



# INTEGRATION PACK FOR TEXT MANIPULATION

*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\_Text\_Manipulation\_3.0**

For System Center 2022 and later, you must use the 64-bit version of the integration pack, which has the name **Keverion\_IP\_Text\_Manipulation\_x64\_3.0**

## User Guide

Version 3.0

# Kelverion Integration Pack for Text Manipulation

Copyright 2012 Kelverion Inc. All rights reserved.

Published: September 2022

*[Feedback](#)*

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

# Contents

Getting Started.....	5
System Requirements.....	5
Registering and Deploying the Integration Pack.....	5
Licensing the Integration Pack.....	6
Text Manipulation Activities.....	7
Common Configuration Instructions for All Activities.....	7
Activity Properties.....	7
General Tab.....	7
Properties Tab.....	7
Run Behavior Tab.....	8
Published Data.....	8
Compare Text Activity.....	10
Contains Text Activity.....	11
Decode Text Activity.....	12
Encode Text Activity.....	13
Ends With Activity.....	14
Extract Text Activity.....	15
Insert Text Activity.....	16
Join Text Activity.....	17
Lowercase Text Activity.....	18
Pad Text Activity.....	19
Regex Match Activity.....	20
Regex Replace Activity.....	21
Regex Split Activity.....	22
Remove Text Activity.....	23
Replace Text Activity.....	24
Reverse Text Activity.....	25
Split Text Activity.....	26

Starts With Activity .....	27
Text Length Activity.....	28
Trim Text Activity.....	29
Truncate Text Activity .....	30
Uppercase Text Activity.....	31

# Getting Started

---

The Integration Pack for Text Manipulation is an add-on for Microsoft System Center Orchestrator that performs a variety of text manipulation tasks, such as splitting, joining and comparing text.

## System Requirements

The Integration Pack for Text Manipulation 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\_Text\_Manipulation (32-bit)*

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

### *Kelverion\_IP\_Text\_Manipulation\_x64 (64-bit)*

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

## 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.

**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\_Text\_Manipulation**
- For System Center 2022 and later, you must use the 64-bit version of the integration pack, which has the name **Kelverion\_IP\_Text\_Manipulation\_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 Kolverion License Terms, and then click **Accept**.
10. The **Log Entries** pane displays a confirmation message when the integration pack is successfully registered

#### *To deploy the integration pack:*

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

## Licensing the Integration Pack

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

# Text Manipulation Activities

---

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

This category contains the following activities:

- Compare Text
- Contains Text
- Decode Text
- Encode Text
- Ends With
- Extract Text
- Insert Text
- Join Text
- Lowercase Text
- Pad Text
- Regex Match
- Regex Replace
- Regex Split
- Remove Text
- Reverse Text
- Split Text
- Starts With
- Text Length
- Trim Text
- Truncate Text
- Uppercase Text

## 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 activity 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 Tab

These tabs contain properties that are specific to the activity.

## 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 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*

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

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

### *To use published data*

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

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

# Compare Text Activity

---

The **Compare Text** activity is used in a runbook to compare two or more text inputs to determine if they are equivalent.

## *Required Properties*

You must configure the following properties:

<b>Number of inputs</b>	The number of text inputs to compare.
<b>Case sensitive</b>	Specifies whether the comparison is case sensitive.
<b>Input Text 1 ... N</b>	The input text items to be compared.

## *Published Data*

The activity publishes the following activity specific data:

<b>Equivalent</b>	Indicates whether all of the input text items are the same.
-------------------	---

## Contains Text Activity

---

The **Contains Text** activity is used in a runbook to test whether some input text contains some text fragment.

### *Required Properties*

You must configure the following properties:

<b>Input text</b>	The input text to be searched
<b>Case sensitive</b>	Specifies whether the search is case sensitive.
<b>Find this</b>	The text fragment to search for

### *Published Data*

The activity publishes the following activity specific data:

<b>Found</b>	Indicates whether the text fragment was found
--------------	---

# Decode Text Activity

---

The **Decode Text** activity is used in a runbook to convert base-64 encoded text input into binary data, which is then written to a specified destination file.

## *Required Properties*

You must configure the following properties:

<b>Destination file</b>	The file to which the binary data will be written
<b>Encoded text</b>	The base-64 encoded text
<b>Overwrite</b>	Specifies whether the target file should be overwritten if it already exists.

## *Published Data*

The activity publishes the following activity specific data:

<b>Name</b>	Description
<b>Bytes written</b>	The number of bytes that were written to the destination file

# Encode Text Activity

---

The **Encode Text** activity is used in a runbook to read the contents of a file and convert the binary data to a base-64 encoded text.

## *Required Properties*

You must configure the following properties:

---

<b>Source file</b>	The file to read
--------------------	------------------

---

## *Published Data*

The activity publishes the following activity specific data:

---

<b>Encoded text</b>	Base-64 encoded text representation of the source file's contents.
---------------------	--

---

## Ends With Activity

---

The **Ends With** activity is used in a runbook to test whether some input text ends with a specified suffix.

### *Required Properties*

You must configure the following properties:

<b>Input text</b>	The input text to be searched
<b>Case sensitive</b>	Specifies whether the search is case sensitive.
<b>Ends with</b>	The text fragment to search for

### *Published Data*

The activity publishes the following activity specific data:

<b>Found</b>	Indicates whether the suffix was found
--------------	--

# Extract Text Activity

---

The **Extract Text** activity is used in a runbook to extract a text fragment from some input text.

## *Required Properties*

You must configure the following properties:

<b>Input text</b>	The input text to be searched
<b>Number of characters</b>	The number of characters to extract
<b>Extract from</b>	Identifies the relative position to extract characters

## *Optional Properties*

You can use the following properties to change the default behavior:

<b>Number of characters</b>	The number of characters to extract.
<b>Starting at</b>	When extracting from the middle, this is the position from which to start extracting characters. The first character is position 1.

## *Published Data*

The activity publishes the following activity specific data:

<b>Output text</b>	The text fragment that was extracted from the input text.
--------------------	---

# Insert Text Activity

---

The **Insert Text** activity is used in a runbook to insert a text fragment into some input text.

## *Required Properties*

You must configure the following properties:

<b>Input text</b>	The input text to be modified
<b>Insert at</b>	The position to insert the text fragment. The position of the first character is 1.
<b>Insert this</b>	The text fragment to be inserted.

## *Published Data*

The activity publishes the following activity specific data:

<b>Output text</b>	The modified text
--------------------	-------------------

# Join Text Activity

---

The **Join Text** activity is used in a runbook to join two or more text inputs using an optional character or text fragment, such as a comma, as a separator.

## *Required Properties*

You must configure the following properties:

<b>Number of inputs</b>	The number of text inputs to join.
<b>Separator</b>	A character or text fragment that will be used to separate the text inputs. Leave empty if you want to concatenate the inputs.
<b>Input Text 1 ... N</b>	The input text items to be joined.
<b>Skip empty inputs</b>	Indicates whether to include empty inputs in the output text

## *Published Data*

The activity publishes the following activity specific data:

<b>Output text</b>	The joined output text
--------------------	------------------------

# Lowercase Text Activity

---

The **Lowercase Text** activity is used in a runbook to convert the case of some input text to lowercase in whole or in part.

## *Required Properties*

You must configure the following properties:

<b>Input text</b>	The input text to be modified
-------------------	-------------------------------

## **Lower Text Optional Properties**

<b>Number of characters</b>	The number of characters to make lowercase.
<b>Starting at</b>	The position to start making characters lowercase. The first character has position 1.

## *Published Data*

The activity publishes the following activity specific data:

<b>Output text</b>	The modified text
--------------------	-------------------

# Pad Text Activity

---

The **Pad Text** activity is used in a runbook to pad some input text, with spaces or some other character that you specify, so that it has a desired length.

## *Required Properties*

You must configure the following properties:

<b>Input text</b>	The input text to be modified
<b>Position</b>	Identifies where padding characters should be inserted.
<b>Total length</b>	The total length of the text with padding.

## *Optional Properties*

You can use the following properties to change the default behavior:

<b>Pad character</b>	The character used to pad the input text.
----------------------	---

## *Published Data*

The activity publishes the following activity specific data:

<b>Output text</b>	The modified text
--------------------	-------------------

# Regex Match Activity

---

The **Regex Match** activity is used in a runbook to test whether some input text matches a specified regular expression and to retrieve the matches.

## *Required Properties*

You must configure the following properties:

<b>Input text</b>	The input text to be tested
<b>Regular expression</b>	The regular expression used to test the input text.

## *Optional Properties*

You can use the following properties to change the default behavior:

<b>Ignore Case</b>	Specifies whether to ignore case when testing the input text.
<b>Multiline</b>	If true, then the ^ and \$ match the beginning and end, respectively, of any line, and not just the beginning and end of the entire string.
<b>Singleline</b>	If true, then the dot (.) matches every character instead of every character except linefeed (\n).

## *Published Data*

The activity publishes the following activity specific data:

<b>Match count</b>	The number of matches that were found.
<b>Matches</b>	The matches that were found.

# Regex Replace Activity

---

The **Regex Replace** activity is used in a runbook to replace a fragment in some input text that matches a specified regular expression.

## *Required Properties*

You must configure the following properties:

<b>Input text</b>	The input text to be modified.
<b>Regular expression</b>	The regular expression used to find the text fragment to replace.
<b>Replace with</b>	The text used to replace the text found by the regular expression.

## *Optional Properties*

You can use the following properties to change the default behavior:

<b>Ignore Case</b>	Specifies whether to ignore case when testing the input text.
<b>Multiline</b>	If true, then the ^ and \$ match the beginning and end, respectively, of any line, and not just the beginning and end of the entire string.
<b>Singleline</b>	If true, then the dot (.) matches every character instead of every character except linefeed (\n).

## *Published Data*

The activity publishes the following activity specific data:

<b>Output text</b>	The modified text.
--------------------	--------------------

# Regex Split Activity

---

The **Regex Replace** activity is used in a runbook split some input text into fragments using a specified regular expression.

## *Required Properties*

You must configure the following properties:

<b>Input text</b>	The input text to be modified.
<b>Regular expression</b>	The regular expression used to split the input text.

## *Optional Properties*

You can use the following properties to change the default behavior:

<b>Ignore Case</b>	Specifies whether to ignore case when testing the input text.
<b>Multiline</b>	If true, then the ^ and \$ match the beginning and end, respectively, of any line, and not just the beginning and end of the entire string.
<b>Singleline</b>	If true, then the dot (.) matches every character instead of every character except linefeed (\n).

## *Published Data*

The activity publishes the following activity specific data:

<b>Output text</b>	The modified text.
--------------------	--------------------

# Remove Text Activity

---

The **Remove Text** activity is used in a runbook to remove a text fragment from some input text.

## *Required Properties*

You must configure the following properties:

<b>Input text</b>	The input text to be split
<b>Remove from</b>	Specifies the relative position from which characters should be removed.

## *Optional Properties*

You can use the following properties to change the default behavior:

<b>Starting at</b>	Specified the position from which characters should be removed. The position of the first character is 1.
<b>Number of characters</b>	The number of characters to remove.

## *Published Data*

The activity publishes the following activity specific data:

<b>Output text</b>	The modified output text
--------------------	--------------------------

# Replace Text Activity

---

The **Replace Text** activity is used in a runbook to replace a text fragment in some input text.

## *Required Properties*

You must configure the following properties:

<b>Input text</b>	The input text to be modified.
<b>Find this</b>	The text fragment to replace.
<b>Replace with</b>	The replacement text.

## *Published Data*

The activity publishes the following activity specific data:

<b>Output text</b>	The modified output text
--------------------	--------------------------

# Reverse Text Activity

---

The **Reverse Text** activity is used in a runbook to reverse some input text in whole or in part.

## *Required Properties*

You must configure the following properties:

<b>Input text</b>	The input text to be reversed.
-------------------	--------------------------------

## *Optional Properties*

You can use the following properties to change the default behavior:

<b>Starting at</b>	The position to start reversing the input text. The position of the first character is 1.
<b>Number of characters</b>	The number of characters to reverse.

## *Published Data*

The activity publishes the following activity specific data:

<b>Output text</b>	The modified output text
--------------------	--------------------------

# Split Text Activity

---

The **Split Text** activity is used in a runbook to split some input text into text fragments using a specified character or text fragment separator, such as a comma. The **Split Text** activity also lets you select which text fragments you want to publish using a range of indices. For example, you could split a comma-separated text input and then select the first three items or the fifth, seventh and ninth.

## *Required Properties*

You must configure the following properties:

<b>Input text</b>	The input text to be split
<b>Separator</b>	The character or text fragment used to split the input text.

## *Optional Properties*

You can use the following properties to change the default behavior:

<b>Range</b>	Specifies which text fragments to publish. <ul style="list-style-type: none"><li>• To specify a range of fragments, use a hyphen. For example 1-3.</li><li>• To specify specific fragments, use a comma. For example: 5,7,9.</li><li>• Combine range and specific fragments. For example: 1-3,5,7,9.</li><li>• To specify the last fragment in a range use END, for example: 3-END.</li></ul>
<b>Skip empty values</b>	Specifies whether to exclude empty text fragments from the output text.

## *Published Data*

The activity publishes the following activity specific data:

<b>Output text</b>	The modified output text
--------------------	--------------------------

# Starts With Activity

---

The **Starts With** activity is used in a runbook to test whether some input text starts with a specified prefix.

## *Required Properties*

You must configure the following properties

<b>Input text</b>	The input text to be searched
<b>Case sensitive</b>	Specifies whether the search is case sensitive.
<b>Starts with</b>	The prefix to search for

## *Published Data*

The activity publishes the following activity specific data:

<b>Found</b>	Indicates whether the prefix was found
--------------	--

# Text Length Activity

---

The **Text Length** activity is used in a runbook to determine the length of some input text.

## *Required Properties*

You must configure the following properties:

---

<b>Input text</b>	The input text to be searched
-------------------	-------------------------------

---

## *Published Data*

The activity publishes the following activity specific data:

---

<b>Text length</b>	The length of the text input
--------------------	------------------------------

---

# Trim Text Activity

---

The **Trim Text** activity is used in a runbook to trim leading and/or trailing whitespace from some input text.

## *Required Properties*

You must configure the following properties:

<b>Input text</b>	The input text to be modified
<b>Trim start</b>	Specifies whether to trim whitespace from the start of the input text.
<b>Trim end</b>	Specified whether to trim whitespace from the end of the input text.

## *Published Data*

The activity publishes the following activity specific data:

<b>Output text</b>	The modified output text
--------------------	--------------------------

# Truncate Text Activity

---

The **Truncate Text** activity is used in a runbook to truncate input text.

## *Required Properties*

You must configure the following properties:

<b>Input text</b>	The input text to be truncated
<b>Text Length</b>	Length for the output text, from the beginning of the string

## *Published Data*

The activity publishes the following activity specific data:

<b>Output text</b>	The truncated output text
--------------------	---------------------------

# Uppercase Text Activity

---

The **Uppercase Text** activity is used in a runbook to convert the case of some input text to uppercase, in whole or in part.

## *Required Properties*

You must configure the following properties:

<b>Input text</b>	The input text to be modified
-------------------	-------------------------------

## *Optional Properties*

You can use the following properties to change the default behavior:

<b>Number of characters</b>	The number of characters to make uppercase.
<b>Starting at</b>	The position to start making characters uppercase. The first character has position 1.

## *Published Data*

The activity publishes the following activity specific data:

<b>Output text</b>	The truncated output text
--------------------	---------------------------