

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 columns 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 constraints 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 an 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

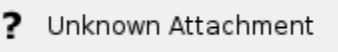
Below is a summary of the new options supported by IDERA SQL Schema Compare 7.0:

- **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.

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

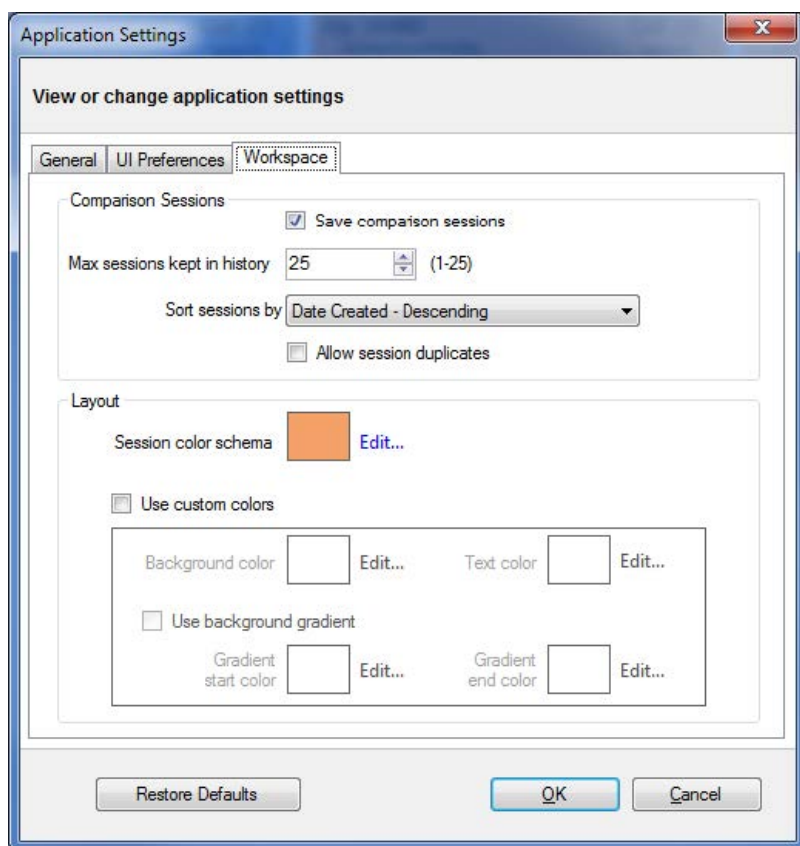
IDERA SQL Comparison Toolset 5.0 offers full support for Microsoft Azure SQL Database v11 and the preview of v12. Comparing cloud-based or on-premise databases uses the same process within IDERA SQL Comparison Toolset. The toolset handles the platform differences automatically and produces any warnings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

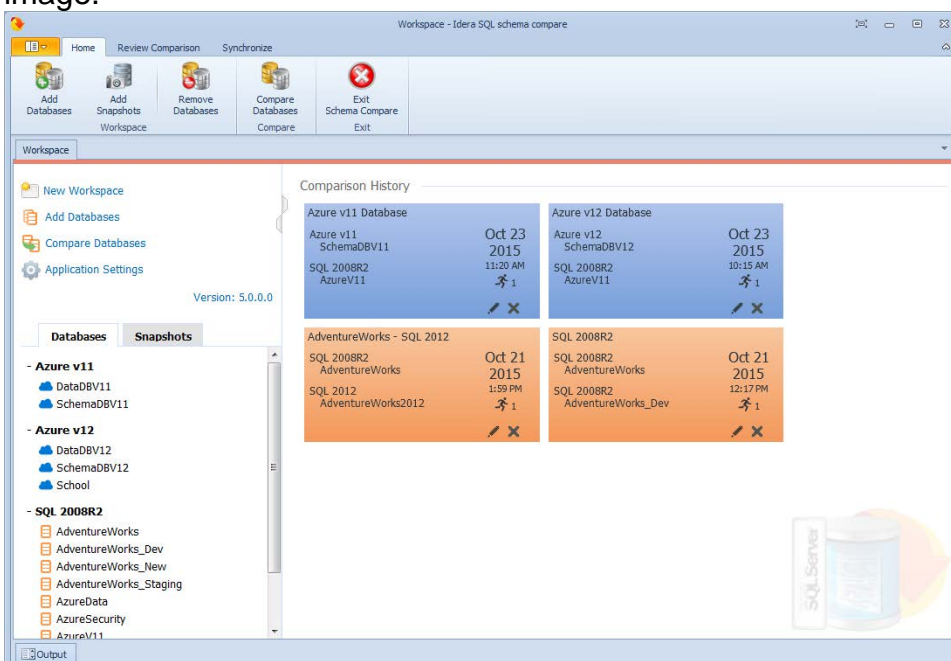
IDERA SQL Comparison Toolset now allows you to have multiple sessions for the same SQL database. You can control this feature on the Workspace tab of the Application Settings window, as show in the following image.



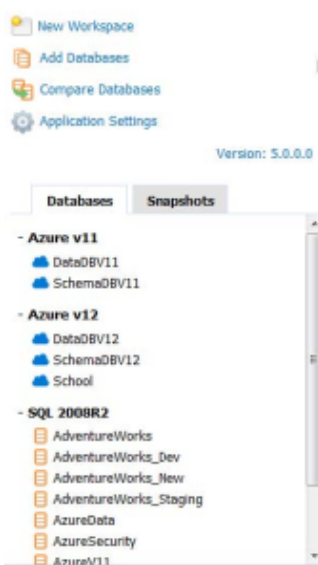
SQL Schema Compare Workspace tab user interface improvement

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.



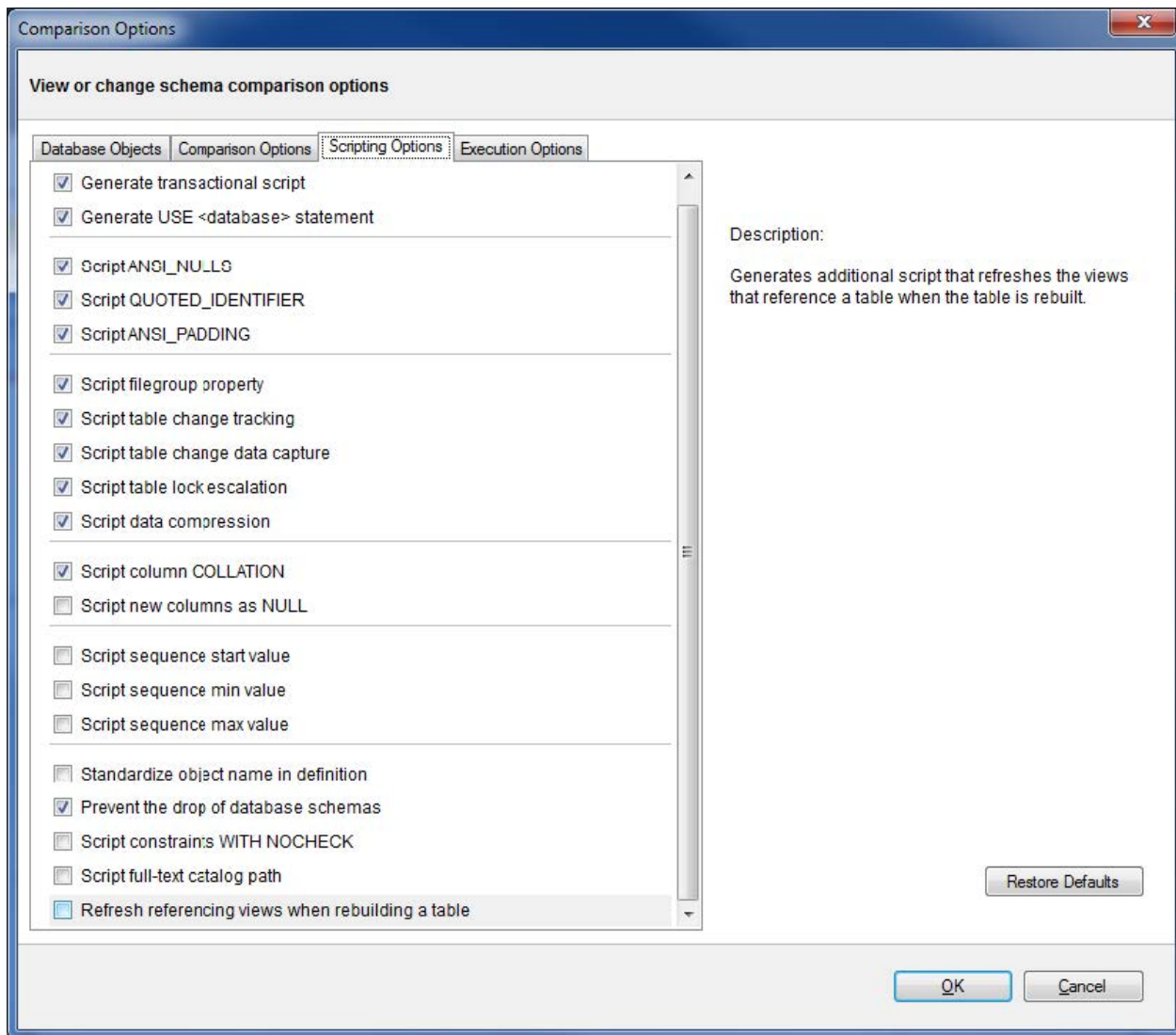
- The Databases tab of the Workspace now includes a cloud icon next to Microsoft Azure SQL Databases for you to distinguish between cloud and on-premise databases, as shown in the following image.



- 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.

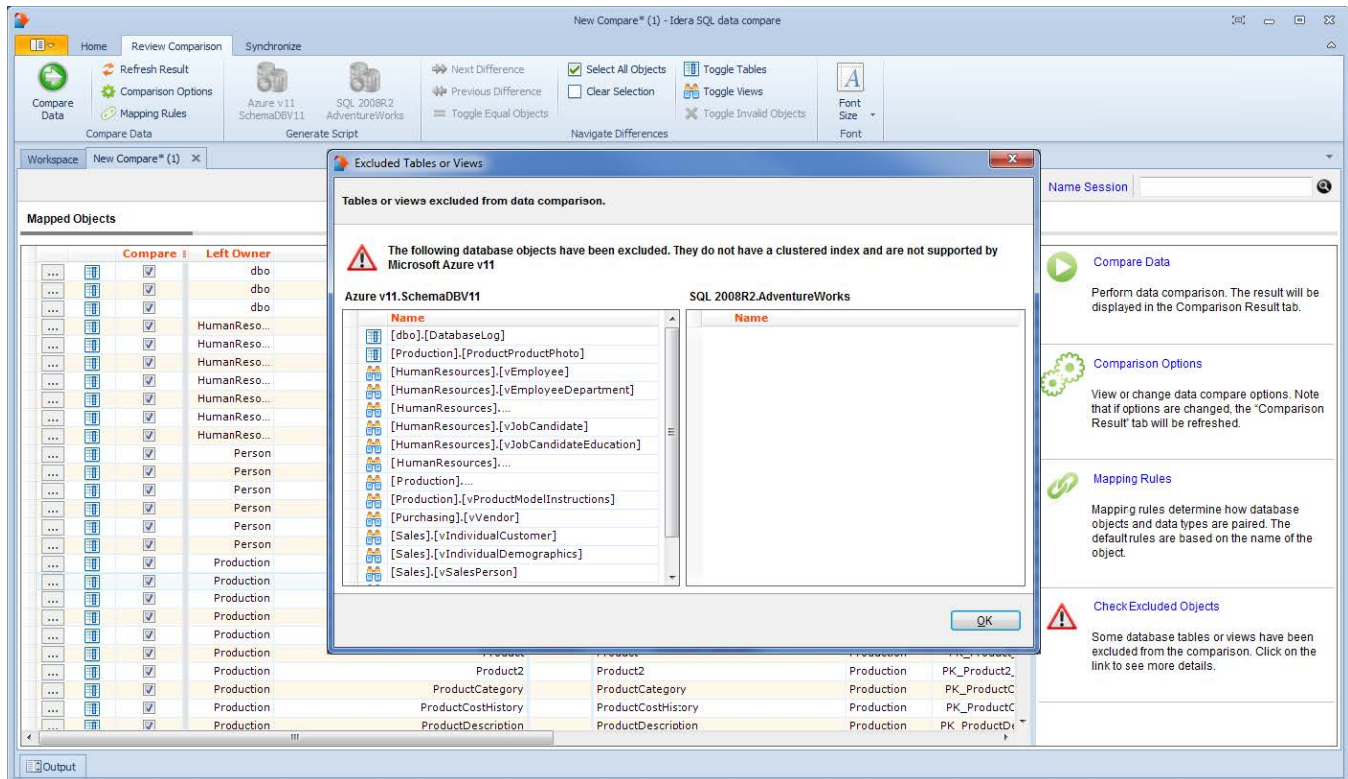
SQL Schema Compare refresh option

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.



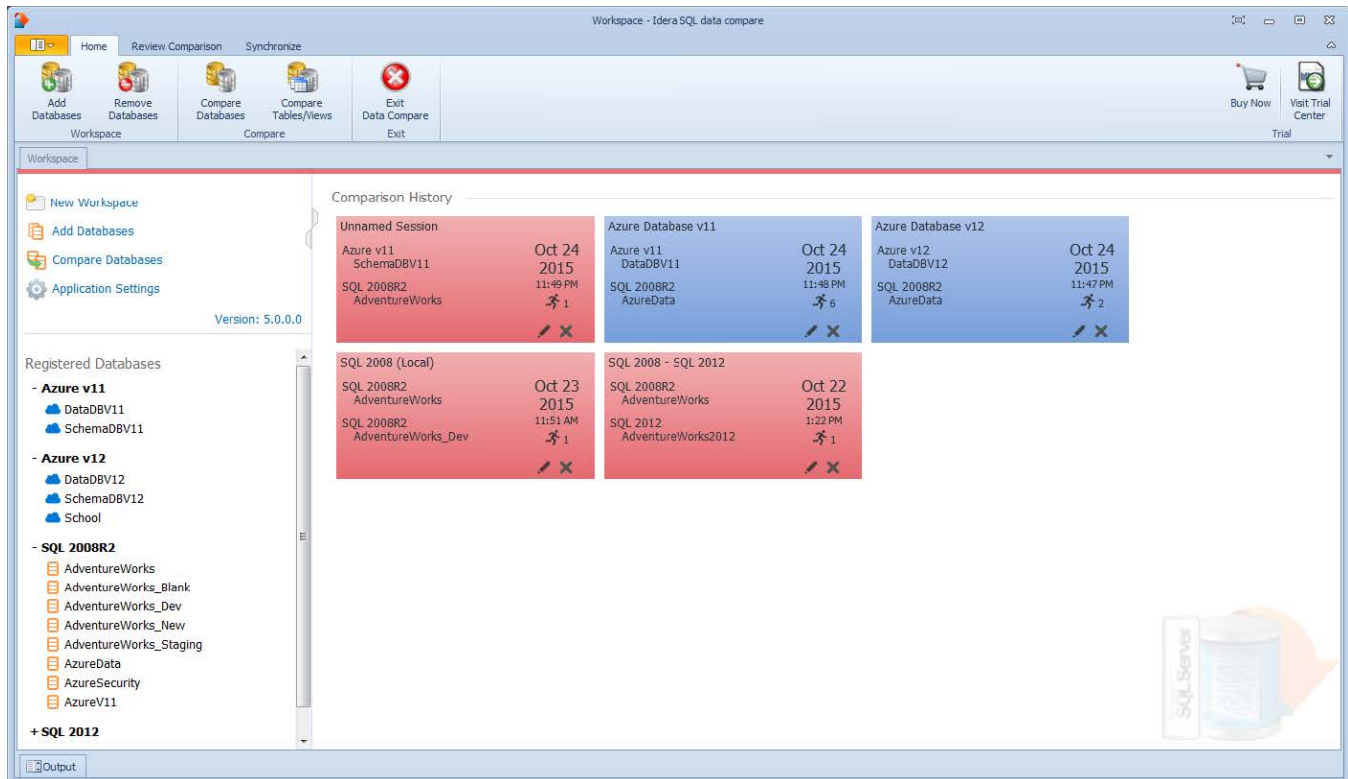
SQL Data Compare user interface improvements for Microsoft Azure SQL Database v11

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
-------------------------------	--------------------------	--------------------------	-------------------------	---------------------------	--------------------------	---------------------------	-----------------------