



INTEGRATION MODULE FOR ORACLE

For Keverion Runbook Studio and Azure Automation

User Guide

Version 2.6



Kelverion Integration Module for Oracle

Copyright 2018 Kelverion Inc. All rights reserved.

Published: January 2024

The Kelverion Integration Module for Oracle is Microsoft Azure Certified

Feedback

Send suggestions and comments about this document to support@kelverion.com

Contents

Getting Started.....	4
System Requirements.....	4
Deploying the Integration Module	4
Licensing the Integration Module	6
Working with Activities in Runbook Studio.....	7
Smart Connections	7
Global Connection Assets.....	8
Activity Properties	9
Smart Discovery.....	9
Smart Parameters.....	9
Smart Filters	11
Retry Behavior	12
Additional Parameters.....	13
Supported Oracle Data Types.....	13
Delete-OracleRow	15
Insert-OracleRow	16
Invoke-OracleProcedure	17
Invoke-OracleCommand	18
Select-OracleRow.....	19
Update-OracleRow.....	21

Getting Started

The following sections outline how to deploy and configure the Keverion Integration Module for Oracle.

System Requirements

The Integration Module for Oracle requires the following software to be installed and configured prior to implementing the integration. For more information about installing and configuring Oracle, refer to the respective product documentation.

- Keverion Runbook Studio 5.6.1
- Microsoft .NET Framework 4.7.2

Oracle database requirements:

- Oracle 18c
- Oracle 19c
- Oracle 21c
- Oracle 23c

Deploying the Integration Module

The easiest way to install and deploy the Integration Module for Oracle is from the PowerShell Gallery, but you can also download the module from Keverion and perform the steps manually.

You must install and deploy the Integration Module to each Azure Automation Account and Hybrid Worker host system that you plan to use to run your runbooks. You must also install the Integration Module on any Runbook Studio host systems that you will be using to build and manage your runbooks.

Using the PowerShell Gallery

Using the commands in the **PowerShellGet** module you can download the Keverion Integration Module for Oracle from the PowerShell Gallery and install it on your local computer. You can also deploy the module directly from the PowerShell Gallery to any of your Azure Automation Accounts.

Install the Integration Module on your local computer:

1. Confirm that the PowerShellGet module is installed.
2. Start a PowerShell window as Administrator and run the command:

```
Install-Module -Name Keverion.Oracle -Scope AllUsers
```

Upload the Integration Module to an Azure Automation Account:

1. Go to the [PowerShell Gallery](#).
2. Click the **Azure Automation** tab.
3. Click **Deploy to Azure Automation**. You will be directed to Microsoft Azure.
4. Select the **Automation Account** that you want to deploy the module to.
5. Click **OK**.

Manual Installation

Alternatively, you can download the Integration Module package from Keverion and deploy it manually to your local computer, hybrid workers and Automation Accounts.

The download package from Keverion includes a **.zip** file containing the Integration Module as well as the User Guide and Release Notes. The following instructions assume that you have unzipped the download package and have access to the **.zip** file containing the Integration Module.

Install the Integration Module on your local computer:

1. Copy the **Keverion.Oracle.zip** file to your local computer.
2. Right-click on the file and select **Properties**.
3. Click the **General** tab. If necessary, click **Unblock**, and then click **OK**.
4. Unzip the **Keverion.Oracle.zip** file.
5. Copy the **Keverion.Oracle** folder to a location in the `%PsModulePath%` path.

Important: When installing the Integration Module on a Hybrid Worker, you must use a location that is accessible to all users of the computer.

Upload the Integration Module to an Azure Automation Account:

1. Sign into [Microsoft Azure](#).
2. Open the Automation Account that you want to upload the module to.
3. Click **Modules** under Shared Resources. The list of installed modules is displayed.
4. Click **Add a module** at the top of the list.
5. In the **Upload File** box, select the **Keverion.Oracle.zip** file that you downloaded.
6. Click **OK**.

Licensing the Integration Module

Licenses for Keverion Integration Modules are managed and deployed using *Keverion Runbook Studio* and *Automation Connection Assets*.

Register an Integration Module license with Runbook Studio:

1. Open **Keverion Runbook Studio**.
2. In the **File** tab, click **About**.
3. Click **License Information**.
4. Click the **Integration Modules** tab, and then click **Add License**.
5. Select the integration module license file (*.kaml) and click **Open**.
6. You should see your entitlements displayed in the list.
7. Click **OK**.

Important: Entitlements will not display until after the Integration Module has been installed on the Runbook Studio computer.

Create a Connection Asset with a license key and upload it to Azure:

1. On the **Home** tab, click **Sign In**. The Sign in dialog appears.
2. Sign into your account.
3. In the **Active Azure Automation Account** box, select that account that you want to add the connection asset. To.
4. Click **New Asset** and then click **Connection**. The New Connection dialog appears.
5. In the **Name** field, enter a name to identify the connection.
6. In the **Connection Type** field, select the desired connection type.
7. Enter the appropriate connection information in the provided fields.
8. Click **OK**.

Update all Connection Assets license keys and upload them to Azure:

1. On the **Home** tab, click **Sign In**. The Sign in dialog appears.
2. Sign into your account.
3. In the Explorer panel, click the **Azure (Online)** group.
4. Right-click the Azure Automation Account that contains the connection assets you want to update, and then and then click **Update License Keys**. A summary is displayed.

Working with Activities in Runbook Studio

The following sections outline some of the common configuration options that are available to you when working with the activities in the Keverion Integration Module for Oracle.

The integration module includes the following activities:

Delete-OracleRow	Delete rows from a database table
Insert-OracleRow	Insert a row into a database table
Invoke-OracleProcedure	Run a stored procedure
Invoke-OracleCommand	Run a script containing SQL statements supported by Oracle
Select-OracleRow	Select rows from a database table using easy-to-use filters
Update-OracleRow	Update rows in a database table

The advanced discovery capabilities provided by the activities in this integration module are only supported when authoring runbooks in Keverion Runbook Studio.

When you publish your runbooks from Keverion Runbook Studio to Azure Automation or when you generate PowerShell code snippets for Service Management Automation, Runbook Studio will automatically convert the dynamically generated parameters and filters of Smart activities into the parameters provided by the underlying command activities.

Smart Connections

In Keverion Runbook Studio you can configure one or more Smart Connections to establish reusable links between Runbook Studio and specific Oracle instances. You can create as many Smart Connections as you require, specifying links to multiple databases. You can also create multiple Smart Connections to the same database to allow for differences in security privileges for different user accounts.

Add a Smart Connection in Keverion Runbook Studio:

1. On the **Home** tab, click **Smart Connections**. The Smart Connections dialog appears.
2. Click **Add a connection** at the top of the list.
3. In the **Name** box, enter the name of the connection.
4. In the **Connection Type** box, select *Keverion.Oracle*.
5. In the **HostName** box, type the IP address or name of the Oracle host.
6. In the **Port** box, type the port used for Oracle. The default is *1521*.
7. In the **ServiceName** box, type the name of the Oracle service you want to connect to.

8. In the **UserName** and **Password** boxes, type the credentials that activity will use to connect to the Oracle database.
9. In the **ConnectionTimeout** box, enter the time in seconds that the Integration Module will wait for a connection to be established.
10. Click **OK**, and then click **OK** again.

Global Connection Assets

The activities in the Kelterion Integration Module for Oracle require connection information to connect to instances of Oracle as well as the Kelterion Management server.

The recommended way to pass connection information to your activities in your runbooks is to use Global Connection Assets. Global connection assets let you securely define connection information in Azure which can then be retrieved on demand using either the *Get-AutomationConnection* cmdlet or Connection Asset Data Source.

Add a global connection asset in Runbook Studio:

1. On the **Home** tab, click **Sign In**. The Sign In dialog appears.
2. Sign into your account.
3. In the **Active Azure Automation Account** box, select the account that you want to add the connection asset to.
4. Click **New Asset** and then click **Connection**. The New Connection dialog appears.
5. In the **Name** box, enter the name of the configuration.
6. In the **Connection Type** box, select *Kelterion.Oracle*.
7. In the **HostName** box, type the IP address or name of the Oracle host.
8. In the **Port** box, type the port used for Oracle. The default is *1521*.
9. In the **ServiceName** box, type the name of the Oracle service you want to connect to.
10. In the **UserName** and **Password** boxes, type the credentials that activity will use to connect to the Oracle database.
11. In the **ConnectionTimeout** box, enter the time in seconds that the Integration Module will wait for a connection to be established.
12. In the **ShowAll** box, select **True** to display all tables and view or **False** to display user tables and view.
13. Click **OK**.

Activity Properties

All activities in Kelterion Integration Module for Oracle have the following properties:

Property	Description
Label	A unique label that identifies the activity in the runbook. Runbook Studio will provide a default name for each activity, but you can provide your own labels to make their role in the runbook more obvious.
Description	An optional description of the activity. Providing a description is a fantastic way to let everyone understand the function of the activity in the runbook.
Checkpoint	<p>Indicates whether a checkpoint is set in the runbook workflow after the activity runs. Checkpoints are only available for Graphical PowerShell Workflow runbooks.</p> <p>If the runbook uses Azure cmdlets you should follow best practices and follow a check-pointed activity with an <u>Add-AzureRMAccount</u> in case the runbook is suspended and restarts from this checkpoint on a different worker.</p>

Smart Discovery

When designing runbooks in Kelterion Runbook Studio, you will notice that the activities in the Kelterion Integration Module for Oracle include a **Discovery** panel instead of the **Parameter Sets** panel that is present for standard command activities. This is because the activities in the Kelterion Integration Module for Oracle support interactive discovery of the database assets in your on-premises and cloud environments.

All activities in the Kelterion Integration Module for Oracle have a **Connection** option on the **Discovery** panel which lets you specify how Runbook Studio should connect to Oracle.

When connected to Oracle, Runbook Studio will provide additional discovery options, such as Database and Table Name, which can be used to specify the database resources that you want to integrate with. Once you have filled in the discovery options Runbook Studio will provide additional parameters, and in some cases, filters which can be used to configure the activity.

Smart Parameters

Unlike standard command activities whose parameters are determined by the Parameter Set that is selected, the parameters in the Kelterion Integration Module for Oracle are determined by the Discovery options that you specify.

For example, when using the **Insert-OracleRow** activity, the Discovery panel will contain options for selecting Database Name and Table Name. Once you have selected a table, Runbook Studio will provide you with parameters that coincide with the columns in the table schema. If you select another table, Runbook Studio will provide you with a separate set of parameters automatically.

You must configure all mandatory parameters. To view the optional parameters that are associated with an activity, click **Optional** at the top of the Parameters tab.

In addition, all activities in the Keverion Integration Module for Oracle include a **Connection** parameter which is used to specify information that the activity will use to connect to Oracle when it is executed as part of a runbook running in Azure. Typically, you will assign a Connection Asset data source to this parameter so that the activity can securely use connection information stored in Azure. The Connection parameter should not be confused with the similarly named Connection option on the Discovery panel which is used to specify how Runbook Studio connects to Oracle to provide design-time configuration options.

Several factors determine the data sources that are available when configuring a parameter. They include: the parameter's data type, whether it is linked to another activity and whether the runbook has any input parameters.

Runbook studio supports the following data source:

Data Source	Description
Activity output	<p>Specify activity whose output will be assigned to the parameter. You may also provide an optional Path to select a specific property of the output objects that are generated by the activity.</p> <p>Available when the activity is linked to a source activity.</p>
Not configured	<p>Clears any value that was previously configured. You must configure all mandatory parameters.</p>
Certificate asset	<p>Specify the name of the global certificate asset that will be used to provide a value for the parameter.</p> <p>If you have connected to Azure and selected a Subscription and Automation Account on the toolbar, the data source will provide the names of the certificates that are available.</p>
Credential asset	<p>Specify the name of the global credential asset that will be used to provide a value for the parameter.</p> <p>If you have connected to Azure and selected a Subscription and Automation Account on the toolbar, the data source will provide the names of the credentials that are available.</p>
Constant	<p>Specify a constant value to assign to the parameter.</p> <p>Available for parameters that have the following data types:</p> <ul style="list-style-type: none">• String• DateTime• Boolean• Char• Byte• SByte• Int16• Int32• Int64• UInt16• UInt32• UInt64• Decimal• Double• Float• SwitchParameter

	When assigning a constant DateTime value, Runbook Studio assumes the value is in UTC.
Connection asset	Specify the name of the global connection asset that will be used to provide a value for the parameter. If you have connected to Azure and selected a Subscription and Automation Account on the toolbar, the data source will provide the names of the connections that are available.
Empty string	An empty string will be assigned to the parameter. Available when the parameter is typing <i>System.String</i>
Null	A null (\$null) value will be assigned to the parameter. Available when the parameter type is a reference type.
PowerShell expression	Specify a <i>simple</i> PowerShell expression whose output will be assigned to the parameter. You can use variables in the expression to access the output of an activity or a runbook parameter.
Runbook input	Specify the name of the runbook input parameter whose value will be assigned to the parameter. Available when the runbook has one or more input parameters.
Variable asset	Specify the name of the global variable asset that will be used to provide a value for the parameter. If you have connected to Azure and selected a Subscription and Automation Account on the toolbar, the data source will provide the names of the variables that are available.

Smart Filters

Some of the activities in the Integration Module include a Filters panel which lets you specify filters that can be used to retrieve or modify specific rows in a database table.

If you do not provide any filters, then the activity will retrieve or modify all records in the target table.

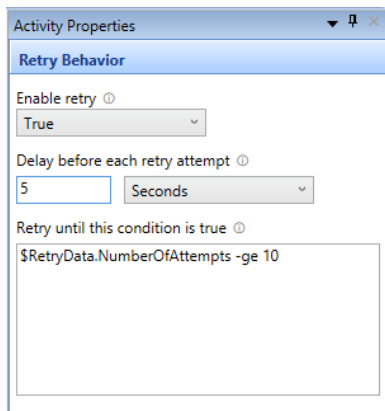
To add a filter to your activity, select the **Filters** panel and click **Add**. Filters have the following properties.

Property	Description
Filter	The name of the filter.
Operation	The operation to evaluate the filter. Different operators will be provided based on the filter that is selected. Filter operators include: <ul style="list-style-type: none"> Equals Does not equal Contains Does not contain

	<ul style="list-style-type: none"> • Is less than • Is less than or equal to • Is greater than • Is greater than or equal to • Matches • Does not match • Starts with • Ends with
Value	<p>The data source used to retrieve the value to evaluate the filter.</p> <p>The value used to evaluate the filter will be obtained. For more information on data sources, please refer to the Parameters section for more information on configuring data sources.</p> <p>If you want to filter for a NULL database value, use a PowerShell Expression data source and enter the expression [DBNull]::Value.</p>

Retry Behavior

The activities in the Kelverion Integration Module for Oracle can be configured to run multiple times until a particular condition, which you specify, is satisfied. You can use the retry behavior options to configure activities that should run multiple times, that are error prone or may need more than one attempt for success.



When you enable retry for an activity, you can configure the runbook to wait a specified number of minutes or seconds before running the activity again. If no delay is specified the runbook will run the activity again, immediately after it completes.

The retry condition lets you specify a PowerShell expression that the runbook will evaluate after each time the activity runs. If the result of the expression is true the activity does not run again, and the runbook moves on to the next child activity in the runbook.

When defining the retry conditions for your activity you can take advantage of a global variable called **\$RetryData**. Specific information about the last time the activity ran can be accessed using the following properties.

Property	Description
NumberOfAttempts	Number of times that the activity has ran
Output	Output that was generated by the activity the last time that it ran
TotalDuration	Time elapsed since the activity was started
StartedAt	Time in UTC when the activity was first started

The following are some examples of activity retry conditions.

```
# Run the activity exactly five times
$RetryData.NumberOfAttempts -eq 5

# Run the activity until it produces some output
$RetryData.Output.Count -ge 1

# Run the activity until at least 2 minutes has elapsed
$RetryData.TotalDuration.TotalMinutes -ge 2
```

Additional Parameters

The activities in the Kelverion Integration Module for Oracle let you specify additional PowerShell parameters that you can use to control the behavior of the activity.

For example, to output detailed information about the operation performed by an activity you would specify **-Verbose:\$True**

Supported Oracle Data Types

The Integration Module for Oracle supports most Oracle built in data type, although only a subset of types can be used with smart activity filters. Tables that contain columns with unsupported data types can still be accessed, but any unsupported columns will not be available for activity input properties, filters, or output.

The following table outlines the built-in Oracle data types are supported:

Oracle Data Type	Notes
BINARY DOUBLE	System.Decimal for input and output
BINARY FLOAT	System.Decimal for input and output
BLOB	System.Byte[] for input and output. Does not support filtering.
CHAR	System.String for input and output.
CLOB	System.String for input and output. Does not support filtering.
DATE	System.DateTime for input and output. Time portion is ignored.
INTERVAL DAY TO SECOND	Accepts System.TimeSpan, a System, Double representing the number of days in the interval or a System.String representing a literal interval*.

	Output is returned as a System.TimeSpan.
INTERVAL YEAR TO MONTH	<p>Accepts System.Int64 number representing the number of months in the interval or System.String that represents a literal interval*.</p> <p>Output is returned as a System.Int64 representing the number of months in the interval.</p>
NCHAR	System.String for input and output
NUMBER	System.Decimal for input and output
NVARCHAR2	System.String for input and output
TIMESTAMP	<p>Accepts System.DateTime, System.DateTimeOffset or System.String values. Values are not changed, and offsets are ignored. String values must be in round-trip format. †</p> <p>Output is returned as System.DateTime values with kind set to <i>Unspecified</i>.</p>
TIMESTAMP WITH TIMEZONE	<p>Accepts System.DateTime, System.DateTimeOffset values. System.DateTime or System.String values are converted to DateTimeOffset values with either an offset of 00:00 or offset of local time zone based on whether the Kind is <i>Utc</i>, <i>Local</i> or <i>Unspecified</i>. Offsets are preserved in Oracle. String values must be in round-trip format. †</p> <p>Output is returned as System.DateTimeOffset values with offset information preserved.</p>
TIMESTAMP WITH LOCAL TIMEZONE	<p>Accepts System.DateTime and System.DateTimeOffset or System.String values. Values are converted to the time zone of the Oracle database instance. String values must be in round-trip format. †</p> <p>Output is returned as System.DateTime values converted to the local time.</p>
VARCHAR2	System.String for input and output

* For more information on Oracle literals refer to

https://docs.oracle.com/cd/B28359_01/server.111/b28286/sql_elements003.htm#SQLRF00221

† For information on the Round-trip date/time format refer to <https://docs.microsoft.com/en-us/dotnet/standard/base-types/standard-date-and-time-format-strings#Roundtrip>

Delete-OracleRow

The **Delete-OracleRow** activity is used in a runbook to delete rows from a database table. Smart Filters are used to select which rows to delete.

Discovery Parameters

You can use the following discovery options to connect to Oracle and configure the activity:

Connection	The name of the Smart Connection used to connect Runbook Studio to Oracle.
Table Name	The name of the database table from which to delete rows.

Required Parameters

You must configure the following parameters:

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
-------------------	--

Optional Parameters

You can use the following optional parameters to control the behavior of the activity:

Command Timeout	The number of seconds that the activity will wait for the database command to be executed before failing with an error.
------------------------	---

Filters

The activity will provide filters based on the columns in the table that you selected. You can configure one or more filters to determine which rows to delete. **If you do not define any filters, the activity will truncate every row in the table.**

Output

The activity outputs the number of rows that were deleted. When no filters are provided, the activity returns -1 to indicate that the table has been truncated.

Insert-OracleRow

The **Insert-OracleRow** activity is used in a runbook to insert a row into a database table.

Discovery Parameters

You can use the following discovery options to connect to Oracle and configure the activity:

Connection	The name of the Smart Connection used to connect Runbook Studio to Oracle.
Table Name	The name of the database table into which the row will be inserted

Required Parameters

If the table you selected has columns that do not allow NULL, then the activity will provide parameters that you must configure. You must also configure the following parameters:

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
-------------------	--

Important: For tables that have an **identity column**, the Insert-Oracle activity can only insert records where the identity column was created with the option GENERATED BY DEFAULT ON NULL. When configuring the activity, assign the required identity column parameter a \$null value.

Optional Parameters

If the table that you selected has any columns that allow NULL, then the activity will provide corresponding parameters that you can configure. You can also use the following parameters to control the behavior of the activity:

Command Timeout	The number of seconds that the activity will wait for the database command to be executed before failing with an error.
------------------------	---

Output

The activity does not generate any output.

Invoke-OracleProcedure

The **Invoke-OracleProcedure** activity is used in a runbook to run a stored procedure or function.

Discovery Parameters

You can use the following discovery options to connect to Oracle and configure the activity:

Connection	The name of the Smart Connection used to connect Runbook Studio to Oracle.
Procedure Name	The name of the database procedure to invoke

Required Parameters

If the procedure or function that you selected has required parameters, then the activity will provide corresponding parameters that you must configure. You must also configure the following parameters:

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
-------------------	--

Optional Parameters

If the procedure or function that you selected has optional parameters, then the activity will provide corresponding parameters that you can use to run the procedure. You can also configure the following parameters to control the behavior of the activity:

Command Timeout	The number of seconds that the activity will wait for the database command to be executed before failing with an error.
------------------------	---

Output

The activity outputs an object that represents the output of the stored procedure or function.

Invoke-OracleCommand

The **Invoke-OracleCommand** activity is used in a runbook to execute an SQL query.

Discovery Parameters

You can use the following discovery options to connect to Oracle and configure the activity:

Connection	The name of the Smart Connection used to connect Runbook Studio to Oracle.
Execute Mode	Specifies the execution mode. Select <i>ExecuteQuery</i> to execute the SQL command and return the rows. Select <i>ExecuteNonQuery</i> to execute the SQL command and return the number of rows that were affected. Select <i>ExecuteScalar</i> to execute the query and returns the first column of the first row in the result set returned by the query (additional columns or rows are ignored).

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
Command	The SQL query to execute

Optional Parameters

You can use the following optional parameters to control the behavior of the activity:

Command Timeout	The number of seconds that the activity will wait for the database command to be executed before failing with an error.
------------------------	---

Output

The activity will have different outputs based on the Execute Mode that was selected:

ExecuteQuery	Zero or more objects, representing the results returned from the SQL query. Each custom object contains properties for each column in the retrieved rows
ExecuteNonQuery	The number of rows that were affected by the SQL command.
ExecuteScalar	The first column of the first row in the result set returned by the query (additional columns or rows are ignored).

Select-OracleRow

The **Select-OracleRow** activity is used in a runbook to select rows from a database table or view using filter criteria that you specify.

Discovery Parameters

You can use the following discovery options to connect to Oracle and configure the activity:

Connection	The name of the Smart Connection used to connect Runbook Studio to Oracle.
Table Name	The name of the target database table or view

Required Parameters

You must configure the following parameters:

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
-------------------	--

Optional Parameters

You can configure the following optional parameters to control the behavior of the activity:

Command Timeout	The number of seconds that the activity will wait for the database command to be executed before failing with an error.
Order By	Indicates the column that should be used to order the results
Top*	Indicates the maximum number or rows to select
Descending	Indicates whether to order the results in ascending order. The alternative is descending order. Only used when the Order By property is defined.
Offset*	Indicates the number of rows to skip before row limiting starts. Used in conjunction with Top. The default is no offset.
Percentage*	Indicates that the value of the Top parameter should be evaluated as a percentage. The default is false.
With Ties*	Indicates that ties should be considered when applying row limits.

** Parameters used for limiting the records that are returned from Oracle, including Top, Offset, Percentage and With Ties are only supported by Oracle 12c Release 1 or greater.*

Filters

The activity will provide filters based on the columns in the table that you selected. You can configure one or more filters to determine which rows to select.

Output

The activity will output objects that represent the rows that were selected. The properties of the objects are based on the columns in the database table that you selected.

Update-OracleRow

The **Update-OracleRow** activity is used in a runbook to update rows in a database table or view.

Discovery Parameters

You can use the following discovery options to connect to Oracle and configure the activity:

Connection	The name of the Smart Connection used to connect Runbook Studio to Oracle.
Table Name	The name of the database table to update

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
-------------------	--

Optional Parameters

The activity will provide parameters that correspond to the columns in the database table that you selected. You can use the following parameters to control the behavior of the activity.

Command Timeout	The number of seconds that the activity will wait for the database command to be executed before failing with an error.
------------------------	---

Filters

The activity will provide filters based on the columns in the database table that you selected. You can configure one or more filters to determine which rows to update. ***If you do not define any filters, the activity will update every row in the table.***

Output

The activity will output the number of records that were updated.

Remarks

If you want to assign a NULL value to a database column, use a PowerShell Expression data source and enter the expression `[DBNull]::Value`.