



# INTEGRATION PACK FOR AUTOTASK

*For Microsoft System Center Orchestrator*

For System Center 2019, you must use the 32-bit version of the integration pack, which has the name **Kelverion\_Integration\_Pack\_for\_Autotask\_1.0**

For System Center 2022 and later, you must use the 64-bit version of the integration pack, which has the name **Kelverion\_IP\_Autotask\_x64\_1.0**

## User Guide

Version 1.0

# Kelverion Integration Pack Autotask

Copyright 2025 Kelverion Inc. All rights reserved.

Published: September 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
Connecting to Autotask .....	6
Prerequisite Configuration .....	6
Working with Configuration Files .....	7
Deploying Your Changes .....	7
Configuration File Schema .....	8
API Concurrency Limitations .....	10
Autotask Activities .....	11
Common Configuration Instructions for All Activities.....	11
Activity Properties.....	11
General Tab.....	11
Properties Tab.....	11
Filters Tab.....	12
Run Behavior Tab.....	12
Published Data .....	13
Create Resource Activity.....	14
Get Resource Activity.....	15
Update Resource Activity.....	16

# Installation and Configuration

---

The Integration Pack for Autotask is an add-on for System Center Orchestrator that enables you to integrate with Autotask and automate service management processes.

## System Requirements

The Integration Pack for Autotask has the following requirements.

### *Kelverion\_Integration\_Pack\_for\_Autotask (32-bit)*

- Autotask 2025.4.5 and later
- Microsoft System Center Orchestrator 2019
- Microsoft .NET Framework 4.7.2

### *Kelverion\_IP\_Autotask (64-bit)*

- Autotask 2025.4.5 and later
- Microsoft System Center Orchestrator 2022, 2025
- Microsoft .NET Framework 4.7.2

## 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](https://technet.microsoft.com/en-us/library/hh420346.aspx) (<https://technet.microsoft.com/en-us/library/hh420346.aspx>).

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

- For System Center 2019, you must use the 32-bit version of the integration pack, which has the name **Kelverion\_Integration\_Pack\_for\_Autotask**
- For System Center 2022 and later, you must use the 64-bit version of the integration pack, which has the name **Kelverion\_IP\_Autotask\_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. Confirm that the file is not set to **Read Only**.
2. Start the **Deployment Manager**.
3. 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.
4. Click **Next**.
5. In the **Select Integration Packs or Hotfixes** dialog box, click **Add**.
6. Locate the **.OIP** file that you copied locally from step 1, click **Open** and then click **Next**.
7. In the **Completing the Integration Pack Wizard** dialog box, click **Finish**.

8. On the **End User Agreement** dialog box, read the Keverion License Terms, and then click **Accept**.
9. 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:*

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.

## Connecting to Autotask

The Keverion Integration Pack for Autotask requires an Autotask API user to authenticate with the Autotask Web Service. Following best practices, it is advisable to create a new API User specifically for use with the Integration Pack.

**Note:** The Autotask API limits incoming REST requests to **three concurrent requests to the same entity endpoint, per API user**. For more details, please refer to the [API Concurrency Limitations](#) section.

### *To create a new API user:*

1. Login into Autotask,
2. Open the Autotask Settings by clicking the **gear** icon in the top right corner.
3. Click the **Admin** menu.
4. Click the **Organization Settings & Users** tab.
5. Expand **Resources/User (HR)**
6. Click **Resources/Users**.
7. Click the **down arrow** on the **New** button.
8. Click **New API User**.
9. Fill General fields according to your requirements.
10. Fill Credential fields according to your requirements.
11. Select **Custom (Internal Integration)** option for **API Tracking Identifier**.
12. In the **Internal Integration Name** box, specify a name to identify the integration that will use this user.
13. Copy the **Tracking Identifier**. This will be used for the **API Integration Code** when configuring a connection to Autotask.
14. Click **Save & Close**.

## Prerequisite Configuration

Prerequisite Configurations establish reusable links between Orchestrator and Autotask. You can create as many configurations as you require. You can also create multiple configurations to allow for different users and/or roles.

### *To set up a prerequisite configuration in Runbook Designer*

1. On the **Options** menu, click **KA Autotask**.
2. In the **KA Autotask** dialog box, click the **Configuration** tab and then click **Add**.
3. On the **Configuration** tab, click **Add**.
4. In the **Add Configuration** dialog box, click the browser next to the **Type** box. Select **Autotask Configuration** and then click **OK**.
5. In the **Configuration File Path** box, type the file path to the desired configuration file. A default configuration file is provided in the Integration Pack installation folder. For details, please refer to the [Working with Configuration Files](#) section.

6. In the **Server URL** box, type the URL of the Autotask web service URL. For example, <https://webservices3.autotask.net>
7. In the **Username** box, type the API username.
8. In the **Secret** box, type the API user secret.
9. In the **API Integration Code**, type the API integration code. For details on how to obtain the API integration code, please refer to [Connecting to Autotask](#) section.
10. Optional. In the **Retry Count** box, you can specify the number of times the integration pack retries an API request, after the request failed due to API concurrency restrictions. The default Retry Count is 5. For details, please refer to the [API Concurrency Limitations](#) section.
11. Optional. In the **Retry Interval (seconds)** box, you can configure the number of seconds the integration pack waits between API request retries, after the request failed due to API concurrency restrictions. The default value is 30 seconds. Note that the specified value may not necessarily be the actual delay value, which the integration pack calculates to optimize retry throughput and can be any value between the specified Retry Interval and Retry Interval x 2. For details, please refer to the [API Concurrency Limitations](#) section.
12. Optional. In the **Web Proxy URL** box, type the optional web proxy URL.
13. Optional. In the **Web Proxy Username** box, type optional web proxy username.
14. Optional. In the **Web Proxy Password** box, type the web proxy password.
15. Optional. In the **HTTP Request Timeout (seconds)** box, you can specify the number of seconds before the HTTP request times out. The default value is 100 seconds.
16. Click **OK**, and then click **Finish**.

## Working with Configuration Files

Integration with Autotask is supported by a JSON (JavaScript Object Notation) formatted configuration file. The configuration file contains entity definitions for the supported Autotask entities. This information is used to support discovery in System Center Orchestrator and to execute runbooks at runtime. The integration pack is deployed with a default configuration file, supporting some important entity types, for your convenience.

The configuration file can also be used to customize the integration with your Autotask instance. Customizations can include adding new field definitions to existing Autotask entities, such as Ticket or Configuration Item, or adding definitions for new entity that are not supported by default.

## Deploying Your Changes

The **Configuration File Path** property from the integration pack configuration options specifies the location of the JSON configuration file that the integration pack will be using. By default, this specifies the configuration file that is installed with the integration pack. If you wish to make customizations to the configuration file, we recommend making a copy of the default configuration file, saving it in a safe location, and then updating the **Configuration File Path** property accordingly.

**Important:** Always back up your custom configuration file(s). When you update the Integration Pack, the default configuration file will be overwritten. If you are using a custom configuration file, we recommend that you make a copy of your file to a safe location that is unaffected by the update.

## Configuration File Schema

The entity definitions in the configuration file used by the Integration Pack for Autotask are formatted using JSON. The following JSON Schema outlines the JSON used for configuration files.

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "properties": {
    "entities": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "displayName": { "type": "string" },
          "name": { "type": "string" },
          "endpoint": { "type": "string" },
          "supportsCreate": { "type": "boolean" },
          "supportsUpdate": { "type": "boolean" },
          "supportsDelete": { "type": "boolean" },
          "supportsQuery": { "type": "boolean" },
          "supportsMonitor": { "type": "boolean" },
          "fields": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "displayName": { "type": "string" },
                "name": { "type": "string" },
                "isFilter": { "type": "boolean" },
                "isDateOnly": { "type": "boolean" }
              },
              "required": ["displayName", "name", "isFilter"]
            }
          }
        },
        "required": ["displayName", "name", "isFilter"]
      }
    },
    "required": [
      "displayName",
      "name",
      "endpoint",
      "supportsCreate",
      "supportsUpdate",

```



```

        "supportsDelete",
        "supportsQuery",
        "supportsMonitor",
        "fields"
    ]
}
},
"required": ["entities"]
}

```

## Defining Entities

Entity definitions provide the integration pack with the information used to integrate with Autotask.

### DisplayName Property

The **displayName** property specifies the name used to display the entity in System Center Orchestrator.

### Name Property

The **name** property specifies the internal Autotask name of the entity.

### Endpoint Property

The **endpoint** property specifies the name of the entity used by the Autotask REST API. For example, Tickets or Configuration Items.

### SupportsCreate Property

The **supportsCreate** property specifies if the entity should be displayed in the Create Resource Activity.

### SupportsUpdate Property

The **supportsUpdate** property specifies if the entity should be displayed in the Update Resource Activity.

### SupportsQuery Property

The **supportsQuery** property specifies if the entity should be displayed in the Get Resource Activity.

### Fields Property

The **fields** property specifies the fields that are available in activities.

## Defining Entity Fields

Field definitions provide the integration pack with information used to support specific entity fields. This information is used by integration pack activities to present appropriate input properties and filter options for the supported entities.

### Name Property

The **name** property specifies the internal Autotask name of the field.

### DisplayName Property

The **displayName** property specifies the name used to display the field in System Center Orchestrator.

### IsFilter Property

The **isFilter** property determines if the field has a filter in the Get Resource Activity. Note that this only applies to fields which also support filtering in the Autotask API.

### IsDateOnly Property

The **isDateOnly** property specifies if the field is a date field with no time portion.

## API Concurrency Limitations

The Autotask API restricts incoming REST requests to a maximum of **three concurrent requests to the same entity endpoint, per API user**. To mitigate this limitation, the Integration Pack provides a retry mechanism which repeats failed requests when the request fails because of API concurrency restrictions.

You can configure request retries by setting the **Retry Count** and **Retry Interval (seconds)** in the [Integration Pack configuration](#). Retry Count specifies how many times a failed request is retried, and Retry Interval configures the how long the IP waits between retries. The actual delay value is calculated for optimal retry throughput and can be any value between the specified Retry Interval and  $\text{Retry Interval} \times 2$ .

**Note:** We highly recommend keeping concurrency limitations in mind when designing your runbooks. The integration pack retry mechanism can handle a limited number of concurrency failures with the default retry settings. For more aggressive concurrency situations, you may need to increase the **Retry Count** and **Retry Interval** values, with activity processing time also suffering.

# Autotask Activities

---

This integration pack adds the KA Autotask category to the Runbook Designer Activities pane. This category contains the following activities:

- Create Resource
- Update Resource
- Get Resource

## Common Configuration Instructions for All Activities

The following configuration instructions apply to all activities in this integration pack.

### 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 Runbook Designer.

#### *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, **Properties** tab, and a **Run Behavior** tab. Some activities may have additional tabs, such as a **Filters** tab.

### 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 Tab

The Properties 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 which prerequisite configuration to use to connect to Autotask.

#### *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 Autotask Connections](#).

## Filters Tab

The Get Resource activities include a filters tab. You can use the Filters tab to add one or more filters to target a specific subset of resources. Each filter includes **Name**, **Relation**, and **Value** properties, that you must configure.

#### *To add a filter to your runbook activity:*

1. Click the **Filters** tab and then click **Add**.
2. In the **Name** box, select the entity field that the filter will target.
3. In the **Relation** box, select the relation used to evaluate the filter value.
4. In the **Value** box, type the value that you want the filter to evaluate.
5. Click **OK**.

The relations that are available when configuring a filter are dependent on the data type of the entity field that the filter is targeting. One or more of the following will be provided.

- **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.

## 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.

### *Published Data Behavior*

By default, Published Data is passed as multiple individual outputs. You can alternatively specify that all values be flattened into a single comm-delimited value (.csv) file.

When you enable the Flatten feature, you also choose a multi-value 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.

**Note:** The Flatten feature does not flatten data across multiple instances of the same activity. It only flattens multiple values returned from a single instance of the 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 fails.

#### *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).

### *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 to it in the runbook. You can use published data to automatically populate the property values needed by activities.

#### *To subscribe to Published Data of an earlier activity in the workflow:*

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. By default, the dialog box only displays Published Data that is specific to that activity. To include Published Data that is common to all activities, click **Show Common Published Data**.
3. Click the published data item 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 see [Published Data](http://technet.microsoft.com/en-us/library/hh403821.aspx) (http://technet.microsoft.com/en-us/library/hh403821.aspx).

# Create Resource Activity

---

The **Create Resource** activity is used to create and initialize a resource, such as a ticket.

## *Required Properties*

You must select the entity type of the resource that you want to create. After you have selected an entity type, the activity will provide additional properties, that you must configure, to initialize the new resource.

Entity	The type of resource to create.
--------	---------------------------------

## *Optional Properties*

The activity will provide optional properties that correspond to the non-mandatory fields in the type of entity that you selected, and these can be configured as needed to initialize the new resource.

## *Published Data*

The activity publishes the following activity specific data items.

<Entity> ID	The unique ID of the resource that was created.
-------------	---

# Get Resource Activity

---

The **Get Resource** activity is used to retrieve resource details. One or more filters can be configured to retrieve a resource subset that matches certain criteria.

## *Required Properties*

You must configure the following properties.

<b>Entity</b>	The type of resource retrieve.
---------------	--------------------------------

## *Optional Properties*

You can use the following properties, as necessary, to control the behavior of the activity.

<b>Limit</b>	The maximum number of resources to retrieve. The maximum value is five hundred. When limit is not specified, the activity retrieves all the resources matching the specified filters.
--------------	---

## *Filters*

The activity provides filters that correspond to the fields in the entity type that you selected. You can combine one or more filters to selectively control which resources to retrieve.

## *Published Data*

The activity publishes data that represents the resource records that were retrieved. Each record has published data items that correspond to the fields in resource that you selected. The activity also publishes the following activity specific data items.

<b>Resource Count</b>	The total number of resources that were retrieved.
-----------------------	--

# Update Resource Activity

---

The **Update Resource** activity is used to update one or more fields in an existing resource.

## *Required Properties*

You must configure the following properties.

<b>Entity</b>	The type of resource to update.
<b>&lt;Entity&gt; ID</b>	The unique ID of the resource to update.

## *Optional Properties*

The activity will provide parameters that correspond to the fields of the entity type that you selected, and these can be used, as necessary, to update the specified resource.

## *Published Data*

The activity publishes the following activity specific data items.

<b>&lt;Resource&gt; ID</b>	The unique ID of the resource that was updated.
----------------------------	---