



# KELVERION INTEGRATION PACK FOR AUTOMATION PORTAL

*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 **Kelverion\_Integration\_Pack\_for\_Automation\_Portal\_1.4**

For System Center 2022 and later, you must use the 64-bit version of the integration pack, which has the name **Kelverion\_IP\_Automation\_Portal\_x64\_1.4**

## User Guide

Version 1.4

August 2025

# Kelverion Integration Pack for Automation Portal

Copyright 2024 Kelverion Inc. All rights reserved.

Published: August 2025

## *Feedback*

Send suggestions and comments about this document to [support@kelverion.com](mailto:support@kelverion.com)

# Contents

Installation and Configuration .....	4
System Requirements.....	4
Registering and Deploying the Integration Pack .....	4
Licensing the Integration Pack.....	5
Configuring the Keverion Integration Pack for Automation Portal .....	6
Automation Portal Activities.....	7
Common Configuration Instructions for All Activities.....	7
Activity Properties.....	7
General Tab.....	8
Properties/Filters Tab .....	8
Run Behavior Tab.....	9
Published Data .....	9
Activity Reference .....	11
Approve Request .....	11
Create Request .....	12
Create Request History.....	17
Delete Attachment .....	18
Delete Request .....	18
Download Attachment .....	19
Get Offering.....	20
Get Request .....	22
Get Request Attachment.....	24
Get Request Data .....	26
Get Request History.....	28
Get Service.....	29
Monitor Request .....	31
Reject Request.....	33
Update Request.....	34

# Installation and Configuration

---

The Keverion Integration Pack for Automation Portal is an add-on for System Center Orchestrator that enables you to work with requests submitted from the Keverion Automation Portal.

## System Requirements

The Keverion Integration Pack for Automation Portal requires the following software to be installed and configured prior to implementing the integration. For more information about installing and configuring System Center Orchestrator and **Keverion Automation Portal** refer to the respective product documentation.

### *Keverion\_Integration\_Pack\_for\_Automation\_Portal (32-bit)*

- Microsoft System Center Orchestrator 2016, 2019
- Microsoft .NET Framework 4.7.2

### *Keverion\_IP\_Automation\_Portal\_x64 (64-bit)*

- Microsoft System Center Orchestrator 2022
- Microsoft .NET Framework 4.7.2

### *This integration is compatible with*

- Keverion Automation Portal 4.3

## Registering and Deploying the Integration Pack

After you download the integration pack, you register the integration pack file with the Orchestrator management server and then deploy it to Runbook servers and computers that have the Runbook Designer installed.

**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 **Keverion\_Integration\_Pack\_for\_Automation\_Portal**
- For System Center 2022 and later, you must use the 64-bit version of the integration pack, which has the name **Keverion\_IP\_Automation\_Portal\_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 Kelverion 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 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. For more information about how to install integration packs, see the [How to Install an Integration Pack](https://technet.microsoft.com/en-us/library/hh420346.aspx) (https://technet.microsoft.com/en-us/library/hh420346.aspx).

## Licensing the Integration Pack

After you register and deploy the integration pack you must provide a valid Kelverion 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)%\Kelverion 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%\Kelverion Automation\Licenses
2. Repeat for each Orchestrator Runbook Server and Runbook Designer host system.

## Configuring the Kelverion Integration Pack for Automation Portal

A configuration establishes a reusable link between System Center Orchestrator and a Kelverion Automation Portal server. You can create as many connections as you require specifying links to multiple servers. You can also create multiple connections to the same server to allow for differences in security permissions for different user accounts.

*To set up a Kelverion Automation Portal configuration:*

1. In the Client, click the **Options** menu, and select **KA Automation Portal**.
2. On the **Configurations** tab, click **Add** to begin the connection setup. The **Add Configuration** dialog box appears.
3. Click the ellipsis (...) button next to the **Type** box and select **Kelverion Automation Portal**.
4. In the **Name** box, enter the name for the connection.
5. In the **Portal URL** box, enter the URL of the Kelverion Automation Portal API server. For example, <https://contoso.com/kelverion-automation-portal:8443>
6. In the **Tenant ID** box, enter the ID for the tenant that is hosting your Automation Portal App Registration.
7. In the **Client ID** box, enter the Client/Application ID of your App Registration.
8. In the **Client Secret** box, enter the client secret used to authenticate with the Automation Portal Registration.
9. Optionally, in the **Proxy Server URL** box, type the URL of your proxy server.
10. Optionally, in the **Proxy Server Username** box, type the UPN of the user that will be used to authenticate with the proxy server. For example, [john.smith@contoso.com](mailto:john.smith@contoso.com).
11. Optionally, in the **Proxy Server Password** box, type the password that will be used to authenticate with the proxy server, for example john
12. Optionally, in the **Proxy Server Bypass Local** box, select **True** to bypass the proxy server for local addresses.
13. Click **OK** to close the configuration dialog box, and then click **Finish**.

# Automation Portal Activities

This integration pack adds the **KA Automation Portal** category to the **Activities** pane in Runbook Designer. This category contains the following activities:

<b>Approve Request</b>	Approves an Automation Portal request that is pending approval
<b>Create Request</b>	Create a new Automation Portal request
<b>Create Request History</b>	Create a new history record for the specified Automation Portal request
<b>Delete Attachment</b>	Removes one or more Automation Portal attachments
<b>Delete Request</b>	Removes one or more Automation Portal requests
<b>Download Attachment</b>	Downloads the contents from an Automation Portal attachment
<b>Get Offering</b>	Gets Automation Portal service offerings
<b>Get Request</b>	Gets Automation Portal requests
<b>Get Request Attachment</b>	Gets attachment records for an Automation Portal request
<b>Get Request Data</b>	Retrieves data from an Automation Portal request.
<b>Get Request History</b>	Gets history records for an Automation Portal request
<b>Get Service</b>	Gets Automation Portal services
<b>Monitor Request</b>	Monitor for new and/or updated Automation Portal requests
<b>Reject Request</b>	Rejects an Automation Portal request that is pending approval
<b>Update Request</b>	Updates an Automation Portal request

## 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.

### 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.

## 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.

### *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 Kelverion Integration Pack for Automation Portal](#).

## Filter Behavior

The Monitor and Get 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 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 causes alerts that do not match the filter to trigger the runbook.

- **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.
- **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.

## 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#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 on 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>).

# Activity Reference

---

## Approve Request

The **Approve Request** activity is used in a runbook to approve an Automation Portal request that is pending approval.

### *Required Properties*

You must configure the following properties.

<b>Request ID</b>	Specifies the ID of the request to be approved.
-------------------	---

### *Published Data*

The activity publishes the following activity specific data items.

<b>Can Approve</b>	Indicates whether the request can be approved.
<b>Can Set Reject Reason</b>	Indicates whether you can set a reject reason
<b>Cost Saved</b>	The amount saved by the request
<b>Created</b>	The date and time that the request was created
<b>Deleted</b>	Indicates whether the request deleted
<b>External ID</b>	The external Ref. ID that was assigned to the request
<b>Message</b>	The update message that was assigned to the request
<b>Offering ID</b>	The ID of the service offering
<b>Offering Name</b>	The name of the service offering
<b>Request ID</b>	The identifier used to uniquely identify the request
<b>Request State</b>	The current state of the request.
<b>Requested By</b>	The email of the user that made the request
<b>Requires Approval</b>	Indicates whether the request requires approval
<b>Runbook Owner</b>	The name of the runbook owner
<b>Service ID</b>	The ID of the service that contains the offering
<b>Service Name</b>	The name of the service that contains the offering
<b>State Reason</b>	The reason for the current request state
<b>Tag</b>	The tag assigned to the request offering
<b>Time Saved (Minutes)</b>	The time saved by the request
<b>Updated</b>	The date and time that the request was last updated

## Create Request

The **Create Request** activity is used in a runbook to submit a new request to the Automation Portal. This activity uses discovery to automatically generate appropriate input properties for the offering that is selected, making it easy to submit new requests.

### Required Properties

You must configure the following properties.

<b>Service</b>	Specifies the service that contains the offering that you want to use.
<b>Folder</b>	Specifies the service folder that contains the offering that you want to use. If the offering that you want to use is not in a folder, then select <b>(None)</b> .
<b>Offering</b>	Specify the offering that you want to submit the request for.

### Additional Properties

This Create Request activity provides additional required and optional properties based on the fields in the offering that you selected. The format of the value(s) that can be provided for each field is determined by the type of field's type. The following table outlines the expected formats for each field type.

<b>Date</b>	A local date string with the format <b>yyyy-MM-dd</b> . If a time portion is provided, it is ignored.
<b>DateTime</b>	A <b>local</b> date string with the format <b>yyyy-MM-ddTHH:mm:ss</b> .
<b>File Attachment</b>	An attachment record as JSON formatted string (See remarks for the JSON schema). The content of attachment must be encoded as a Base-64 encoded string.
<b>Hidden</b>	String
<b>ListMultipleSelection</b>	Comma separated values (CSV). Values containing commas or double quotes must be enclosed with double quotes. If double-quotes are used to enclose a value, then a double-quote appearing within the value must be escaped by preceding it with another double-quote.
<b>ListSingleSelection</b>	String
<b>Radio Button</b>	String
<b>SecureTextBox</b>	String
<b>Switch</b>	<b>True</b> or <b>False</b> .
<b>TableMultipleSelection</b>	An array of table rows formatted in JSON (See remarks for the JSON schema).
<b>TableSingleSelection</b>	A table row formatted in JSON (See remarks for the JSON schema).
<b>TextArea</b>	String
<b>TextBox</b>	String

<b>Time</b>	A time string with the format <b>HH:mm</b>
-------------	--

### *Published Data*

The activity publishes the following activity specific data items.

<b>Can Approve</b>	Indicates whether the request can be approved.
<b>Can Set Reject Reason</b>	Indicates whether you can set a reject reason
<b>Cost Saved</b>	The amount saved by the request
<b>Created</b>	The date and time that the request was created
<b>Deleted</b>	Indicates whether the request deleted
<b>External ID</b>	The external Ref. ID that was assigned to the request
<b>Message</b>	The update message that was assigned to the request
<b>Offering ID</b>	The ID of the service offering
<b>Offering Name</b>	The name of the service offering
<b>Requested By</b>	The email of the user that made the request
<b>Request ID</b>	The identifier used to uniquely identify the request
<b>Request State</b>	The current state of the request.
<b>Requested By</b>	The email of the user that made the request
<b>Requires Approval</b>	Indicates whether the request requires approval
<b>Runbook Owner</b>	The name of the runbook owner
<b>Service ID</b>	The ID of the service that contains the offering
<b>Service Name</b>	The name of the service that contains the offering
<b>State Reason</b>	The reason for the current request state
<b>Tag</b>	The tag assigned to the request offering
<b>Time Saved (Minutes)</b>	The time saved by the request
<b>Updated</b>	The date and time that the request was last updated

### *Remarks*

The following JSON schemas can be used to insert complex-data, such as file attachments and table rows.

#### *File Attachment Schema*

The following JSON schema defines the format for strings used to insert attachment data.

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$title" : "Attachment",
  "$description": "A Keverion Automation Portal attachment record",
  "type": "object",
  "properties": {
```

```

    "name": {
      "description": "The name given to the attachment",
      "type": "string"
    },
    "contentType": {
      "description": "The MIME content-type",
      "type": "string"
    },
    "content": {
      "description": "The content as a base-64 encoded string",
      "type": "string"
    }
  },
  "required": ["name", "content"]
}

```

For example:

```

{
  "name": "Test.txt",
  "contentType": "text/plain",
  "content": "TG9yZW0gaXBzdW0gZG9sb3Igc2l0IGFtZXQ="
}

```

## Table Single Selection Schema

The following JSON schema defines the format for strings used to insert data for single selection table fields.

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$title" : "Table Row",
  "$description": "A Keverion Automation Portal table row",
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "name": {
        "description": "The column name",
        "type": "string"
      },
      "value": {
        "description": "The value to insert",
        "type": "string"
      }
    },
    "required": ["name", "value"]
  }
}

```

For example:

```

[
  {
    "name": "ProductID",

```

```

        "value": "707"
    },
    {
        "name": "Name",
        "value": "Sport-100 Helmet, Red"
    },
    {
        "name": "ProductNumber",
        "value": "HL-UF09-R"
    },
    {
        "name": "Color",
        "value": "Red"
    },
    {
        "name": "ListPrice",
        "value": "34.99"
    }
]

```

## Table Multiple Selection Schema

The following JSON schema defines the format for strings used to insert values for multiple selection table fields.

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$title": "Table Row",
  "$description": "A Keverion Automation Portal table row",
  "type": "array",
  "items": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "name": {
          "description": "The column name",
          "type": "string"
        },
        "value": {
          "description": "The value to insert",
          "type": "string"
        }
      },
      "required": ["name", "value"]
    }
  }
}

```

For example:

```
[
```

```
[
  {
    "name": "ProductID",
    "value": "707"
  },
  {
    "name": "Name",
    "value": "Sport-100 Helmet, Red"
  },
  {
    "name": "ProductNumber",
    "value": "HL-UF09-R"
  },
  {
    "name": "Color",
    "value": "Red"
  },
  {
    "name": "ListPrice",
    "value": "34.99"
  }
],
[
  {
    "name": "ProductID",
    "value": "708"
  },
  {
    "name": "Name",
    "value": "Sport-100 Helmet, Black"
  },
  {
    "name": "ProductNumber",
    "value": "HL-UF09"
  },
  {
    "name": "Color",
    "value": "Black"
  },
  {
    "name": "ListPrice",
    "value": "34.99"
  }
]
]
```



## Create Request History

The **Create Request History** activity creates a new history record for the specified Automation Portal request.

### *Required Parameters*

This activity requires the following parameters.

<b>Action</b>	Specify the action to assign the history record
<b>Request ID</b>	Specify the ID of the request to create the history record for

### *Outputs*

This activity outputs an object that represents that request history record that was created. The history object has the following properties.

<b>Action</b>	The action that was assigned to the history record
<b>Created</b>	The date and time that the history record was created
<b>Created By</b>	The email of the user that created the history record
<b>Deleted</b>	Indicates whether the request deleted
<b>History ID</b>	The unique ID of the history record
<b>Request ID</b>	The unique ID of the request that the history is associated with

## Delete Attachment

The **Delete Attachment** activity is used in a runbook to remove an attachment from the Automation Portal.

### *Required Properties*

You must configure the following properties.

<b>Attachment ID</b>	Specifies the ID of the attachment to remove.
----------------------	---

### *Published Data*

The activity publishes the following activity specific data item.

<b>Attachment ID</b>	The ID of the attachment that was deleted.
----------------------	--

## Delete Request

The **Delete Request** activity is used in a runbook to remove a request from the Automation Portal.

### *Required Properties*

You must configure the following properties.

<b>Request ID</b>	Specifies the ID of the request to delete.
-------------------	--

### *Published Data*

The activity publishes the following activity specific data item.

<b>Request ID</b>	The ID of the request that was deleted.
-------------------	---

## Download Attachment

The **Download Attachment** activity is used in a runbook to download the content of a request attachment. The downloaded content is published as a base-64 encoded string.

### *Required Properties*

You must configure the following properties.

<b>Attachment ID</b>	Specifies the ID of the attachment to download from.
<b>Destination Folder</b>	Specifies the folder where the attachment should be downloaded to.
<b>Overwrite</b>	Indicates whether to overwrite the file if it already exists.

### *Published Data*

The activity publishes the following activity specific data items.

<b>Attachment ID</b>	The ID of the attachment that was downloaded.
<b>Content-Type</b>	The MIME type of the attachment.
<b>Destination Filename</b>	The name of the file that that attachment was saved to.
<b>Destination Filepath</b>	The full path of the file that the attachment was saved to, including file name.
<b>Destination Folder</b>	The full path of the folder where the attachment was downloaded to.

## Get Offering

The **Get Offering** activity is used in a runbook to retrieve offering records from the Automation Portal. You can retrieve a specific offering by its unique ID, retrieve all offerings or retrieve a subset of offerings using filters.

### Required Properties

You must configure the following property.

<b>Search By</b>	Indicates whether to retrieve an offering by its <b>Offering ID</b> or to retrieve a collection of offerings using <b>Filters</b> .
------------------	---

### Optional Properties

This activity provides the following optional properties when **Search By** is set to **Filters**.

<b>Descending Order</b>	Indicates that offering records should be sorted in descending order according to the <b>OrderBy</b> value. The default is ascending order.
<b>Limit</b>	The maximum number of offering records to retrieve. When zero is specified, all offerings are retrieved. The default value is <b>one thousand</b> .
<b>Order By</b>	Specifies the property used to order the results. The default value is <b>Offering ID</b> .

### Filters

When **Search By** is set to **Filters**, this activity provides filters that can be used to control which offering records to retrieve from the Automation Portal. The following filters are available.

<b>Active</b>	Filter on whether the offering is active.
<b>Cost Saved</b>	Filter on the cost that has been saved by using the offering.
<b>Created</b>	Filter on the date and time that the offering was created.
<b>Created By</b>	The email of the user that created the history record
<b>Description</b>	Filter on the offering's description
<b>Folder ID</b>	Filter on the ID of the folder that contains the offering.
<b>Mobile Active</b>	Filter on whether the folder is available for mobile devices.
<b>Name</b>	Filter on the offering's name.
<b>Offering ID</b>	The offering's unique ID.
<b>Requires External ID</b>	Filter on whether the offering requires an External Ref. ID.
<b>Service ID</b>	Filter on the ID of the service that contains the offering.
<b>Service Name</b>	Filter on the name of the service that contains the offering.
<b>Tag</b>	Filter on the tag that has been assigned to the offering.
<b>Time Saved</b>	Filter on the time, in minutes, that has been saved by using the offering.

<b>Updated</b>	Filter on the date and time that the offering was last updated.
----------------	---

### *Published Data*

The activity publishes the following activity specific data items.

<b>Active</b>	Indicates whether the offering is active.
<b>Cost Saved</b>	The cost that has been saved by using the offering.
<b>Created</b>	The date and time that the offering was created.
<b>Created By</b>	The name of the user that created the offering.
<b>Deleted</b>	Indicates whether the offering has been deleted.
<b>Description</b>	The offering's description
<b>Folder ID</b>	The ID of the folder that contains the offering.
<b>Mobile Active</b>	Indicates whether the folder is available for mobile devices.
<b>Name</b>	The offering's name.
<b>Offering Count</b>	The number of offerings that were retrieved.
<b>Offering ID</b>	The offering's unique ID.
<b>Requires Approval</b>	Indicates whether the offering requires approval.
<b>Requires External ID</b>	Indicates whether the offering requires an External Ref. ID.
<b>Service ID</b>	The ID of the service that contains the offering.
<b>Service Name</b>	The name of the service that contains the offering.
<b>Tag</b>	The tag that has been assigned to the offering.
<b>Time Saved</b>	The time, in minutes, that has been saved by using the offering.
<b>Updated</b>	The date and time that the offering was last updated.

## Get Request

The **Get Request** activity is used to retrieve requests from the Automation Portal. You can retrieve a specific request by its unique ID, retrieve all requests or retrieve a subset of requests that match filter criteria.

### Required Properties

You must configure the following property.

<b>Search By</b>	Identifies whether to retrieve a request by its <b>Request ID</b> or to retrieve a collection of requests matching one or more <b>Filters</b> .
------------------	---

### Optional Properties

This activity provides the following optional properties when **Search By** is set to **Filters**.

<b>Descending Order</b>	Indicates that request records should be sorted in descending order according to the <b>Order By</b> value. The default is ascending order.
<b>Limit</b>	The maximum number of request records to retrieve. When zero is specified, all requests are retrieved. The default value is <b>1000</b> .
<b>Order By</b>	Specifies the property used to order the results. The default value is <b>Request ID</b> .

### Filters

When **Search By** is set to **Filters**, this activity provides filters that can be used to control which request records are retrieved from the Automation Portal. The following filters are available.

<b>Can Approve</b>	Filter on whether you can approve or reject the request
<b>Can Set Reject Reason</b>	Filter on whether you can set a reason for rejecting the request
<b>Cost Saved</b>	Filter on the cost that the request has saved
<b>Created</b>	Filter on the date and time that the request was created.
<b>External ID</b>	Filter on the External Ref. ID that was assigned to the request.
<b>Message</b>	Filter on the update message assigned to the request.
<b>Offering ID</b>	Filter on the ID of the offering that the request is for.
<b>Offering Name</b>	Filter on the name of the offering that the request is for.
<b>Request ID</b>	Filter on the request's unique ID
<b>Request State</b>	Filter on the request state (e.g., New, Pending Approval, Closed, etc.)
<b>Requested By</b>	Filter on the name of the user that made the request.
<b>Requires Approval</b>	Filter on whether the request requires approval

<b>Runbook Owner</b>	Filter on the name of the user that owns the runbook responding to the request.
<b>Service ID</b>	Filter on the ID of the service that contains the offering that the request is for.
<b>Service Name</b>	Filter on the name of the service that contains the offering that the request is for.
<b>State Reason</b>	Filter the reason that has been given for the state the request is in.
<b>Tag</b>	Filter on the tag that has been assigned to the request.
<b>Time Saved (Minutes)</b>	Filter on the time that the request has saved
<b>Updated</b>	Filter on the date and time that the request was last updated.

### *Published Data*

The activity publishes the following activity specific data items.

<b>Can Approve</b>	Indicates whether the request can be approved.
<b>Can Set Reject Reason</b>	Indicates whether you can set a reject reason
<b>Cost Saved</b>	The amount saved by the request
<b>Created</b>	The date and time that the request was created
<b>Data</b>	The values that were assigned to the request fields
<b>Deleted</b>	Indicates whether the offering has been deleted.
<b>External ID</b>	The external Ref. ID that was assigned to the request
<b>Message</b>	The update message that was assigned to the request
<b>Offering ID</b>	The ID of the service offering
<b>Offering Name</b>	The name of the service offering
<b>Request Count</b>	The number of requests that were retrieved.
<b>Requested By</b>	The email of the user that made the request
<b>Request ID</b>	The identifier used to uniquely identify the request
<b>Request State</b>	The request state
<b>Requires Approval</b>	Indicates whether the requires approval
<b>Runbook Owner</b>	The name of the runbook owner
<b>Service ID</b>	The ID of the service that contains the offering
<b>Service Name</b>	The name of the service that contains the offering
<b>State Reason</b>	The reason for the current request state
<b>Tag</b>	The tag assigned to the request offering
<b>Time Saved (Minutes)</b>	The time saved by the request
<b>Updated</b>	The date and time that the request was last updated

## Get Request Attachment

The **Get Request Attachment** activity is used in a runbook to retrieve attachment records from the Automation Portal. You can retrieve a specific attachment by its unique ID, retrieve all attachment records for a request or retrieve a subset of attachment records that match one or more filter criterion.

### Required Properties

You must configure the following properties.

<b>Request ID</b>	Specifies the ID of the request from which to retrieve attachments.
-------------------	---

### Optional Properties

This activity provides the following optional properties.

<b>Descending Order</b>	Indicates that attachments records should be published in descending order according to the <b>Order By</b> property. The default is ascending order.
<b>Limit</b>	The maximum number of attachments records to retrieve. When zero is specified, all records are retrieved. The default value is <b>1000</b> .
<b>Order By</b>	Specifies the property used to order the results. The default value is <b>Attachment ID</b> .

### Filters

When **Search By** is set to **Filters**, this activity provides filters that can be used to control which attachment records to retrieve from the Automation Portal. The following filters are available.

<b>Attachment ID</b>	Specifies the ID of the attachment to download from.
<b>Content Type</b>	Filter on the MIME content type.
<b>Created</b>	Filter on the date and time that the attachment was created.
<b>Field ID</b>	Filter on the attachment field that the attachment is for.
<b>Name</b>	Filter on the name assigned to the attachment.
<b>Updated</b>	Filter on the date and time that the request was last updated.

### Published Data

The activity publishes the following activity specific data items.

<b>Attachment Count</b>	The number of attachment records that were retrieved.
<b>Attachment ID</b>	The ID of the attachment record.
<b>Content Type</b>	The MIME content type assigned to the attachment.
<b>Created</b>	The date and time that the attachment was created.
<b>Deleted</b>	Indicates whether the attachment has been deleted.



<b>Field ID</b>	The ID of the field that the attachment is for.
<b>Name</b>	The name given to the attachment.
<b>Request ID</b>	The ID of the request that the attachment is for.
<b>Updated</b>	The date and time that the attachment was last updated.

## Get Request Data

The **Get Request Data** activity is used in a runbook to retrieve request data from the Automation Portal. This activity uses discovery to automatically generate appropriate published data items for the offering that was selected, making it much easier to use request data in child activities.

### *Required Properties*

You must configure the following properties.

<b>Request ID</b>	Specifies the ID of the request that you want to retrieve data for.
<b>Service</b>	Specifies the service that contains the offering that you want to use.
<b>Folder</b>	Specifies the service folder that contains the offering that you want to use. If the offering that you want to use is not in a folder, then select <b>(None)</b> .
<b>Offering</b>	Specifies the offering that you want to use.
<b>Include Secure Data</b>	Indicates whether you want to retrieve data from Secure Text Box fields.

### *Published Data*

This activity generates activity specific data items that correspond to the fields in the offering that was selected, and the published values are determined by the underlying field type.

<b>Date</b>	A date time string in the format yyyy-MM-ddTHH:mm:ss in local time
<b>DateTime</b>	A date time string in the format yyyy-MM-ddTHH:mm:ss
<b>File Attachment</b>	A JSON formatted string describing the file attachment record. See remarks section for the New Request activity for details.
<b>Hidden</b>	String
<b>ListMultipleSelection</b>	Comma separated values. Values containing command and/or double quotes are enclosed in double quotes. Double quotes within a value are escaped by preceding it with another double quote.
<b>ListSingleSelection</b>	String
<b>Radio Button</b>	String
<b>SecureTextBox</b>	String
<b>Switch</b>	<b>True or False.</b>
<b>TableMultipleSelection</b>	A JSON formatted string describing the row(s) that were selected. See remarks section for the New Request activity for details.
<b>TableSingleSelection</b>	A JSON formatted string describing the row that was selected. See remarks section for the New Request activity for details.
<b>TextArea</b>	String
<b>TextBox</b>	String

Time	A string with the format HH:mm:ss
------	-----------------------------------

## Get Request History

The **Get Request History** activity is used in a runbook to retrieve request history records for an Automation Portal request. You can use filters to control which history records are retrieved.

### Required Properties

You must configure the following property.

<b>Request ID</b>	Specifies the ID of the request that you want to retrieve history records for.
-------------------	--

### Optional Properties

This activity provides the following optional properties.

<b>Descending Order</b>	Indicates that history records should be sorted in descending order. The default is ascending order.
<b>Limit</b>	The maximum number of history records to retrieve. When zero is specified, all records are retrieved. The default value is <b>1000</b> .
<b>Order By</b>	Specifies the property used to order the results. The default value is <b>History ID</b> .

### Filters

This activity provides filters that can be used to control which history records to retrieve from the Automation Portal. The following filters are available.

<b>Action</b>	Filter on the action assigned to the history record.
<b>Created</b>	Filter on the date and time that the history record was created.
<b>Created By</b>	Filter on the user the created the history record.
<b>History ID</b>	The ID of the history record.

### Published Data

The activity publishes the following activity specific data items.

<b>Action</b>	The action assigned to the history record.
<b>Action Type</b>	An integer indicating the type of action that was performed
<b>Created</b>	The date and time that the history record was created.
<b>Created By</b>	The user that created the history record.
<b>Deleted</b>	Indicates whether the request has been deleted.
<b>History Count</b>	The number of history records.
<b>History ID</b>	The ID of the history record.
<b>Request ID</b>	The ID of the request that the history record is for
<b>Request State</b>	The state of the request.

## Get Service

The **Get Service** activity is used in a runbook to retrieve service records from the Automation Portal. You can retrieve a specific service by its unique ID, all services or a subset of services using filters.

### Required Properties

You must configure the following property.

<b>Search By</b>	Indicates whether you want to retrieve a specific service by its unique <b>Service ID</b> or retrieve a collection of services matching one or more <b>Filters</b> .
------------------	--

### Optional Properties

This activity provides the following optional properties when **Search By** is set to **Filters**.

<b>Descending Order</b>	Indicates that service records should be published in descending order according to the <b>OrderBy</b> value. The default is ascending order.
<b>Limit</b>	The maximum number of service records to retrieve. When zero is specified, all requests are retrieved. The default value is <b>1000</b> .
<b>Order By</b>	Specifies the property used to order the results. The default value is <b>Service ID</b> .

### Filters

When **Search By** is set to **Filters**, this activity provides filters that can be used to control which service records to retrieve from the Automation Portal. The following filters are available.

<b>Active</b>	Filter on whether the service is active.
<b>Created</b>	Filter on the date and time that the service was created.
<b>Description</b>	Filter on the description given to the service.
<b>Name</b>	Filter on the name given to the service.
<b>Service ID</b>	Filter on the ID of the service.
<b>Updated</b>	Filter on the date and time that the service was last updated.

### Published Data

The activity publishes the following activity specific data items.

<b>Active</b>	Indicates whether the service is active.
<b>Created</b>	The date and time that the service was created.
<b>Created By</b>	The user that created the service.
<b>Deleted</b>	Indicates whether the service has been deleted.

<b>Description</b>	The description given to the service.
<b>Name</b>	The name given to the service.
<b>Service ID</b>	The unique ID used to identify the service.
<b>Updated</b>	The date and time that the service was last updated.

## Monitor Request

The **Monitor Request** activity is to monitor the Automation Portal for new and/or updated requests.

### *Required Properties*

You must configure the following properties.

<b>Monitor New Requests</b>	Indicates whether to monitor the Automation Portal for new requests. The default is True.
<b>Monitor Updated Requests</b>	Indicates whether to monitor the Automation Portal for updated requests. The default is True.
<b>Monitor Interval (seconds)</b>	Specifies the polling interval in seconds.

### *Filters*

This activity provides filters that can be used to control which request records will trigger the monitor. The following filters are available.

<b>Created</b>	Filter on the date and time that the request was created.
<b>External ID</b>	Filter on the External Ref. ID that was assigned to the request.
<b>Message</b>	Filter on the update message assigned to the request.
<b>Offering ID</b>	Filter on the ID of the offering that the request is for.
<b>Offering Name</b>	Filter on the name of the offering that the request is for.
<b>Requested By</b>	Filter on the name of the user that made the request.
<b>Runbook Owner</b>	Filter on the name of the user that owns the runbook responding to the request.
<b>Service ID</b>	Filter on the ID of the service that contains the offering that the request is for.
<b>Service Name</b>	Filter on the name of the service that contains the offering that the request is for.
<b>State</b>	Filter on the state that the request is in (e.g., New, Pending Approval, Closed)
<b>State Reason</b>	Filter the reason that has been given for the state the request is in.
<b>Tag</b>	Filter on the tag that has been assigned to the request.
<b>Updated</b>	Filter on the date and time that the request was last updated.

### *Published Data*

The activity publishes the following activity specific data items.

<b>Can Approve</b>	Indicates whether the request can be approved.
<b>Can Set Reject Reason</b>	Indicates whether you can set a reject reason
<b>Cost Saved</b>	The amount saved by the request
<b>Created</b>	The date and time that the request was created

<b>Data</b>	The values that were assigned to the request fields
<b>External ID</b>	The external Ref. ID that was assigned to the request
<b>Message</b>	The update message that was assigned to the request
<b>Offering ID</b>	The ID of the service offering
<b>Offering Name</b>	The name of the service offering
<b>Request Count</b>	The number of requests that were retrieved.
<b>Requested By</b>	The email of the user that made the request
<b>Request ID</b>	The identifier used to uniquely identify the request
<b>Requires Approval</b>	Indicates whether the requires approval
<b>Runbook Owner</b>	The name of the runbook owner
<b>Service ID</b>	The ID of the service that contains the offering
<b>Service Name</b>	The name of the service that contains the offering
<b>State</b>	The request state
<b>State Reason</b>	The reason for the current request state
<b>Tag</b>	The tag assigned to the request offering
<b>Time Saved</b>	The time saved by the request
<b>Updated</b>	The date and time that the request was last updated



## Reject Request

The **Reject Request** activity is used in a runbook to reject a pending Automation Portal request.

### *Required Properties*

You must configure the following property.

<b>Request ID</b>	Specifies the ID of the request to reject
-------------------	---

### *Optional Properties*

This activity provides the following optional properties.

<b>Reason</b>	Specifies the reason for rejecting the request.
---------------	---

### *Published Data*

The activity publishes the following activity specific data items.

<b>Can Approve</b>	Indicates whether the request can be approved.
<b>Can Set Reject Reason</b>	Indicates whether you can set a reject reason
<b>Cost Saved</b>	The amount saved by the request
<b>Created</b>	The date and time that the request was created
<b>Deleted</b>	Indicates whether the request has been deleted.
<b>External ID</b>	The external Ref. ID that was assigned to the request
<b>Message</b>	The update message that was assigned to the request
<b>Offering ID</b>	The ID of the service offering
<b>Offering Name</b>	The name of the service offering
<b>Request ID</b>	The identifier used to uniquely identify the request
<b>Request State</b>	The current state of the request.
<b>Requested By</b>	The email of the user that made the request
<b>Requires Approval</b>	Indicates whether the requires approval
<b>Runbook Owner</b>	The name of the runbook owner
<b>Service ID</b>	The ID of the service that contains the offering
<b>Service Name</b>	The name of the service that contains the offering
<b>State Reason</b>	The reason for the current request state
<b>Tag</b>	The tag assigned to the request offering
<b>Time Saved (Minutes)</b>	The time saved by the request
<b>Updated</b>	The date and time that the request was last updated

## Update Request

The **Update Request** activity is used in a runbook to update an Automation Portal request.

### *Required Properties*

You must configure the following property.

<b>Request ID</b>	Specifies the ID of the request to be updated.
-------------------	--

### *Optional Properties*

This activity provides the following optional properties.

<b>Message</b>	Specifies a message to assign to the request.
<b>Request State</b>	Specifies the new state to assign to the request.
<b>Runbook Owner</b>	Specifies the owner of the runbook that is handling the request.
<b>State Reason</b>	Specifies the reason for assigning the state. This parameter is only if the required if you want to change the state to <b>Rejected</b> and the offering requires a reason for rejecting a request.

### *Published Data*

The activity publishes the following activity specific data items.

<b>Can Approve</b>	Indicates whether the request can be approved.
<b>Can Set Reject Reason</b>	Identifies whether you can set a reject reason
<b>Created</b>	The date and time that the request was created
<b>Deleted</b>	Indicates whether the request has been deleted.
<b>External ID</b>	The external Ref. ID that was assigned to the request
<b>Message</b>	The update message that was assigned to the request
<b>Offering ID</b>	The ID of the service offering
<b>Offering Name</b>	The name of the service offering
<b>Request ID</b>	The identifier used to uniquely identify the request
<b>Request State</b>	The request state
<b>Requested By</b>	The email of the user that made the request
<b>Requires Approval</b>	Indicates whether the requires approval
<b>Runbook Owner</b>	The name of the runbook owner
<b>Service ID</b>	The ID of the service that contains the offering
<b>Service Name</b>	The name of the service that contains the offering
<b>State Reason</b>	The reason for the current request state

<b>Tag</b>	The tag assigned to the request offering
<b>Time Saved (Minutes)</b>	The time saved by the request
<b>Updated</b>	The date and time that the request was last updated