



INTEGRATION PACK FOR RUNBOOK MANAGEMENT

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_Runbook_Management_4.0**

For System Center 2022 and later, you must use the 64-bit version of the integration pack, which has the name **Keverion_IP_Runbook_Management_x64_4.0**

User Guide

Version 4.0

Kelverion Integration Pack for Runbook Management

Copyright 2013 Kelverion Inc. All rights reserved.

Published: August 2024

Feedback

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

Contents

Introduction	4
Managing Orchestrator	4
System Requirements	5
Registering and Deploying the Integration Pack	5
Licensing the Integration Pack	6
Configuring the Runbook Management Options	7
KA Runbook Management Activities	9
Common Configuration Instructions for All Activities.....	9
Activity Properties	9
General Tab.....	9
Properties Tab.....	9
Run Behavior Tab	11
Published Data	11
Get Events Activity.....	13
Get Runbooks Activity	15
Get Runbook Jobs Activity.....	17
Monitor Events Activity	18
Start Runbook Activity.....	19
Stop Runbook Job Activity.....	21

Introduction

The Kolverion Integration Pack for Runbook Management is an add-on for System Center Orchestrator that enables you to automate operations in Orchestrator such as starting or stopping a Runbook.

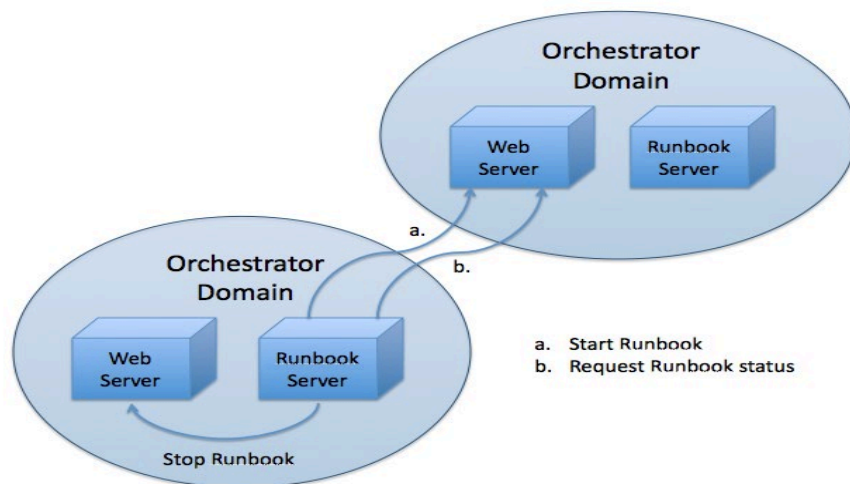
Managing Orchestrator

The Integration Pack for Runbook Management offers an uncomplicated way to make use of the Web Services without the knowledge of a programming language, or indeed intimate details of the Web Service itself. It allows administrators to perform functions which otherwise cannot be executed without access to and manipulation of the Orchestrator databases. It replaces use of strongly discouraged hacks by best practice methods.

Like the Orchestration Console, the web service is organized into Runbooks, Jobs, and Instances entities. Runbooks, as you would expect, represent the runbooks in your Orchestrator environment. The differences between Jobs and Instances are not so clear, however you can think of Jobs as runbooks that have been run or are scheduled to run, whereas Instances are runbooks that have been run to completion.

The Integration Pack for Runbook Management focuses on the management of Runbooks and Jobs.

A System Center Orchestrator installation is an isolated instance. Management access from the outside, from a different Orchestrator domain, for example, is only possible using the Web Service. As RMA simplifies access to the Web Service, even allowing access to several of them in parallel, it is now possible to build whole hierarchies of Orchestrator domains. This offers the separation, which might be required by the environment and solutions Orchestrator is dealing with, while keeping the management control needed to run Orchestrator securely and efficiently.



System Requirements

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

Kelverion_Integration_Pack_for_Runbook_Management (32-bit)

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

Kelverion_IP_Runbook_Management_x64 (64-bit)

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

Both versions support integrating with an environment with at least one of these:

- System Center Orchestrator Web Service (2016, 2019)
- System Center Orchestrator Web API Service (2022)

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 2016 and 2019, you must use the 32-bit version of the integration pack, which has the name **Kelverion_Integration_Pack_for_Runbook_Management**
- For System Center 2022 and later, you must use the 64-bit version of the integration pack, which has the name **Kelverion_IP_Runbook_Management_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 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 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 Runbook Management Options

A configuration establishes a reusable link between Orchestrator and the Orchestrator Web Service. You can create multiple configurations to link to multiple servers or to allow for differences in security permissions for different user accounts.

To setup a Runbook Management configuration:

1. In the Runbook Designer, click the **Options** menu, and select *KA Runbook Management*. The KA Runbook Management dialog box appears.
2. On the **Configurations** tab, click **Add** to begin the connection setup. The **Add Configuration** dialog box appears.
3. In the **Name** box, enter a name for the connection. This could be the name of the Orchestrator server or a descriptive name to distinguish the type of configuration.
4. Click the ellipsis (...) button next to the **Type** field and select *Runbook Management*.
5. In the **Web Service URL** box:
 - a. If targeting the legacy Web Service (Orchestrator 2016 or 2019) type the URL, including port number. For example,
<https://example.com:81/Orchestrator2012/Orchestrator.svc>.
Note that the legacy web service is supported by System Center Orchestrator 2016 or 2019 and can also be installed on System Center Orchestrator 2022.

OR

 - b. If targeting the new Web API Service (Orchestrator 2022), type the URL, including port number. For example, <https://example.com:81/api>.
6. In the **Username**, **Domain**, and **Password** boxes, type the credentials that will be used to connect to Orchestrator.
7. In the **Skip Certificate Validation** box, specify if you want the IP to perform server SSL certificate validation or not. This applies only when connecting to the server over HTTPS. The default value for this property is **False** and IP activities may fail, if you are using a self-signed certificate, or if the specified server name is not listed on the certificate. Please, ensure that the server is configured with a valid certificate signed by a valid certificate authority and that the specified server name is listed on the certificate. Alternately, for a secure network environment when working with at trusted server, you may choose not to validate the server certificate and use a value of **True**.
8. In the **Use Legacy Web Service** box, specify if the integration pack will be using the Orchestrator legacy Web Service (Orchestrator 2016 or 2019) **True** or the new Web API Service (Orchestrator 2022) **False**. This value should be set according to the **Web Service URL** setting.

Note: When targeting the new Web API Service in System Center 2022 and later you may encounter an error when requesting Runbooks. To work around this issue, run the following SQL against the Orchestrator Database:

```
GRANT CONTROL ON ASYMMETRIC KEY::[ORCHESTRATOR_ASYM_KEY] TO  
[Microsoft.SystemCenter.Orchestrator.Admins]
```

```
GRANT CONTROL ON SYMMETRIC KEY::[ORCHESTRATOR_SYM_KEY] TO  
[Microsoft.SystemCenter.Orchestrator.Admins]
```

```
GRANT EXECUTE ON  
object::[Microsoft.SystemCenter.Orchestrator].[GetSecurityToken] TO  
[Microsoft.SystemCenter.Orchestrator.Operators]
```

```
GRANT SELECT ON  
object::[Microsoft.SystemCenter.Orchestrator.Internal].[Settings] TO  
[Microsoft.SystemCenter.Orchestrator.Operators]
```

```
GRANT SELECT ON  
object::[Microsoft.SystemCenter.Orchestrator.Internal].[AuthorizationCa  
che] TO [Microsoft.SystemCenter.Orchestrator.Admins]
```


KA Runbook Management Activities

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

- Get Runbooks
- Get Runbook Jobs
- Start Runbook
- Stop Runbook Job
- Get Events
- Monitor Events

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 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, a **Details** 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 Tab

This tab contains 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 System Center Orchestrator.

To configure the Configuration Name property:

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

Filters Tab

The **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 property of the instance exactly matches the text or number specified in the filter.
- **Does not equal:** the property of the instance does not exactly match the text or number specified in the filter.
- **Is less than:** the property of the instance is less than the number specified in the filter.
- **Is less than or equal to:** the property of the instance is less than or equal to the number specified in the filter.
- **Is greater than:** the property of the instance is greater than the number specified in the filter.
- **Is greater than or equal to:** the property of the instance is greater than or equal to the number specified in the filter.
- **Contains:** the property of the instance 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 property of the instance does not contain the exact text specified in the filter. Unlike the Equals behavior, there can be other text surrounding the matching text.
- **Matches pattern:** use regular expressions to specify a pattern that the text must match.
Note: **Matches pattern** filter operation is always performed client-side.
- **Does not match pattern:** use regular expressions to specify a pattern that the text must not match.
Note: **Does not match pattern** filter operation is always performed client-side.
- **Starts with:** the property of the instance starts with the exact text specified in the filter.
- **Ends with:** the property of the instance ends with the exact text specified in the filter.

Server-Side vs Client-Side Filters

When a **Get** activity performs client-side filtering, it first retrieves the data from Orchestrator, and then filters records in the activity. When a **Get** activity performs server-side filtering, the data is filtered on the server, thus reducing the amount of data returned from Orchestrator to the activity.

When using the new Web API Service (Orchestrator 2022), most of the filtering is performed server-side. When using the legacy Web Service (Orchestrator 2016 and 2019), some filtering is performed server-side and some client-side. Please refer to specific activity reference for details on which filtering operations are performed server-side vs client-side.

IMPORTANT: It is highly recommended that you always configure activity filters to reduce the amount of data returned from Orchestrator and to reduce the amount of data published on the Orchestrator data bus.

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 of time.

Multi-Value Published Data Behavior

The Get Runbooks and Get Runbook Jobs 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 Runbooks activity, the data output from that activity contains data such as the Runbook ID, Path, and Last Modified Time.

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 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 policies.

An activity can only subscribe 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>)

Get Events Activity

The **Get Events** activity is used in a runbook to retrieve information about the Orchestrator platform events in your environment using filter criteria that you specify.

Filters

You can use the following filters to control which events to retrieve from Orchestrator.

Name	Description	Legacy Web Filtering	Web API Filtering
Created	The date and time the event was created	Client-side	Server-side
Details	The event details	Client-side	Server-side**
Event ID	The unique ID of event	Sever-side*	Server-side
Source ID	The unique ID of source	Client-side	Server-side
Source Name	The name of the source	Client-side	Server-side**
Summary	A summary of the event	Sever-side*	Server-side**
Event Type	The type of event	Sever-side*	Server-side**

* Server-side filtering **only** for operators **Equals** and **Does Not Equal**. Client-side filtering for other operators.

****Matches pattern** and **Does not match pattern** filter operations are always performed client-side.

Published Data

The activity publishes the following data to the Orchestrator data bus.

Created	The date and time that the event was created
Details	The event details.
Event ID	The unique ID of event
Source ID	The unique ID of source
Source Name	The name of the source
Summary	A summary of the event
Event Type	The type of event

Event Count	The number of events returned
Domain	The domain name
Username	The username used to login to the Web Service
Web Service URL	The URL used to integrate with Orchestrator Web Services

Get Runbooks Activity

The **Get Runbooks** activity is used in a runbook to retrieve information about the runbooks in your Orchestrator environment using filter criteria that you specify.

Optional Properties

You can use the following optional properties to control the behavior of the activity.

Path	The path of a runbook to be retrieved, e.g., <code>\MyRunbooks\VM\StartVMs</code> . Note: Filtering operation is performed server-side.
Folder Path	Specifies the path of a runbook folder. The activity will return all the runbooks in that folder, e.g., <code>\MyRunbooks\VM</code> . Note: Filtering operation is performed server-side.

Filters

You can use the following filters to control which runbooks to retrieve from Orchestrator.

Name	Description	Legacy Web Filtering	Web API Filtering
Created	The date and time that the runbook was created	Client-side	Server-side
Folder ID	The Unique ID of the folder that contains the runbook	Server-side*	Server-side
Folder Path	The path to the folder containing the runbook	Server-side*	Server-side*
Is Monitor	Indication of whether the runbook has a monitor	Client-side	Server-side
Last Modified Time	The date and time that the runbook was last modified	Client-side	Server-side
Path	The full path to the runbook's location	Server-side*	Server-side**
Runbook ID	The unique ID of the Runbook	Server-side*	Server-side
Runbook Name	The name of the runbook	Server-side*	Server-side**

* Server-side filtering **only** for operators **Equals** and **Does Not Equal**. Client-side filtering for other operators.

****Matches pattern** and **Does not match pattern** filter operations are always performed client-side.

Tip: Use the **Folder Path** filter to distinguish multiple runbooks with the same name that exist in different folders.

Published Data

The activity publishes the following data to the Orchestrator data bus.

Created	The date and time that the runbook was created
Folder ID	The Unique ID of the folder in which the runbook is located
Folder Path	The path to the folder containing the runbook.
Is Monitor	Indication whether the runbook has a monitor
Last Modified Time	The date and time that the runbook was last modified
Path	The full path to the runbook's location
Runbook Count	The number of runbooks that were retrieved
Runbook ID	The unique ID of the Runbook
Runbook Name	The runbooks' name
Web Service URL	The URL used to integrate with Orchestrator Web Services

Get Runbook Jobs Activity

The **Get Runbook Jobs** activity is used in a runbook to retrieve job information, such as Job ID and Job Status, using filter criteria that you specify.

Filters

You can use the following filters to control which runbooks jobs to retrieve from Orchestrator.

Name	Description	Legacy Web Filtering	Web API Filtering
Job ID	The unique ID of the runbook job	Server-side*	Server-side
Last Modified Time	The date and time that the job was last modified	Client-side	Server-side
Runbook ID	The unique ID of the Runbook	Server-side*	Server-side
Runbook Server ID	The ID of the Runbook Server that is running the job	Client-side	Server-side
Start Time	The date and time that the job was created	Client-side	Server-side
Status	The status of the job	Server-side*	Server-side**

* Server-side filtering **only** for operators **Equals** and **Does Not Equal**. Client-side filtering for other operators.

****Matches pattern** and **Does not match pattern** filter operations are always performed client-side.

Tip: Use the **Status** filter to publish job information for jobs with a particular state, such as *Running*. This is recommended when combining the Get Runbook Jobs activity with other activities, such as Stop Runbook Job.

Published Data

The activity publishes the following data to the Orchestrator data bus.

Job Count	The number of jobs that were retrieved
Last Modified Time	The date and time that the job was last modified
Runbook ID	The unique ID of the Runbook
Runbook Server ID	The ID of the Runbook Server that is running the job
Start Time	The date and time that the job was created
Status	The status of the job

Web Service URL	The URL used to integrate with Orchestrator Web Services
------------------------	--

Monitor Events Activity

The **Monitor Events** activity is used in a runbook to monitor Orchestrator platform events in your environment using filter criteria that you specify.

Required Properties

You must configure the following properties.

Polling Interval	The time in seconds the monitor will poll for new events
-------------------------	--

Filters

You can use the following filters to control which Orchestrator events will trigger the monitor.

Name	Description	Legacy Web Filtering	Web API Filtering
Created	The date and time the event was created	Client-Side	Server-side
Details	The event details	Client-Side	Server-side*
Event ID	The unique ID of event	Client-Side	Server-side
Source ID	The unique ID of source	Client-Side	Server-side
Source Name	The name of the source	Client-Side	Server-side*
Summary	A summary of the event	Client-Side	Server-side*
Event Type	The type of event	Client-Side	Server-side*

* **Matches pattern** and **Does not match pattern** filter operations are always performed client-side.

Published Data

The activity publishes the following data to the Orchestrator data bus.

Created	The date and time that the event was created
Details	The event details
Event ID	The unique ID of event
Source ID	The unique ID of source
Source Name	The name of the source

Summary	A summary of the event
Event Type	The type of event
Event Count	The number of events that triggered the monitor
Domain	The domain name
Username	The username used to login to the Web Service
Web Service URL	The URL used to integrate with Orchestrator Web Services

Start Runbook Activity

The **Start Runbook** activity is used in a runbook to start a Runbook job.

Tip: When using the Web API Service (Orchestrator 2022), to start a runbook, the runbook must have unique input and output parameter names.

Required Properties

You must configure the following properties.

Parameter Count	The number of parameters that are required to start the runbook
Parameter Name N	The name of the nth parameter, generated dynamically based on Parameter Count
Parameter Value N	The value of the nth parameter, generated dynamically based on Parameter Count
Runbook ID	The ID of the Runbook to start
Wait for Completion	Indicates whether the activity waits for the runbook job to be completed

Optional Properties

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

Runbook Server	<p>The runbook server where the runbook should run.</p> <p>Note: that the server name(s) are case-sensitive and should appear exactly as displayed in the Runbook Designer or Orchestration Console. Also, when specifying multiple runbook servers, they must be separated with semi-colons (;).</p>
-----------------------	--

Published Data

The activity publishes the following data to the Orchestrator data bus.

Job ID	The ID of the Job that was created.
---------------	-------------------------------------

Return Data	The data returned from the runbook. The runbook must be configured to return data outputs and it must end with a Return Data activity. Only available when Wait for Completion is set to <i>True</i> .
Runbook ID	The ID of the Runbook to start
Runbook Server	The name of the runbook server where the runbook ran. Only available when Wait for Completion is set to <i>True</i> . <i>Important:</i> Do not confuse this property with Runbook Server Name, which is a property published by Orchestrator and represents the runbook server where this activity ran.
Runbook Server ID	The ID of the runbook server where the runbook ran. Only available when Wait for Completion is set to <i>True</i> .
Web Service URL	The URL used to integrate with Orchestrator Web Services
Status	The status of the runbook job. Only available when Wait for Completion is set to <i>True</i> .

Important: If the runbook publishes multiple outputs from multiple **Return Data** activity instances, the output values in **Return Data** are not correlated.

Stop Runbook Job Activity

The **Stop Runbook Job** activity is used in a runbook to stop a specified runbook job.

Required Properties

You must configure the following properties.

Job ID	The ID of the runbook job to stop
---------------	-----------------------------------

Published Data

The activity publishes the following data to the Orchestrator data bus.

Job ID	The ID of the Job to stop
Web Service URL	The URL used to integrate with Orchestrator Web Services

Important: Ensure that the **Job ID** for the **Stop Runbook Job** activity is associated with a runbook job that is *Pending* or *Running*.

In most scenarios, we expect you will use the **Get Runbook Jobs** activity to retrieve the **Job ID** of the jobs that you want to stop, and we recommend using **Status** filters to ensure that *Completed*, *Canceled* and *Failed* jobs are excluded. In this scenario, we also recommend using a link condition between the **Get Runbook Jobs** and **Stop Runbook Job** activity to ensure that the **Stop Runbook Job** activity does not run when the **Job Count** published from **Get Runbook Jobs** is zero.