

New features and fixed issues

Idera Comparison Toolset provides the following new features and fixed issues.

10.0 New features

Supports Azure Active Directory Authentication for SQL Server

SQL Comparison Toolset includes a new system check for the **Microsoft Active Directory Authentication Library for SQL Server** component, which supports the new Azure Active Directory Authentication feature. If the library is not found, a warning appears during the setup process. You still can continue installation without this component, but note that the Azure AD Authentication is unavailable without the library. SQL Comparison Toolset does not install this library automatically due to the licensing restrictions imposed by Microsoft.

Azure Active Directory Authentication

In addition to traditional Windows Authentication and SQL Server Authentication, SQL Comparison Toolset now supports:

- Azure Active Directory Integrated Authentication
- Azure Active Directory Password Authentication

The new authentication types require the **Microsoft Active Directory Authentication Library for SQL Server** library. When Azure Authentication is selected for a SQL Server, the schema and data interface checks for this component, and then displays a warning popup if the library is missing. The window displays the download URL. Other SQL Comparison Toolset functionalities are not impacted by the absence of this component.

Supports SQL Server graph structure

SQL Comparison Toolset now supports the new SQL Server graph structure. This includes the node and the edge tables, introduced on SQL Server 2017, and the edge constraints, added by SQL Server 2019. New comparison options allow users to exclude these objects if necessary.

Careful consideration and extensive testing went into ensuring the proper synchronization of graph data in particular. Unlike traditional tables, which have only user-columns, graph tables contain system generated columns, such as \$node_id, \$edge_id, \$from_id and \$to_id. SQL Server places restrictions on these columns. It allows, for example, inserts and deletes, but it doesn't permit updates. The toolset takes great care to properly update the graph data, so that the graph structure is maintained and remains consistent between the databases that are being synchronized.

Includes new assembly scripting options

Starting with SQL Server 2017, the code access security in .NET framework is no longer used as a security boundary for CLR assemblies. A new configuration option, named **clr strict security**, further enhances the security of CLR assemblies. When this option is ON (default value), SQL Server treats all assemblies, even those marked with the SAFE or the EXTERNAL_ACCESS attribute, as if they were UNSAFE.

With this new security policy in place, creating an assembly in the same way as it was done prior to SQL Server 2017, may not be sufficient. For this reason, Schema Compare includes four new options for the CLR assemblies. Each of the following options generates additional statements, so that the assembly is not rejected by SQL Server:

1. **Add assembly to the trusted assembly list.** White-list the assembly by adding it to the trusted assembly list. This is done by calling `sp_add_trusted_assembly` with the correct hash value generated from the assembly binaries. The user must be a sysadmin or have CONTROL SERVER permission to add assemblies to the trusted list.
2. **Set the database TRUSTWORTHY ON.** Generates a statement that makes the database TRUSTWORTHY. The user must be a sysadmin to use this option.
3. **Disable "CLR strict security" option.** Generates a statement that disables the option **CLR strict security**. The user must have ALTER SETTINGS server-level permission to change this option.
(ALTER SETTINGS permission is implicitly held by the sysadmin and serveradmin fixed server roles).
4. **None of the above.** This option should be used if the user has taken the necessary steps to allow CLR assemblies in the database. Microsoft provides a guideline to accomplish this task.

Users can also remove an assembly from the trusted list, when the assembly is dropped.

Includes new reports

Interactive Schema Differences report

SQL Comparison Toolset contains a new interactive Schema Differences report. The new report is generated as an html file, with embedded JavaScript for navigation and filtering, useful for large databases with many objects.

The UI provides the following options that allow users to customize the report:

- Objects can be excluded from the report by status, such objects that are equal, different or missing.
- Additional properties can be included for columns, such as the data type, nullability, identity settings and other properties.
- Columns can be included with indexes and constraints (primary keys, foreign keys, unique and edge constraints).
- The SQL expressions can be scripted for check constraints and default constraints.
- The SQL definition can be scripted for views, procedures, functions and triggers.

Interactive Data Differences report

SQL Comparison Toolset also contains a new Data Differences report. The report is generated and saved as an html file, with embedded JavaScript for filtering and navigation. It provides a table summary, which could include views depending on the comparison options, with their data differences, such as the number of rows that are equal, different or missing on one databases. The user can filter out tables that are equal and leave only those that are different or vice versa.

Supports external users and groups for Azure databases

SQL Schema Compare supports the new external users and external groups for Azure databases.

Improved other areas

- **Check constraint mappings.** Previous versions of the toolset mapped the check constraints by name or by the column on which the constraint is defined. This feature however was supported for check constraints on a single column. Version 10 removes this limitation and supports the column mapping of the check constraints on multiple columns as well.
- **Unique constraint mappings.** Schema compare now supports column-mapping of the unique constraints, in addition to mapping by name. Unique constraints mapped by columns do not trigger a schema difference if their names are different.
- **Synchronizing data changes that do not allow updates.** Data compare can synchronize changes between tables, which, for various reasons, do not support updates. An identity column, that is not part of the data key, is one example. The toolset scans the database tables for these cases and adjusts the script automatically to accommodate them.
- **Using delete/insert instead of updates.** Data compare adds a new option that allows users to synchronize data changes using inserts/deletes instead of updates. This could prove useful for databases that do not permit or "prefer" updates, such as databases containing financial data.
- **Improved mapping.** Data compare contains an improved mapping functionality, which handles duplicated objects better than the previous versions. Duplicates are logged and reported, instead of aborting the entire operation.
- **Unique constraint on user-defined table types.** Schema compare fixes an issue with the unique constraints on user-defined table types. The table type is no longer flagged as different when the constraints on the source and target database have a different name, since the name of these constraints can't be scripted in this case.
- **IGNORE_DUP_KEY index option on in-memory table types.** Schema compare fixes an issue with the IGNORE_DUP_KEY option for indexes created on in-memory table types. This option is omitted since it is not supported by SQL Server.