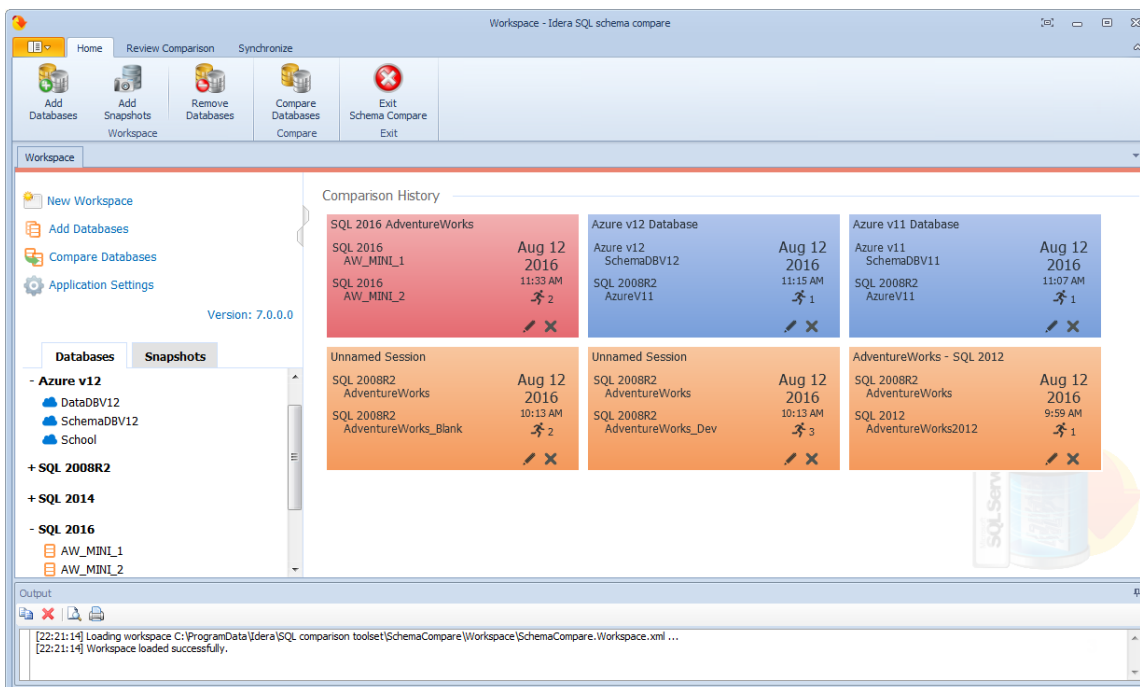
IDERA SQL Schema Compare Workspace is the storage and the interface for saving and managing SQL Server databases, credentials, comparison sessions and other UI options.

The Workspace window is divided into four sections:

1. **Workspace actions.** Provides links that allow you to create a new workspace, add databases to the current workspace, compare schema, or change application settings. The **New Workspace** action resets the current workspace by erasing the comparison history and removing all registered databases.
2. **Databases/Snapshots.** Contains the list of databases and snapshots that have been registered in the workspace. On mouse over two action links appear for a registered server:
 - **Edit.** Allows you to change the connection settings for that server and/or add and remove databases.
 - **Remove.** Un-registers a server (deletes all the server information from the workspace).

When you move your cursor over a database name, the following action links related to that database appear:

 - **Take a Snapshot.** Allows you to take a snapshot of the schema of that database.
 - **Remove Database.** Removes the database from the workspace.
3. **Comparison History.** Displays up to 25 comparison sessions from the most recent one to the oldest one. You can rerun a session by clicking on it.
4. **Output.** Displays each workspace event as it occurs.

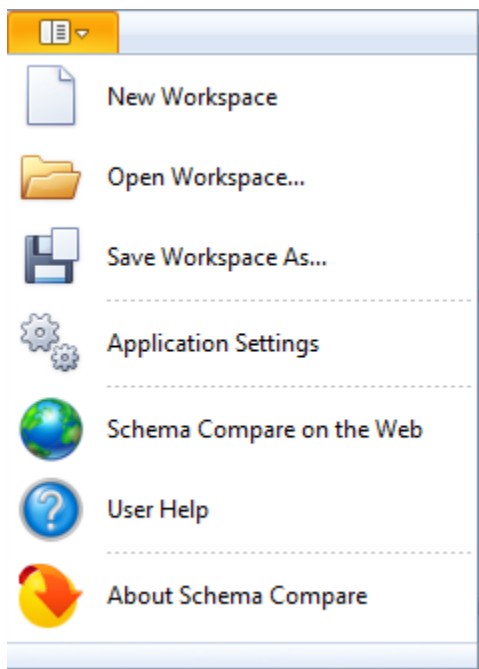


Saving and loading workspaces

SQL Schema Compare Workspace stores the data in a set of xml files. There is always one master file that keeps SQL Server databases and credentials, and a separate xml file for each

comparison session. The workspace is saved by default in a subfolder under the common folder for application data that is used by all users. It is usually found under \ProgramData\Idera\SQL comparison toolset\SchemaCompare\Workspace.

The workspace is managed automatically, so no action is required to save it or load it. However, if you wish to have multiple copies of the workspace, you can save it to a custom path. Functions for saving and loading the workspace are located in the quick menu of the application button:



i When you choose to save the workspace to a custom location, you specify the path and the name of the workspace master file. The session files are saved in the same directory as the master file, but their names will be auto-generated by SQL Schema Compare. Similarly, when you load a workspace from a custom location, you select the workspace master file. SQL Schema Compare expects the session files to be in the same folder as the master file.

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

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

SQL Schema Compare Add Databases

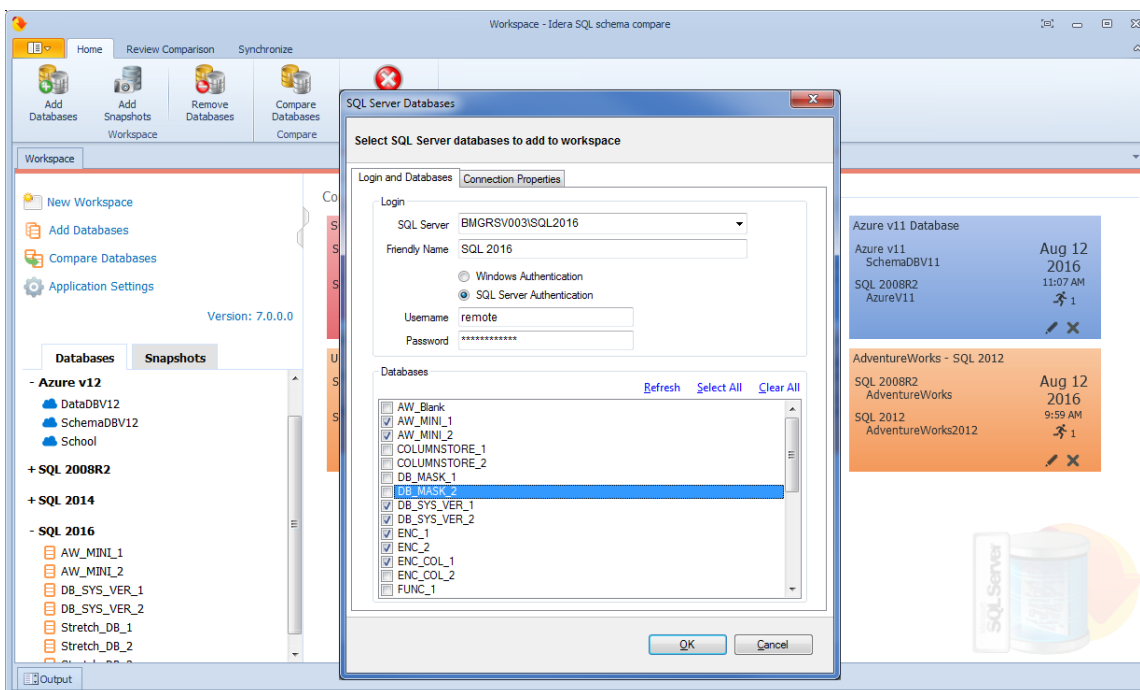
Before comparing databases, you will need to add them to the workspace. The **Add Databases** functionality can be accessed from the ribbon (the **Home** tab) or from the **Add Databases** action link at the top left section of the Workspace tab.

On the Add Database dialog window, choose the server where you want the new database(s) to reside. You can click on the **Refresh** item of the SQL Server drop down to discover the SQL Server instances running on your network. If the application is not able to discover the SQL Server instance you are looking for because it is not exposing itself or it is not running on your local area network, or it is not listening on the default port, then you can type in [ServerName or IP Address]\InstanceName on the drop-down box. If the SQL Server is not listening on the default port you can also specify a custom port on the following format: [ServerName or IP Address]\[Instance Name],[PortNumber]

Once you have chosen the server and the authentication method (including login name and password if you chose SQL Server Authentication), you can click **Refresh** at the top of the Databases box at which point the application will connect to the server using the information you provided and populate the Databases box with the list of all user databases in that server. You can select one or more databases to add to the workspace.

By default, IDERA SQL Schema Compare will use the connection properties set at the Application Settings, however, you can change the connection properties for this server only if you need to.

In the Other Properties edit box, you can specify additional connection properties as name=value pairs separated by semicolons: name1=value1;name2=value2;...



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

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

SQL Schema Compare Add Snapshots

A schema snapshot is a compact file that contains all the schema information for a database. The schema information is stored in a proprietary format and can only be read by IDERA SQL Schema Compare. The schema snapshot does not contain data, only schema information.

Schema snapshots allow you to maintain a history of schema changes. You can take schema snapshots at different points in time and then you can compare those snapshots to each other or compare the snapshots to the live database to get a clear and accurate picture of the "evolution" of the database schema.

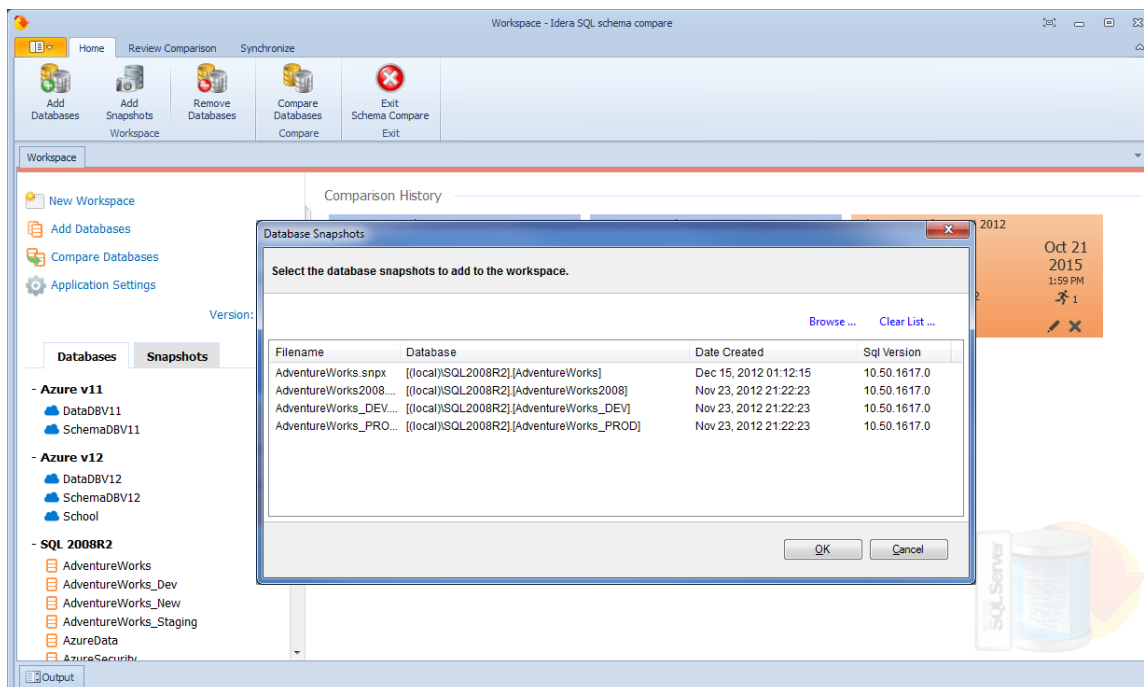
Take a snapshot

To take a schema snapshot, simply move the mouse over the name of the database on the left panel of the Workspace and you will see a "camera" icon appear on the side of the database name. Click on that icon to take the snapshot - you will be asked to choose the name of the snapshot and the location where you wish to save it.

Adding snapshots to the workspace

Before comparing snapshots to each other or to live databases you will need to add the snapshots you need to the workspace. The **Add Snapshots** functionality can be accessed from the ribbon (the Home tab).

On the Add Snapshots dialog window that appears, click **Browse...** to browse to the location where the snapshot(s) you wish to add to the workspace have been stored. You can select as many snapshots as you want from different locations. For each snapshot you select the application will display the file name, the database the snapshot was taken from, the date when the snapshot was taken, and SQL Server version.



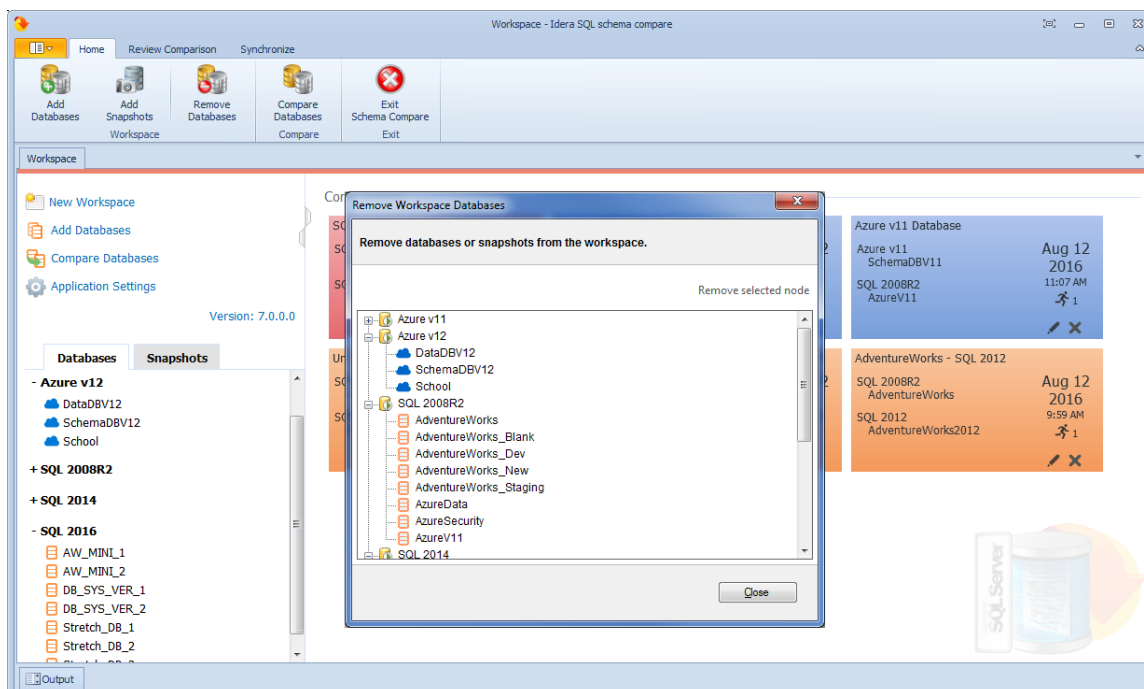
Total compare from data to schema with IDERA [SQL Comparison Toolset](#). [Learn more >>](#)

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

SQL Schema Compare Remove Databases

There are two ways to remove a database or a schema snapshot from the workspace:

- Move the mouse over the name of database or the schema snapshot you wish to remove under the Registered Databases on the left panel of the Workspace. The database/snapshot name will be highlighted and a **Remove** icon/link will become visible. Simply click on that icon and confirm the removal when prompted. Note, that in the same way you can remove a registered server and all its databases from the workspace.
- Click **Remove Databases** on the Home tab of the ribbon. A remove databases dialog window appears showing all the registered servers/databases - select the server or database you wish to remove and click on the **Remove Selected SQL Server / Database...** link at the top right corner and then confirm the removal when prompted.



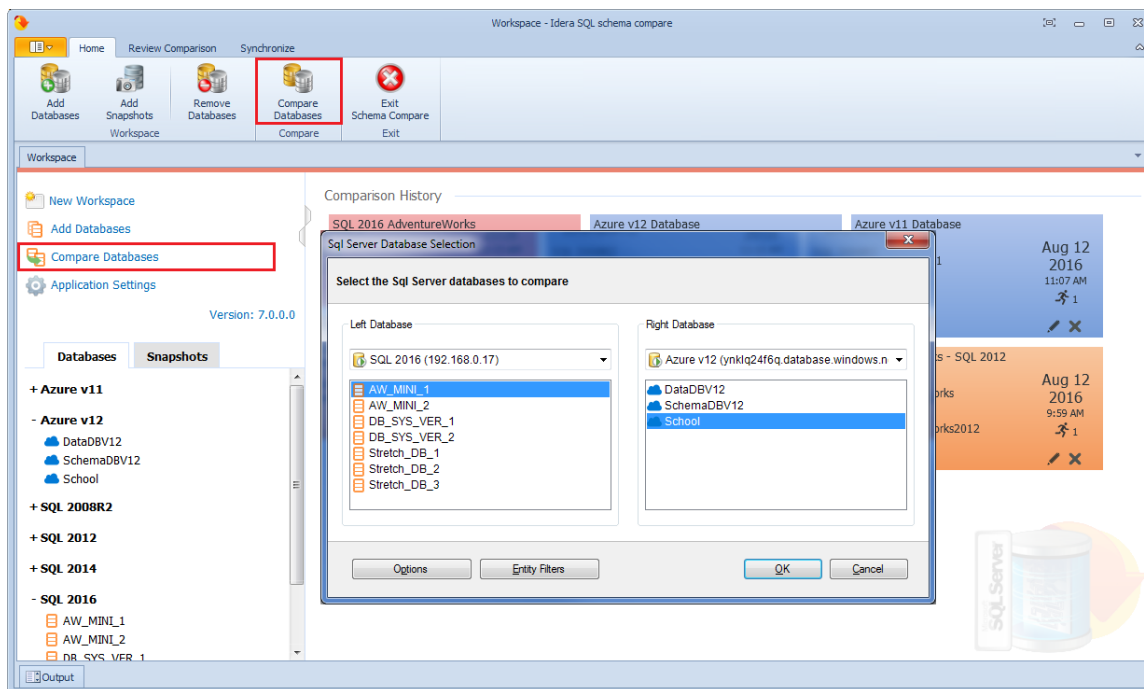
Total compare from data to schema with IDERA [SQL Comparison Toolset](#). [Learn more >>](#)

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

SQL Schema Compare Comparing Schemas

To use SQL Schema Compare to start a comparison between two databases, click **Compare Databases** on the Home tab of the ribbon. The SQL Server Database Selection dialog window appears and allows you to choose the server(s) where the databases you wish to compare reside as well as the databases themselves.

From this window, you can also access the Comparison Options that will be applied during the comparison.



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

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

SQL Schema Compare Object Types

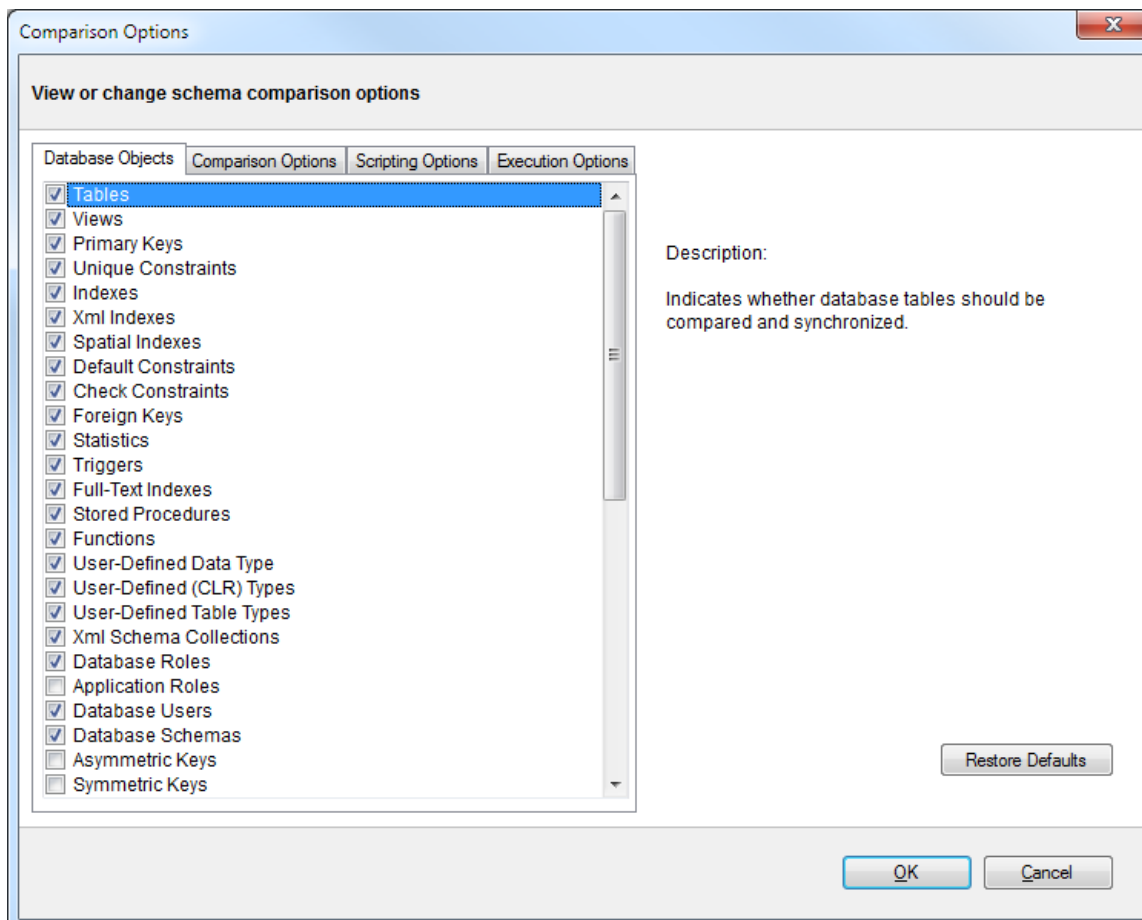
IDERA SQL Schema Compare supports over 45 different types of database objects. By default, all but a handful of object types will be included in the comparison and synchronization. The only categories that are excluded by default are:

- Asymmetric Keys
- Symmetric Keys
- Certificates
- Broker Priorities

In the majority of the cases, you should not need to change the default settings. There are scenarios in which you might wish to exclude certain categories that are included by default or include those that are excluded by default.

To exclude or include certain object types, click **Comparison Options**. The first tab on the Comparison Options window that appears is the Database Objects tab which provides a brief description for each object type and allows you to select or clear the object types you want.

! Because of the dependencies between database objects, excluding all objects of a given type may result on a synchronization script that will fail to execute due to invalid dependencies. For more information about the handling of dependencies, see [SQL Schema Compare Comparison Options](#).



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

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

SQL Schema Compare Options

IDERA SQL Schema Compare exposes over 60 different options that provide the user with highly granular control of the behavior of the comparison and synchronization engine.

The defaults have been carefully selected to match the most common scenarios so that in the majority of the cases you don't even need to look at those options at all, however, every SQL Server implementation is different and there is no "one size fits all" scenario.

For clarity and easy access, the options have been organized on three different tabs:

- [Comparison Options](#)
- [Scripting Options](#)
- [Execution Options](#)

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

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

SQL Schema Compare Comparison Options

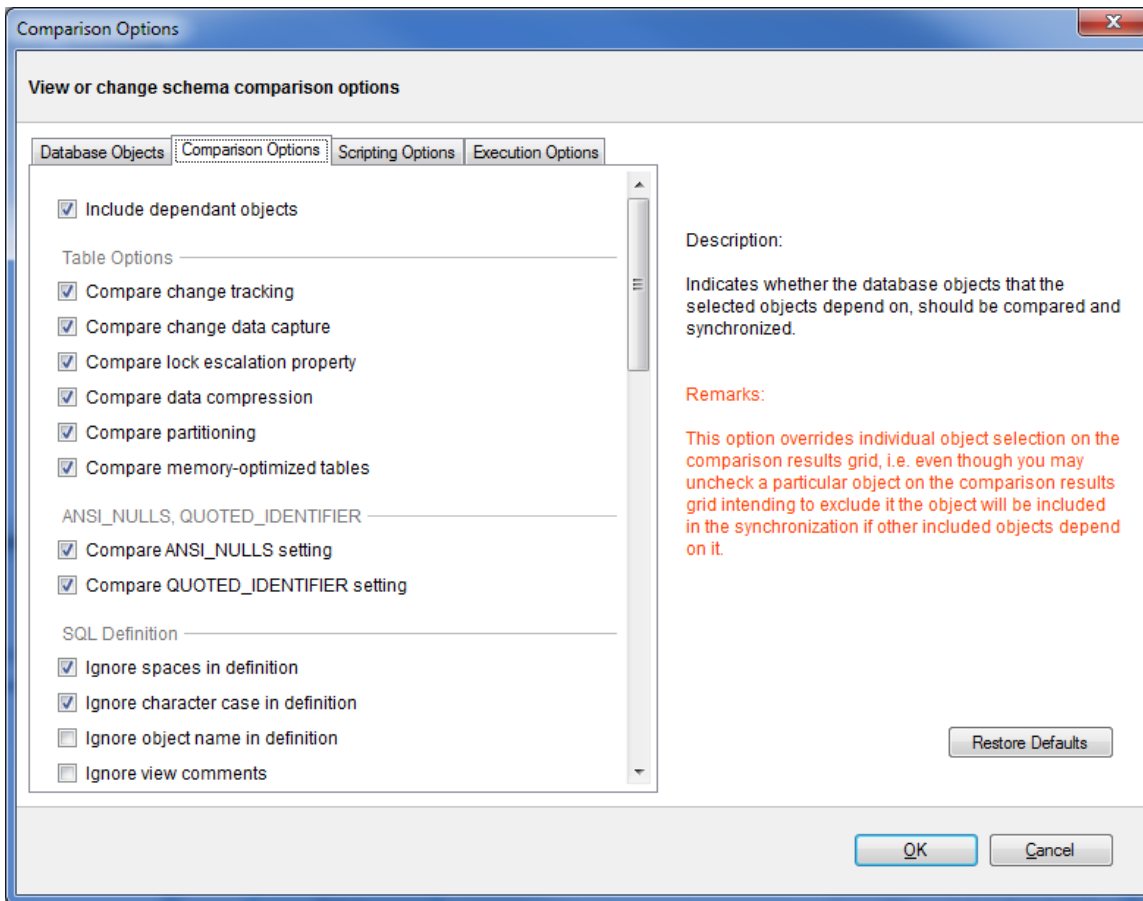
When are two database objects equal? The answer is: it depends! It depends on what you want to compare those objects on - for example: is "Equal" equal to "equal"? The comparison options allow you to have complete control over the comparison engine which considers those options before deciding to mark a pair of objects as equal or different.

So, when two object are found to be different by the comparison engine while you believe they should be equal, or they are found to be equal while you believe they should be different don't panic - investigate the differences and carefully review the comparison options - more likely than not one of those options holds the answer to your surprise.

The comparison options tab provides a description and important remarks for each option while also allowing you to turn those options ON or OFF. To view the description and remarks for a given option simply move the mouse over that option as shown on the image below.

Most of the comparison options are self-explanatory, and for those that are not the description and remarks provided on the "comparison options" window are sufficient. However, there are two options that merit special attention:

- **Script New Columns As NULL.** In cases when the synchronization of the schema requires adding a NOT NULL column on the target but, no default constraint has been specified then SQL Server will reject the action and the synchronization will fail. In such scenarios the best course of action would be to add a default constraint on the source, however, if for one reason or another you cannot add the constraint on the source you can add the column to the target as a "NULL ALLOWED" column to ensure the schema synchronization operation completes successfully. After the synchronization you can connect to the target and handle the changing of that particular column from NULL ALLOWED to NOT NULL.
- **Script New Constraints WITH NOCHECK.** When the schema synchronization requires adding new constraints like foreign key constraints and check constraints on tables that contain data there is a it is possible that the existing data violates those constraints in which case SQL Server will reject the action and the synchronization script will fail to execute. To bypass this obstacle and allow the synchronization operation to go through you can choose to add those constraints WITH NO CHECK first and then check to see if the existing data violates the constraint. When this option is ON then Idera SQL schema compare will generate two synchronization scripts that are executed one after the other:
 - **Synchronization Script.** This will create the new constraints with NOCHECK so the script will execute successfully.
 - **Constraint Enabling Script.** This script enables the constraints that were created WITH NOCHECK and runs on a separate transaction. If the existing data violates one or more of those constraints then the script will fail but that failure does not affect the overall synchronization of the schemas. In case of failure you will need to address the constraint violations on case by case bases.



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

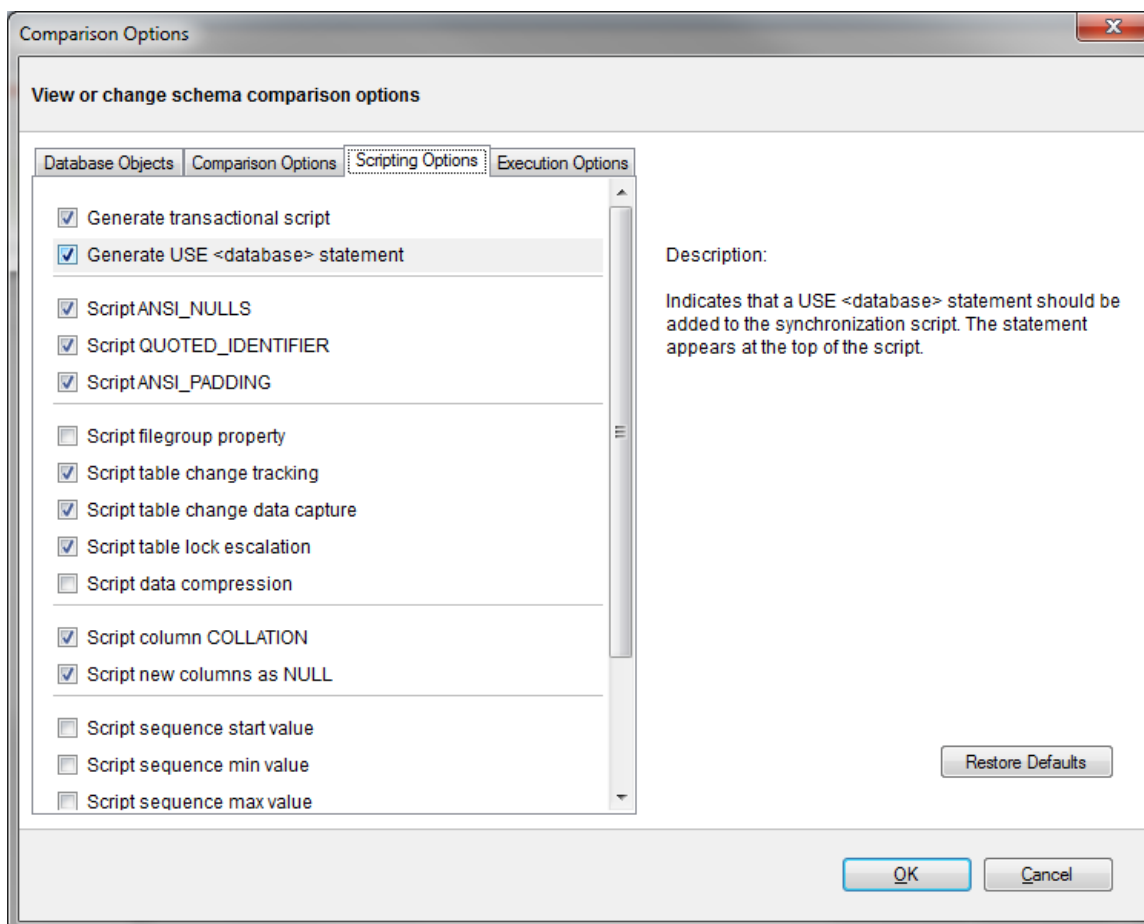
IDERA Website	Products	Purchase	Support	Community	About Us	Resources	Legal
-------------------------------	--------------------------	--------------------------	-------------------------	---------------------------	--------------------------	---------------------------	-----------------------

SQL Schema Compare Scripting Options

IDERA SQL Schema Compare exposes a wide variety of options to let you tweak the way the comparison engine works. From ignoring white spaces to performing automated file group translation you have control over it all. The default options have been carefully selected to match the most common scenarios however, every SQL Server implementation is different and there is no "one size fits all" scenario.

When an object is found to be different by the comparison engine while you believe it should be equal, or the other way around, don't panic - investigate the differences and carefully review those options - more likely than not one of those options holds the answer to your surprise.

Rule of thumb: whenever you are not sure what options would be better for your scenario, click **Default Options**.



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

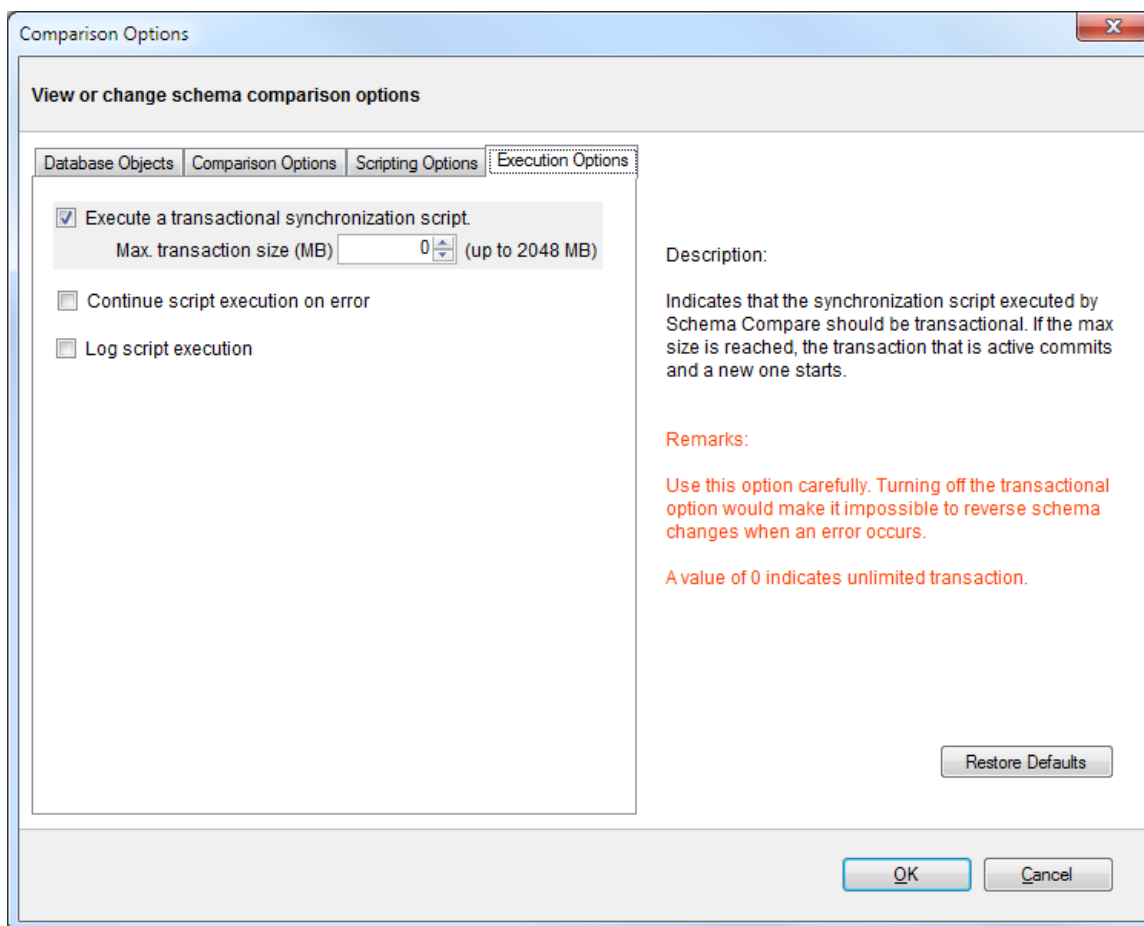
IDERA Website	Products	Purchase	Support	Community	About Us	Resources	Legal
-------------------------------	--------------------------	--------------------------	-------------------------	---------------------------	--------------------------	---------------------------	-----------------------

SQL Schema Compare Execution Options

IDERA SQL Schema Compare exposes a wide variety of options to let you tweak the way the comparison engine works. From ignoring white spaces to performing automated file group translation you have control over it all. The default options have been carefully selected to match the most common scenarios however, every SQL Server implementation is different and there is no "one size fits all" scenario.

When an object is found to be different by the comparison engine while you believe it should be equal, or the other way around, don't panic - investigate the differences and carefully review those options - more likely than not one of those options holds the answer to your surprise.

Rule of thumb: whenever you are not sure what options would be better for your scenario, click **Default Options**.



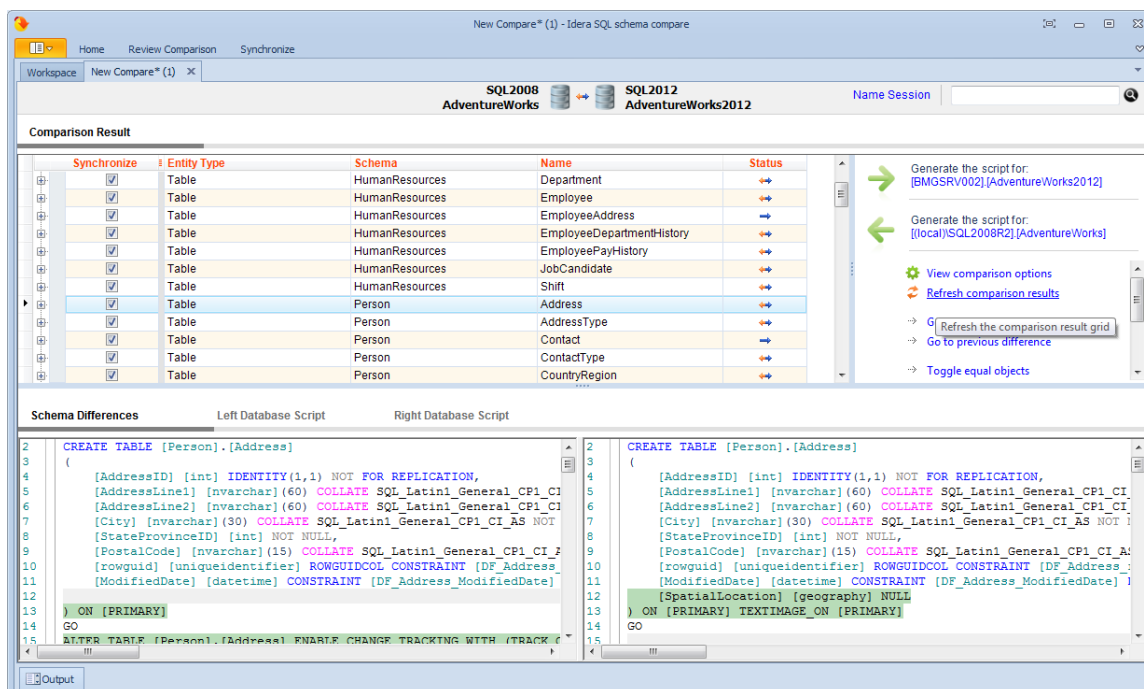
Total compare from data to schema with IDERA [SQL Comparison Toolset](#). [Learn more >>](#)

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

SQL Schema Compare Comparison Results

Upon completion of the schema compare operation, the results of the comparison are displayed on a new comparison session tab. The Comparison Results window is divided into three main sections:

1. **Comparison summary.** The main section displays the list of all objects that were compared, grouped by object type, indicating for each object the result of the comparison (equal, different, exists only on one of the databases). By default, all the objects are displayed including the ones that are identical, however, a handy context menu allows you to filter objects based on object type or/and based on the result of the comparison.
2. **Differences.** The bottom section of the results window contains three tabs.
 - The Differences tab displays, side by side, the scripts of the object pair that has been selected in the top section of the results grid in both of the databases being compared. The differences are highlighted using a simple color coding scheme.
 - The Left Database Script contains the t-sql script that you would need to execute on the left database to make the selected object on the left side the same as the corresponding object on the right side.
 - The Right Database Script contains the t-sql script that you would need to execute on the right database to make the selected object on the right side the same as the corresponding object on the left side.
3. **Action.** The right panel provides quick access to action buttons/links that allow you to generate synchronization scripts, view/change comparison options, refresh comparison results etc.



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

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

SQL Schema Compare Equal but Different

Making sense of the comparison results

Columns are marked as different but they seem to be identical! In some cases you may see a table pair marked as different and when you look at the "Schema Differences" tab on the bottom you may notice that one or more columns have been highlighted as being different but there is no visible difference on the script! In these cases you will need to dig deeper to find out why those columns are marked as different. To do that expand the table on the top section and then click on the column you are interested on - on the bottom section of the screen you will see the script for that particular column with the different section highlighted. This difference will be visible on the table script as well however it is a lot easier to "spot" on the column script.

Differences on default constraints, check constraints, default bindings, rule bindings, or extended properties can cause the columns to be flagged as different.

Columns are different but are not marked as such! The highlighting of the differences on the script of an object is controlled by the comparison engine so if the comparison engine marks two objects as equal then nothing will be highlighted on the scripts of those objects even though there might be visible differences on the script. Why does this happen? The answer lies in the comparison options - there are a few options related to collation, nullability, replication, identity etc. that affect the column comparison. So if you have for example chosen to ignore the column collation then even though the column on the left might have a different collation from that on the right the columns will be marked as equal and therefore the collation difference on the script will not be highlighted.

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

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

SQL Schema Compare Refreshing Results

When you make changes to comparison options after you have compared two databases the comparison results might no longer reflect your choices. For example, when you compared the databases, the option **Ignore spaces on definition** was ON so let's say a view named VW1 is marked as being equal on both sides. Now you go to options and uncheck the **Ignore spaces on definition** at which point the VW1 is technically different since let's say its definition on the left database contains a few extra spaces compared to its definition on the right database. Any time you make such changes after the comparison IDERA SQL Schema Compare considers the effect of the changes on the comparison results and if necessary prompts you to refresh the results. The refreshing of the results is accomplished by either a partial or a full re-compare of the schemas.

You can also refresh the comparison results on demand by clicking **Refresh Results** on the ribbon, or by clicking **Refresh Comparison Grid** on the action panel on the lower-right hand side of the comparison results window.

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

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

SQL Schema Compare Working with Filegroups

The following topics describe the filegroup options in detail:

- [Comparing and Synchronizing Filegroups and Data Files](#)
- [Working with Filegroup Mapping Rules](#)

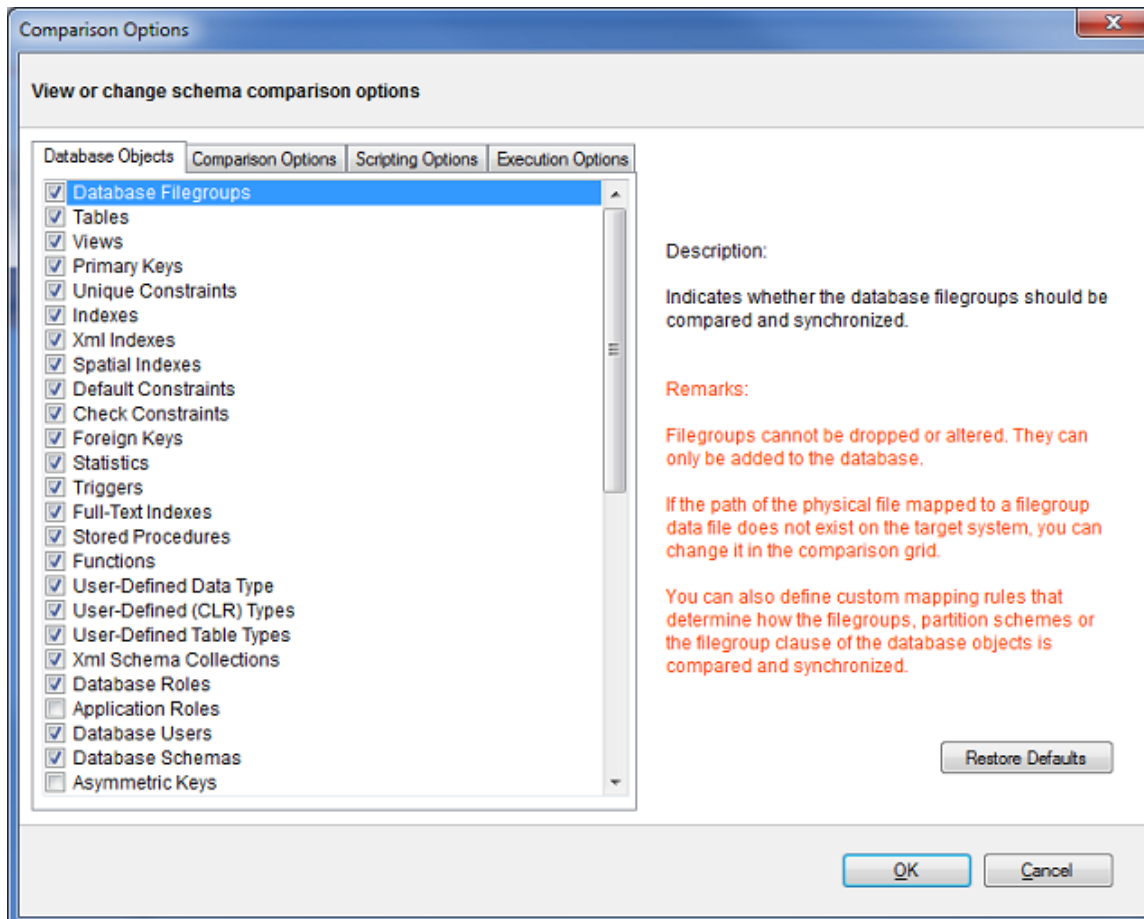
Total compare from data to schema with IDERA [SQL Comparison Toolset](#). [Learn more >>](#)

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

SQL Schema Compare Comparing Filegroups

Beginning with version 4, IDERA SQL Schema Compare allows you to compare and synchronize database filegroups including the filegroup data files. Database log files are not supported.

Comparing filegroups is similar to other database objects. You start by selecting the **Database Filegroups** option in the Comparison Options window (by default filegroups are not selected):

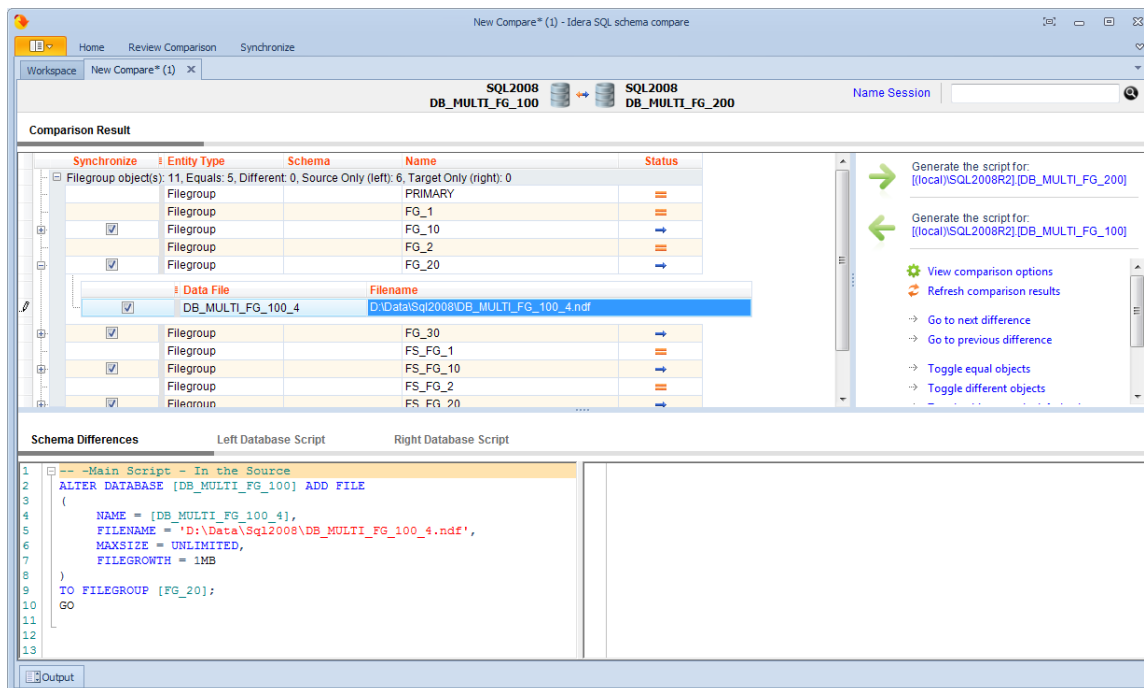


When the filegroup option is selected, a filegroup category appears in the comparison grid. The plus sign adjacent to the filegroup allows you to view the data files associated with the filegroup. Depending on the type of the filegroup, the data file is mapped to one of the following:

- A file, usually with the .mdf or .ndf extension, for a data filegroup.
- A path for a filestream or a filegroup that contains memory-optimized tables.

When you choose to synchronize the database filegroups, it is important to remember that the path of the physical file mapped to the data files must exist on the target system. If this is not the case, you can change the path and filename of the data file in the comparison grid as follows:

- Expand the filegroup that you wish to change
- In the data file section of the comparison grid, enter the new filename in the Filename column



You can also exclude filegroups or specific data files, similar to other database objects. Excluded items will not be synchronized.

The following restrictions apply to filegroup synchronization:

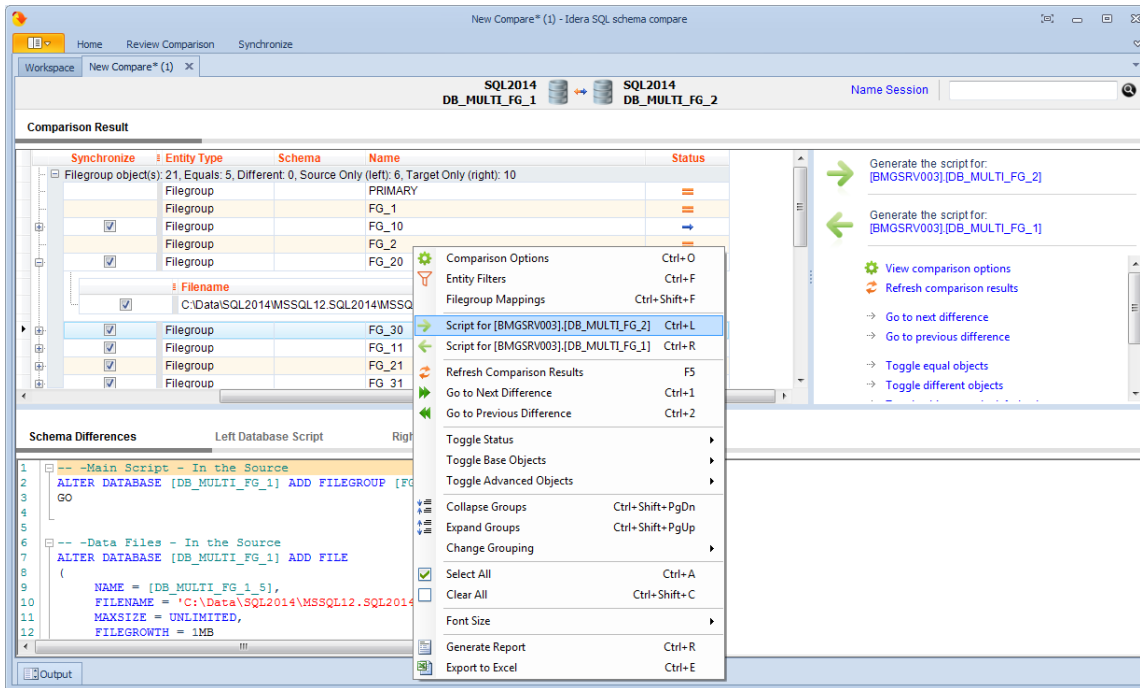
- Filegroups can only be added to the database. They cannot be altered or removed.
- Data files are only added to the new filegroups. Data files on existing filegroups are not modified.
- SQL Server does not support transactional execution of DDL statements for filegroups or data files, which means that changes to the filegroup structure cannot be rolled back.

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

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

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:

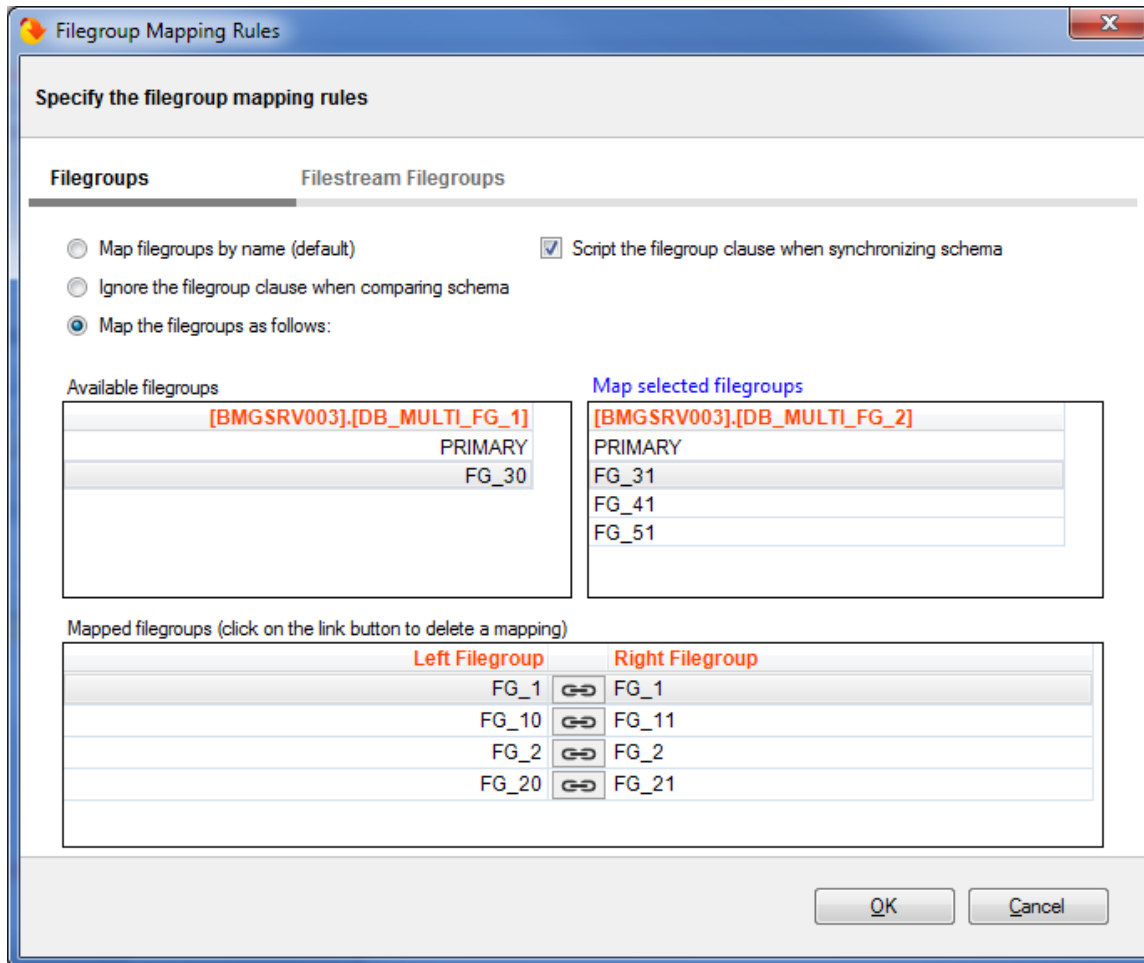
- FG_1 and FG_2 are considered equivalent, which means that no schema difference is triggered by the filegroup clause of the Employees table.
- If other schema differences require the Employees table to be

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.



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	Purchases	Support	Community	About Us	Resources	Legal
-------------------------------	--------------------------	---------------------------	-------------------------	---------------------------	--------------------------	---------------------------	-----------------------

SQL Schema Compare Entity Filters

Entity filters are a new feature introduced in the version 4 of IDERA SQL Schema Compare. They allow you to include or exclude database objects based on some predefined criteria. Entity filters have replaced the schema filters supported by the previous versions of SQL Schema Compare. You can access the entity filters from the context menu of the comparison grid.

To learn more about entity filters, click on the following topics:

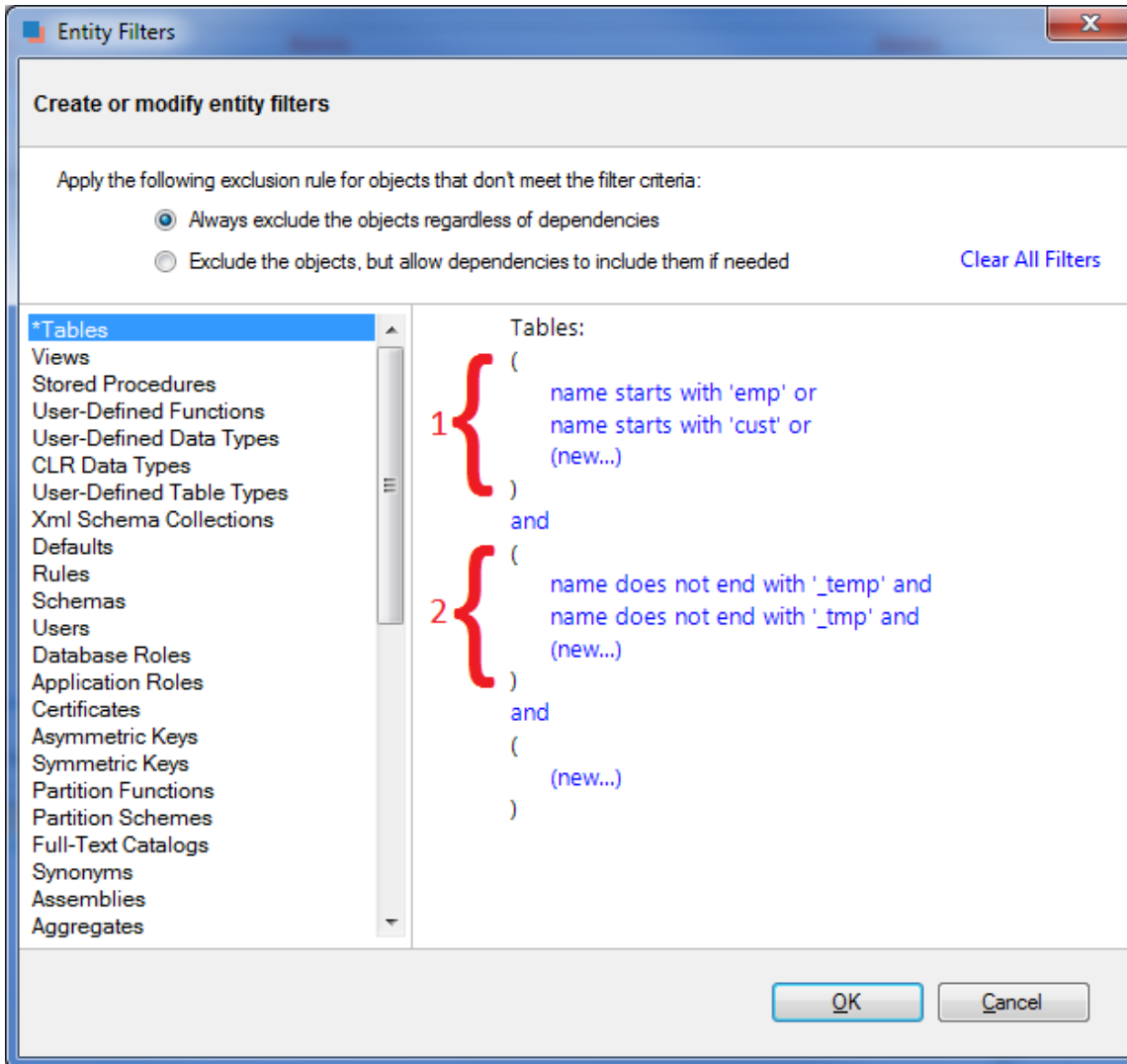
- [Entity filters overview](#)
- [Creating exclusion rules](#)
- [Working with entity filters](#)

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

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

SQL Schema Compare Entity Filters Overview

The entity filter is a collection of conditions that allows you to include or exclude database objects. Conditions are organized into units called groups. A single group contains conditions combined with the same operator: AND/OR. The groups themselves can be combined with an operator that may be the same as or different from the conditions' operator.



There is no limit to the number of groups that a filter can have or the number of conditions contained in a single group. In order to keep the filter simple, we recommend that you place all conditions that can be combined with a common operator in the same group. Create additional groups only if you need conditions that require a different operator.

The following filter, for example, defines two conditions combined with the AND operator. It requires only one group.

- name does not start with "v_temp" and
- name does not contain "temp"

The following filter, on the other hand, contains conditions that requires different operators, therefore two groups are needed.

Group 1:

- name starts with "emp" or
- name ends with "emp"

Group 2:

- name does not start with "tmp" and
- name does not start with "#" and
- name does not start with "##"

i An entity filter, similar to other comparison settings, is saved in the comparison session and it is applied automatically when you launch the session.

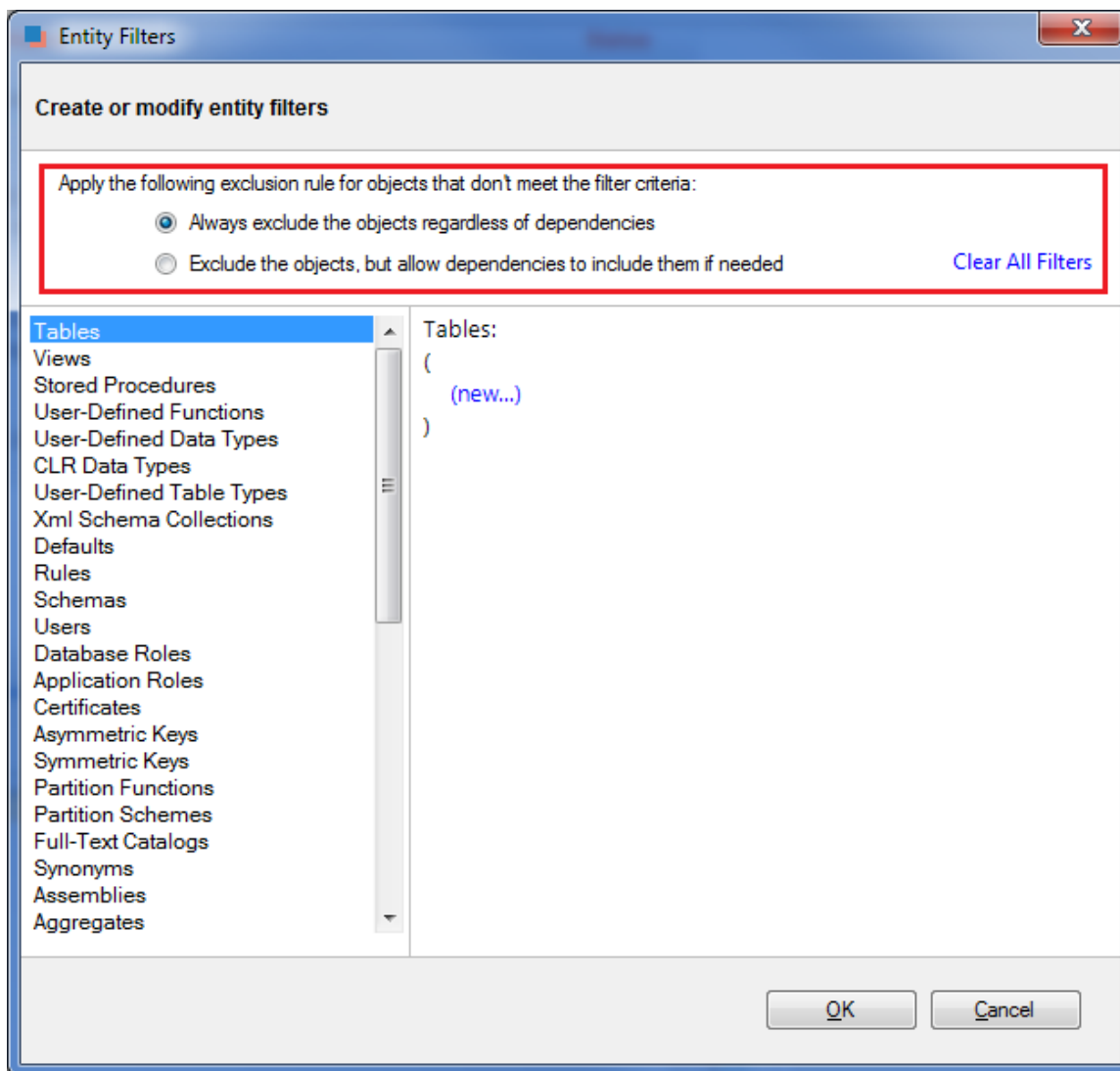
Total compare from data to schema with IDERA [SQL Comparison Toolset](#). [Learn more >>](#)

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

SQL Schema Compare Exclusion Rules

An entity filter provides two options, located at the top of the filter form, that determine the method by which database objects are excluded:

- The first option always excludes objects that do not meet the filter criteria. The excluded objects do NOT appear in the comparison grid at all.
- The second option excludes objects that do not meet the filter criteria, but allow the database dependencies to include them (if these objects are needed). Excluded objects in this case appear unchecked in the comparison grid.



Please note that the second exclusion rule could produce, in some cases, unexpected results. You might notice that database objects, explicitly excluded by the filter, still appear in the synchronization script.

Consider, for example, the following scenario:

A database contains a table named "employees" and a view named "v_employees" that depends on it. A table filter has been defined that excludes the "employees" table. Since the view "v_employees" depends on the "employees" table, the "employees" table will be included if the view needs to be synchronized.

If you wish to always exclude database objects that do not meet the filter criteria, regardless of the dependencies, do one of the following:

- Select the first exclusion rule OR
- Leave the second rule selected, but go to the Comparison Options form and uncheck the option "Include dependent objects".

The first option is recommended.

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

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

SQL Schema Compare Working with Entity Filters

Entity filters can be accessed from the context menu of the comparison grid. The conditions for an entity filter can be created or modified via the links provided in the filter definition section.

To create a new condition:

1. From the list of objects, select the object type to which the filter applies.
2. Click on the (new...) link.
3. Enter the condition criteria and click **OK**.

i Database object types with an associated filter have a star * in front of the name and appear at the top of the list.

To modify or delete an existing condition:

1. In the filter definition, click on the condition that you wish to modify.
2. In the condition dialog, change the condition's property and click **OK**, or click **Delete** to remove it.
3. To change an operator, simply click on it and the operator flips from AND to OR or vice versa.

Filter Conditions

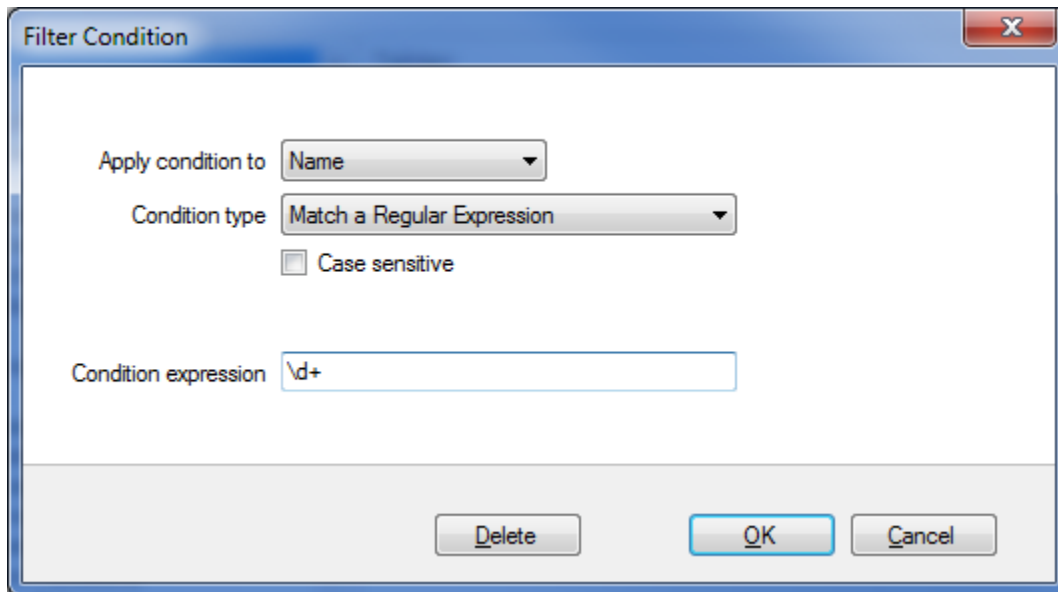
The conditions are the basic filter criteria that are evaluated to determine whether an object should be included or excluded. A filter condition exposes the following properties:

- **Name/Schema.** Indicates whether the condition will be checked against the name or the schema of the database object
- **Condition type.** Determines the type of the condition, such as "Starting With", "Equals To" etc.
- **Case sensitive.** Indicates whether the condition should honor the character case
- **Condition expression.** Defines the condition expression

Most of the conditions types are straightforward. The filter supports also two conditions that are

based on regular expression for more complex criteria.

The following condition, for example, selects the database objects that have at least one digit in their name:



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

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

SQL Schema Compare Synchronizing Schemas

Propagating database schema changes from one environment to another (often from development to production) safely and efficiently is the most critical functionality that IDERA SQL Schema Compare provides to software developers and database administrators.

After the comparison of the schemas of two databases has been completed you can select the objects you want to include in the synchronization and generate the schema synchronization script. The generated script will be displayed on a new tab where you will be able to review and execute it.

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

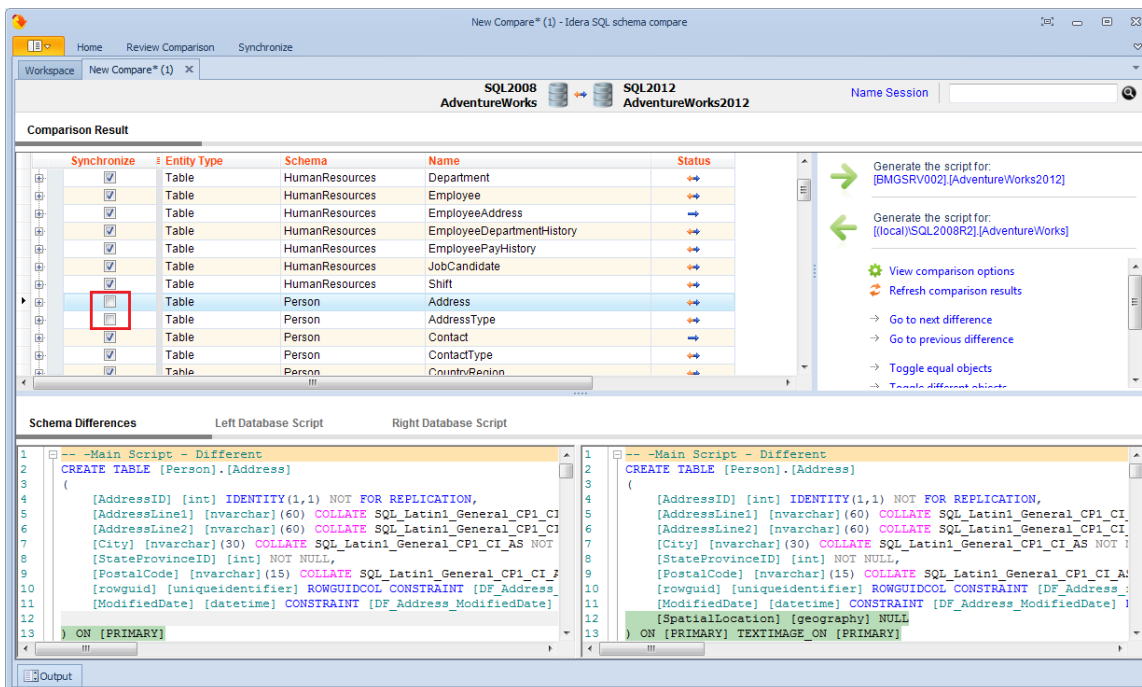
IDERA Website	Products	Purchase	Support	Community	About Us	Resources	Legal
-------------------------------	--------------------------	--------------------------	-------------------------	---------------------------	--------------------------	---------------------------	-----------------------

SQL Schema Compare Selecting Objects

By default all the objects that were found to be different during the comparison of the two databases are marked for inclusion in the synchronization script. However, you can choose to exclude or include certain objects from the comparison by checking or clearing the **Synchronize** checkbox next to each object. Selecting or clearing a parent object also affects the dependent objects. For example, if the checkbox for a table is cleared, then all of its primary keys, foreign keys, indexes etc. are also cleared.

Cleared but included

You may be surprised to see included in the synchronization script an object that you explicitly excluded. Why are those objects included? To answer this you need to look under the comparison / synchronization options and see if the option to "include dependent objects" is **ON**. When that option is **ON**, an included object will cause all the objects on which it depends to also be included in the synchronization script. For example, if you have a table T1 that contains foreign keys coming from tables T2 and T3 then, even though you may have cleared tables T2 and T3 if T1 is checked then both T2 and T3 will be included. When the "include dependent objects" option is **OFF**, then your object selection is respected, however, in this case you should be aware that the resulting synchronization script might not execute successfully.



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

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

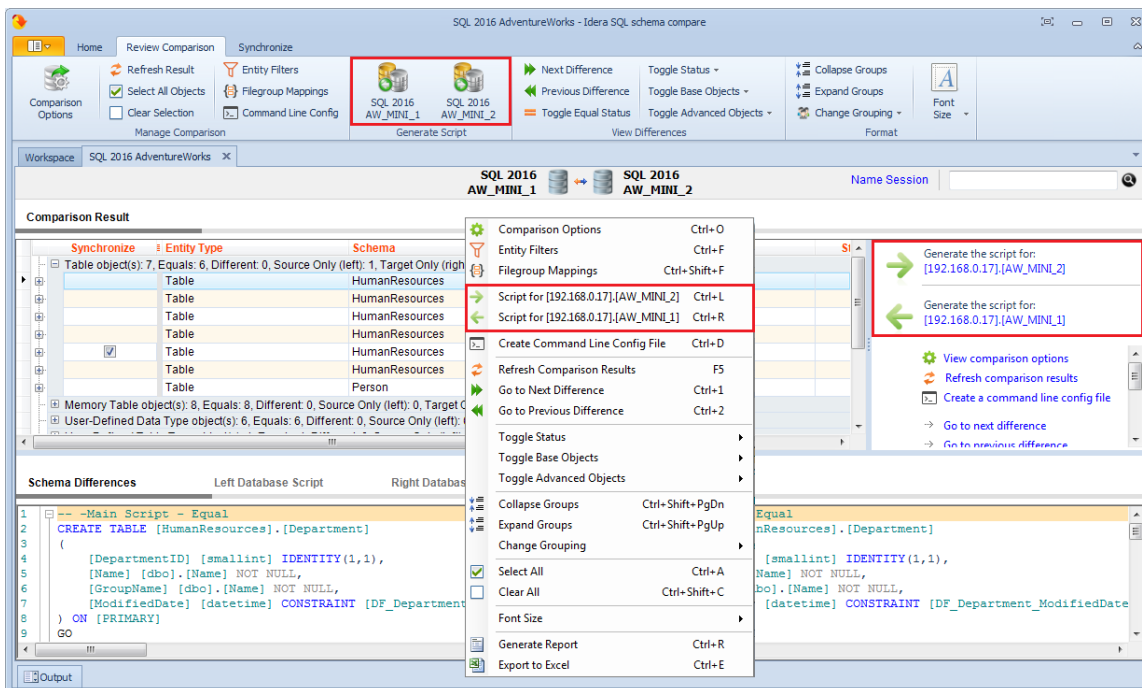
SQL Schema Compare Synchronization Script

Once the schema comparison operation is finished you are now ready to start the synchronization operation.

By default all objects that are found to have differences are automatically marked to be included in the synchronization. You can however easily change which objects should or should not be included in the script by checking or clearing the Synchronization checkbox (see [SQL Schema Compare Selecting Objects](#)).

Once you have made your selections you can generate the synchronization script to either change the database on the right of the comparison to make it the same as that on the left or the other way around. There are three ways to start the script generation:

- **Ribbon.** In the Review Compare tab of the ribbon there are two big buttons in the middle under the grouping **Generate Script**. Each button is labeled with the name of the server plus the name of the database for which the script will be generated.
- **Action Links** on the right panel. (listed under the label **Generate the script for**) Clicking on the [server].[database] link generates the synchronization script that will make that database the same as the other one.
- **Context menu.** When you right click on the comparison results grid a context menu will pop up. The context menu allows you to access the comparison options and trigger the generation of the synchronization script in addition of other context relevant actions.



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

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

SQL Schema Compare Working with the Script

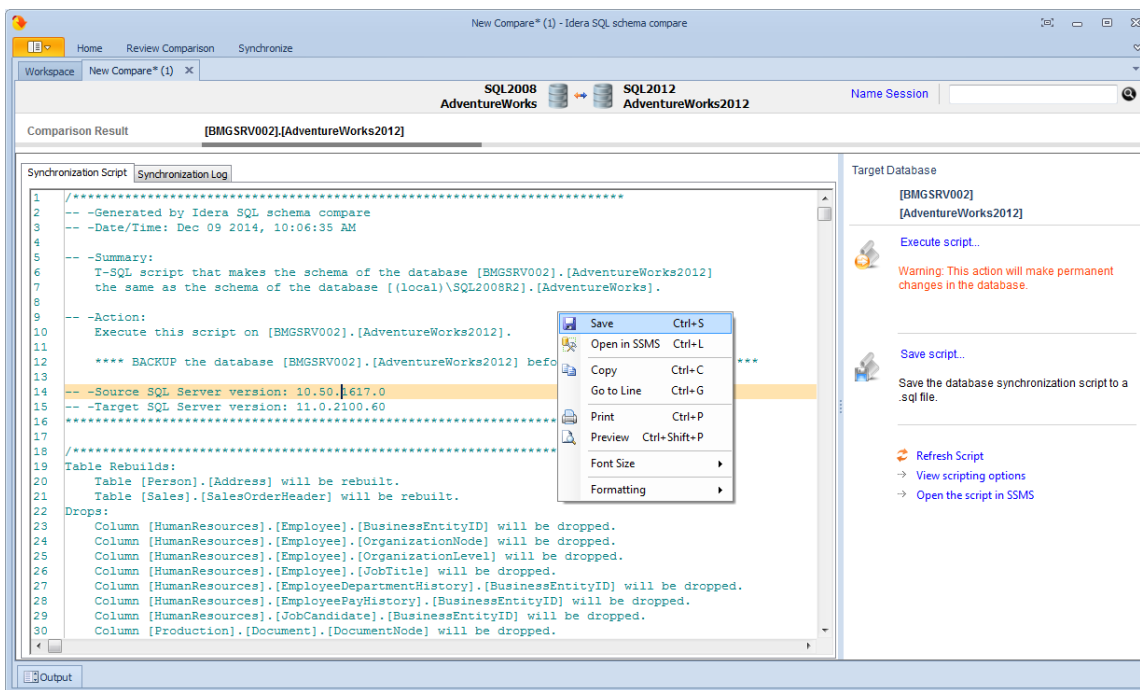
The data synchronization script will be displayed on a new window within the comparison sessions tab. The new window is named with the [server] . [database] for which the script was generated and it contains two sub-tabs. The Synchronization Script tab contains the actual script whereas the Synchronization Log tab contains a summary of the actions that will be performed on the target database.

Warnings section

It is important that you pay special attention to the warnings displayed at the top section of the synchronization script window. When the synchronization requires dropping objects such as columns or tables that may contain data or in situations when the synchronization has to be performed through an intermediary table IDERA SQL Schema Compare will display a warning indicating the potential risk that such action carries.

Both the ribbon and context menu allow you to open the script in SSMS, save the script, copy, print, or jump to a certain line in the script.

Note that if you go back to the comparison results tab and redo the comparison the corresponding synchronization script may be out of sync as the script is not automatically re-generated. To refresh the script, click **generate the script...** on the Comparison Results tab.



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

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

SQL Schema Compare Executing Synchronization Script

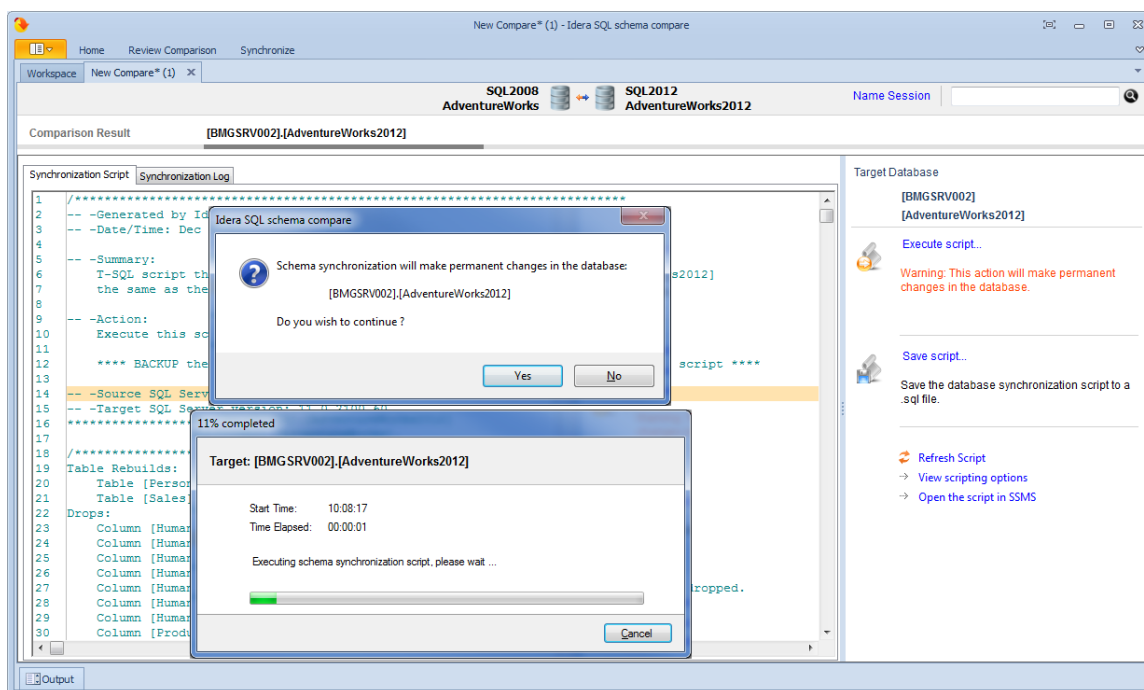
After reviewing the synchronization script and making sure that the script will do what you intend it to do, you can execute it.

Warning! This action will make permanent changes to the content of the target database. It is common sense and we strongly recommend that before executing the script you:

- Make a full backup of the target database. Keep in mind that if columns or tables that contain data will be dropped by the synchronization script the only way to reverse the changes will be to restore the data from a full backup of the database.
- Save the script you are about to execute so that you know exactly what was done to the database and when.

You can execute the script directly from IDERA SQL Schema Compare SSMS and execute it from there. Please pay attention to the Action section on the header of the synchronization script to make sure that the script will be executed against the correct database.

Note that by default, the synchronization script will be executed within a transaction so if you cancel the execution of the script before it completes the transaction will roll back and undo all the changes that were done up to that point. Therefore, the cancellation of the script execution is not instantaneous as depending on the scenario it may take significant amount of time to roll the changes back.



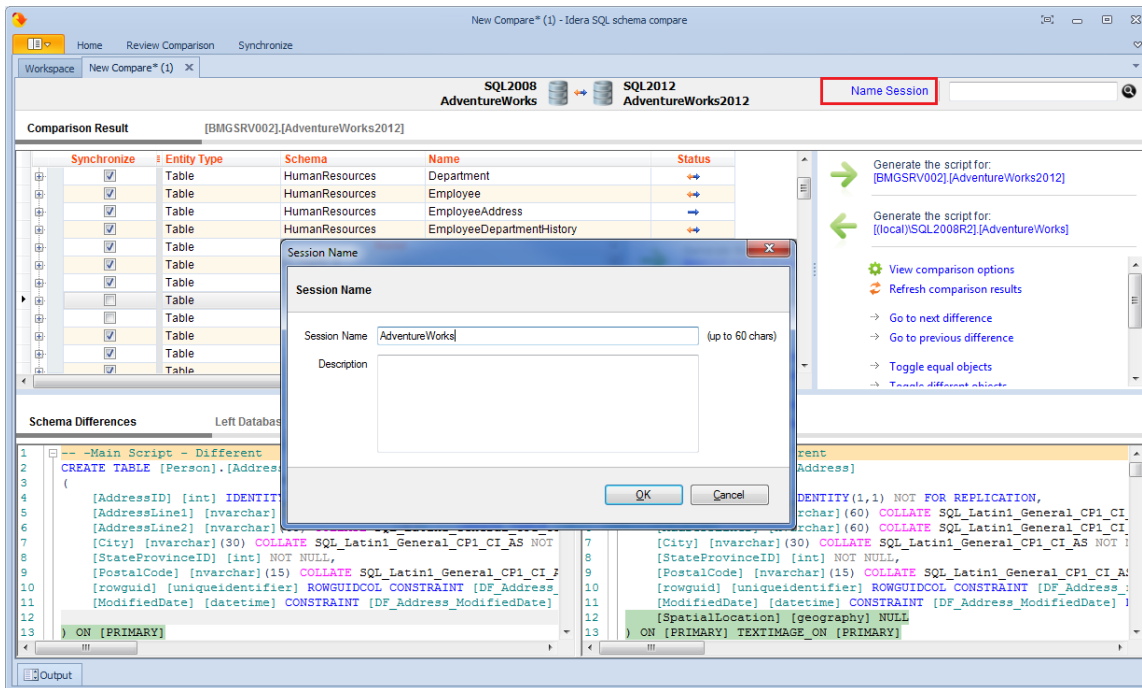
Total compare from data to schema with IDERA [SQL Comparison Toolset](#). [Learn more >>](#)

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

SQL Schema Compare Comparison Sessions

Depending on the scenario, the first time you compare two particular databases you might have to spend some time to configure the comparison and synchronization just the way you want it. IDERA SQL Schema Compare automatically saves all your configuration choices on a "comparison session" so that next time you might need to compare and synchronize those same databases you can do that with one click without having to go through that configuration effort again.

You can also name the comparison sessions to make it easier to identify them. Up to 25 sessions are saved and displayed on the Workspace tab.



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

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

SQL Schema Compare Tools and Utilities

IDERA SQL Schema Compare provides the following tool to help you work with database snapshots. In the default installation path, tools are installed under All Programs > IDERA > SQL Comparison Toolset > Tools menu. Click on the following link to view details about the tool:

- [IDERA Snapshot Utility](#)

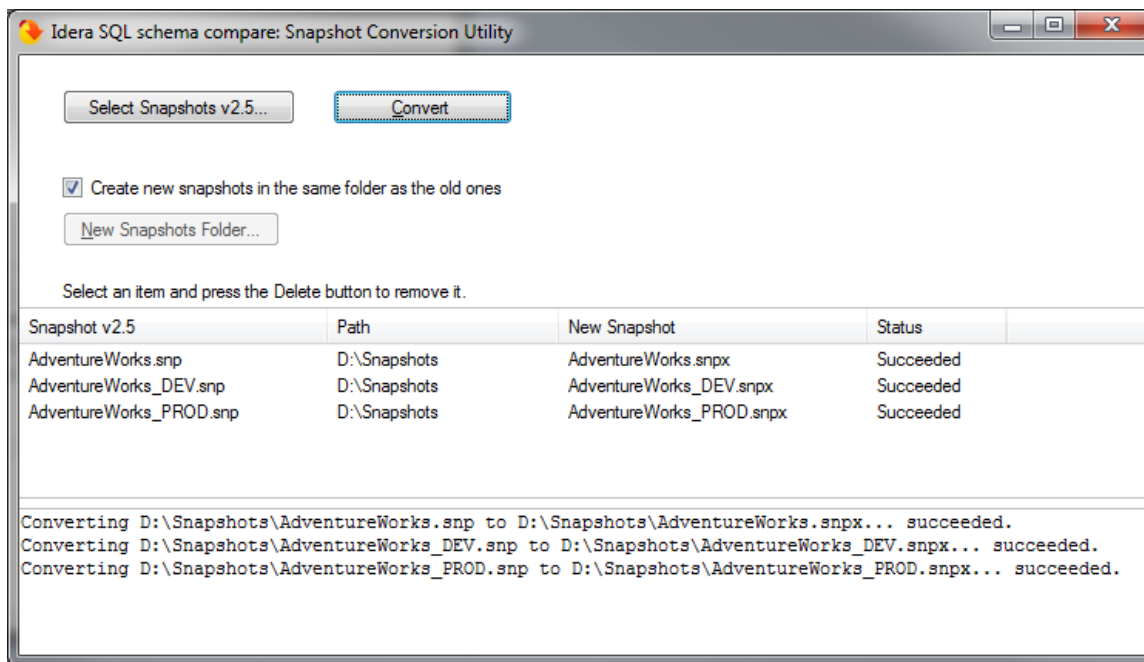
Total compare from data to schema with IDERA [SQL Comparison Toolset](#). [Learn more >>](#)

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

SQL Schema Compare IDERA Snapshot Converter

⚠ IDERA Snapshot Converter is no longer available as of SQL Comparison Manager 7.0.

IDERA Snapshot Converter is a small utility that allows you to convert database snapshots from previous versions of SQL Schema Compare to version 4. This utility automatically converts a snapshot when you add it to the workspace and also features a bulk conversion. Use the bulk snapshot conversion when you have many files and wish to avoid adding them to the workspace one by one.



Begin by adding the snapshots that you intend to convert to the snapshot grid. To remove a snapshot, select the snapshot row in the grid and press **Delete** on your keyboard. You can choose to save the new snapshots in the same folder as the old ones, or you can select a different folder. The converted snapshots will have the same name as the old ones and the extension .snpx. The old snapshot files are preserved.

To start the conversion, click **Convert**. The **Status** column in the grid indicates the status of the conversion as the tool progresses. If a snapshot fails to convert, details about the error can be viewed by clicking on the snapshot row.

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

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

SQL Schema Compare IDERA Snapshot Utility

IDERA Snapshot Utility is a tool that allows you to create database snapshots via the command line. A snapshot is a file that contains all the schema information for a database. The schema information is stored in a proprietary format and can only be read by the IDERA SQL Schema Compare user interface or command line. The schema snapshot does not contain data, only schema information.

The Snapshot Utility accepts parameters in the format `/name:value`. There are no spaces between the name and the value and the value is separated from the name with a colon `<:>`. If the value itself contains spaces, it should be surrounded in double quotes.

IDERA Snapshot Utility supports the following arguments:

- `/s:<server>` Specifies the SQL Server instance.
- `/d:<database>` Specifies the database name.
- `/t`: Indicates that the connection to the database should be a trusted connection. If not specified, the connection by default is trusted.
- `/u:<username>` Specifies the username when the database uses the SQL Server authentication.
- `/p:<password>` Specifies the password when the database uses the SQL Server authentication.
- `/sp:<filename>` Saves the snapshot to the specified filename. If the filename is not set, the snapshot file will be named as `database.snpx`.
- `/cs:<connection_string>` Specifies the connection string to the database. Use this option when you need to set additional settings that are not supported directly such as the connection timeout.
- `/f:<setting_file>` Creates the snapshot using the settings in the specified file. When the `/f` parameter is specified, all other parameters are ignored. In addition of being easier to manage, using a setting file allows you also to create snapshots for multiple databases. In this case use one line per database.

Samples:

The following command creates a snapshot for the database AdventureWorks on the local SQL Server using a trusted connection:

```
xSQLSnapshot /s:(local) /d:AdventureWorks
/sp: "c:\AdventureWorks.snpx"
```

The following command creates the snapshot by specifying a connection string:

```
xSQLSnapshot /cs:"Persist Security Info=False;User
ID=<username>;Password=<password>;Initial
Catalog=AdventureWorks;Data Source=(local)"
/sp: "c:\AdventureWorks.snpx"
```

The following command uses the setting file `snapshot.txt` to create the snapshot for two databases:

```
xSQLSnapshot /f:"c:\snapshot.txt"
```

The setting file `snapshot.txt` contains the following lines:

```
/s:(local) /d:AdventureWorks_DEV /sp:"c:\AdventureWorks_DEV.snpx"
/s:(local) /d:AdventureWorks_PROD /sp:"c:\AdventureWorks_PROD.snpx"
```

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

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

IDERA SQL Schema Compare command line

IDERA SQL Schema Compare command line allows you to compare and synchronize database schema via the command line. With the command line you can:

- Perform automated schema comparison and synchronization;
- Compare and synchronize database schema as part of your setup and deployment solution;
- Compare database schemas and execute the synchronization scripts in batch files;
- Schedule the comparison.

All command line settings are specified in an xml config file. You can create an xml config file via one of the following methods:

- **Using the SQL Schema Compare command line wizard.** You can access the wizard from the start menu, under **Start > All Programs > IDERA > SQL Server comparison tools > Command Lines**.
- **Using the SQL Schema Compare interface.** Simply compare the databases for which you wish to create a command line config file, set the options and other comparison preferences and click **Create a command line config file** in the Ribbon, context menu or in the right panel.
- **Manually, using any text editor, such as Notepad.**

The command line comes with samples for most typical data comparison scenarios. The sample configuration files are located under **\Program Files\Idera\SQL comparison toolset\Command Line Samples\Schema Compare**.

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

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

SQL Schema Compare Command Line Arguments

The command line supports the following arguments:

Argument	Description
xmlfile	Specifies the xml file that contains comparison settings, such as databases, credentials, options, object exclusion, schema filters and more.
/v, /validate	Validates the xml configuration file without performing a schema comparison.
/i, /logfile:filename	Specifies the filename where operational messages should be saved. If no log is specified, messages are displayed on the screen.
/e, /error:errorlog	Specifies the error log. Depending on the /t option, this parameter could indicate a file or a Windows event log.
/t, /logtype:(windowslog)	Indicates the type of the error log. If /t:windowslog is specified, the error log set via /e parameter is considered a Windows event log; otherwise the log is considered a file. By default errors are logged to a file.
/o, /overwrite	Indicates whether the log files should be overwritten. By default the command line appends to the log files.
/q, /quiet	Indicates the "quiet" mode of operation, which suppresses all messages from being displayed on the screen or logged to a file.
/en, /encoding:(UTF7 UTF8 UNICODE ASCII)	Specifies the encoding that should be used for all output files. ANSI encoding is used if no encoding is specified.
/? , /h, /help	Displays the command line arguments.

The following command performs a schema comparison using the settings in the config.xml file:

```
xSQLSchemaCmd config.xml
```

The following command validates the config.xml file:

```
xSQLSchemaCmd config.xml /v
```

i Apart from **/v** and **/?**, all other command line arguments are supported and can be set in the xml config file. We strongly recommend, as a good practice, that you specify the

comparison options in the xml config file instead of using the command line arguments. When an option is set via the command line parameter and in the xml file, the command line parameter takes precedence.

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

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

SQL Schema Compare XML Config File

The command line uses an xml config file for all comparison options and settings. The xml file allows you to:

- Specify SQL Server databases, credentials and connection settings
- Specify the snapshots that contain the database schema
- Change the comparison options
- Customize the comparison by excluding objects, setting schema filters or changing object mappings
- Specify filegroup comparison options
- Specify the files for the synchronization script, logs, warnings and errors
- Execute the schema synchronization script

The xml config file contains the following sections:

Section	Description
Database <code><LeftDatabase></code> <code><RightDatabase></code>	Use this element to specify SQL Server databases, credentials or the snapshots containing the schema that should be compared.
Comparison Options <code><ComparisonOptions></code> Selected Comparison Options <code><SelectedComparisonOptions></code> Excluded Comparison Options <code><ExcludedComparisonOptions></code>	Use these elements to select or exclude comparison options.
Object Type Selection <code><ObjectTypeSelection></code>	Use this element to include or exclude database objects by type.
Object Selection <code><ObjectSelection></code>	Use this element to include or exclude database objects by name.
Schema Filters <code><SchemaFilters></code> Schema Filter Expressions <code><SchemaFilterExpressions></code>	Use these elements to specify schema filters. A schema filter allows you to include or exclude database objects based on matching criteria.
Filegroup Settings <code><FileGroupSettings></code>	Use this element to specify the filegroup options.
Filestream Filegroup Settings <code><FileStreamFileGroupSettings></code>	Use this element to specify the filestream filegroup options.
Object Mappings <code><Mappings></code>	Use this element to specify the mapping type for various database objects. Command line supports mappings by name or schema and name.

Command Line Settings <CommandLineSettings>	Use this element to specify various comparison settings such as the output files, the target database, whether databases should only be compared or compared and synchronized, and more.
--	--

The following is a typical xml config file:

```
<?xml version="1.0" encoding="utf-8" ?>
<CommandLineParameters xmlns="http://www.xsql.com/sqlschemacmd.xsd">
  <LeftDatabase>
    <SqlServer>(local)</SqlServer>
    <DatabaseName>Left_DB</DatabaseName>
    <TrustedConnection>>true</TrustedConnection>
  </LeftDatabase>
  <RightDatabase>
    <SqlServer>(local)</SqlServer>
    <DatabaseName>Right_DB</DatabaseName>
    <TrustedConnection>>true</TrustedConnection>
  </RightDatabase>
  <CommandLineSettings>
    <SchemaScriptFile>script.sql</SchemaScriptFile>
    <SchemaLogFile>log.txt</SchemaLogFile>
    <SchemaWarningsFile>warnings.txt</SchemaWarningsFile>
    <ErrorLogName>error.txt</ErrorLogName>
    <CompareSchema Direction="RightAsLeft "
Synchronize="true"></CompareSchema>
  </CommandLineSettings>
</CommandLineParameters>
```

This config file forces the command line to perform the followings:

- It compares database **Left_DB** and **Right_DB** on the local SQL Server instance. A trusted connection is used for both databases.
- It saves the synchronization script to the file **script.sql** indicated by the <SchemaScriptFile> element.
- It saves the synchronization log to the file **log.txt** indicated by the <SchemaLogFile> element.
- It saves any warnings that are generated during the comparison to the file **warnings.txt** indicated by the <SchemaWarningsFile> element.
- It saves errors that might occur during the execution of the synchronization script to the file **error.txt** indicated by the <ErrorLogName> element.
- It sets the Right_DB as the target database indicated by the attribute **Direction="RightAsLeft"** under the element <CompareSchema>. The command line will generate a script that makes the Right_DB the same as the Left_DB.
- It executes the synchronization script and makes permanent changes in target database indicated by the attribute **Synchronize="true"** under the element <CompareSchema>. By default the command line generates the synchronization script, but does not execute it.

The following command performs a schema comparison using the settings in config.xml:

```
xSQLSchemaCmd config.xml
```

The following command validates the file config.xml:

```
xSQLSchemaCmd config.xml /v
```

 The xml elements and attributes are case-sensitive.

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

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

SQL Schema Compare Comparing Databases

With the command line you can compare live databases or snapshot files. A snapshot is a file where the database schema is serialized. It does not contain data, just the schema information. The information is stored in a proprietary format and can only be read by the IDERA SQL Schema Compare user interface or command line. Snapshots do not require authentication, live databases or the other hand can use either Windows or SQL Server authentication.

All database-related settings are specified in the xml config file under the <LeftDatabase> and <RightDatabase> elements. The following xml fragment specifies Left_DB and Right_DB on the local SQL Server. Trusted connection is used for the Left_DB database and SQL Server authentication is used for the Right_DB database:

```
<LeftDatabase>
  <SqlServer>(local)</SqlServer>
  <DatabaseName>Left_DB</DatabaseName>
  <TrustedConnection>>true</TrustedConnection>
</LeftDatabase>

<RightDatabase>
  <SqlServer>(local)</SqlServer>
  <DatabaseName>Right_DB</DatabaseName>
  <TrustedConnection>>false</TrustedConnection>
  <Username>username</Username>
  <Password>password</Password>
</RightDatabase>
```

To compare snapshots instead of live databases, use the <SnapshotFile> element as in the following xml fragment:

```
<LeftDatabase>
  <SnapshotFile>AdventureWorks_DEV.snp</SnapshotFile>
</LeftDatabase>

<RightDatabase>
  <SnapshotFile>AdventureWorks_PROD.snp</SnapshotFile>
</RightDatabase>
```

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

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

SQL Schema Compare Creating Snapshots

You can use the command line to create a database snapshot before comparing the schema.

The following xml config creates a snapshot for the database AdventureWorks_DEV and compares AdventureWorks_DEV and AdventureWorks_PROD database in the local SQL Server:

```
<?xml version="1.0" encoding="utf-8" ?>
<CommandLineParameters xmlns="http://www.xsql.com/sqlschemacmd.xsd">
  <LeftDatabase>
    <SqlServer>(local)</SqlServer>
    <DatabaseName>AdventureWorks_DEV</DatabaseName>
    <TrustedConnection>>true</TrustedConnection>
    <CreateSnapshot>AdventureWorks_DEV.snpx</CreateSnapshot>
  </LeftDatabase>

  <RightDatabase>
    <SqlServer>(local)</SqlServer>
    <DatabaseName>AdventureWorks_PROD</DatabaseName>
    <TrustedConnection>>true</TrustedConnection>
  </RightDatabase>

  <CommandLineSettings>
    <SchemaScriptFile>script.sql</SchemaScriptFile>
    <SchemaLogFile>log.txt</SchemaLogFile>
    <SchemaWarningsFile>warnings.txt</SchemaWarningsFile>
    <ErrorLogName>error.log</ErrorLogName>
    <CompareSchema Direction="RightAsLeft "
Synchronize="false"></CompareSchema>
  </CommandLineSettings>
</CommandLineParameters>
```

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

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

Customizing SQL Schema Compare

The command line provides a great deal of customization to meet your comparison requirements. It allows you to:

- Compare live databases or snapshot files
- Take a snapshot before comparing live databases.
- Include or exclude an entire class of objects by type, such as tables, views and more
- Include or exclude database objects by name
- Include or exclude database objects via schema filter
- Change the comparison options
- Change the filegroup comparison options
- Change the filestream filegroup comparison options
- Change the default mapping of various database objects

The following sections describe each scenario in details.

- [SQL Schema Compare Excluding Objects](#)
- [SQL Schema Compare Changing Comparison Options](#)
- [SQL Schema Compare Entity Filters customization](#)
- [SQL Schema Compare Schema Filters](#)
- [SQL Schema Compare Changing Filegroup Settings](#)
- [SQL Schema Compare Changing Object Mappings](#)

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

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

SQL Schema Compare Excluding Objects

The command line provides two methods for excluding database objects:

- [Excluding objects by type](#)
- [Excluding objects by name](#)

The following topics describe each method in detail.

- [SQL Schema Compare Excluding Objects by Type](#)
- [SQL Schema Compare Excluding Objects by Name](#)

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

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

SQL Schema Compare Excluding Objects by Type

Type-based method allows you to include or exclude an entire class of objects, such as tables or views. To exclude objects by type, use the <ObjectTypeSelection> element and specify two attributes:

- **Type**. Indicates the object type
- **Include="true|false"**. A flag that indicates whether the specified type should be included or excluded

The following xml fragment excludes tables and views from the comparison.

```
<ObjectTypeSelection>
  <ObjectType Type="Table" Include="false" />
  <ObjectType Type="View" Include="false" />
</ObjectTypeSelection>
```

The command line provides a list of predefined types that you can choose to include or exclude:

- Table
- MemoryTable
- View
- StoredProcedure
- UserDefinedFunction
- UserDefinedDataType
- ClrDataType
- UserDefinedTableType
- XmlSchemaCollection
- Default
- Rule
- Schema
- User
- DatabaseRole
- ApplicationRole
- Certificate
- AsymmetricKey
- SymmetricKey
- PartitionFunction
- PartitionScheme
- FullTextCatalog
- Synonym
- Assembly
- Aggregate
- Sequence
- DatabaseTrigger
- PrimaryKey
- UniqueConstraint
- Index
- XmlIndex
- SpatialIndex

- FullTextIndex
- ForeignKey
- DefaultConstraint
- CheckConstraint
- Trigger
- Statistic
- BrokerMessageType
- BrokerContract
- BrokerQueue
- BrokerRoute
- BrokerService
- BrokerRemoteServiceBinding
- BrokerPriority
- Permission
- ExtendedProperty
- FullTextStopList
- FullTextStopListWord
- SearchPropertyList
- SecurityPolicy
- ColumnMasterKey
- ColumnEncryptionKey

In addition to these types, the command line supports also three macro-types in order to facilitate the selection:

- **All**. When specified it includes all object types
- **None**. When specified it excludes all object types
- **DefaultObjects**. When specified it includes the most common objects.

The Include attribute is ignored for macro-types. Object types included by DefaultObjects macro-option are those that you see in the Schema Compare User-Interface option screen when you click on the **Restore Defaults** button. More details are provided [here](#).

Depending on your scenario, you could combine one of the macro-types with the actual types to achieve the desired result. For example, to exclude only views and procedures, you could use the following xml fragment:

```
<ObjectTypeSelection>
  <ObjectType Type="DefaultObjects" />
  <ObjectType Type="View" Include="false" />
  <ObjectType Type="StoredProcedure" Include="false" />
</ObjectTypeSelection>
```

or just:

```
<ObjectTypeSelection>
  <ObjectType Type="View" Include="false" />
  <ObjectType Type="StoredProcedure" Include="false" />
</ObjectTypeSelection>
```

To include only tables and exclude all other objects, you could combine the None macro-type with the Table type as in the following xml fragment:

```
<ObjectTypeSelection>
  <ObjectType Type="None" />
  <ObjectType Type="Table" Include="true" />
</ObjectTypeSelection>
```

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

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

SQL Schema Compare Excluding Objects by Name

While the type-based selection excludes an entire class of objects, the name-based method allows you to exclude specific objects by name.

To exclude an object by name, use the <ObjectSelection> element of the xml config file and specify one or multiple <Object> elements with the following attributes:

- **Schema.** Contains the object schema. If not specified, the command line will locate the object by name. We recommend you always specify the schema.
- **Name.** Contains the object name
- **Type.** Contains the object type
- **Include="true|false".** Indicates whether the object should be included or excluded.

The following xml fragment excludes the table employees and the view v_employees:

```
<ObjectSelection>
  <Object Schema="dbo" Name="employees" Type="Table"
  Include="false"/>
  <Object Schema="dbo" Name="v_employees" Type="View"
  Include="false"/>
</ObjectSelection>
```

The following list contains the object type that you can exclude:

- Table
- MemoryTable
- View
- StoredProcedure
- UserDefinedFunction
- UserDefinedDataType
- ClrDataType
- UserDefinedTableType
- XmlSchemaCollection
- Default
- Rule
- Schema
- User
- DatabaseRole
- ApplicationRole
- Certificate
- AsymmetricKey
- SymmetricKey
- PartitionFunction
- PartitionScheme
- FullTextCatalog
- Synonym
- Assembly
- Aggregate
- DatabaseTrigger
- Sequence

- BrokerMessageType
- BrokerContract
- BrokerQueue
- BrokerRoute
- BrokerService
- BrokerRemoteServiceBinding
- BrokerPriority
- FullTextStopList
- SearchPropertyList
- SecurityPolicy
- ColumnMasterKey
- ColumnEncryptionKey

In addition to these types, one macro-type is supported as well:

- **None.** Excludes all database objects

You can combine the macro-type None with the real types to achieve the desired results. For example, to include a table named employees and exclude all other objects, you could use following xml fragment:

```
<ObjectSelection>
  <Object Name="" Type="None" />
  <Object Schema="dbo" Name="employees" Type="Table"
Include="true"/>
</ObjectSelection>
```

i Level-two objects, such as columns, constraints, indexes, triggers, statistics, extended properties, permissions can't be excluded directly. These objects are excluded when the parent object is excluded.

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

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

SQL Schema Compare Changing Comparison Options

You can change the comparison options using the following elements:

- **<ComparisonOptions>** or **<SelectedComparisonOptions>** Contain the options that should be selected or set to true.
- **<ExcludedComparisonOptions>** Contains the comparison options that should be excluded or set to false.

The following xml fragment selects the options **IgnoreSpacesInDefinition**, **IgnoreCaseInDefinition**, **IgnoreDefaultConstraintName** and excludes the option **StandardizeNameInDefinition**. The max transaction size is also set to 65536 bytes:

```
<SelectedComparisonOptions TransactionSize="65536">
  <ComparisonOption>IgnoreSpacesInDefinition</ComparisonOption>
  <ComparisonOption>IgnoreCaseInDefinition</ComparisonOption>

  <ComparisonOption>IgnoreDefaultConstraintName</ComparisonOption>
</SelectedComparisonOptions>

<ExcludedComparisonOptions>

  <ComparisonOption>StandardizeNameInDefinition</ComparisonOption>
</ExcludedComparisonOptions>
```

The options you can choose from are predefined and can be one of the following:

- IncludeDependantObjects
- IgnoreSpacesInDefinition
- IgnoreCaseInDefinition
- IgnoreNameInDefinition
- StandardizeNameInDefinition
- CompareAnsiNullsSetting
- CompareQuotedIdentifierSetting
- ComparePartitioning
- CompareDataCompression
- CompareTableChangeTracking
- CompareTableChangeDataCapture
- CompareTableLockEscalation
- IgnorePrimaryKeyName
- CompareIndexPadding
- CompareIndexStatsNoRecompute
- CompareIndexRowLocks
- CompareIndexPageLocks
- CompareIndexColumnOrder
- IgnoreDefaultConstraintName
- IgnoreCheckConstraintName
- CompareColumnOrdinal
- CompareColumnCollation
- CompareIdentity
- CompareIdentitySeed

- CompareIdentityIncrement
- CompareNullability
- IgnoreViewComments
- AlterViewWithDropCreate
- IgnoreProcedureComments
- AlterProcedureWithDropCreate
- IgnoreFunctionComments
- AlterFunctionWithDropCreate
- IgnoreTriggerComments
- CompareUserLogin
- PreventSchemaDrop
- CompareSynonymServer
- CompareSynonymDatabase
- CompareSequenceStartValue
- CompareSequenceMinValue
- CompareSequenceMaxValue
- GenerateTransactionalScript
- GenerateUseDatabaseStatement
- ScriptFileGroup
- ScriptTableChangeTracking
- ScriptTableChangeDataCapture
- ScriptTableLockEscalation
- ScriptConstraintsWithNoCheck
- ScriptDataCompression
- ScriptAnsiNullsSetting
- ScriptQuotedIdentifierSetting
- ScriptAnsiPaddingSetting
- ScriptCollation
- ScriptNewColumnsAsNull
- ScriptFullTextCatalogPath
- ScriptSequenceStartValue
- ScriptSequenceMinValue
- ScriptSequenceMaxValue
- ExecuteTransactionalScript
- LogScriptExecution
- ContinueExecutionOnError
- CompareNotForReplication
- CompareIndexFillFactor
- CompareReplicatedTables
- CompareDiagramObjects
- TransactionSize
- CompareColumnEncryption
- CompareColumnMask
- CompareColumnMasterKeyProvider
- CompareColumnMasterKeyPath
- CompareSystemVersioning
- IgnoreSystemVersioningHistoryTableName
- ScriptColumnEncryption

- ScriptColumnMask
- ScriptSystemVersioning
- DropSystemVersioningHistoryTable
- CompareHashIndexBucketCount
- CompareRemoteDataArchiving
- ScriptRemoteDataArchiving
- ScriptHistoryRemoteDataArchiving
- AbandonRemoteDataWhenDisablingArchiving

i Both elements **<ComparisonOptions>** and **<SelectedComparisonOptions>** can be used to set the selected comparison options. Use the **<SelectedComparisonOptions>** element in order to avoid ambiguities.

TransactionSize is the only comparison option that must be set as an attribute of **<ComparisonOptions>**, **<SelectedComparisonOptions>**, or **<ExcludedComparisonOptions>** element. Specify a decimal value that indicates the max size of the schema transaction size in bytes. The default value is 0, which indicates an "unlimited" transaction. Use this option carefully. In most comparison scenarios, the default setting is sufficient.

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

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

SQL Schema Compare Entity Filters customization

An entity filter is a new feature introduced in the version 4 of the command line that replaces the schema filters supported by the previous versions. It allows you to include or exclude database objects based on some predefined criteria.

i Schema filters supported by the previous versions of the command line are depreciated and will be removed in a future release.

An entity filter is composed by groups combined with an operator: AND/OR. Each group contains conditions, which act as the filter criteria. When the filter is applied, the conditions are evaluated against the name or the schema of a database object. If the object meets the condition criteria, it is included in the comparison, otherwise it is excluded.

The following xml fragment defines a filter that selects views the name of which starts with the prefix v_emp:

```
<EntityFilters>
  <filter entitytype="View">
    <condition>
      <conditiontype>StartingWith</conditiontype>
      <expression>v_emp</expression>
    </condition>
  </filter>
</EntityFilters>
```

The following xml fragment defines a filter that contains two conditions. It selects tables the name of which does not start with "#" and "##":

```
<entityfilters>
  <filter entitytype="Table">
    <group conditionoperator="And">
      <condition>
        <conditiontype>NotStartingWith</conditiontype>
        <expression>#</expression>
      </condition>
      <condition>
        <conditiontype>NotStartingWith</conditiontype>
        <expression>##</expression>
      </condition>
    </group>
  </filter>
</entityfilters>
```

i The command line provides samples for the most common entity filter scenarios. You will find them in the command line sample folder under **\Program Files\Idera\SQL comparison toolset\Command Line Samples\Schema Compare\EntityFilters**.

The structure of the entity filters supported by the command line xml config file is as follows:

```

<entityfilters exclusionrule="ExcludeAlways|ExcludeIfPossible">
  <filter entitytype="object_type" groupoperator="And|Or">
    <group conditionoperator="And|Or"
includematches="true|false">
      <condition casesensitive="true|false"
namecondition="true|false">
        <conditiontype>condition_type</conditiontype>
        <expression>expression</expression>
      </condition>
    </group>
  </filter>
</entityfilters>

```

xml Element/Attribute	Description	Usage
EntityFilters	The starting element of the entity filters.	
ExclusionRule: <i>ExcludeAlways</i> <i>ExcludeIfPossible</i>	Determines the rule that should be used to exclude objects that don't meet the filter criteria with respect to dependencies: <ul style="list-style-type: none"> • ExcludeAlways indicates that objects should be excluded regardless of dependencies. • ExcludeIfPossible indicates that objects should be excluded, but database dependencies might include them if needed. 	Optional. Default value is <i>ExcludeAlways</i> .
Filter	Starts a filter element. Multiple filters are supported, each for a specific object type.	
EntityType	Indicates the type of the database objects associated with the filter. Nearly all database are supported, except for the following: <ul style="list-style-type: none"> • Filegroups • Full-Text Indexes • Permissions • Extended Properties 	Required.

GroupOperator: <i>And/Or</i>	Determines the operator that combines the groups that participate in the filter.	Optional. Default value is <i>And</i> .
Group	Starts a filter group. Multiple groups can be included in a single filter.	Optional when the filter contains only one condition, otherwise required.
ConditionOperator: <i>And/Or</i>	Determines the operator that combines filter conditions associated with a group.	Optional. Default value is <i>Or</i> .
IncludeMatches: <i>true/false</i>	Indicates whether the database objects that meet the group criteria should be included in the comparison.	Optional. Default value is <i>true</i> .
Condition	Starts a filter condition.	Required.
CaseSensitive: <i>true/false</i>	Indicates whether the condition should honor the character case.	Optional. Default value is <i>false</i> .
NameCondition: <i>true/false</i>	Determines whether the filter condition should be evaluated against the name of a database object. If false, the condition is checked against the schema.	Optional. Default value is <i>true</i> .
ConditionType	One of the following condition types: <ul style="list-style-type: none"> • StartingWith • NotStartingWith • EndingWith • NotEndingWith • Containing • NotContaining • EqualsTo • DifferentFrom • RegExMatching • RegExNotMatching 	Required.
Expression	A text expression that defines the condition criteria.	Required.

The following xml fragment is a more complete example that defines three entity filters:

- The first filter creates two conditions that select database views starting with `v_emp` or `v_cust`. Both conditions are case-sensitive.
- The second filter selects stored procedures that belong to `HumanResource` or `Personal`

schema. In addition to schema, the procedure name should contain at least one digit.

- The third filter selects functions that do not end with `_temp` and `_tmp`

```
<entityfilters exclusionrule="ExcludeAlways">
  <filter entitytype="View">
    <group conditionoperator="Or">
      <condition casesensitive="true">
        <conditiontype>StartingWith</conditiontype>
        <expression>v_emp</expression>
      </condition>
      <condition casesensitive="true">
        <conditiontype>StartingWith</conditiontype>
        <expression>v_cust</expression>
      </condition>
    </group>
  </filter>
  <filter entitytype="StoredProcedure" groupoperator="And">
    <group conditionoperator="Or">
      <condition namecondition="false">
        <conditiontype>EqualsTo</conditiontype>
        <expression>HumanResource</expression>
      </condition>
      <condition namecondition="false">
        <conditiontype>EqualsTo</conditiontype>
        <expression>Personal</expression>
      </condition>
    </group>
    <group>
      <condition namecondition="true">
        <conditiontype>RegExMatching</conditiontype>
        <expression>\d+</expression>
      </condition>
    </group>
  </filter>
  <filter entitytype="UserDefinedFunction">
    <group conditionoperator="And">
      <condition>
        <conditiontype>NotEndingWith</conditiontype>
        <expression>_temp</expression>
      </condition>
      <condition>
        <conditiontype>NotEndingWith</conditiontype>
        <expression>_tmp</expression>
      </condition>
    </group>
  </filter>
</entityfilters>
```

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

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

SQL Schema Compare Schema Filters

Schema filters allow you to include or exclude database objects based on matching criteria. You can exclude, for example, tables the name of which start with "tmp_" or include views that belong to the schema HumanResources.

i Schema filters are deprecated and will be removed in a future version of the command line utility. Use [entity filters](#) instead.

A schema filter affects only the objects that belong to the class of objects for which the filter is defined. A table filter, for example, affects tables only. When a schema filter is defined, the command line performs one of the following actions:

- If the filter is defined as an "Include" filter, objects that pass the filter criteria are included; all other objects that do not pass the filter criteria are excluded.
- If the filter is defined as an "Exclude" filter, objects that do NOT pass the filter criteria are included; all other objects that pass the filter criteria are excluded.

The command line provides two elements for defining schema filters:

- Simple filters that contain just a few criteria can be defined under the element `<SchemaFilters>`
- Complex filter with multiple criteria, different validation types and specific case-sensitivity can be defined under the element `<SchemaFilterExpressions>`

Simple filters are defined as `<SchemaFilter>` under `<SchemaFilters>` elements with the following attributes:

- **Criteria** Contains the filter matching criteria.
- **ObjectType** Indicates the type of object on which the filter will be applied.
- **Action="Exclude|Include"** Indicates whether the object that satisfies the filter criteria should be included or excluded.
- **FilterType** Indicates the filter type which can be one of the following:
 - StartsWith
 - EndsWith
 - Contains
 - EqualsTo
 - Schema
 - RegularExpression

The following xml fragment adds a filter that excludes all tables whose name ends with "_tmp":

```
<SchemaFilters>
  <SchemaFilter Criteria="_tmp" FilterType="EndsWith"
  ObjectType="Table" Action="Exclude" />
</SchemaFilters>
```

Schema filters defined under `<SchemaFilters>` element are sufficient in most comparison scenarios. If you need more control over the filter criteria, you can define the filters as `<SchemaFilterExpression>` under the `<SchemaFilterExpressions>` element. The `<SchemaFilterExpression>` element contains two attributes:

- **ObjectType** Indicates the type of object on which the filter should be applied
- **FilterValidationType="AllCriteria|AtLeastOneCriteria"** Indicates the type of validation that should be performed when the filter contains multiple criteria. If AllCriteria is specified, an object must pass all criteria in order to satisfy the filter; otherwise one criteria is sufficient. If not specified the validation is set by default to AtLeastOneCriteria.

Under the <SchemaFilterExpression> element, you can defined one or multiple filter criteria. Each criteria is a <SchemaFilterCriteria> element with the following attributes:

- **Criteria** Contains the filter criteria
- **Action="Exclude|Include"** Indicates whether the object that satisfies the filter criteria should be included or excluded.
- **FilterType** Indicates the filter type and can be one of the followings:
 - **StartsWith**
 - **EndsWith**
 - **Contains**
 - **EqualsTo**
 - **Schema**
 - **RegularExpression**
 - **IgnoreCase="true|false"** Indicates whether the character case should be ignored when the criteria is applied. By default the filter ignores the case.

The following xml fragment defines two filters: one for views and one for stored procedures. The view filter has three criteria. It selects only views that start with v1_, v2_ or v3_. The procedure filter has two criteria. It excludes procedures the name of which contain _p1_ AND _p2_. The first criteria of the procedure filter is case-sensitive:

```
<SchemaFilterExpressions>
  <SchemaFilterExpression ObjectType="View">
    <SchemaFilterCriteria Action="Include"
FilterType="StartsWith" Criteria="v1_" />
    <SchemaFilterCriteria Action="Include"
FilterType="StartsWith" Criteria="v2_" />
    <SchemaFilterCriteria Action="Include"
FilterType="StartsWith" Criteria="v3_" />
  <SchemaFilterExpression>
    <SchemaFilterExpression ObjectType="StoredProcedure"
FilterValidationType="AllCriteria">
      <SchemaFilterCriteria Action="Exclude"
FilterType="Contains" Criteria="_p1_" IgnoreCase="false" />
      <SchemaFilterCriteria Action="Exclude"
FilterType="Contains" Criteria="_p2_" />
    </SchemaFilterExpression>
  </SchemaFilterExpressions>
```

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

IDERA Website	Products	Purchas e	Support	Commun ity	About Us	Resourc es	Legal
-------------------------------	--------------------------	-------------------------------	-------------------------	--------------------------------	------------------------------	--------------------------------	-----------------------

SQL Schema Compare Changing Filegroup Settings

The command line allows you to customize the way the filegroups are mapped and compared. By default, filegroups are mapped by name, but you can choose to ignore them or map them differently.

Consider, for example, a table named employees that in the source database is stored on the filegroup LEFT_FG and in the target database on the filegroup RIGHT_FG. When the table is compared, the difference in the filegroup will trigger a table rebuild. If you change the filegroup settings and map LEFT_FG to RIGHT_FG, the filegroup difference will not trigger any action on the table.

You can change the comparison settings for regular filegroups via the elements <FileGroupSettings>. Filestream filegroup settings can be set under the <FileStreamFileGroupSettings> element. To alter the settings, specify the following elements:

- **ComparisonType="Ignore|ByName|ByMappings"** Indicates the type of comparison that should be applied to filegroups.
- **Mappings** Contains the filegroup mappings when the ComparisonType is set to "ByMappings". Mappings are set as <LEFT_FG>:<RIGHT_FG>. When more than one mapping is required, separate them with a semicolon <;>

The following xml fragment sets the filegroup comparison option to a compare by mappings:

```
<FileGroupSettings>
  <ComparisonType>ByMappings</ComparisonType>
  <Mappings>LEFT_FG_1:RIGHT_FG_1;LEFT_FG_2:RIGHT_FG_2</Mappings>
</FileGroupSettings>
```

i Filegroup settings affect the existing database filegroups and the filegroup clause of tables and indexes. Missing database filegroups are NOT created by default.

To create missing database filegroups, include the database objects DatabaseFileGroup and DatabaseDataFile as in the following xml fragment:

```
<ObjectTypeSelection>
  <ObjectType Type="DatabaseFileGroup" Include="true" />
  <ObjectType Type="DatabaseDataFile" Include="true" />
</ObjectTypeSelection>
```

We strongly recommend that you create the missing filegroups manually.

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

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

SQL Schema Compare Changing Object Mappings

Mapping determines the way database objects are paired together. The command line supports two types of mapping: by name, or by schema and name. The default mapping is by schema and name.

The following xml fragment changes the mapping for tables and views to name-based mapping:

```
<Mappings>
  <Mapping MatchingType="Name" ObjectType="Table" />
  <Mapping MatchingType="Name" ObjectType="View" />
</Mappings>
```

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

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

SQL Schema Compare Error Codes

The command line returns the following error codes:

Code	Description
0	Indicates success.
1	Invalid command line switches.
2	Invalid xml file.
4	Invalid output file.
8	Generic error.
12	Missing or expired license.

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

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

IDERA SQL Data Compare

Welcome to IDERA SQL Data Compare, the tool of choice for thousands of SQL Server DBAs and developers around the globe.

Version 7 of IDERA SQL Data Compare is an updated, more efficient and robust comparison framework. There are no restrictions on the size of the tables / databases that can be compared while the memory requirements are kept at a bare minimum.

IDERA SQL Data Compare supports all editions of SQL Server from SQL Server 2000 to SQL Server 2016, including Azure v11 and v12. It allows you to compare and synchronize data across databases, versions, on-premise SQL Server or on Azure.

We are indebted to you, users of our tools, for all the comments and suggestions you have given us over time – we truly appreciate your help and attribute much of our success to you! Please continue to help us in our quest to make IDERA SQL Data Compare even better by sending us unfiltered feedback on the current version and suggestions for the future versions at support@idera.com.

Everything in this document, unless it is explicitly stated otherwise, applies to all editions of IDERA SQL Data Compare.

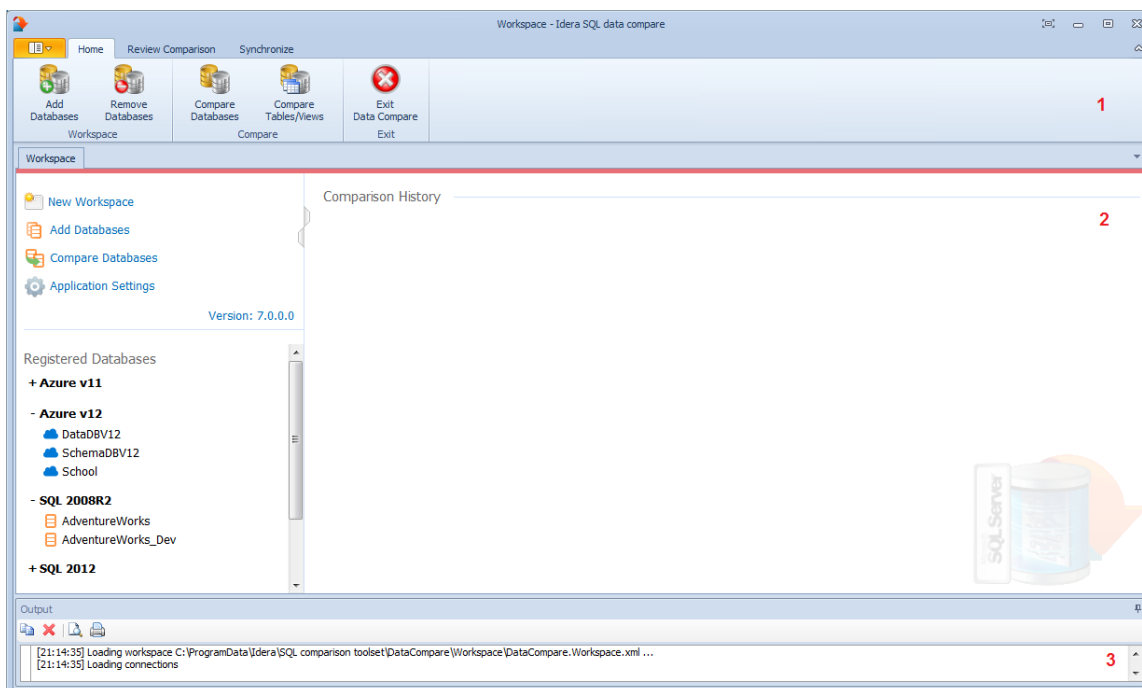
Total compare from data to schema with IDERA [SQL Comparison Toolset](#). [Learn more >>](#)

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

SQL Data Compare Application window

IDERA SQL Data Compare divides the main application window into three areas:

1. **The ribbon.** The context sensitive ribbon provides all the command buttons and icons you need to do your job quickly. The ribbon is organized in three distinct tabs each of which corresponding to the main steps of the database comparison and synchronization process and "housing" the relevant command buttons.
2. **Main panel.** The main panel contains a permanent tab called the **Workspace** and a tab for each active comparison session.
3. **Output window.** The output window provides a running log of the tasks being performed in IDERA SQL Data Compare.



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

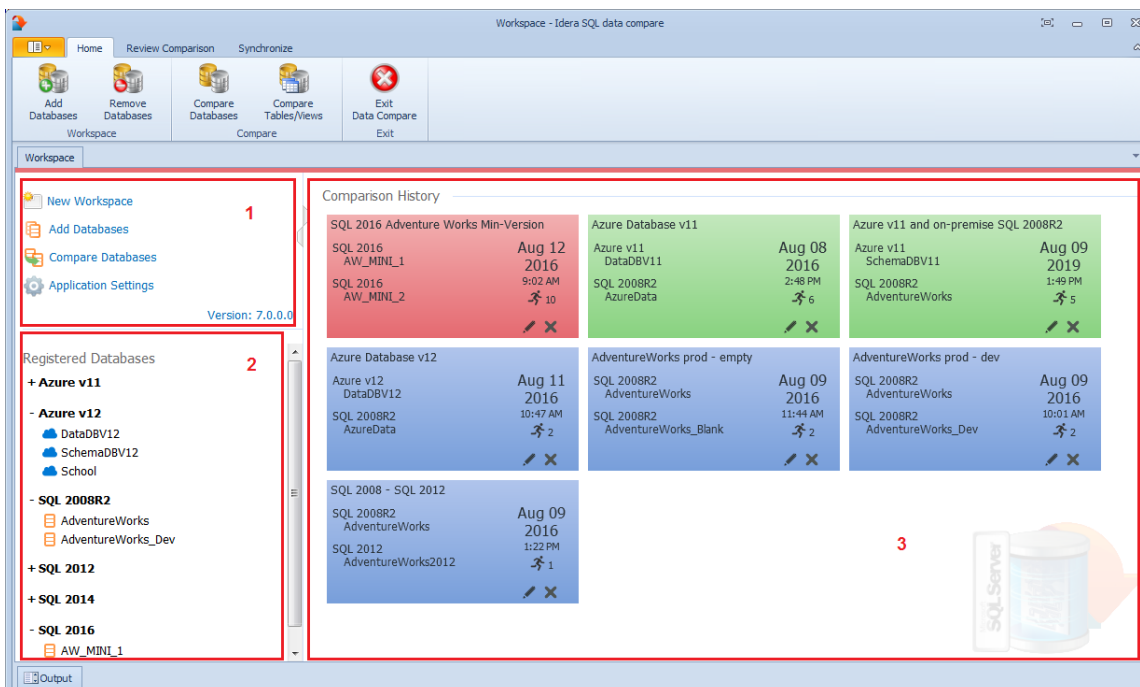
IDERA Website	Products	Purchase	Support	Community	About Us	Resources	Legal
-------------------------------	--------------------------	--------------------------	-------------------------	---------------------------	--------------------------	---------------------------	-----------------------

SQL Data Compare Workspace

IDERA SQL Data Compare workspace is the storage and the interface for saving and managing SQL Server databases, credentials, comparison sessions and other UI options.

The Workspace window is divided into the following sections:

- **Workspace actions.** This section provides links that allow you to create a new workspace, add databases to the current workspace, compare database and view application settings. The **New Workspace** action resets the current workspace by erasing the comparison history and the registered databases.
- **Registered Databases.** Shows the list of servers/databases that have been registered. On mouse over two action links appear for a registered server: **Edit** allows you to change the connection settings for that server and/or add and remove databases; and **Remove** un-registers a server (deletes all the server information from the workspace).
- **Comparison History.** Lists up to 25 comparison sessions from the most recent one to the oldest one. To re-open a comparison session, simply click it. The two links at the bottom-right of each session allow you to edit a session or delete it. For each session, data compare displays the following information:
 - The left and the right databases included in the session. These are the databases that are compared when you launch the session.
 - The date and time when the session was created
 - The session execution count, which indicates the number of times to session has been opened. A new sort order, under the application settings, allows you to arrange the sessions in the workspace by execution count, so that the ones that you use the most are always at the top. You can also change the execution count by editing the session.

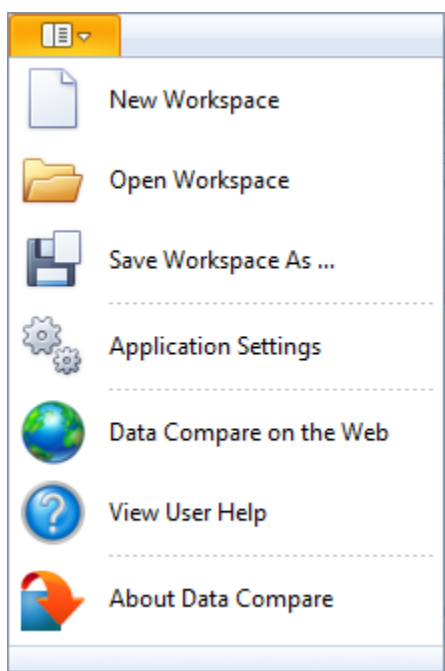


Saving and Loading Workspaces

Data Compare workspace stores the data in a set of xml files. There is always one master file

that keeps SQL Server databases and credentials, and a separate xml file for each comparison session. The workspace is saved by default in a subfolder under the common folder for application data that is used by all users. It is usually found under \ProgramData\Idera\SQL comparison toolset\DataCompare\Workspace.

The workspace is managed automatically, so no action is required to save it or load it. However, if you wish to have multiple copies of the workspace, you can save it to a custom path. Functions for saving and loading the workspace are located in the quick menu of the application button:



i When you choose to save the workspace in a custom location, you specify the path and the name of the workspace master file. The session files will be saved in the same directory as the master file, but their names will be auto-generated. Similarly, when you load a workspace from a custom location, you select the workspace master file. Data Compare expects the session files to be in the same folder as the workspace master file.

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

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

SQL Data Compare Add Databases

Before comparing databases, you will need to add them to the workspace. The IDERA SQL Data Compare Add Databases functionality can be accessed from the ribbon (the Home tab) or from the **Add Databases** action link at the top left section of the Workspace tab.

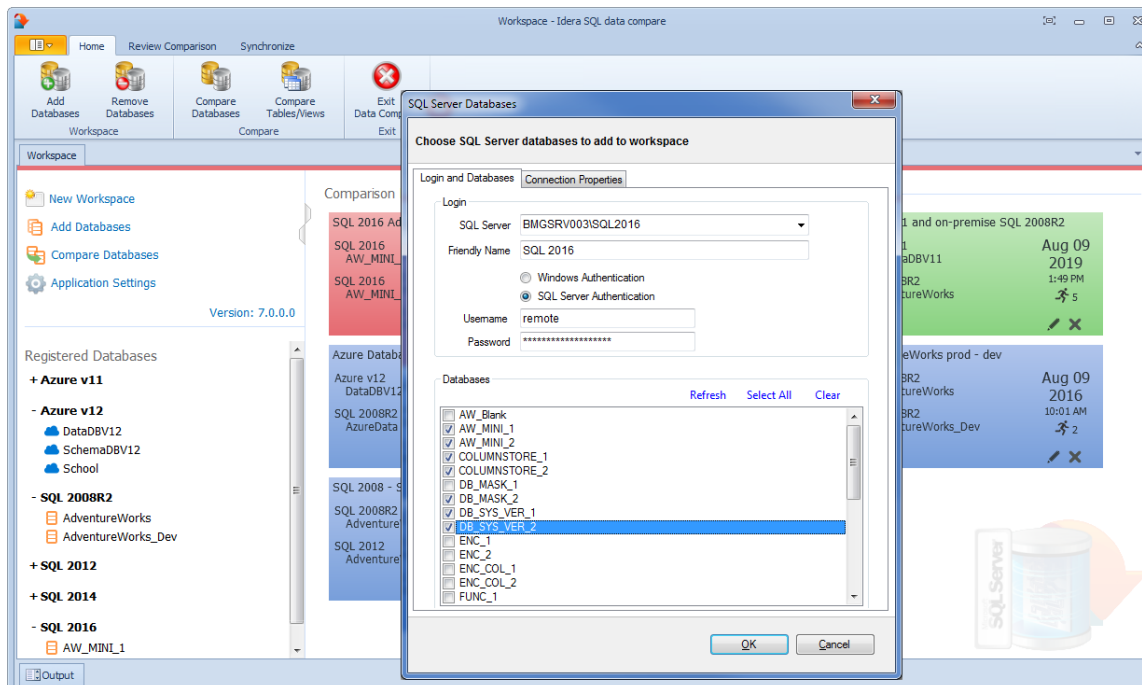
On the Add Database dialog window choose the server where the database(s) you want to add resides. You can click on the **Refresh** item of the SQL Server drop down to discover the SQL Server instances running on your network. If the application is not able to discover the SQL Server instance you are looking for either because it is not exposing itself, or it is not running on your local area network, or it is not listening on the default port then you can type in ServerName (or IP address)\Instance Name on the drop down box. If the SQL Server is not listening on the default port you can specify a custom port on the following format:

[ServerName or IP Address]\[Instance Name],PortNumber

Once you have chosen the server and the authentication method (including login name and password if you chose SQL Server Authentication) you can click on the **Refresh** link at the top of the Databases box and the application will connect to the server using the information you provided and populate the Databases box with the list of all user databases in that server. You can select one or more databases to add to the workspace.

By default IDERA SQL Data Compare will use the connection properties set at the Application Settings, however, you can change the connection properties for this server only if you need to.

In the other properties edit box, you can specify additional connection properties as name=value pairs separated by semicolons: name1=value1;name2=value2;...



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

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

SQL Data Compare Remove Databases

Using IDERA SQL Data Compare, there are two ways to remove a database from the workspace:

- Move the mouse over the name of database you want to remove under the Registered Databases on the left panel of the Workspace. The database name is highlighted and a **Remove** link is available.
- Click **Remove Databases** on the Home tab of the IDERA SQL Data Compare ribbon. A remove databases dialog window appears showing all the registered servers/databases. Select the database you want to remove, and then click **Remove Selected Database...** at the top right corner.

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

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

SQL Data Compare Application Settings

Application settings allow you to tweak the behavior of the IDERA SQL Data Compare and can be accessed from the drop down menu off of the top left corner product icon or from the Application Settings link at the top left corner on the Workspace tab.

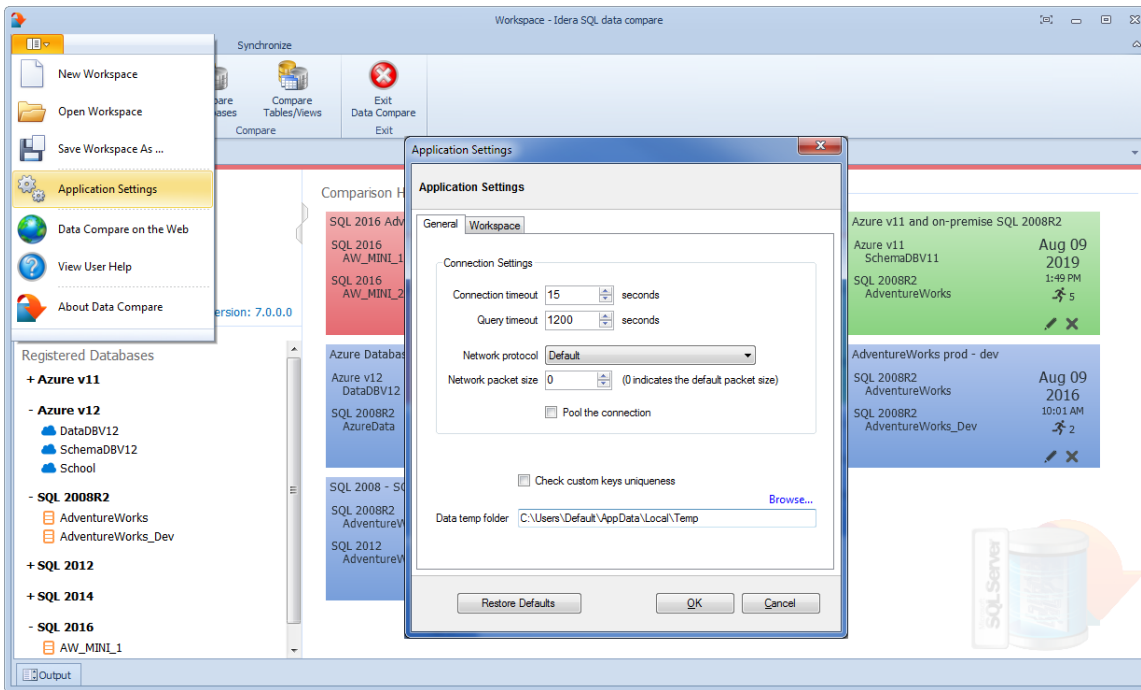
Under the General tab, you will find:

- **Connection Settings** If you are reading this you are already familiar with the database connection settings. Under normal circumstances you do not need to change any of these settings.
- **Check custom keys uniqueness** Before comparing the data IDERA SQL Data Compare will pair the rows based on a unique key that is automatically determined. However, you can choose to pair the rows based on some custom key that you define either because the tables in question do not have a predefined unique key or because you wish to use a different key. In this case there is an option to validate the chosen key. By default the validation simply checks for type compatibility. If you wish to ensure that the chosen key is unique you can check this option. Please note however that depending on the size of those tables the validation for uniqueness may take from a few seconds to minutes to complete.
- **Data temp folder** Indicates the temp folder that Data Compare uses for row and script serialization. Note that if you change the temp folder, you need to restart Data Compare for the new setting to take effect.

Under the Workspace tab you will find:

- **Sessions** Contains settings related to comparison sessions:
 - **Save sessions** By default it is checked. It instructs the IDERA SQL Data Compare to store comparison sessions for future use. A stored session contains all the necessary information to allow you to repeat the comparison with one click. Stored sessions will appear on the main panel of the Workspace tab.
 - **Max sessions kept in history** By default IDERA SQL Data Compare will store the last 25 comparison sessions. Although there is no noticeable performance difference on application launch related to the number of sessions stored you may choose to set this parameter to a lower number. Allowable values are 1 to 25.
 - **Sort sessions by** Change the order that stored sessions appear on the workspace tab.
- **Layout** Allows you to change the color schema of a session. You can choose from built-in colors or pick custom ones.

To revert all settings back to their default values, click **Restore Defaults**.



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

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

SQL Data Compare Comparing

Comparing the data in two databases is part of the core functionality of IDERA SQL Data Compare.

The databases you wish to compare (or tables within them) can be virtually anywhere – as long as you have access to them you can compare and synchronize them. Furthermore, the databases do not have to be of the same version or flavor – you can compare between any “flavors” of the SQL Server: Express, Developer, Standard, Enterprise.

In the case of cross version comparison IDERA SQL Data Compare will map the data types that can be mapped and ignore the columns that are of a type that cannot be mapped (let's say a data type that only exists in SQL Server 2012 but does not exist in 2005 and there is no equivalent type to which it can be mapped).

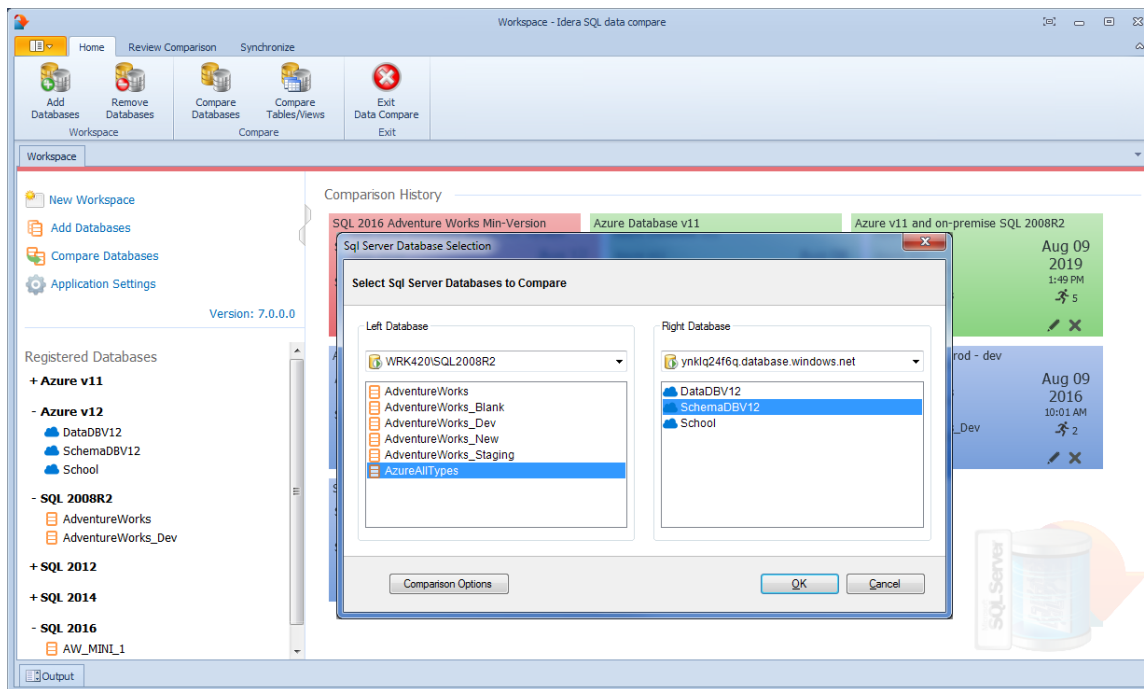
Total compare from data to schema with IDERA [SQL Comparison Toolset](#). [Learn more >>](#)

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

SQL Data Compare Comparing Databases

To start a comparison between two databases click on the IDERA SQL Data Compare **Compare Databases** button on the Home tab of the ribbon. The SQL Server Database Selection dialog window that appears allows you to chose the server(s) where the databases you wish to compare reside as well as the databases themselves.

From this window, you can also access the [Comparison Options](#) that are applied during the comparison.



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

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


SQL Data Compare Comparing Selected Tables

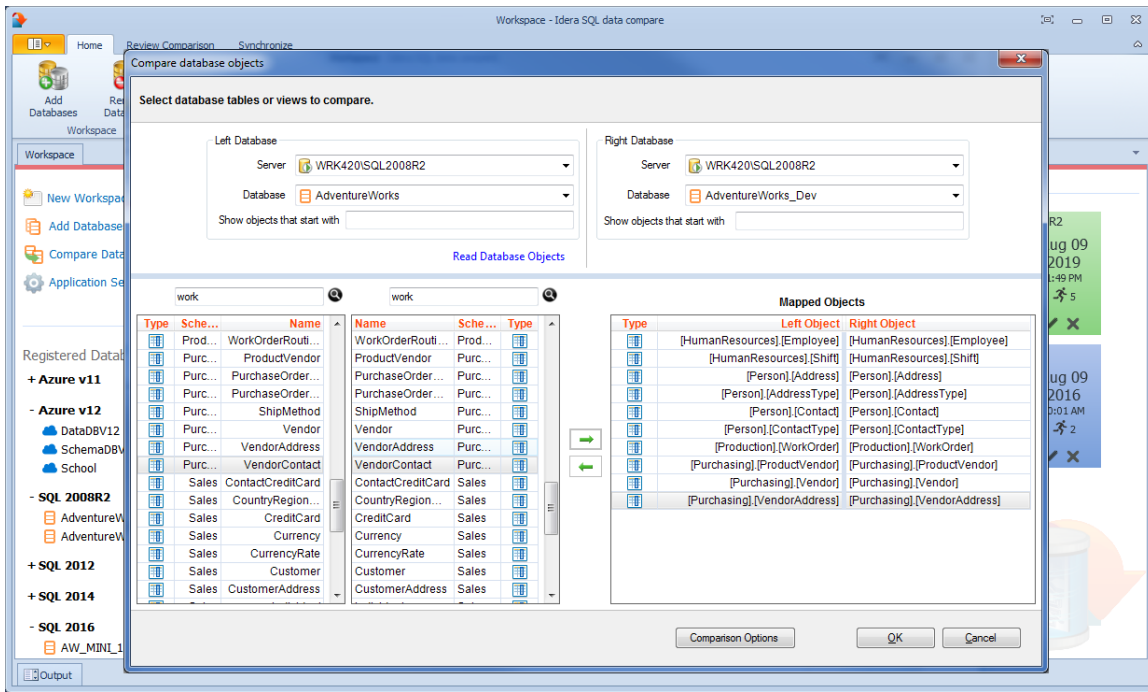
To start a comparison between select tables and/or indexed views click on the IDERA SQL Data Compare **Compare Tables/Views** button on the Home tab of the ribbon. On the Object Compare dialog window that appears you can first select the servers and databases on which the objects you wish to compare reside. In case there is a large number of objects in those databases you may wish to specify a pattern for the name of the objects you are looking for (ex. all tables that start with a). That will make it easier to find those objects.

- **Comparing Views.** IDERA SQL Data Compare, in addition to tables, has the ability to compare views as well. The option **Compare and Synchronize Views** determines whether views should be included in the comparison and synchronization. However, use this option with caution. There are several limitations that SQL Server imposes on views:
 - Not all views are updatable.
 - If a view has indexes, Data Compare will select one in the same order as table's indexes; otherwise you must define a custom index in order to be able to compare the view.
 - Since SQL Server does not support SET IDENTITY_INSERT ON|OFF on views, the insert statements might fail if one of the view's underlying tables contains an identity column.
 - Data Compare cannot synchronize views that contain large binary fields such as varbinary(max) and image, or views with large text field such as varchar(max), nvarchar(max), text and ntext.

Once you have made your selections above you can click on the **Read Database Objects** link. IDERA SQL Data Compare will read the schemas of the two databases and populate the two left bottom windows with the list of tables / views available.

Now you can map those objects one at a time irrespective of the object names. Select an object on the left, then select an object on the right and click the left to right arrow button to map those two objects.

 The [SQL Data Compare Comparison Options](#) affect the ability to compare objects. For example, if you clear the **Compare and synchronize tables** option, then you cannot compare the tables but can only view and map them. The same holds true for indexed views which by default are not included in the comparison.



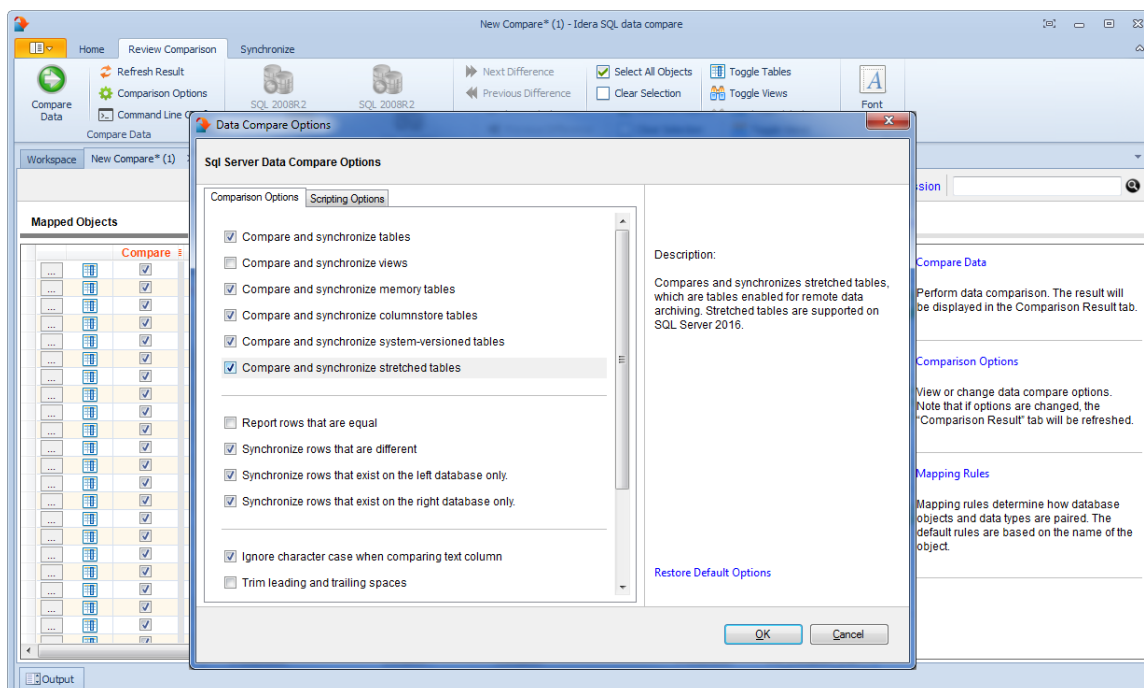
Total compare from data to schema with IDERA [SQL Comparison Toolset](#). [Learn more >>](#)

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

SQL Data Compare Comparison Options

IDERA SQL Data Compare exposes a series of options that allow you to customize the way the comparison is performed. On mouse over a description of the highlighted option is displayed on the right panel of the options window.

You can access the Data Compare options from the Compare Databases and Compare Tables dialog windows before the object mapping operation. After the objects (tables and indexed views) have been mapped you can access the Comparison Options from the Review Comparison tab of the ribbon as well as from the action links on the right panel of the Mapped Objects tab.



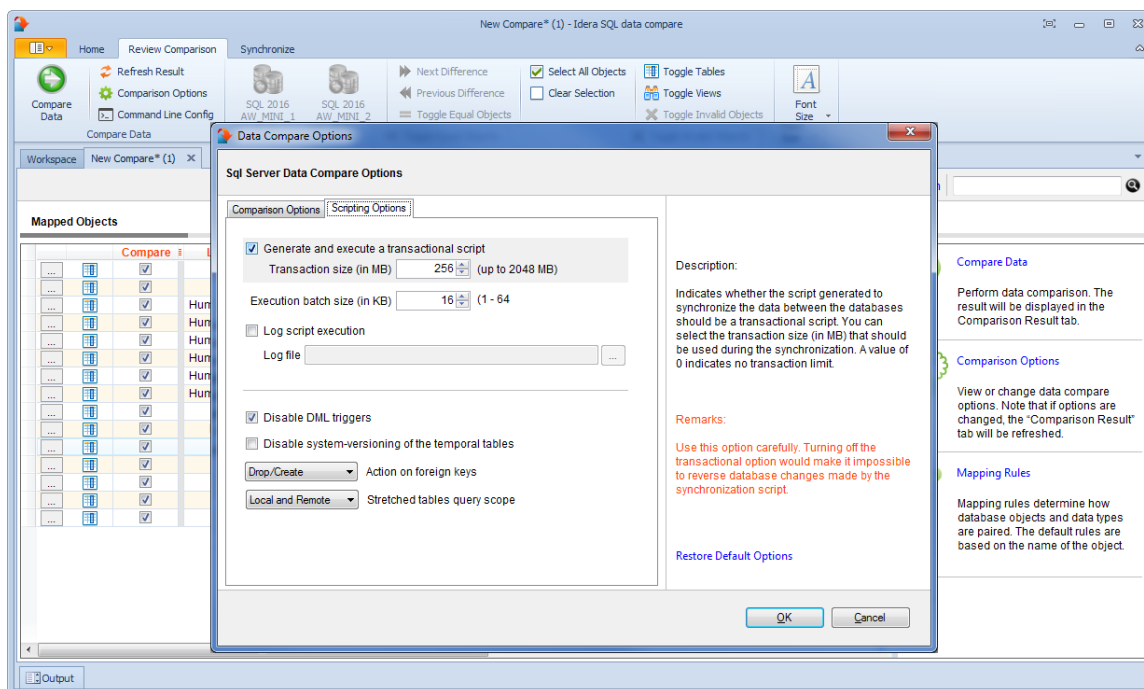
Total compare from data to schema with IDERA [SQL Comparison Toolset](#). [Learn more >>](#)

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

SQL Data Compare Scripting Options

Scripting Options are listed on the second tab of the IDERA SQL Data Compare Options window. Those options affect the way the synchronization script is generated but do not affect the comparison operation.

- Transaction size.** Determines the size of the script that will be wrapped in a single transaction. Depending on the size of the databases/tables being compared the synchronization script may get to be extremely large (for example, millions of insert/update/delete statements are needed to make database DB2 the same as DB1). Wrapping such large scripts in a single transaction may not be possible due to SQL Server transaction size limitations, and may not be acceptable for performance and other reasons. So, it is necessary to split the script into smaller "chunks" and wrap each of those "chunks" in individual transactions. If the synchronization process is canceled or fails due to errors, only the current open transaction is rolled back. Changes committed prior to that transaction are permanent and cannot be reversed. To have an unlimited transaction size specify 0 as the max value. Having an unlimited transaction size ensures that all changes in the database are rolled back if an error occurs, but might be rejected by SQL Server. Finding the right transaction size depends on many factors - please adjust this option carefully to suit your specific environment.
- Batch size.** Determines the size of data that is transmitted to SQL Server in one execution. A transaction normally contains many batches, but the batch itself is a single unit of work that cannot be divided any further.



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

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

SQL Data Compare Preparing the Comparison

Before comparing the content of two databases, IDERA SQL Data Compare will analyze the schemas of both databases (or the selected tables). If you manually map the tables between the two databases, SQL Data Compare will use that mapping as is and proceed with analyzing the pairs of tables. If you compare whole database it will first perform an automatic mapping of the tables based on their names (owner name + table name) - every table on one database is paired with a table (when possible) on the other database. Then, it proceeds to analyze the pairs of tables to map individual columns, determine the available comparison keys and choose a key that will be used to pair rows during the data compare.

During this preparation phase you will be able to customize the object mapping, include and exclude certain tables and certain columns within those tables, choose comparison keys and define data filters to compare only subsets of rows from certain tables.

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

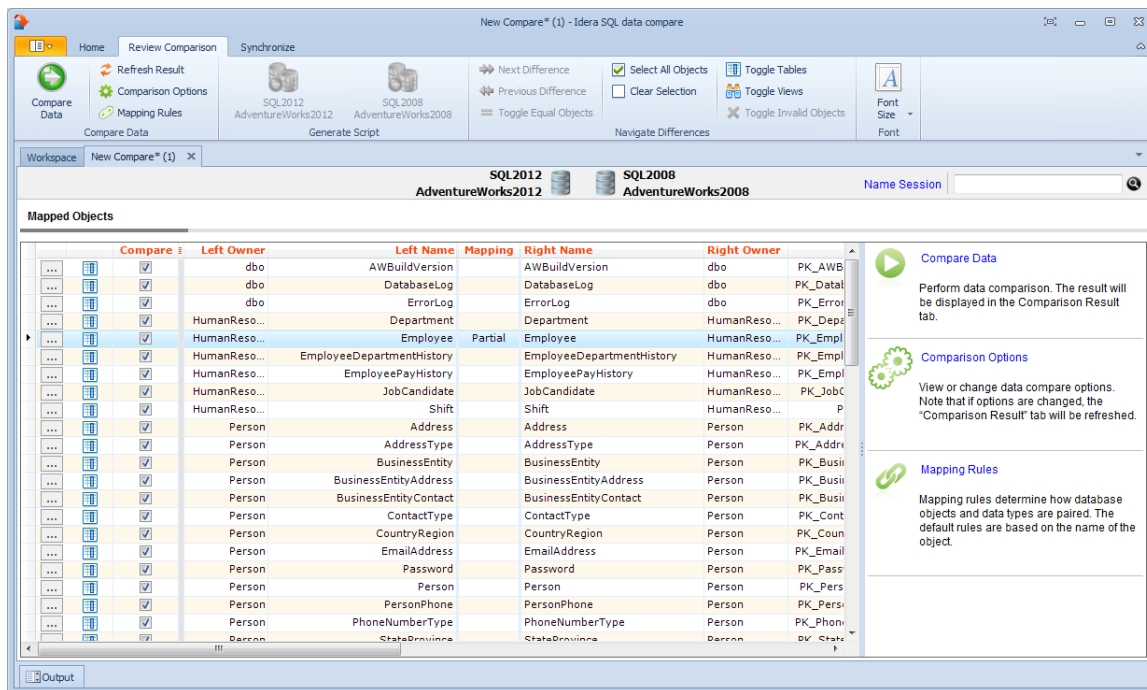
IDERA Website	Products	Purchase	Support	Community	About Us	Resources	Legal
-------------------------------	--------------------------	--------------------------	-------------------------	---------------------------	--------------------------	---------------------------	-----------------------

SQL Data Compare Mapped Objects

The results of the schema analyses and the automatic mapping operation are displayed on the IDERA SQL Data Compare Mapped Objects tab. For each pair of mapped objects the following information is displayed:

- **Object Type.** The object can be a table or an indexed view. The type is indicated through the small icon in the second column of the grid.
- **Compare.** A checkbox indicating whether the pair should be included in the comparison. By default, all mapped pairs will be included in the comparison. You can quickly select / deselect all objects using the **Select All Objects** and **ClearSelection** commands in the ribbon.
- **Owner and object name.** For both sides. In a default mapping the Left Owner and Left Name would be identical to respectively Right Owner and Right Name.
- **Mapping Status.** A blank value indicates that the objects have been fully mapped, that is, there was a complete match of columns. A **Partial** value indicates that some columns were not matched and will not be included in the comparison.
- **Left/Right Keys.** Those are the unique keys that have been selected by SQL Data Compare to be used in the mapping of the rows during the data comparison. If those two fields are empty for a given pair that means that SQL Data Compare was not able to identify a unique key that could be used. If you do not choose a custom key for this pair then those objects will not be included in the comparison.

i **Changing the comparison options.** Please note that changing the comparison options after the mapping operation has completed will trigger a refresh of the Mapped Objects tab. For example if you uncheck the **Compare and synchronize tables** option the table pairs will be removed from the mapped objects grid.



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

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

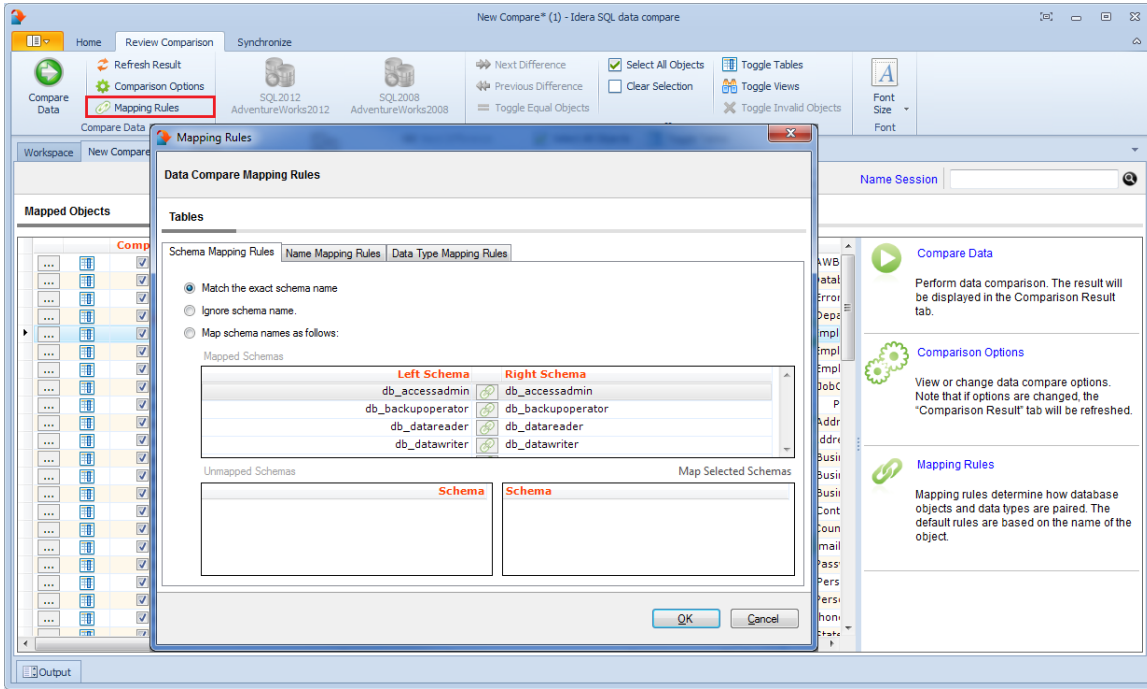
SQL Data Compare Mapping Rules

In most cases the default mapping mechanism that pairs the objects based on the schema name and the object name is what is needed. However, there are often scenarios in which the same table may be owned by different schemas in different databases, or the tables on the development server for example may have the names prefixed with dev_ whereas the same tables in the production server do not have that prefix. In such cases you may want to define certain mapping rules that would allow you to pair objects that would otherwise not be paired together by IDERA SQL Data Compare.

The Mapping Rules can be accessed from the ribbon or from the action links on the right panel of the comparison tab.

There are three types of rules you can set:

- **Schema Mapping Rules.** By default SQL Data Compare performs an exact match of the schema names, however you have two additional options:
 - **Ignore schema name.** In this case table [dbo].[T1] on database 1 will be paired with table [user1].[T1] on database 2, in other words the schema name is irrelevant.
 - **Manually map the schema names.** This allows you to choose which schema name from database 1 to map to which schema name from database 2. If you map schema [S1] from DB1 to schema [S2] from DB2 then [S1].[T1] from DB1 will be paired with [S2].[T1] from DB2. To un-map a pair of schemas double click on the icon in the middle of the schema pairs on the Mapped Schemas box; that will break the mapping and move the pair of schemas to the **Unmapped Schemas** area. To map two schemas click on the schema on the left to select it; click on the schema you want to map it to on the right box and then click on the **Map Selected Schemas** link.
- **Name Mapping Rules.** By default SQL Data Compare performs an exact match on the object names, however, you can choose to ignore certain prefix and/or postfix. For example you can indicate that on the left database the dev_ prefix and the _old postfix and on the right database the qa_ prefix and the _new postfix should be ignored. In such case table [dev_T1_old] from the left database will be paired with [qa_T1_new].
- **Data Type Mapping Rules.** By default data types are mapped by compatibility. If column C1 of table T1 on DB1 is of the type varchar but the same column of table T1 on DB2 is of type nvarchar then those two columns will be paired together since those two data types are compatible with each other. However, you have the option to enforce an exact type match for your comparison. When you choose the **Map data types by name** then two columns will only be paired if they are of the same exact type.



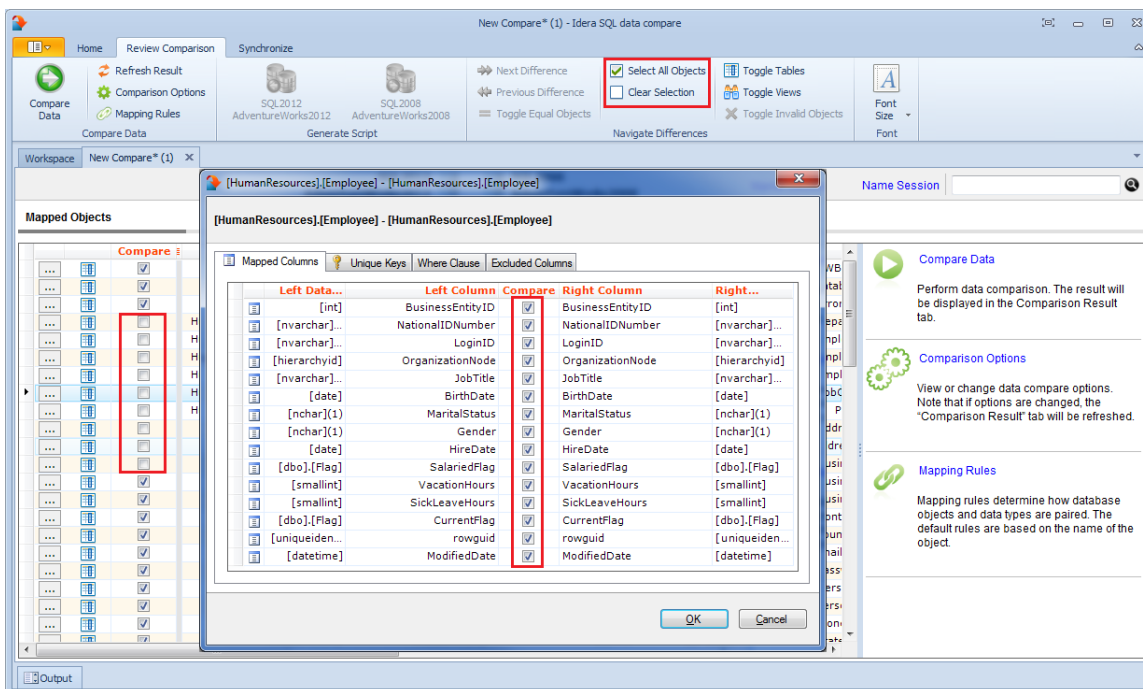
Total compare from data to schema with IDERA [SQL Comparison Toolset](#). [Learn more >>](#)

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

SQL Data Compare Include/Exclude Objects

The first step in customizing your comparison after the objects have been paired is to choose which objects you wish to include in the comparison and for each object pair which columns you wish to include or exclude. The options available in IDERA SQL Data Compare:

- Include/Exclude object pairs.** By default all objects (tables and indexed views) are marked to be included in the comparison. You can decide which pairs you want to include/exclude by checking / un-checking the **Compare** checkbox for each object pair. To de-select or select all object pairs in one click you can use the **Select all objects** and **Clear selection** buttons in the ribbon.
- Include/Exclude mapped columns.** For each object pair you can click on the details button (the first column of the Mapped Objects grid) to further customize the comparison. By default all the mapped columns are marked to be included in the comparison. However, you may choose to exclude certain columns from the comparison (example: image or text type columns that would take a long time to compare). To exclude columns simply uncheck the **Compare** checkbox for that column pair. Important: note that the column(s) that are part of the key that has been selected to be used as the comparison key will always be included in the comparison regardless of whether you have checked or un-checked them in the Mapped Columns tab.



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

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

SQL Data Compare Comparison Keys


The IDERA SQL Data Compare Unique Keys tab is divided in two main sections. The top section contains two boxes with the list of unique keys that are potential candidates to be used as the comparison key for each table in the pair. Initially the boxes contain only the predefined unique keys that SQL Data Compare found, but you can add other unique keys that can be a combination of any and all the columns on the given table. The bottom section contains a box with the pair of unique keys that have been selected as the key to be used for the data comparison operation. SQL Data Compare picks the comparison key in the following order (whichever is found first):

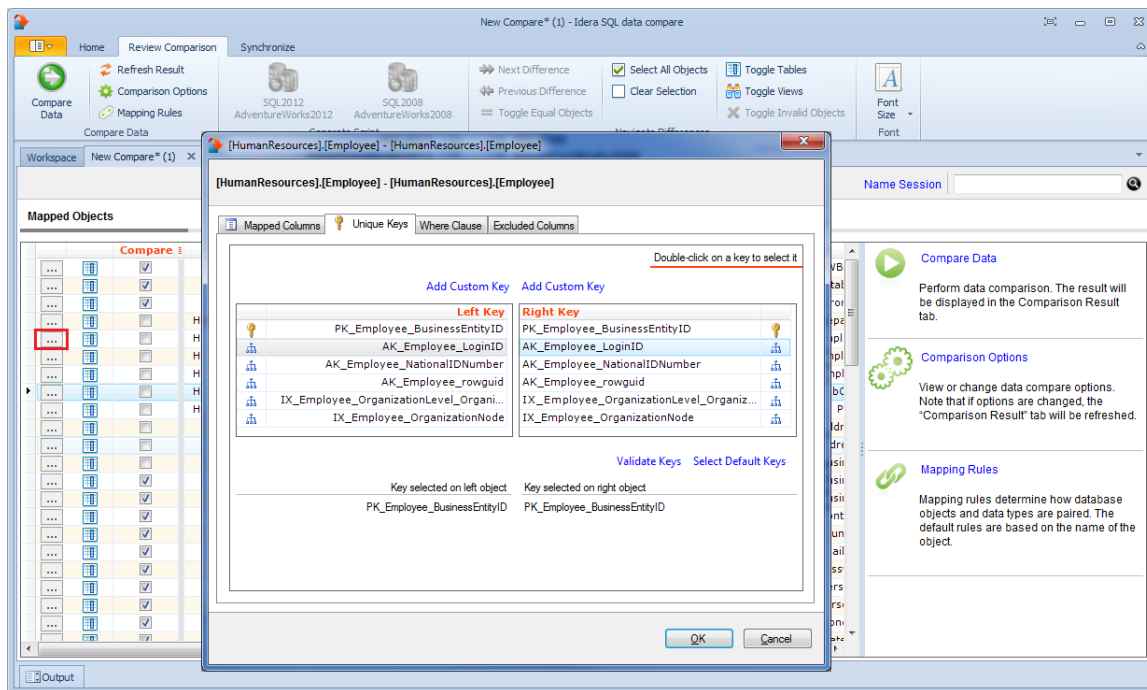
1. User-defined keys. User-defined keys (or custom keys) takes precedence over the built-in keys and indexes.
2. Primary Key
3. Unique Constraint
4. Unique Index. Indexes that are not unique are not considered.

When the selected keys are not those that were picked by SQL Data Compare you should validate the keys by clicking on the **Validate** button. If the selected keys are not valid you will not be able to compare the those tables. By default the keys are only validated for type compatibility. If you wish to validate your custom defined keys for uniqueness then you should check that option in the Application Settings. User-defined key validation is done as follows:

1. Columns participating in the the user-defined key in both objects in the pair must have the same names
2. Columns that participate in the key must have the same data type. Only the name of the type is checked. The length and other type properties are not considered.
3. The uniqueness of the key is checked only if the **Check custom key uniqueness** option in the application settings under Miscellanies section is checked.

Selecting a key for the comparison. Double-clicking on a unique key on the top section will replace the selected key with the new one.

 User-defined keys are not supported on Memory Tables.



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

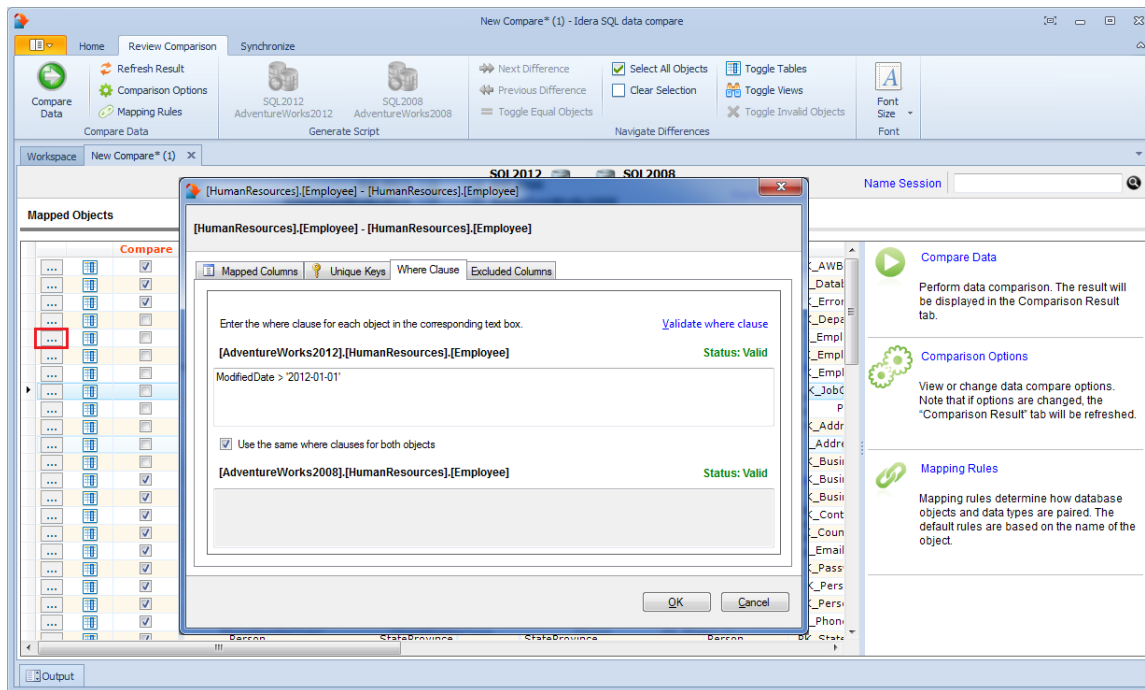
IDERA Website	Products	Purchase	Support	Community	About Us	Resources	Legal
-------------------------------	--------------------------	--------------------------	-------------------------	---------------------------	--------------------------	---------------------------	-----------------------

SQL Data Compare Where Clause

For large tables the comparison operation may take time since all the rows from both sides will have to be read and compared. So, if you know for example that out of 10 million rows that 9 million of them are equal it would be very inefficient to compare those equal rows. Furthermore, regardless of how many rows may be different you might need to do a quick comparison of just a subset of rows, for example, only rows that were modified today.

IDERA SQL Data Compare allows you to set where clauses for each object being compared so that you can read and compare only the rows you need to. When defining a where clause by default the same filter will be applied to both tables in the pair, however, if you wish to define different filters for each of the tables you can un-check the **Use the same where clause** checkbox. We suggest you validate the data filters to ensure they have been defined correctly.

i Do not write the "where" word on the clause box. Simply write the condition the way you would write it in T-SQL (see the screen shot below for an example).

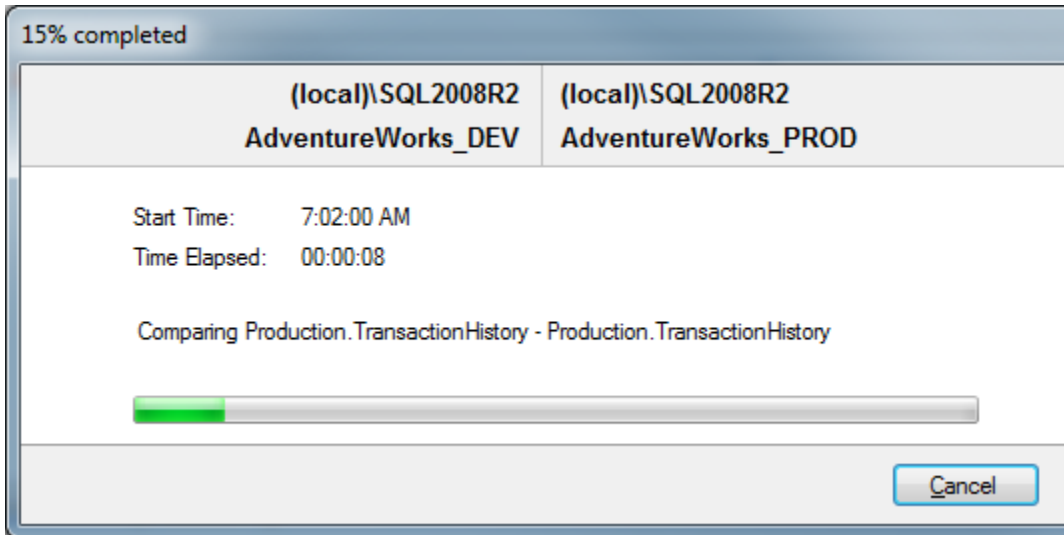


Total compare from data to schema with IDERA [SQL Comparison Toolset](#). [Learn more >>](#)

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

SQL Data Compare Comparing Data

After completing the preparation phase of the comparison you are ready to compare the data in all those mapped objects (tables and views). To start the comparison either click on the **Compare Data** button on the ribbon (under the Review Comparison tab) or click on the **Compare Data** on the right panel of the Mapped Objects window. A progress window displays the start time, elapsed time and the names of the objects being compared. If you cancel the comparison operation IDERA SQL Data Compare will display partial comparison results.



On completion of the data compare operation the results are displayed on a new window called "Comparison Result" within that comparison session's tab. The "Comparison Result" window is divided into five sections:

1. **Comparison summary.** Displays the list of table/view pairs that were compared indicating the total number of rows on each of the tables in the pair, the number of rows that were found to be equal ("Equal Rows"), number of rows that were found only on the left/right table/view and could not be paired with a corresponding row on the right/left table/view ("Left Rows", "Right Rows"), and number of rows that exist in both the left and the right table/view but have at least one column that contains a different value on the left table/view from the right.

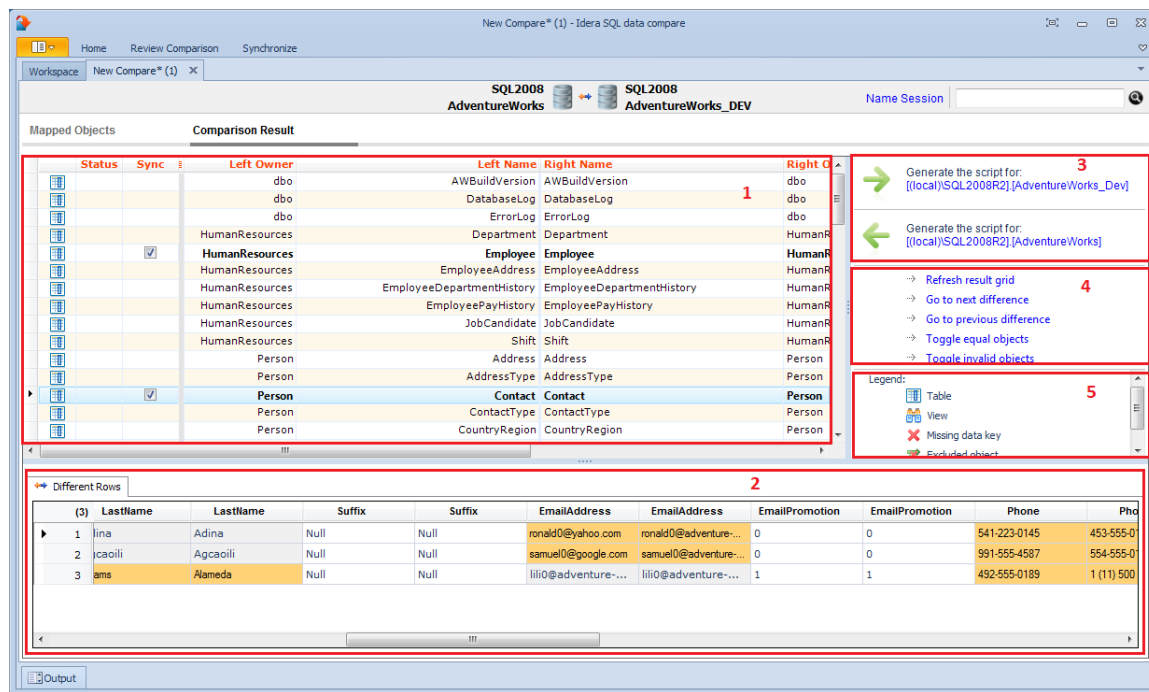
Notes on the comparison summary grid:

- Database objects that have some differences are displayed in bold.
 - The check box, which indicates whether the a pair of objects can be synchronized, appears only for the object that have a valid data compare key and for which the comparison completed without error.
 - If the comparison is interrupted, SQL Data Compare displays the result of the process up to the point of cancellation and marks the objects that were not compared with a question mark.
 - In the comparison summary grid, an invalid object is considered an object without a valid key.
2. **Comparison details.** When you select a pair of objects that were found to be different in the comparison results grid, depending on the differences found one to three tabs will appear on the bottom section of the Comparison Results window: Left Rows, Right Rows, and Different Rows. For the different rows the paired columns from both tables are shown

next two each other so that you can easily see the differences. You can zoom in on a particular row to take a closer look - right click on the row you wish to inspect and click on **View Row Details**.

Notes on details grid:

- Columns colored in the light-blue represent columns that participate in the selected data key for that particular table or view. The key label appears in the column header as well.
 - Cells colored in orange represent data that found different.
 - The grey shades that alternate between columns are for displaying purposes only to distinguish between pairs of mapped columns.
 - By default rows that are equal are not displayed in the row grid unless the data comparison option **Report rows that are equal** is checked.
3. **Generate synchronization script.** Those links will trigger the generation of the synchronization script for one direction or the other.
 4. **Comparison action links.** The action items provided in this section are also available on the Review Comparison tab of the ribbon. Clicking on the **Refresh Results** link will trigger a re-compare of the data.
 5. **Legend.** Provides a list of symbols used in the data compare results grid. The first two symbols, Database table and Database view are used in the first column which is un-named and indicate whether the objects in that pair are tables or indexed views. The other three symbols are used in the **Status** column to indicate cases when the pair of objects was not included in the comparison.



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

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

SQL Data Compare Synchronizing

Identifying the data differences between two databases helps you as a DBA/Developer in many ways but often your job does not end there - you need to migrate those changes from one database to another. IDERA SQL Data Compare allows you to do just that - after comparing the contents of two databases you can use it to generate the synchronization script - a T-SQL script that will update the target database (or just the selected tables/rows) to make it the same as the source database.

The data synchronization operation consists of three key actions:

- **Update.** All rows that exist in both sides will be updated on the target database to match the source.
- **Insert.** All rows that exist only on the source database but not on the target will be inserted on the target database.
- **Delete.** All the rows that exist only on the target database but not on the source will be deleted from the target database.

In case you do not wish to delete the rows that only exist on the target database you can change the comparison options so that those rows are not included in the synchronization.

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

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

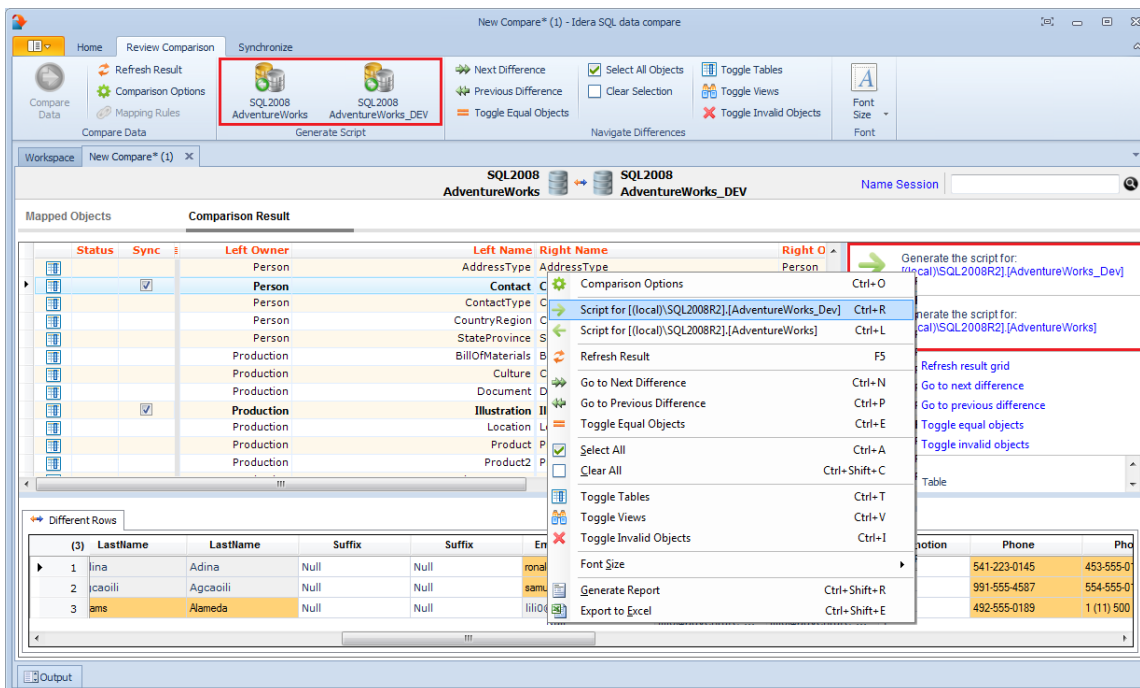
SQL Data Compare Generating Synchronization Script

Once the data comparison operation is finished you are now ready to start the data synchronization operation.

By default, all tables that are found to have differences and all the different rows in those tables are automatically marked to be included in the synchronization. You can however easily change which tables, and which rows in those tables, should or should not be included in the script by checking or un-checking the **Sync** checkbox.

Once you have made your selections you can generate the synchronization script to either change the database on the right of the comparison to make it the same as that on the left or the other way around. There are three ways to start the script generation:

- **Ribbon.** In the Review Comparison tab of the ribbon there are two big buttons in the middle under the grouping **Generate Differences Script**. Each button is labeled with the name of the server + name of the database for which the script will be generated.
- **Action Links on the right panel.** Are listed under the label **Generate the script for:.** Clicking on the [server].[database] link generates the synchronization script that will make that database the same as the other one.
- **Context menu.** When you right click on the comparison results grid a context menu will pop up. The context menu allows you to access the comparison options and trigger the generation of the synchronization script in addition of other context relevant actions.



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

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

SQL Data Compare Working with the Script

IDERA SQL Data Compare displays the data synchronization script on a new window within the comparison session tab. The new window name includes the [server].[database] for which the script was generated.

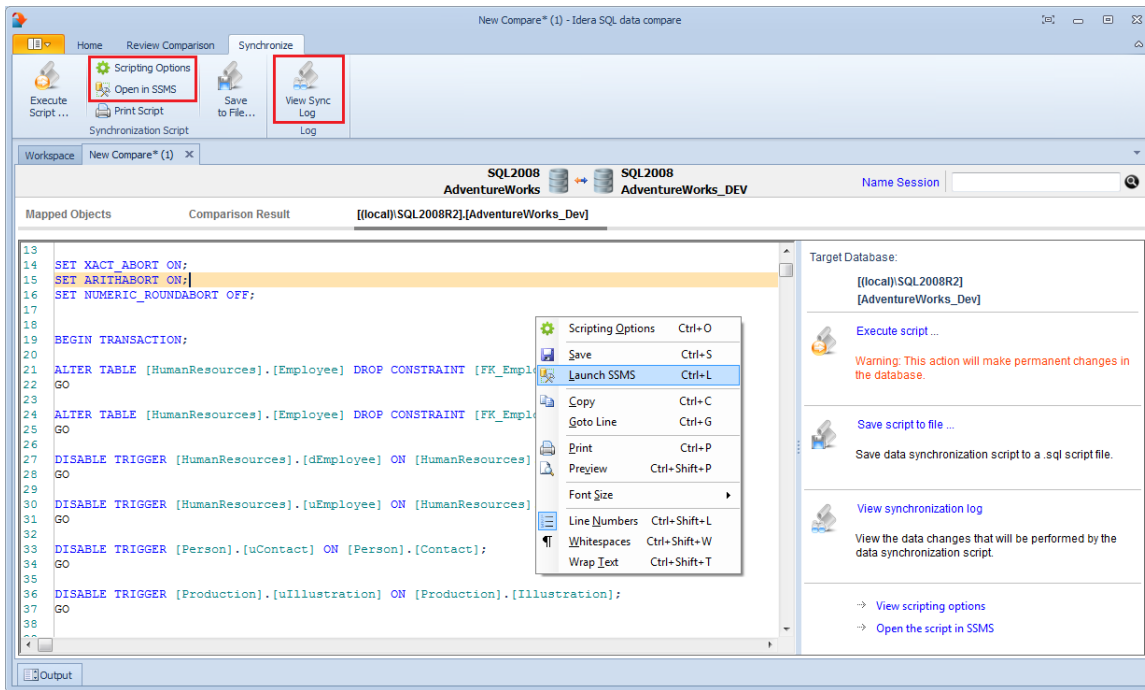
i If the data synchronization script exceeds 32MB, IDERA SQL Data Compare displays only the first 32MB-size block of the script in order to preserve the memory. However, if you choose to execute the script, the entire script and not just the first block is executed. If you want to view the whole script, you can save it to a file, and then open the file using a text editor.

A handy context menu allows you to save, copy, print, and add line numbers to the script as well as jump to a certain line in the script.

Here you can review the script to ensure that it is doing what you intend it to do. You may want to start the review from the Synchronization Log so that you can quickly identify the areas of the script that may require a closer look. you can access the synchronization log from the ribbon by clicking **View Sync Log** or from the action links on the right panel. The synchronization log contains a summary of the actions that you want to perform on the target database.

Other actions include:

- **Refreshing the comparison and regenerating the script.** While reviewing the script you may discover that you should have done something different during the preparation phase before running the compare. IDERA SQL Data Compare allows you to change the options at any point during the process and simply click **Refresh Result** on the ribbon to redo the comparison without having to go through the whole exercise of selecting columns and keys etc. Note however that this refresh action does not affect the synchronization script already generated. To refresh the script window, simply re-generate the script.
- **Opening the script in SSMS.** From the script window you can transfer the script directly to SQL Server Management Studio (SSMS) assuming you have SSMS installed on your machine. Simply right click anywhere on the script, and then click **Launch SSMS** or click **Open in SSMS** in the ribbon (under the Synchronize tab).



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

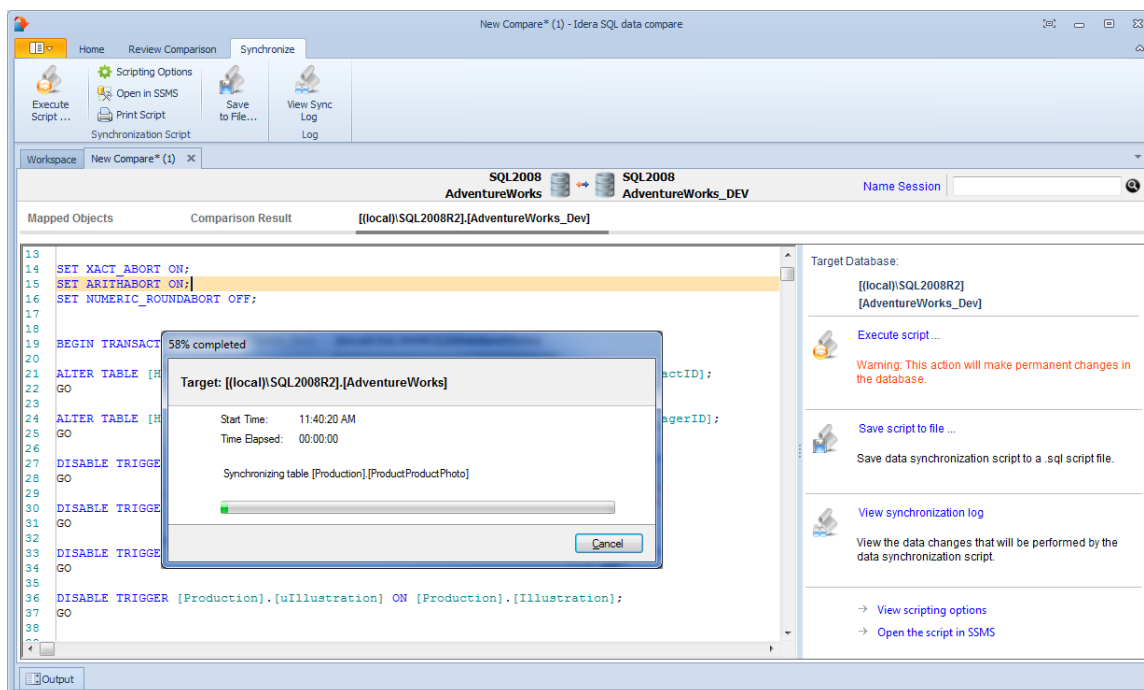
IDERA Website	Products	Purchase	Support	Community	About Us	Resources	Legal
-------------------------------	--------------------------	--------------------------	-------------------------	---------------------------	--------------------------	---------------------------	-----------------------

SQL Data Compare Executing Synchronization Script

Once you have reviewed and you are comfortable with the synchronization script you can execute it on the target database.

! This action will make permanent changes to the content of the target database. It is common sense and we strongly recommend that before executing the script you:

- Make a full backup of the target database;
- Save the script you are about to execute so that you know exactly what was done to the database and when.



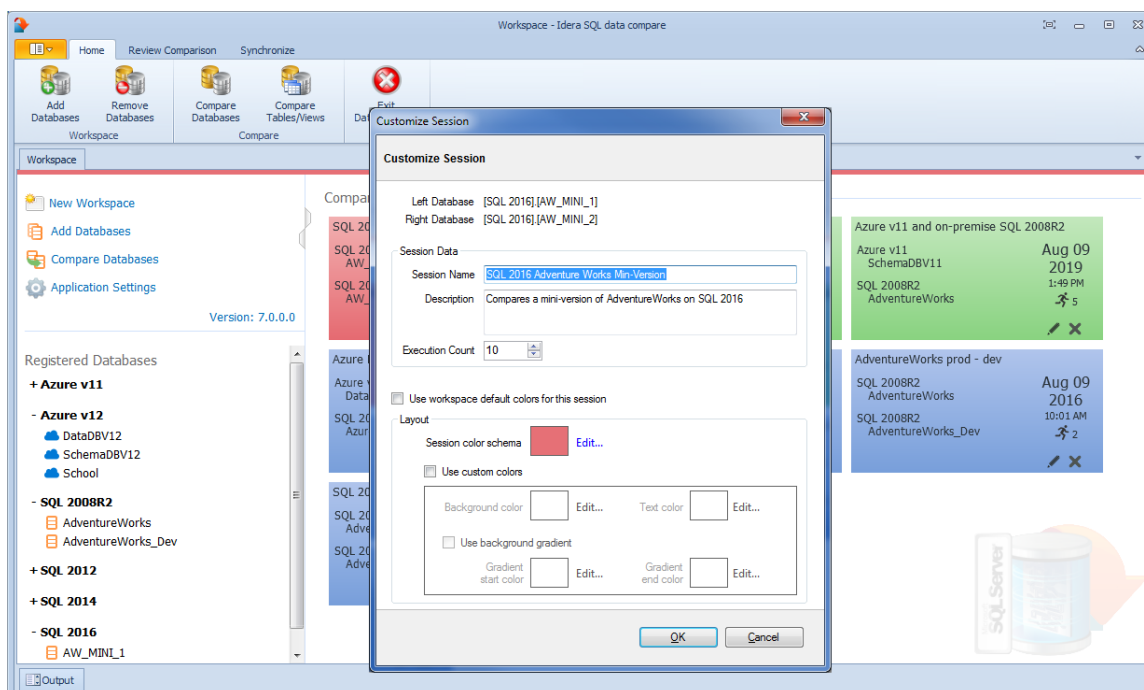
Total compare from data to schema with IDERA [SQL Comparison Toolset](#). [Learn more >>](#)

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

SQL Data Compare Comparison Sessions

The preparation phase of the data compare can be a time consuming process if you need to exclude certain objects, do custom mapping, define custom comparison keys and set data filters. IDERA SQL Data Compare automatically saves all your preparation work so that you can repeat the same comparison in the future with a single click without having to go through that process again.

You can also name the comparison sessions to make it easier to identify them. Up to 25 sessions are saved and displayed on the Workspace tab.



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

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

IDERA SQL Data Compare command line

IDERA SQL Data Compare command line provides for comparing and synchronizing the content of SQL Server databases via the command line. With the command line you can:

- Perform automated data comparison and synchronization;
- Compare and synchronize database content as part of your setup and deployment solution;
- Compare database content and execute the data synchronization script in batch files;
- Schedule the comparison.

All command line settings are specified in an xml config file. You can create an xml config file via one of the following methods:

- Using the data compare command line wizard. You can access the wizard from the start menu, under **Start > All Programs > IDERA > SQL Server comparison tools > Command Lines**.
- Using the data compare interface. Simply compare the databases for which you wish to create a command line config file, set the options and other comparison preferences and click **Create a command line config file** link in the Ribbon, context menu or in the right panel.
- Manually, using any text editor, such as Notepad.

The command line comes with samples for most typical data comparison scenarios. The sample configuration files are located under **\Program Files\Idera\SQL comparison toolset\Command Line Samples\Data Compare**.

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

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

SQL Data Compare Command Line Arguments Overview

Command line switches

The command line supports two categories of options.

- Options that are specified as command line switches. Most of the command line switches have a short and a long name and separate the name from the value by a semicolon.
- Options that are specified via an xml configuration file. We strongly recommend that you set all command line options via the xml configuration file.

When the option can be specified via the command line and xml file, the command line switch takes precedence.

IDERA SQL Data Compare command line supports the following command line switches:

Option	Description
xmlfile	Performs a data comparison using settings in the specified xmlfile .
/datascript:filename	Specifies a file name for the data synchronization script. We recommend you set this option in the xml configuration file.
/datalog:filename	Specifies a file name for the data synchronization log. We recommend you set this option in the xml configuration file.
/en, /encoding: (UTF7 UTF8 UNICODE ASCII)	Indicates the encoding that should be used for output files. ANSI encoding is used if no encoding is specified.
/t, /logtype:(windowslog)	Indicates the type of error log. You can specify windowslog to log the errors in Windows Event log.
/e, /error:errorlog	Specifies the error log. Depending on the /t option, it could indicate a file or a Windows Event log.
/l, /logfile:filename	Specifies a file for the data compare operational log. We recommend you specify the option in the xml configuration file.
/o, /overwrite	Indicates whether the log files should be overwritten. By default command line appends to output logs.
/q, /quiet	Specifies the "quiet" mode of operation, which suppresses all messages from being displayed on the console.

<p>/d, /direction: (leftasright rightasleft)</p>	<p>Indicates whether the command line should make the right database the same as the left database or vice versa. You can specify leftasright or rightasleft. We recommend you set this option in the xml configuration file.</p>
<p>/s, /sync</p>	<p>Indicates whether the command line should synchronize databases. By default data is only compared but the synchronization script is NOT executed. Use this option carefully since it makes permanent changes in the target database.</p>
<p>/mappingtype: (auto custom)</p>	<p>Indicates the mapping of database tables and views. If auto is specified, objects are mapped based on the default mapping: schema.name. If custom is selected, the pairs of objects must be specified manually. The default mapping type is auto. We recommend you set this option in the xml file.</p>
<p>/v, /validate</p>	<p>Validates the xml configuration file.</p>
<p>/? , /h, /help</p>	<p>Displays the command line options.</p>

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

<p>IDERA Website</p>	<p>Products</p>	<p>Purchase</p>	<p>Support</p>	<p>Community</p>	<p>About Us</p>	<p>Resources</p>	<p>Legal</p>
-----------------------------	------------------------	------------------------	-----------------------	-------------------------	------------------------	-------------------------	---------------------

SQL Data Compare XML Configuration File Overview

Xml configuration file allows you to specify all data compare options and settings. Even though some settings can be set via command line switches, we recommend you specify them in the xml file.

With the xml configuration file you can:

- Specify SQL Server databases and login credentials;
- Define the comparison options;
- Customize data compare by excluding objects, defining custom keys and where clauses;
- Specify mapping rules;
- Specify the files for the synchronization script and logs;
- Execute the data synchronization script.

The xml file groups the comparison options into the following sections:

Section	Description
Database <LeftDatabase> <RightDatabase>	Contains the source and target database.
Data Compare Options <DataCompareOptions>	Contains the data compare options.
Table Mapping Rules <TableMappingRules>	Contains the rules that determine how database tables are mapped.
View Mapping Rules <ViewMappingRules>	Contains the rules that determine how database views are mapped.
Table Pairs <TablePairs>	Contains the pairs of tables that should be mapped manually or pairs that should be excluded.
View Pairs <ViewPairs>	Contains the pairs of views that should be mapped manually or pairs that should be excluded.
Command Line Settings <CommandLineOptions>	Contains additional data compare settings.

A typical xml configuration file has the following structure:

```
<?xml version="1.0" encoding="utf-8" ?>
<SqlDataCompareCmdConfig xmlns="http://www.xsqlsoftware.com/schemas/xSQLDataCompareCmdSchema.xsd">
  <LeftDatabase>
    <SqlServer>(local)</SqlServer>
    <DatabaseName>DB_Test1</DatabaseName>
    <TrustedConnection>true</TrustedConnection>
  </LeftDatabase>
```

```

<RightDatabase>
<SqlServer>(local)</SqlServer>
<DatabaseName>DB_Test2</DatabaseName>
<Username>username</Username>
<Password>password</Password>
</RightDatabase>
<CommandLineOptions>
<Synchronize>>true</Synchronize>
<DataScriptFile>script.sql</DataScriptFile>
<DataLogFile>script.log</DataLogFile>
<ComparisonLogFile>log.txt</ComparisonLogFile>
<ErrorLogName>error.log</ErrorLogName>
</CommandLineOptions>
</SqlDataCompareCmdConfig>

```

- <LeftDatabase> and <RightDatabase> elements specify the source and the target database referred to as the left and the right database respectively.
- No explicit data compare options are set. The command line will use the default options.
- <CommandLineSettings> contains the following settings:
 - <Synchronize> true: Indicates that the command line should synchronize the target database by executing the data synchronization script. This setting forces command line to make permanent changes in the target database
 - <DataScriptFile>: Specifies a file name for data synchronization script.
 - <DataLogFile>: Specifies a file name for data synchronization log.
 - <ComparisonLogFile>: Specifies a file name for data compare operational log.
 - <ErrorLogName>: Specifies a file name for logging the errors that might occur during the comparison.

You can run the command line with a xml configuration file as follows:

```
IderaDataCmd <config.xml>
```

You can validate an xml file as follows:

```
IderaDataCmd <config.xml> /v
```

where: <config.xml> is the name of the xml file.

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

IDERA Website	Products	Purchas e	Support	Commun ity	About Us	Resourc es	Legal
--------------------------	-----------------	----------------------	----------------	-----------------------	---------------------	-----------------------	--------------

Customizing SQL Data Compare

There are various comparison options that you can customize via the command line. You can:

- Choose specific data comparison options instead of the default ones.
- Select the mapping type between database objects. Instead of default mapping, you can define a custom mapping and manually specify the pair of tables or views.
- Exclude specific tables, views or columns from the comparison.
- Override default comparison keys with custom keys.
- Specify where clauses for tables or views.
- Define mapping rules for tables or views.

The following sections describe the most common scenarios in details.

- [SQL Data Compare Modifying Comparison Options](#)
- [SQL Data Compare Changing Mapping Type](#)
- [SQL Data Compare Excluding Database Objects](#)
- [SQL Data Compare Excluding Columns](#)
- [SQL Data Compare Defining Custom Keys](#)
- [SQL Data Compare Defining Mapping Rules](#)
- [SQL Data Compare Specifying WHERE Clauses](#)

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

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

SQL Data Compare Modifying Comparison Options

Comparison options can be specified under the xml element <DataCompareOptions>.

The following xml fragment specifies two additional comparison options: GenerateDetailedLog and LogScriptExecution

```
<?xml version="1.0" encoding="utf-8" ?>
<SqlDataCompareCmdConfig
xmlns="http://www.xsqlsoftware.com/schemas/xSQLDataCompareCmdSchema.xsd"
">
  <LeftDatabase>
    <SqlServer>(local)</SqlServer>
    <DatabaseName>DB_Test1</DatabaseName>
    <TrustedConnection>>true</TrustedConnection>
  </LeftDatabase>
  <RightDatabase>
    <SqlServer>(local)</SqlServer>
    <DatabaseName>DB_Test2</DatabaseName>
    <Username>username</Username>
    <Password>password</Password>
  </RightDatabase>
  <DataCompareOptions>
    <GenerateDetailedLog>>true</GenerateDetailedLog>
    <LogScriptExecution>>true</LogScriptExecution>
  </DataCompareOptions>
  <CommandLineOptions>
    <Synchronize>>true</Synchronize>
    <DataScriptFile>DataScript.sql</DataScriptFile>
    <Direction>RightAsLeft</Direction>
    <ComparisonLogFile>log.txt</ComparisonLogFile>
  </CommandLineOptions>
</SqlDataCompareCmdConfig>
```

For the full list of supported options, check [Xml Reference](#).

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

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

SQL Data Compare Changing Mapping Type

Mapping type determines how database tables and views are paired together. Default mapping is based on the full name of the object schema.name. A table/view on the "left" database can only be mapped to a table/view with the same name on the "right" database. You can change the mapping type by choosing a "Custom" mapping and specify the objects that should be paired together that might not necessarily have the same name.

To define custom mappings, you need to:

- Set the `<TableMappingSchema>` under `<CommandLineOptions>` element to `Custom`.
- Specify the custom pairs of Tables/Views under the element `<TablePairs>` or `<ViewPairs>`

The following xml fragment defines custom mappings:

```
<CommandLineOptions>
  <TableMappingSchema>Custom</TableMappingSchema>
</CommandLineOptions>
<TablePairs>
  <Pair>
    <LeftObject Name="dbo.Table_A"></LeftObject>
    <RightObject Name="dbo.Table_AA"></RightObject>
  </Pair>
  <Pair>
    <LeftObject Name="dbo.Table_B"></LeftObject>
    <RightObject Name="dbo.Table_BB"></RightObject>
  </Pair>
</TablePairs>
```

 Specify the full name of the objects in the custom mappings set.

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

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

SQL Data Compare Excluding Database Objects

You can exclude specific tables or views from data compare by listing them under the xml element `<TablePairs>` or `<ViewPairs>`.

Rules for excluding database objects are as follows:

- The attribute Action of `<Pair>` element should be set to Exclude
- The left and right object must be both specified for each excluded pair

The following xml fragment excludes the tables `dbo.Employees` and `dbo.Customers` from the comparison:

```
<TablePairs>
  <Pair Action="Exclude">
    <LeftObject Name="dbo.Employees"></LeftObject>
    <RightObject Name="dbo.Employees"></RightObject>
  </Pair>
  <Pair Action="Exclude">
    <LeftObject Name="dbo.Customers"></LeftObject>
    <RightObject Name="dbo.Customers"></RightObject>
  </Pair>
</TablePairs>
```

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

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

SQL Data Compare Excluding Columns

Command line allows you to exclude specific columns from data compare. In order to exclude columns, you must identify the pair of entities that columns belong to.

Rules for excluding columns are as follows:

- Excluded columns should be listed under the xml element `<ColumnPair>` of the `<Columns>` element
- The attribute `Action` of the `<Columns>` element should be set to `Exclude`

The following xml fragment excludes the columns `DateCreated`, `CreatedBy` from the table `dbo.Employees` and the column `Address` from the table `dbo.Customers`:

```
<TablePairs>
  <Pair>
    <LeftObject Name="dbo.Employees"></LeftObject>
    <RightObject Name="dbo.Employees"></RightObject>
    <Columns Action="Exclude">
      <ColumnPair>
        <LeftColumn>DateCreated</LeftColumn>
        <RightColumn>DateCreated</RightColumn>
      </ColumnPair>
      <ColumnPair>
        <LeftColumn>CreatedBy</LeftColumn>
        <RightColumn>CreatedBy</RightColumn>
      </ColumnPair>
    </Columns>
  </Pair>
  <Pair>
    <LeftObject Name="dbo.Customers"></LeftObject>
    <RightObject Name="dbo.Customers"></RightObject>
    <Columns Action="Exclude">
      <ColumnPair>
        <LeftColumn>Address</LeftColumn>
        <RightColumn>Address</RightColumn>
      </ColumnPair>
    </Columns>
  </Pair>
</TablePairs>
```

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

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

SQL Data Compare Defining Custom Keys

You can specify a custom key when the table or view has no built-in key or when the existing key is not suitable for data compare.

The following rules apply to custom keys:

- A custom key can be defined under the xml element <LeftObject> or <RightObject> of the <Pair> element
- A custom key must have a name and one or more columns
- The columns that participate in the custom key should exist in the parent table or view
- The key must be defined for each object in the pair of entities.
- The columns that participate in the custom key must be identical for both objects in the specified pair.

The following xml fragment defines a custom key for the pair of table dbo.Employees:

```
<TablePairs>
  <Pair>
    <LeftObject Name="dbo.Employees">
      <Key Name="Custom_Key">
        <Column>EmployeeID</Column>
      </Key>
    </LeftObject>
    <RightObject Name="dbo.Employees">
      <Key Name="Custom_Key">
        <Column>EmployeeID</Column>
      </Key>
    </RightObject>
  </Pair>
</TablePairs>
```

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

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

SQL Data Compare Defining Mapping Rules

Mapping rules are used to override the default data compare mapping that is based on schema.name. Mapping rules are usually created to reflect the characteristics of a particular environment. If you have, for example, two databases that have the same structure but different schema, you can create a mapping rule that ignores the schema.

The command line supports three types of mapping rules:

- Mapping rules that apply to object name
- Mapping rules that apply to object scheme
- Mapping rules that apply to data types.

Mapping rules that apply to the name of objects should be specified for each database that participates in the comparison.

The following xml fragment defines two mapping rules: A name rule that ignores the name postfix "_temp" on the left database and a schema rule that ignores the schema on both databases:

```
<TablesMappingRules>
  <NameMapping MappingRule="IgnorePostfix" IgnoreLeftName="_temp" />
  <SchemaMapping MappingRule="IgnoreSchema"></SchemaMapping>
</TablesMappingRules>
```

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

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

SQL Data Compare Specifying WHERE Clauses

A where clause is an SQL expression that is used as a data filter. Only the records that meet the where clause criteria are compared and synchronized. You can specify any valid SQL expression as the where clause. The "where" keyword should not be included in the expression.

The command line supports where clauses via three elements under the <Pair> element:

- WhereClause: sets a where clause for both members of a pair
- LeftWhereClause: sets a where clause for the left member of the pair
- RightWhereClause: sets a where clause for the right member of the pair

The following xml fragment defines a where clauses on the employees table that excludes records with emp_id = 3. The where clause is set for both the left and the right database:

```
<TablePairs>
  <Pair>
    <LeftObject Name="dbo.employees"/>
    <RightObject Name="dbo.employees"/>
    <WhereClause>emp_id != 3</WhereClause>
  </Pair>
</TablePairs>
```

The following xml fragment defines separate where clauses on the employees table for the left and the right database. On the left database, it excludes records with emp_id = 3; on the right database, it excludes records with emp_id = 4.

```
<TablePairs>
  <Pair>
    <LeftObject Name="dbo.employees"/>
    <RightObject Name="dbo.employees"/>
    <LeftWhereClause>emp_id != 3</LeftWhereClause>
    <RightWhereClause>emp_id != 4</RightWhereClause>
  </Pair>
</TablePairs>
```

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

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

SQL Data Compare XML Configuration File Reference

```

<SqlDataCompareCmdConfig>
  <LeftDatabase>
    <SqlServer>string</SqlServer>
    <DatabaseName>string</DatabaseName>
    <Username>string</Username>
    <Password>string</Password>
    <TrustedConnection>boolean</TrustedConnection>
    <ConnectionTimeout>int</ConnectionTimeout>
    <CommandTimeout>int</CommandTimeout>
  </LeftDatabase>
  <RightDatabase>
    <SqlServer>string</SqlServer>
    <DatabaseName>string</DatabaseName>
    <Username>string</Username>
    <Password>string</Password>
    <TrustedConnection>boolean</TrustedConnection>
    <ConnectionTimeout>int</ConnectionTimeout>
    <CommandTimeout>int</CommandTimeout>
  </RightDatabase>
  <DataCompareOptions>
    <CompareTables>boolean</CompareTables>
    <CompareMemoryTables>boolean</CompareMemoryTables>
    <CompareColumnStoreTables>boolean</CompareColumnStoreTables>
    <CompareViews>boolean</CompareViews>
    <DisableTriggers>boolean</DisableTriggers>
    <ForeignKeyAction>ForeignKeyActions</ForeignKeyAction>
    <GenerateDetailedLog>boolean</GenerateDetailedLog>
    <LogScriptExecution>boolean</LogScriptExecution>
    <ScriptExecutionLogFile>string</ScriptExecutionLogFile>
  </DataCompareOptions>
  <GenerateTransactionalScript>boolean</GenerateTransactionalScript>
  <IgnoreCharCase>boolean</IgnoreCharCase>
  <TrimSpaces>boolean</TrimSpaces>
  <PerformDependencyAnalysis>boolean</PerformDependencyAnalysis>
  <ReportEqualRows>boolean</ReportEqualRows>
  <SynchronizeDifferentRows>boolean</SynchronizeDifferentRows>
  <SynchronizeLeftRows>boolean</SynchronizeLeftRows>
  <SynchronizeRightRows>boolean</SynchronizeRightRows>
  <TransactionSize>int</TransactionSize>
  <BatchSize>int</BatchSize>
  <IgnoreDatabaseCollation>boolean</IgnoreDatabaseCollation>
  <RoundApproximateNumber>boolean</RoundApproximateNumber>
  <RoundPrecision>int</RoundPrecision>
</DataCompareOptions>
<CommandLineOptions>

```

```

<ComparisonLogFile>string</ComparisonLogFile>
<OverwriteComparisonLog>boolean</OverwriteComparisonLog>
<ErrorLogName>string</ErrorLogName>
<ErrorLogType>ErrorLogTypes</ErrorLogType>
<QuietMode>boolean</QuietMode>
<DataScriptFile>string</DataScriptFile>
<DataLogFile>string</DataLogFile>
<FileEncoding>Encodings</FileEncoding>
<Direction>CompareDirections</Direction>
<Synchronize>boolean</Synchronize>

```

```

<TableMappingSchema>TableMappingSchemaTypes</TableMappingSchema>
  </CommandLineOptions>
  <TablesMappingRules>
    <NameMapping>NameMappingType</NameMapping>
    <SchemaMapping>SchemaMappingType</SchemaMapping>
    <TypeMapping>TypeMappingType</TypeMapping>
  </TablesMappingRules>
  <ViewsMappingRules>
    <NameMapping>NameMappingType</NameMapping>
    <SchemaMapping>SchemaMappingType</SchemaMapping>
    <TypeMapping>TypeMappingType</TypeMapping>
  </ViewsMappingRules>
  <TablePairs>Pairs</TablePairs>
  <ViewPairs>Pairs</ViewPairs>
  <NameMappingType>
    <MappingRule>NameMappingRules</MappingRule>
    <IgnoreLeftName>string</IgnoreLeftName>
    <IgnoreRightName>string</IgnoreRightName>
  </NameMappingType>
  <SchemaMappingType>
    <MappingRule>SchemaMappingRules</MappingRule>
    <SchemaPairs>SchemaPairs</SchemaPairs>
  </SchemaMappingType>
  <TypeMappingType>
    <MappingRule>TypeMappingRules</MappingRule>
  </TypeMappingType>

```

```

<NameMappingRules>ExactMatch | IgnorePrefix | IgnorePostfix</NameMappingRules>

```

```

<SchemaMappingRules>ExactMatch | IgnoreSchema | Custom</SchemaMappingRules>
  <SchemaPairs>[1..*]SchemaPairType</SchemaPairs>
  <SchemaPairType>
    <LeftSchema>string</LeftSchema>
    <RightSchema>string</RightSchema>
  </SchemaPairType>

```

```

<TypeMappingRules>ByDefinition|ByCompatibilityMatrix</TypeMappingRules>
  <Pairs>[0..*]Pair</Pairs>
  <Pair>
    <LeftObject>TableOrView</LeftObject>
    <RightObject>TableOrView</RightObject>
    <Columns>Columns</Columns>
    <WhereClause>string</WhereClause>
    <LeftWhereClause>string</LeftWhereClause>
    <RightWhereClause>string</RightWhereClause>
    <Action>Actions</Action>
  </Pair>
  <TableOrView>
    <Key>DataKey</Key>
    <Name>string</Name>
  </TableOrView>
  <DataKey>
    <Name>string</Name>
    <Column>[1..*]string</Column>
  </DataKey>
  <Columns>
    <Action>Actions</Action>
    <ColumnPair>[1..*]ColumnPair</ColumnPair>
  </Columns>
  <Actions>Include|Exclude</Actions>
  <ColumnPair>
    <LeftColumn>string</LeftColumn>
    <RightColumn>string</RightColumn>
  </ColumnPair>

<ForeignKeyActions>DisableAndEnable|DropAndCreate|None</ForeignKeyActions>
  <ErrorLogTypes>File|WindowsLog</ErrorLogTypes>
  <Encodings>Unicode|UTF7|UTF8|ASCII|ANSI</Encodings>
  <CompareDirections>LeftAsRight|RightAsLeft</CompareDirections>
  <TableMappingSchemaTypes>Auto|Custom</TableMappingSchemaTypes>
</SqlDataCompareCmdConfig>

```

The main elements of xml schema are described in the following section:

Element	Description
SqlDataCompareCmdConfig	Root element.
LeftDatabase, RightDatabase	The elements that specify the left and the right database that participate in the data comparison.
SqlServer	Defines the name of the SQL Server instance.
DatabaseName	Defines the database name.

Username, Password	Define database credentials.
TrustedConnection	Defines the connection type.
ConnectionTimeout	Defines the connection timeout in seconds, which the amount of time that command line waits before aborting a connection.
CommandTimeout	Defines the execution timeout of SQL queries in seconds, which the amount of time that command line waits for a query to complete before aborting it.
CompareTables	Indicates whether database tables should be compared and synchronized.
CompareMemoryTables	Indicates whether the memory tables supported by SQL 2014 should be compared and synchronized.
CompareColumnStoreTables	Indicates whether the columnstore tables supported by SQL 2014 should be compared and synchronized.
CompareViews	Indicates whether database views should be compared and synchronized.
DisableTriggers	Specifies the type of action on DML triggers that should be performed by synchronization script.
ForeignKeyAction	Specifies the type of action on foreign keys that should be performed by synchronization script.
GenerateDetailedLog	Indicates whether the command line should generate a detailed synchronization log.
LogScriptExecution	Specifies whether the command line should save to a log file SQL statements executed during data synchronization.
ScriptExecutionLogFile	Defines the file name of the execution log.
GenerateTransactionalScript	Indicates whether the generated synchronization script should be transactional.
IgnoreCharCase	Indicates whether the character chase should be ignored when comparing string columns.
TrimSpaces	Indicates whether the leading and trailing spaces should be ignored when comparing string columns.

PerformDependencyAnalysis	Specifies whether command line should analyze object dependencies (based on foreign key relationship between them) and order the synchronization script accordingly.
ReportEqualRows	Indicates whether the command line should keep the equal rows in the internal row collections. By default equal rows are discarded.
SynchronizeDifferentRows	Indicates whether the rows that have been found different should be synchronized.
SynchronizeLeftRows	Indicates whether the rows found on the left database should be synchronized.
SynchronizeRightRows	Indicates whether the rows found on the right database should be synchronized.
TransactionSize	Defines the max size of SQL Server transaction that should be used when executing synchronization script.
BatchSize	Defines the max batch size that should be used when executing synchronization script.
IgnoreDatabaseCollation	Indicates whether collation should be ignored when comparing databases.
RoundApproximateNumber	Indicates whether approximate SQL Server types such as float and real should be rounding to a specified precision and then compared.
RoundPrecision	Defines the rounding precision for approximate data types such as real and float. Rounding effects only the comparison of approximate types, not the scripting of these types.
ComparisonLogFile	Defines a file name for data compare operational messages.
OverwriteComparisonLog	Indicates whether the command line should overwrite all log files. By default command line appends to log file.
ErrorLogName	Defines a file name for errors that might occur during data compare.
ErrorLogType	Indicates whether errors should be logged to file or Windows event log.

QuietMode	Indicates the quiet mode which suppresses messages from being displayed on the console.
DataScriptFile	Defines a file name for the data synchronization script.
DataLogFile	Defines a file name for the data synchronization log.
FileEncoding	Defines the encoding of all output file.
Direction	Indicates the "direction" of synchronization whether the left database is synchronized with the right database or vice versa.
Synchronize	Indicates whether the command line should synchronize databases. By default databases are only compared.
TableMappingSchema	Defines the mapping type of database objects whether is Auto or Custom.
TablesMappingRules	Defines the mapping rules for tables.
ViewsMappingRules	Defines the mapping rules for views.
NameMapping	Defines the mapping rule that applies to objects name.
SchemaMapping	Defines the mapping rule that applies to objects schema (owner).
TypeMapping	Defines the mapping rule that applies to data types.
TablePairs	Defines that pair of tables that should be mapped manually. Manual pairs require the option TableMappingSchema to be set to Custom.
ViewPairs	Defines that pair of views that should be mapped manually. Manual pairs require the option TableMappingSchema to be set to Custom.

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

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

SQL Data Compare Error Codes

Data compare command line returns the following error codes:

Error Code	Description
0	No error
1	Invalid command line switches
2	Invalid xml file
4	Invalid output files
8	Invalid or missing options
16	Generic error
32	License missing or expired

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

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