



INTEGRATION PACK FOR ORACLE

For Microsoft System Center Orchestrator

For System Center 2016 and 2019, you must use the 32-bit version of the integration pack, which has the name **Keverion_Integration_Pack_for_Oracle_3.2**

For System Center 2022 and later, you must use the 64-bit version of the integration pack, which has the name **Keverion_IP_for_Oracle_x64_3.2**

User Guide

Version 3.2

Kelverion Integration Pack for Oracle

Copyright 2012 Kelverion Inc. All rights reserved.

Published: January 2024

[Feedback](#)

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

Contents

Kelverion Integration Pack for Oracle	4
Introduction.....	4
System Requirements.....	5
Registering and Deploying the Integration Pack	5
Licensing the Integration Pack.....	6
Configuring the Kelverion Integration Pack for Oracle.....	6
Oracle Activities	8
Common Configuration Instructions for All Activities.....	8
Supported Oracle Data Types	8
Activity Properties.....	8
General Tab.....	9
Properties/Filters Tab	9
Run Behavior Tab	10
Published Data	11
Delete Rows Activity	12
Insert Rows Activity.....	13
Monitor Rows Activity.....	14
Select Rows Activity	15
Update Rows Activity	16

Kelverion Integration Pack for Oracle

The Integration Pack for Oracle is an add-on for Microsoft System Center Orchestrator that lets you query and manipulate Oracle databases tables.

Introduction

This Integration Pack was created from experience using Orchestrator in large-scale automation scenarios. The objective was to make the interaction with databases more efficient and easier to use by building more capability into the individual activities and reducing effort to construct and post-process the delimited data, as individual published data items further down the workflow.

When implementing Orchestrator one of the key best practices for success is to implement a database alongside to provide a persistence store, and audit report and to increase the power and flexibility of the resulting workflows.

When adding a database alongside Orchestrator there needs to be an efficient connection mechanism. The existing activities create a connection for each transaction and close it after use, even if there is more data to process to the target table. In scenarios of heavy database interaction this puts an unsustainable load on the database access authentication method and for large updates this can increase the workflow run time. An ability to maintain the connection is required to increase performance.

To improve ease of use a key requirement was to remove the need to understand and build SQL statements that would be executed in an activity. Once executed ideally the activity should exit published data items without any further processing steps or activities. This simplification of workflows would be powerful for both users experienced in Orchestrator workflow development and those new to Orchestrator. The resulting benefits would be to remove any potential errors in the SQL statements, faster workflow development and simpler workflows to maintain and support.

A database is frequently used to track the steps of a process whether it is to do with provisioning, service requests or standard run books. There is a need to be able to flexibly link to the data sources being populated by portals or other request mechanisms. An example here is a service request for a new Virtual Machine within a Virtual Lifecycle Management process or Private Cloud implementation. Typically, this request has multiple steps to perform starting as 'new' resulting in 'completed.'

Orchestrator not only needs to monitor for new and updated records but also has different workflows that would need to be triggered at each stage. A flexible monitor activity is required, where the user can define which columns represent the status of the record, either as a numeric or text-based field. This monitor should not only recognize a matching condition to initiate a workflow but also return an appropriate update to that record.

The Kelverion Integration Pack for Oracle adds these capabilities to the Orchestrator workflow designer and installs them as a compliant Integration Pack. The rest of this guide describes how to install and use the five new activities provided.

System Requirements

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

Kelverion_Integration_Pack_for_Oracle (32-bit)

- Microsoft System Center Orchestrator 2016, 2019
- Microsoft .NET Framework 4.6.2

Kelverion_IP_for_Oracle_x64 (64-bit)

- Microsoft System Center Orchestrator 2022
- Microsoft .NET Framework 4.6.2

The following Oracle databases are supported:

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

Registering and Deploying the Integration Pack

After you download the integration pack file, you must register it with the Orchestrator management server and then deploy it to Runbook Servers and Runbook Designers. For more information about how to install integration packs, see the [How to Install an Integration Pack](#) in the online documentation for System Center Orchestrator.

IMPORTANT: Ensure that you are deploying the correct version of the Integration Pack.

- For System Center 2016 and 2019, you must use the 32-bit version of the integration pack, which has the name **Kelverion_Integration_Pack_for_Oracle**
- For System Center 2022 and later, you must use the 64-bit version of the integration pack, which has the name **Kelverion_IP_for_Oracle_x64**

To register the integration pack:

1. On the management server, copy the **.OIP** file for the integration pack to a local hard drive or network share.
2. Confirm that the file is not set to **Read Only** to prevent unregistering the integration pack later.
3. Start the **Deployment Manager**.
4. In the navigation pane of the Deployment Manager, expand **Orchestrator Management Server**, right-click **Integration Packs** to select **Register IP with the Orchestrator Management Server**. The **Integration Pack Registration Wizard** opens.
5. Click **Next**.
6. In the **Select Integration Packs or Hotfixes** dialog box, click **Add**.

7. Locate the **.OIP** file that you copied locally from step 1, click **Open** and then click **Next**.
8. In the **Completing the Integration Pack Wizard** dialog box, click **Finish**.
9. On the **End User Agreement** dialog box, read the Keverion License Terms, and then click **Accept**.
10. The **Log Entries** pane displays a confirmation message when the integration pack is successfully registered.

To deploy the integration pack:

1. In the navigation pane of the **Deployment Manager**, right-click **Integration Packs**, click **Deploy IP to Runbook Server or Runbook Designer**.
2. Select the integration pack that you want to deploy, and then click **Next**.
3. Enter the name of the runbook server or computers with the Runbook Designer installed, on which you want to deploy the integration pack, click **Add**, and then click **Next**.
4. Continue to add additional runbook servers and computers running the Runbook Designer, on which you want to deploy the integration pack. Click **Next**.
5. In the **Installation Options** dialog box configure the following settings.
6. To choose a time to deploy the integration pack, select the **Schedule installation** check box, and then select the time and date from the **Perform installation** list.
7. Click one of the following:
 - a. **Stop all running runbooks before installing the integration pack** to stop all running runbooks before deploying the integration pack.
 - b. **Install the Integration Packs without stopping the running Runbooks** to install the integration pack without stopping any running runbooks.
8. Click **Next**.
9. In the **Completing Integration Pack Deployment Wizard** dialog box, Click **Finish**.
10. When the integration pack is deployed, the **Log Entries** pane displays a confirmation message.

Licensing the Integration Pack

After you register and deploy the integration pack you must provide a valid Keverion license before running any runbooks that contain activities from the integration pack

To deploy the integration pack license file to System Center Orchestrator 2019 or earlier:

1. Copy the **.KAL** license file to %PROGRAMFILES(X86)%\Keverion Automation\Licenses
2. Repeat for each Orchestrator Runbook Server and Runbook Designer host system.

To deploy the integration pack license file to System Center Orchestrator 2022 or later:

1. Copy the **.KAL** license file to %PROGRAMFILES%\Keverion Automation\Licenses
2. Repeat for each Orchestrator Runbook Server and Runbook Designer host system.

Configuring the Keverion Integration Pack for Oracle

A configuration establishes a reusable link between Orchestrator and a specific Oracle database. You can create as many configurations as you require, specifying links to multiple databases. You can also create multiple configurations to the same database to allow for differences in security privileges for different user accounts.

To set up an Oracle configuration

1. In the Client, click the **Options** menu, and select *KA Oracle*. The **KA Oracle** dialog box appears.
2. On the **Configurations** tab, click **Add** to begin the configuration setup. The **Add Configuration** dialog box appears.
3. In the **Name** box, enter a name for the configuration. This could be the name of the database or a descriptive name to distinguish the type of configuration.
4. Click the ellipsis button (...) next to the **Type** box and select *Oracle Configuration*.
5. In the **Host Name** box, type the name or address of the Oracle host system.
6. In the **Server Port** box, type the port used to access the Oracle database.
7. In the **SID/Server Name** box, type the name of the database you want to connect to.
8. Optionally, in the **Connection Timeout** box, type the time in seconds that the Integration Pack will wait for a connection to be established.
9. Add additional connections if applicable.
10. Click **OK** to close the configuration dialog box, and then click **Finish**.

Oracle Activities

This integration pack adds the **KA Oracle** category to the **Activities** pane in the Client.

This category contains the following activities:

- Delete Rows
- Insert Rows
- Monitor Rows
- Select Rows
- Update Rows

Common Configuration Instructions for All Activities

The following configuration instructions apply to all activities in this integration pack. Links to this section are included in the configuration instructions for each activity.

Supported Oracle Data Types

The Integration Pack for Oracle supports most Oracle built in data types. However, some types, such as RAW, CLOB and NCLOB, are excluded due to lack of .NET support or because they cannot be stored in Orchestrator data bus. 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 published data.

The following built-in Oracle data types are supported:

- VARCHAR2
- NVARCHAR2
- NUMBER
- DATE
- BINARY_FLOAT
- BINARY_DOUBLE
- TIMESTAMP
- INTERVAL YEAR TO MONTH
- INTERVAL DAY TO SECOND
- CHAR
- NCHAR
- TIMESTAMP WITH TIMEZONE
- TIMESTAMP WITH LOCAL TIMEZONE

Activity Properties

Each activity has a set of required or optional properties that define the configuration of that activity. This includes how it connects to other activities or how the activity performs its actions. You can view or modify activity properties in the Orchestrator Client.

To configure the properties for an activity

1. Double-click the activity. Alternatively, you can right-click the activity, and then click **Properties**.
2. To save your configuration entries, click **Finish**.

In the activity properties dialog box, several tabs along the left side provide access to general and specific settings for the activity. Although the number of available tabs for activity properties differs

from activity to activity, all activities will have a **General** tab, a **Properties** tab and/or **Filters** tab, and a **Run Behavior** tab. Some activities may have additional tabs.

Working with NULL Values

For columns that allow NULL values, you can specify that you want to assign a NULL value to a column by using the **NULL** keyword in the appropriate activity property.

General Tab

This tab contains the **Name** and **Description** properties for the activity. By default, the **Name** of the activity is the same as its activity type, and the **Description** is blank. You can modify these properties to create more descriptive names or provide detailed descriptions of the actions of the activity.

Properties/Filters Tab

These tabs contain properties that are specific to the activity.

All activities in this integration pack have the **Configuration Name** property at the top of the **Properties** tab. This property is used to specify the connection to an Oracle table.

To configure the Configuration Name property

- Click the ellipsis (...) button next to the **Name** field, and then select the applicable connection name. Connections displayed in the list have been previously configured as described in [Configuring the Oracle Connections](#).

Filter Behavior

The Select and Monitor activities use filters to determine the values that will invoke a runbook or retrieve activities. Property values of potential candidates are compared to the values of the filters to determine if they meet the criteria. When matching against values, you can select one of the available methods of comparison. An option is provided to either match or not match the filter using each method. For example, the "Does not" version of a method finds messages that do not match the filter to start the activity. All text filters are case sensitive.

- **Equals:** the field of the record exactly matches the text or number specified in the filter.
- **Does not equal:** the field of the record does not exactly match the text or number specified in the filter.
- **Is less than:** the field of the record is less than the number specified in the filter.
- **Is less than or equal to:** the field of the record is less than or equal to the number specified in the filter.
- **Is greater than:** the field of the record is greater than the number specified in the filter.
- **Is greater than or equal to:** the field of the record is greater than or equal to the number specified in the filter.
- **Contains:** the field of the record contains the exact text specified in the filter. Unlike the Equals behavior, there can be other text surrounding the matching text.

- **Does not contain:** the field of the record does not contain the exact text specified in the filter. Unlike the Equals behavior, there can be other text surrounding the matching text.
- **Matches:** the field of the record matches the text specified in the filter. Uses syntax and semantics comparable to the SQL LIKE operator.
- **Does not match:** the field of the record does not match the text specified in the filter. Uses syntax and semantics comparable to the SQL LIKE operator.
- **Starts with:** the field of the record starts with the exact text specified in the filter. Unlike the Equals behavior, there can be other text following the matching text.
- **Ends with:** the field of the record ends with the exact text specified in the filter. Unlike the Equals behavior, there can be other text preceding the matching text.

The following table describes the wildcards that are support when using the **Matches** and **Does not match** filter operators.

Wild Card Character	Description
%	Any string of zero or more characters
_ (underscore)	Any single character

Working with NULL Values

For columns that allow NULL values, you can specify that you want to filter on NULL values by using the **NULL** keyword when specifying the filter value. Note that the NULL keyword is only valid when using the **Equals** or **Does not equal** filter operators.

Run Behavior Tab

This tab contains the properties that determine how the activity handles multi-value published data and what notifications will be sent if the activity fails or runs for an excessive period.

Multi-Value Published Data Behavior

The Get activities retrieve information from another activity or outside source and can return one or more values in the published data. For example, when you use the Get Collection Member activity, the data output from that activity might be a list of computers that belong to the specified collection.

By default, the data from the Get activity will be passed on as multiple individual outputs. This invokes the next activity as many times as there are items in the output. Alternatively, you can provide a single output for the activity by enabling the **Flatten** option. When you enable this option, you also choose a formatting option:

- **Separate with line breaks.** Each item is on a new line. This format is useful for creating human-readable text files for the output.
- **Separate with _.** Each item is separated by one or more characters of your choice.
- **Use CSV format.** All items are in CSV (comma-separated value) format. This format is useful for importing data into spreadsheets or other applications.

The activity will produce a new set of data every time it runs. The **Flatten** feature does not flatten data across multiple instances of the same activity.

Event Notifications

Some activities are expected to take a limited amount of time to complete. If they do not complete within that time they may be stalled or there may be another issue preventing them from completing. You can define the number of seconds to wait for completion of the action. After this period, a platform event will be sent, and the issue will be reported. You can also choose whether to generate a platform event if the activity returns a failure.

To be notified when the activity takes longer than a specified time to run or fails to run:

1. In the **Event Notifications** box, enter the **number of seconds** of run time before a notification is generated.
2. Select **Report if activity fails to run** to generate run failure notifications.

For more information about Orchestrator events, see the "Event Notifications " topics in the [Runbook Properties](https://technet.microsoft.com/en-us/library/hh489610.aspx#EventNotifications) ([https://technet.microsoft.com/en-us/library/hh489610.aspx#Event Notifications](https://technet.microsoft.com/en-us/library/hh489610.aspx#EventNotifications)).

Published Data

Published data is the foundation of a working runbook. It is the data produced because of the actions of an activity. This data is published to an internal data bus that is unique for each runbook. Subsequent activities in the runbook can subscribe to this data and use it in their configuration. Link conditions also use this information to add decision-making capabilities to runbooks.

An activity can subscribe only to data from the activities that are linked before it in the runbook. You can use published data to automatically populate the property values needed by activities.

To use published data

1. Right-click the property value box, click **Subscribe**, and then click **Published Data**.
2. Click the **Activity** drop-down box and select the activity from which you want to obtain the data.
3. To view additional data elements common to all activities, select **Show Common Published Data**.
4. Click the published data element that you want to use, and then click **OK**.

For a list of the data elements published by each activity, see the Published Data tables in the activity topic. For information about the common published data items, see the [Published Data](http://technet.microsoft.com/en-us/library/hh403821.aspx) (<http://technet.microsoft.com/en-us/library/hh403821.aspx>).

Delete Rows Activity

The **Delete Rows** activity is used in a runbook to delete rows from a database table.

Required Properties

You must configure the following properties:

Table Name	The name of the database table
-------------------	--------------------------------

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.**

Published Data

The activity has the following published data:

Table Name	The name of the database table
Record Count	The number of records that were deleted
Host Name	The name of the Oracle instance
SID/Server Name	The name of the database that contains the table or view

Insert Rows Activity

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

Required Properties

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

Table Name	The name of the database table to update
-------------------	--

Important: For tables that have an identity column, the Insert Rows 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 property the value *NULL*.

Optional Properties

If the table that you selected has any columns that allow NULL, then the activity will provide corresponding properties that you can configure.

Published Data

The activity generates the following published data:

Table Name	The name of the database table to update
Server Name	The name of the Oracle instance
SID/Server Name	The name of the database that contains the table or view

Monitor Rows Activity

The **Monitor Rows** activity is used in a runbook to monitor a database table or view for changes based on criteria that you specify.

Important: The Monitor Table activity is designed to work with database tables that have user-defined 'state' columns. State columns are typically integer or bit columns whose values are changed by the system to reflect the status of a table row. As a user you can configure the Monitor Table activity to trigger when the values in these state columns change and then use the optional properties to reset the rows to their un-triggered state.

Required Properties

You must configure the following properties:

Monitor Interval	The time in seconds between subsequent attempts to poll the database table
Table/View Name	The name of a database table or view

Optional Properties

The activity will provide optional properties that correspond to the columns in the table that was selected. You can use these properties to update the rows that triggered the monitor.

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 monitor.

Published Data

The activity will generate published data that corresponds to the columns in the table that was selected. Note that the activity will exclude columns whose data type cannot be published to the Orchestrator data bus. The activity also generates the following published data:

Table/View Name	The name of a database table or view
Record Count	The number of records that were updated
Server Name	The name of the Oracle instance
SID/Server Name	The name of the database that contains the table or view

Select Rows Activity

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

Required Properties

You must configure the following properties:

Table/View Name	The name of a database table or view.
------------------------	---------------------------------------

Optional Properties

You can configure the following properties to control the behavior of the activity:

Ascending Order	Indicates whether to order the results in ascending order. The default is descending order.
By Percentage	Indicates whether to fetch a percentage of rows. The default is to fetch a specified number of rows.
Offset	The number of rows to skip before row limiting starts. The default is zero.
Order By	The column that should be used to order the results.
Select Top	The maximum number of rows to fetch.
With Ties	Indicates whether to include additional rows with the same Order By value as the last row selected.

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.

Published Data

The activity will generate published data that corresponds to the columns in the table that was selected. Note that the activity will exclude columns whose data type cannot be published to the Orchestrator data bus. The activity also generates the following published data:

Table/View Name	The name of a database table or view
Record Count	The number of records that were updated
Server Name	The name of the Oracle instance
SID/Server Name	The name of the database that contains the table

Update Rows Activity

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

Required Properties

You must configure the following properties:

Table/View Name	The name of a database table or view
------------------------	--------------------------------------

Optional Properties

The activity will provide optional properties that correspond to the columns in the database table that you selected.

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 update.

Important: Filters are used to control which rows of the table/view are updated. If you do not define any filters, the Update Table activity will **update every row** in the target table/view.

Published Data

The activity generates the following published data:

Table/View Name	The name of a database table or view
Record Count	The number of records that were updated
Server Name	The name of the Oracle instance
SID/Server Name	The name of the database that contains the table