Blue Prism - Data OLEDB

Introduction

This document is a guide to using OLEDB within the Blue Prism Robotic Process Automation solution. The OLEDB libraries allow interaction with a number of target data files (including text, CSV, Excel, and Access) via a subset of SQL. This has advantages in both speed and in efficiencies in file handling without fully loading the file into memory at any given time.

Pre Requisites

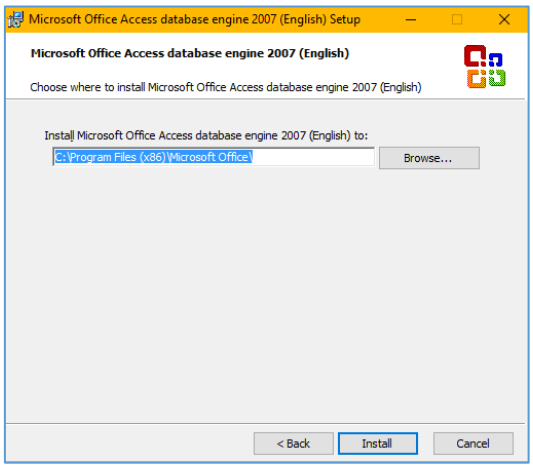
DLL Installation

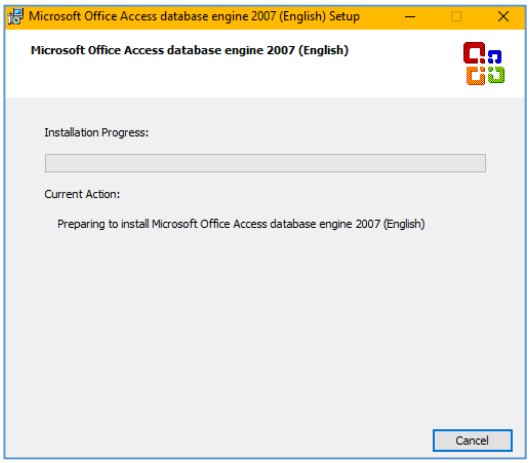
There are multiple versions of the OLEDB library available, which are based out of different versions of Access. Depending on what software is installed, OLEDB may already be installed.

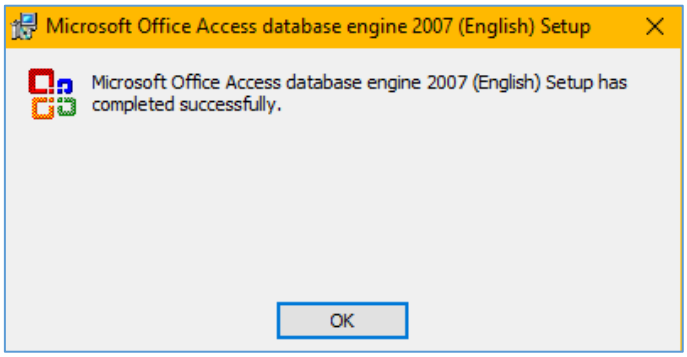
To check go to Programs and Features within the Control Panel and look for Microsoft Access Database Engine. If not installed, it can be downloaded from Microsoft. A usable version (Office 2007) is available here:

https://www.microsoft.com/en-us/download/confirmation.aspx?id=23734

Clicking on the link will allow you to download the required software which in turn will need installing:

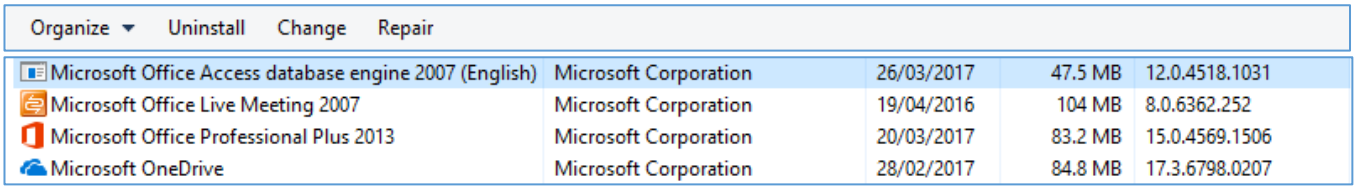






Follow the wizard through and the installation should take no more than 60 seconds to install.

Once installed you will a new entry in the Programs and Features within the Control Panel called Microsoft Access Database Engine 2007 (English).



If necessary, other versions can be used (e.g. a newer version of the Database Access Library) for compatibility or other purposes.

VBO Installation

Once the OLEDB DLL is installed, ensure that the "Data – OLEDB" VBO is imported correctly into Blue Prism. If not, it should be present in the following location depending on the architectural installation of Blue Prism used:

64 Bit - C:\Program Files\Blue Prism Limited\Blue Prism Automate\VBO\

32 Bit - C:\Program Files (x86)\Blue Prism Limited\Blue Prism Automate\VBO\

Standard OLEDB Usage

The following details the basic usage of the OLEDB in Blue Prism.

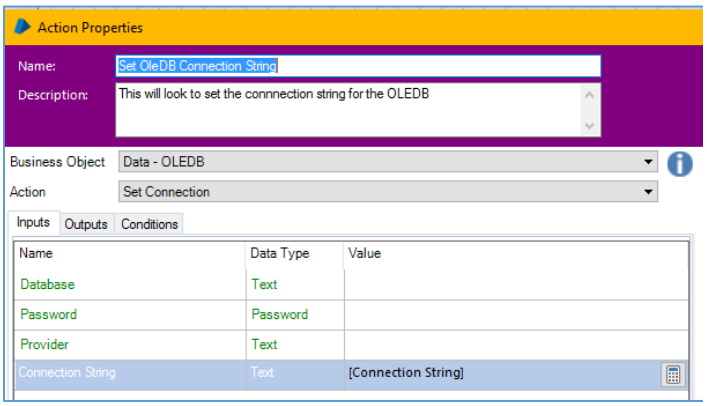
Set Connection

The OLEDB connection string must be set in the VBO as the first step.

Example Connection String:

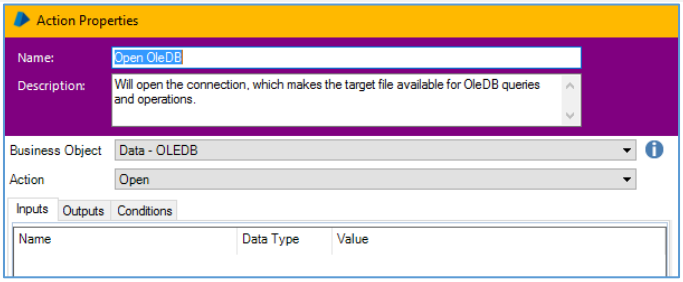
Provider=Microsoft.ACE.OLEDB.12.0; Data Source="C:\BluePrism\Training\Blue Prism - Sample Order Data.xlsx";Extended Properties="Excel 12.0 Xml;HDR=YES;"

"HDR=Yes;" indicates that the first row contains column names, not data. "HDR=No;" indicates the opposite.



Open Connection

This will open the connection, which makes the target file available for OLEDB queries and operations.

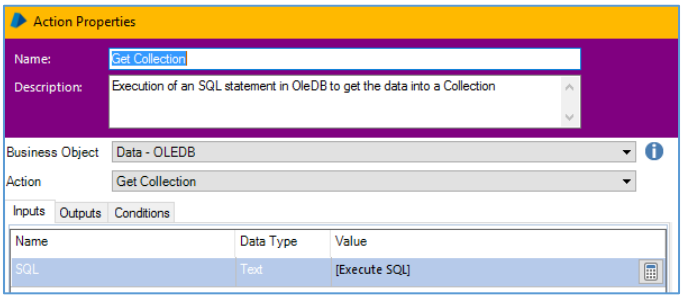


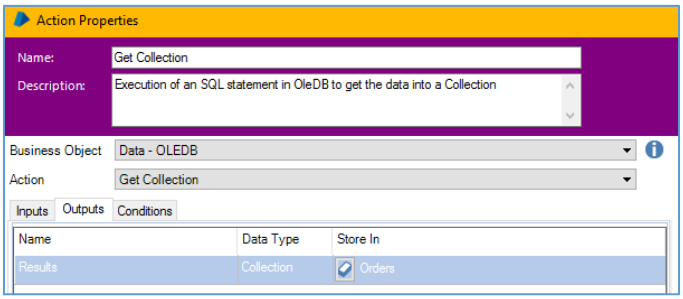
Get Collection

Execution of an SQL statement in OLEDB to get the data into a Collection

Example: *"SELECT [ID No], [Cost Centre], [Value] FROM [Orders$] WHERE [Type] = 'New Order'"*

The Output is a collection of the data

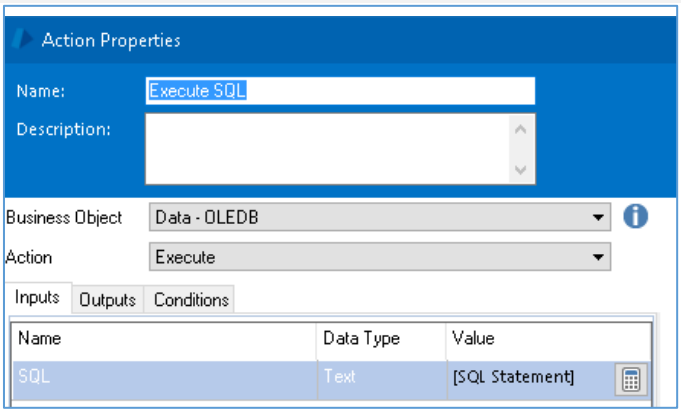




Execute

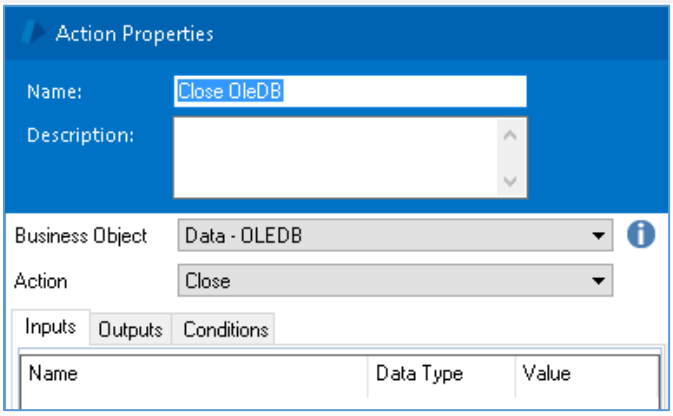
Execute one or more SQL statements in OLEDB. This can be used to write data to a worksheet.

Example of inserting values from a Collection into a worksheet without defined column headers: *INSERT INTO [Sheet1$] ([F1], [F2], [F3]) VALUES ("" & [Results.ID] & "","" & [Results.First Name] & "","" & [Results.Surname] & "")"*



Close OLEDB Connection

When finished, you must close the OLEDB connection to correctly finalize operations.



Implementation Tips

This section details some useful tips and things to remember while working with the OLEDB library.

a. Sheets and Columns

- For Excel files, sheet names are table names. Wrap sheet names in square brackets with a trailing dollar sign to refer to existing sheets, e.g. [Sheet1$]
- For files with column names, complex column names are possible. Again, enclose them in square brackets for proper usage, e.g. [Date of Project Start]
- For Excel files, if column headers are not specified you must use the following syntax:

  [F1] would be column A, [F2] would be column B, [F5] would be column E, etc.

b. Connection Strings

- An excellent resource for helping construct connection strings is https://www.connectionstrings.com/excel/.

c. SQL Dialect

- OLEDB uses a subset of SQL from Microsoft Access, not a full SQL implementation.
- DELETE statements do not function.
- DROP TABLE statements work to remove all data from a table, but will not drop the table itself.
- Depending on the file format in use, more than one table may or may not be available.
  - E.G. CSV files only have one table available, and CREATE TABLE statements will fail.
- Statements to modify table structure generally fail through OLEDB.

Issues and Support

For support with using the OLEDB library, please reference the Blue Prism Knowledge Base on the Portal. If a specific issue is not addressed on the Knowledge Base, contact Blue Prism Support for further help.

Common Issues

The following are some issues you may encounter with OLEDB:

1. Error “the 'microsoft.ace.oledb.12.0' provider is not registered on the local machine”
    - If the above error occurs, this generally means the OLEDB provider is not properly installed on the client machine. See https://social.msdn.microsoft.com/Forums/en-US/1d5c04c7-157f-4955-a14b-41d912d50a64/how-to-fix-error-the-microsoftaceoledb120-provider-is-not-registered-on-the-local-machine?forum=vstsdb for further details.
2. Unable to Open Document Via OLEDB
    - OLEDB generally tries to hold an exclusive OS-level lock on any file it operates against. If it cannot get that lock, it can fail fatally with an error message similar to this. In this case, try to find if any process has the file open and close it to release it for OLEDB’s lock. Consider using Environment Locks to manage robot’s access to files to be accessed via OLEDB.
3. Updates/inserts are not saved to document
    - If OLEDB is not explicitly closed, it will not finalize updates to the document. In this case inserts and updates can be entirely lost. Ensure both that your process has an explicit close and that it occurs without error.
4. Data Type Mismatch on Column
    - OLEDB will attempt to automatically determine the data type of a column based on values in that column; however, it will only consider the first approximately 10 rows (depending on the exact implementation). Because of this, it can detect the data type incorrectly.
    - Excel can hide the true data type of a column; it is common, for example, for a date column to be identified as mixed because the actual data in the cells is a combination of numbers and strings in different rows.