



SSIS for Azure SQL Data Warehouse

Transform your data integration with SSIS for Azure SQL Data Warehouse. Seamlessly orchestrate and manage your data workflows in Azure SQL DW, unlocking unparalleled efficiency and performance. Our solution empowers you to design, deploy, and automate data integration processes with ease.

Configuring Connections

Devart SSIS Data Flow Components provide a number of connection manager components for SSIS data flows. To add a Devart connection manager to an Integration Services project, perform the following steps:

1. Right-click anywhere in the Connection Managers area, and then click **New Connection**.
2. Select the Devart connection manager for the necessary data source in the Add SSIS Connection Manager dialog box.
3. After this, in the opened connection manager editor dialog box, specify the necessary connection settings on the **General** tab.
4. Optionally switch to other tabs and set advanced parameters.
5. Click **OK** to save the connection settings.

You can the information about the connection parameters in the corresponding topics of the Data Source Connection Settings section.

Devart Source Components

Devart Source components are data flow components that make data from the corresponding data sources available to other data flow components. Each Devart Source component provides a convenient editor dialog box, which allows you to easily define which data to load.

Note that Devart Source components support only the corresponding Devart connection managers.

How to use Devart Source Components

To add a Devart Source component to a Data Flow task, perform the following steps:

1. Drag the corresponding Devart Source component from the SSIS Toolbox window to the Data Flow diagram.
2. Double-click the added component to open its editor.
3. In the **<Data Source> Connection** list (where **<Data Source>** is the corresponding data source) select the corresponding Devart Connection Manager or create a new one.
4. In the **Query** box enter an SQL query that selects the required data. You can also drag the corresponding table from the **<Data Source> Objects** tree to the **Query** box.
5. Click **OK**.

After this you may add other Data Flow components and connect Devart Source data flow path to them.

Devart Source Components

Devart Source Component Editor

This dialog box has the following components:

- The **<Data Source> Connection** list allows selecting a connection manager for the Source component.

Batch Size - Available for Devart Salesforce Source. Specifies the number of records to retrieve from Salesforce as one batch.

Include Deleted Records - Available for Devart Salesforce Source. Specifies whether to include records with IsDeleted = true to the query results.

- The **<Data Source> Objects** (for cloud sources) or Tables (for databases) area allows you to view the existing tables, expand their nodes to view columns, and drag them from this area to the Query area.

Show all schemas - Available for databases. Enables displaying all the objects of the selected connection; otherwise, only the objects of the default schema for the connection are displayed.

Refresh Objects - Updates the list of data source tables.

- The **Variables** area allows you to view the system and user variables grouped in the corresponding nodes, and if necessary, drag them to the Query area.
- The **Query** area is intended for creating a SQL query to be executed against the data source. You can drag the tables and variables to this area, or type the query yourself.

Preview Data - Opens the Preview Data window that displays the first records of the query execution result. By default, it displays the first 100 records. If necessary, you may change the **Number of Rows** to display in this window.

- The **Properties** area displays properties of the currently selected table, column or variable.

Devart Source Components

SQL Support

Devart Source components provide advanced SQL support for cloud sources even when a source does not support SQL natively. Note that you should use SQLite compliant SQL, not the Transact-SQL. Simple SELECT statements are translated into cloud API calls and executed against the cloud source directly. For more complex statements, all the data from cloud objects, mentioned in the query, is queried to the local cache, and the query is executed locally, against the cache.

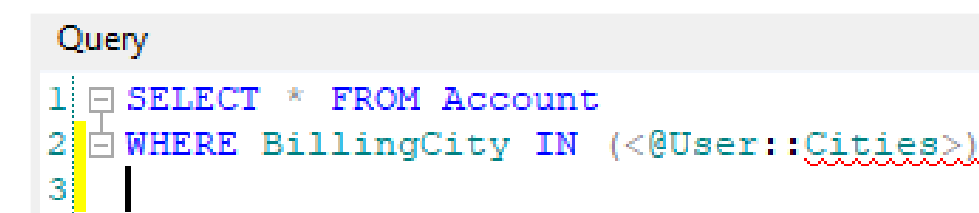
You can disable local SQL execution by setting the Local SQL Engine parameter in the advanced parameters of your cloud connection manager to False. In this case more complex statements that cannot be converted to API calls will simply fail.

Using Variables

Variables can be added to the query text in two ways. By default, they are added quoted with single quotation marks - as SQL string literals.



In this way, the variable value is also enclosed in single quotation marks and any single quotation marks in it are escaped, i. e. the variable value is substituted as an SQL string. However, if you need, you may even pass a set or even an arbitrary SQL fragment as a variable value. For this, just remove these single quotation marks:



The query editor highlights such a SQL statement as invalid, but this SQL can be used if it is valid after substituting a variable value. You can use the following variable value: *cities* = 'Berlin', 'Hamburg', 'Hannover', and as the result, the following SQL will be run:

```
SELECT * FROM Account
WHERE BillingCity IN ('Berlin', 'Hamburg', 'Hannover')
```

Devart Destination Components

Devart Destination components are designed for loading data into the corresponding data sources.

Note that Devart Destination components support only the corresponding Devart connection managers.

How to use Devart Destination Components

To add a Devart Destination component to a Data Flow task, perform the following steps:

1. Drag the corresponding Devart Destination component from the SSIS Toolbox window to the Data Flow diagram.
2. Connect the necessary data flow path to it.
3. Double-click the added component to open its editor.
4. On the **Connection Managers** tab select the corresponding Devart Connection Manager or create a new one.
5. Click the **Component Properties** tab.
6. In the **<Data Source> Object** (for cloud sources) or **<Data Source> Table** (for databases) property type or select the target object or table (where **<Data Source>** is the corresponding data source).
7. In the **Action** property select the required DML operation to apply to the data source. Supported DML operations are described below.
8. Optionally configure other custom properties of the component. Custom properties are source-specific. They are described in the Custom Properties Reference section. For example,
9. Click the **Column Mappings** tab.
10. Map the input columns to the corresponding destination columns. The columns with the same names in input and destination are mapped automatically.
11. Click **OK**.

Devart Destination Components

Supported DML Operations

- Devart Destination components support all the three basic DML operations: insert, update, and delete. Destination components for some sources support additional operations.
- **Insert** - This is the most common operation. See a tutorial on using it.
- **Update** - Devart Destination component in the Update mode enables updating data in a data source. See a tutorial on using it. Peculiarity of this action consists in the fact, that performing it requires using primary key values of the target table (the Id value for cloud sources). Thus, when performing an update, you must know and specify the primary key values of the target table to update a record.
- **Delete** - Devart Destination component in the Delete mode enables deleting data from a data source. As well as for the update operation, you must know and specify the primary key values of the target table to delete a record.
- **Upsert** - This operation is supported for Salesforce. When executing an upsert, records are inserted if they do not exist in the Salesforce object; in case records do exist, then data update is performed. This action is based on the External ID field. In order to use the Upsert option, the External ID field must be added to the Salesforce object, and the local storage must contain the corresponding unique column (usually this is a primary key in the database table). When configuring the component, you must map this unique column to the External ID field in the target Salesforce object.
- **BulkInsert** - This operation is supported for relational databases except DB2 and cloud data warehouses. It uses database-specific optimization techniques to speed up data loading. You can see the description of these techniques below.
- **HardDelete** - This operation is supported for Salesforce. It completely deletes data rows without the possibility to restore them. In Salesforce, the Delete operation actually keeps the records and just sets their IsDeleted property to True, the HardDelete operation actually deletes the data.

Devart Destination Components

Performing Update and Delete Operations

Performing an update or delete operation requires mapping for the target table primary key columns. A destination component determines a record to update or delete by their ID/primary key values. Thus, the updated table must have a primary key. For the UPDATE operation you also need to provide values for the columns you want to change the value of, and you can additionally map any target object columns. For the delete operation, it is enough to map only the target primary key.

When ID/Primary Key values are known and present in the source, you can simply map them to the corresponding target columns. However, in many cases there are no target primary key values in a source. In this case you may retrieve the target primary key values by some other identifying columns using Devart Lookup Components. You can see an example of update operation with using lookup in our Devart Lookup Tutorial - Inserting and Updating Contact Data.

Source-specific Optimization Techniques

Devart Destination components for relational database can use certain database-specific techniques to speed up data loading:

- Devart Oracle Destination uses direct path load interface to speed up data loading in the OCI connection mode. In the Direct connection mode it uses Array Binding - a feature that allows executing several INSERT SQL statements with the different parameters at once. Oracle server is accessed only once per a batch of inserted records, and that increases the speed a lot. See details on the Direct and OCI modes here.
- Devart MySQL Destination uses INSERT statements that insert data by several rows at the same time.
- Devart PostgreSQL Destination uses the COPY command.
- Devart AzureDWH Destination uses PolyBase technique to speed up data loading. It uploads data as CSV files to Azure Blob Storage and tells Azure SQL Data Warehouse to import data. After the data is imported, the CSV files are deleted.
- Devart Redshift Destination uploads data as CSV files to an Amazon S3 bucket and then tells Redshift to import data using the COPY command. After the data is imported, the CSV files are deleted.
- Devart BigQuery Destination also uploads data as CSV files to a Google Cloud Storage bucket and tells BigQuery to import data. After the data is imported, the CSV files are deleted.

Devart Destination Components

Outputs

Devart Destination components support error outputs, which allow you to manage how the component treats row-level errors in both incoming and outgoing data.

The error output includes the column set taking part in the data flow and additional columns, they are ErrorCode that identifies the error, ErrorColumn that contains the lineage identifier of the error column and ErrorDescription that provides a detailed description of the error.

The ErrorCode column can accept the following values:

ErrorCode	Description
-1	Means that the record is valid.
>0	Means that the record causes an error.

Additionally, when the primary key values are automatically generated for the target table, Devart Destination component retrieves the generated values and adds them to its output columns.

Devart Lookup Components

Devart Lookup components are designed for joining the input data with the data from objects in the corresponding data source.

Note that Devart Lookup components support only the corresponding Devart connection managers.

How to use Devart Lookup Components

To add a Devart Lookup component to a Data Flow task, perform the following steps:

1. Drag the corresponding Devart Lookup component from the SSIS Toolbox window to the Data Flow diagram.
2. Connect the necessary data flow path to it.
3. Double-click the added component to open its editor.
4. On the **Connection Managers** tab select the corresponding Devart Connection Manager or create a new one.
5. In the **Lookup Object** (for cloud sources) or **Lookup Table** (for databases) list select the lookup object.
6. In the **Lookup Columns** list select check boxes for the columns to add to the Lookup match output.
7. In the **Referential Constraint** grid select the corresponding input columns and lookup key columns to match.
8. Click **OK**.
9. Connect the necessary lookup component outputs to other components.

Devart Lookup components provide the following outputs:

- **Lookup Match Output** - processes rows that have at least one matched row in the lookup object/table.
- **Lookup No Match Output** - processes rows that do not match to any of lookup object/table rows.
- **Lookup Error Output** - by default this output is not used. You can set the **NoMatchAsError** property to true in order to send rows that do not match to any of lookup object/table rows to this output.

Performance Optimizations

Devart Lookup components use the following techniques to provide best performance for matching rows:

Batch Check

Devart Lookup components can check several input rows for matches at once with a single query to reduce the number of queries. You can set the maximal number of rows to check in a batch in the **BatchSize** custom property of the Lookup component.

Please note that assigning bigger values to this property increases the query size, and for some sources too large queries may cause errors. In case of errors reduce the value of this property.

This property and this functionality are not supported for Zoho CRM.

Advanced Caching

Devart Lookup components cache the queried rows from the lookup object/table, so if multiple input rows have the same lookup key values, Devart Lookup components uses cached rows instead of querying the source again. You can set the number of rows to cache in the **CacheSize** custom property of the Lookup component.