



# INTEGRATION MODULE FOR AZURE DEVOPS

*For Keverion Runbook Studio and Azure Automation*

**User Guide**

Version 1.2

Microsoft  
Azure

Certified

# Kelverion Integration Module for Azure DevOps

Copyright 2019 Kelverion Inc. All rights reserved.

Published: February 2021

[\*Feedback\*](#)

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

# Contents

Getting Started.....	4
System Requirements.....	4
Deploying the Integration Module .....	4
Using the PowerShell Gallery.....	4
Manual Installation .....	5
Licensing the Integration Module .....	5
Working with Activities in Runbook Studio.....	7
Smart Connections .....	7
Global Connection Assets.....	8
Activity Properties .....	9
Smart Discovery.....	9
Smart Parameters.....	10
Smart Filters .....	12
Retry Behavior .....	12
Additional Parameters.....	13
Add-DevOpsAttachmentContent .....	14
Remove-DevOpsWorkItem .....	15
Get-DevOpsAttachmentContent.....	16
Get-DevOpsAttachmentInfo .....	17
Get-DevOpsWorkItem.....	18
New-DevOpsWorkItem .....	19
Remove-DevOpsAttachment .....	20
Set-DevOpsWorkItem .....	21

# Getting Started

---

The following sections outline how to deploy and configure the Keverion Integration Module for Azure DevOps.

## System Requirements

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

- Keverion Runbook Studio 3.4
- Microsoft .NET Framework 4.6.2

### *Azure DevOps or Microsoft TFS:*

- Azure DevOps Services
- Azure DevOps Server 2020
- TFS 2017 Update 3, or TFS 2018 Update 3.2

## Deploying the Integration Module

The easiest way to install and deploy the Integration Module for Azure DevOps is from the PowerShell Gallery, but you can also download the module from Keverion and perform the steps manually.

You must install and deploy the integration module to each Azure Automation Account and hybrid worker host system that you plan to use to run your runbooks. You must also install the integration module on any Runbook Studio host systems that you will be using to build and manage your runbooks.

## Using the PowerShell Gallery

Using the commands in the **PowerShellGet** module you can download the Keverion Integration Module for Azure DevOps from the PowerShell Gallery and install it on your local computer. You can also deploy the module directly from the PowerShell Gallery to any of your Azure Automation Accounts.

### *Install the integration module on your local computer or hybrid worker:*

1. Confirm that the latest PowerShellGet module is installed.
2. Start a PowerShell window as an Administrator and run the command:

```
Install-Module -Name Keverion.Azure.DevOps -Scope AllUsers
```

#### *Upload the integration module to an Azure Automation account:*

1. Go to the [PowerShell Gallery](#).
2. Click the **Azure Automation** tab.
3. Click **Deploy to Azure Automation**. You will be directed to Microsoft Azure.
4. Select the **Automation Account** that you want to deploy the module to.
5. Click **OK**.

## Manual Installation

Alternatively, you can download the Integration Module package from Keverion and deploy it manually to your local computer, hybrid workers and Automation Accounts.

The download package from Keverion includes a **.zip** file containing the integration module as well as the User Guide and Release Notes. The following instructions assume that you have unzipped the download package and have access to the **.zip** file containing the Integration module.

#### *Install the integration module on your local computer or hybrid worker:*

1. Copy the **Keverion.Azure.DevOps.zip** file to your local computer.
2. Right-click on the file and select **Properties**.
3. Click the **General** tab. If necessary, click **Unblock**, and then click **OK**.
4. Unzip the **Keverion.Azure.DevOps.zip** file.
5. Copy the **Keverion.Azuer.DevOps** folder to a location in the `%PsModulePath%` path.

**Important:** When installing the Integration module on a hybrid worker, you must use a location that is accessible to all users of the computer.

#### *Upload the integration module to an Azure Automation account:*

1. Sign into [Microsoft Azure](#).
2. Open the Automation Account that you want to upload the module to.
3. Click **Modules** under Shared Resources. The list of installed modules is displayed.
4. Click **Add a module** at the top of the list.
5. In the **Upload File** box, select the **Keverion.Azure.DevOps.zip** file that you downloaded.
6. Click **OK**. Importing the module may take several minutes.

## Licensing the Integration Module

Licenses for Keverion Integration modules are managed and deployed using the **Keverion Runbook Studio** and Azure Automation connection assets.

*Register an Integration Module license with Runbook Studio:*

1. Open **Kelverion Runbook Studio**.
2. On the **File** tab, click **About**.
3. Click **License Information**.
4. Click the **Integration Modules** tab, and then click **Add License**.
5. Select the integration module license file (.kaml) and click **Open**.
6. You should see your entitlements displayed in the list.
7. Click **OK**.

**Important:** Entitlements will not display until after the Integration module has been installed on the Runbook Studio computer.

*Create a connection asset with a license key and upload it to an Azure Automation account:*

1. On the **Home** tab, click **Sign In**. The Sign In dialog appears.
2. Sign into your account.
3. In the **Active Azure Automation Account** box, select the account that you want to add the connection asset to.
4. Click **New Asset** and then click **Connection**. The New Connection dialog appears.
5. In the **Name** field, enter a name to identify the connection.
6. In the **Connection Type** field, select the desired connection type.
7. Enter the appropriate connection information in the provided fields.
8. Click **OK**.

*Update all connection asset license keys and upload them to an Azure Automation account:*

1. On the **Home** tab, click **Sign In**. The Sign In dialog appears.
2. Sign into your account.
3. In the Explorer panel, click the **Azure (Online)** group.
4. Right-click the Azure Automation Account that contains the connection assets you want to update, and then click **Update License Keys**. A summary is displayed.

# Working with Activities in Runbook Studio

The following sections outline some of the common configuration options that are available to you when working with the activities in the Keverion Integration Module for Azure DevOps.

*The integration module includes the following activities:*

Activity	Description
Add-DevOpsAttachmentContent	Add attachment content on work item
Get-DevOpsAttachmentContent	Get attachment content
Get-DevOpsAttachmentInfo	Get attachment information
Get-DevOpsWorkItem	Get a work item
New-DevOpsWorkItem	Create a new work item
Remove-DevOpsAttachment	Remove attachment on work item
Remove-DevOpsWorkItem	Delete a work item
Set-DevOpsWorkItem	Update a work item

**The advanced discovery capabilities provided by the activities in this integration module are only supported when authoring runbooks in Keverion Runbook Studio.**

When you publish your runbooks from Keverion Runbook Studio to Azure Automation or when you generate PowerShell code snippets for Service Management Automation, Runbook Studio will automatically convert the dynamically generated parameters and filters of Smart activities into the parameters provided by the underlying command activities.

## Smart Connections

In Keverion Runbook Studio you can configure one or more Smart Connections to establish reusable links between Runbook Studio and a specific Azure DevOps or Microsoft TFS instance. You can create as many Smart Connections as you require, specifying links to multiple instances. You can also create multiple Smart Connections to the same instance to allow for differences in security privileges for different user accounts.

*Add a smart connection in Keverion Runbook Studio:*

1. On the **Home** tab, click **Smart Connections**. The Smart Connections dialog appears.
2. Click **Add a connection** at the top of the list.
3. In the **Name** box, enter a name for the connection.
4. In the **Connection Type** box, select *Keverion.Azure.DevOps*.
5. In the **AuthenticationType** box, select the type of authentication:
  - a. Basic

- b. Windows
  - c. Personal Access Token
- 6. In the **WebServiceType** box, select the type of web service end point:
  - a. TFS (Select this when targeting TFS 2017 or 2018)
  - b. DevOps (Select this when targeting Azure DevOps Online)
  - c. DevOps Server (Select this when targeting Azure DevOps Server 2020)
- 7. In the **WebServiceUrl** box, type the DevOps or TFS endpoint URL.
- 8. In the **Username** and **Password** boxes, type the credentials that activity will use to connect to the Azure DevOps instance. Note: this only applies to Basic and Windows authentication types.
- 9. In the **PersonalAccessToken** box, type the personal access token. Note: this only applies to Personal Access Token authentication type.
- 10. Click **OK** to close the configuration dialog box, and then click **OK**.

## Global Connection Assets

The activities in the Kelverion Integration Module for Azure DevOps require connection information to connect to instances of Azure DevOps.

The recommended way to pass connection information to your activities in your runbooks is to use Global Connection Assets. Global connection assets let you securely define connection information in Azure which can then be retrieved on demand using either the **Get-AutomationConnection** cmdlet or Connection Asset Data Source.

### *Add a global connection asset in Runbook Studio:*

1. On the **Home** tab, click **Sign In**. The Sign In dialog appears.
2. Sign into your account.
3. In the **Active Azure Automation Account** box, select the account that you want to add the connection asset to.
4. Click **New Asset** and then click **Connection**. The New Connection dialog appears.
5. In the **Name** box, type the name for the configuration.
6. In the **Connection Type** box, select *Kelverion.Azure.DevOps*.
7. In the **AuthenticationType** box, select the type of authentication:
  - a. Basic
  - b. Windows
  - c. Personal Access Token
8. In the **WebServiceType** box, select the type of web service end point:



- a. TFS (Select this when targeting TFS 2017 or 2018)
  - b. DevOps (Select this when targeting Azure DevOps Online)
  - c. DevOps Server (Select this when targeting Azure DevOps Server 2020)
9. In the **WebServiceUrl** box, type the DevOps or TFS endpoint URL.
10. In the **Username** and **Password** boxes, type the credentials that activity will use to connect to the Azure DevOps instance. Note: this only applies to Basic and Windows authentication types.
11. In the **PersonalAccessToken** box, type the personal access token. Note: this only applies to Personal Access Token authentication type.
12. Click **OK** to close the New Connection dialog box.

## Activity Properties

All activities in the Keverion Integration Module for Azure DevOps have the following properties:

Property	Description
Label	A unique label that identifies the activity in the runbook. Runbook Studio will provide a default name for each activity, but you can provide your own labels to make their role in the runbook more obvious.
Description	An optional description of the activity. Providing a description is a great way to let everyone understand the function of the activity in the runbook.
Checkpoint	Indicates whether a checkpoint is set in the runbook workflow after the activity runs. Checkpoints are only available for Graphical PowerShell Workflow runbooks. If the runbook uses Azure cmdlets, you should follow best practices and follow a check-pointed activity with an <a href="#">Add-AzureRMAccount</a> in case the runbook is suspended and restarts from this checkpoint on a different worker.

## Smart Discovery

When designing runbooks in Keverion Runbook Studio, you will notice that the activities in the Keverion Integration Module for Azure DevOps include a **Discovery** panel instead of the **Parameter Sets** panel that is present for standard command activities. This is because the activities in the Keverion Integration Module for Azure DevOps support interactive discovery of the Azure DevOps assets in your environments.

All activities in the Keverion Integration Module for Azure DevOps have a **Connection** option on the **Discovery** panel which lets you specify how Runbook Studio should connect to Azure DevOps.

When connected to Azure DevOps, Runbook Studio will provide additional discovery options, such as **Collection**, **Project**, and **Work Item Type** which can be used to specify the resources that you want to integrate with. Once you have filled in the discovery options Runbook Studio will provide additional parameters and, in some cases, filters which can be used to configure the activity.

## Smart Parameters

Unlike standard command activities, whose parameters are determined by the Parameter Set that is selected, the parameters in the Keverion Integration Module for Azure DevOps are determined by the Discovery options that you specify.

For example, when using the **New-DevOpsWorkItem** activity, the Discovery panel will contain options for selecting an Azure DevOps Project and Work Item Type. Once you have selected a form, Runbook Studio will provide you with parameters that coincide with the fields in the issue type. If you select another issue type, Runbook Studio will provide you with a different set of parameters automatically.

**You must configure all mandatory parameters.** To view the optional parameters that are associated with an activity, click **Optional** at the top of the Parameters tab.

In addition, all activities in the Keverion Integration Module for Azure DevOps include a **Connection** parameter which is used to specify information that the activity will use to connect to Azure DevOps when it is executed as part of a runbook running on a hybrid worker. Typically, you will assign a Connection Asset data source to this parameter so that the activity can securely use connection information stored in Azure.

The Connection parameter should not be confused with the similarly named Connection option on the Discovery panel which is used to specify how Runbook Studio connects to Azure DevOps in order to provide design-time configuration options.

Several factors determine the data sources that are available when configuring a parameter. They include: the parameter's data type, whether it is linked to another activity and whether the runbook has any input parameters.

Runbook studio supports the following data sources.

Data Source	Description
<b>Activity output</b>	<p>Specify activity whose output will be assigned to the parameter. You may also provide an optional Path to select a specific property of the output objects that are generated by the activity.</p> <p>Available when the activity is linked to a source activity.</p>
<b>Not configured</b>	<p>Clears any value that was previously configured. You must configure all mandatory parameters.</p>
<b>Certificate asset</b>	<p>Specify the name of the global certificate asset that will be used to provide a value for the parameter.</p> <p>If you have connected to Azure and selected a Subscription and Automation Account on the toolbar, the data source will provide the names of the certificates that are available.</p>
<b>Credential asset</b>	<p>Specify the name of the global credential asset that will be used to provide a value for the parameter.</p> <p>If you have connected to Azure and selected a Subscription and Automation</p>

	Account on the toolbar, the data source will provide the names of the credentials that are available.
<b>Constant</b>	<p>Specify a constant value to assign to the parameter.</p> <p>Available for parameters that have the following data types:</p> <ul style="list-style-type: none"> <li>• String</li> <li>• DateTime</li> <li>• Timespan</li> <li>• Decimal</li> <li>• Double</li> </ul> <p>When assigning a constant DateTime and Time values, Runbook Studio assumes the value is in UTC.</p>
<b>Connection asset</b>	<p>Specify the name of the global connection asset that will be used to provide a value for the parameter.</p> <p>If you have connected to Azure and selected a Subscription and Automation Account on the toolbar, the data source will provide the names of the connections that are available.</p>
<b>Empty string</b>	An empty string will be assigned to the parameter. Available when the parameter is type <i>System.String</i>
<b>Null</b>	A null (\$null) value will be assigned to the parameter. Available when the parameter type is a reference type.
<b>PowerShell expression</b>	<p>Specify a <i>simple</i> PowerShell expression whose output will be assigned to the parameter.</p> <p>You can use variables in the expression to access the output of an activity or a runbook parameter.</p>
<b>Runbook input</b>	<p>Specify the name of the runbook input parameter whose value will be assigned to the parameter.</p> <p>Available when the runbook has one or more input parameters.</p>
<b>Variable asset</b>	<p>Specify the name of the global variable asset that will be used to provide a value for the parameter.</p> <p>If you have connected to Azure and selected a Subscription and Automation Account on the toolbar, the data source will provide the names of the variables that are available.</p>

## Smart Filters

Some of the activities in the Kelverion Integration Module for Azure DevOps include a **Filters** panel which lets you specify filters that can be used to retrieve specific issues in Azure DevOps.

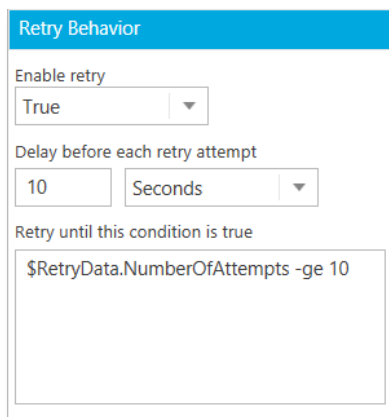
To add a filter to your activity, select the **Filters** panel and click **Add**. Filters have the following properties.

Property	Description
<b>Filter</b>	The name of the filter.
<b>Operation</b>	<p>The operation used to evaluate the filter. Different operators will be provided based on the filter that is selected. Possible filter operators include:</p> <ul style="list-style-type: none"><li>• Equals</li><li>• Does not equal</li><li>• Is less than</li><li>• Is less than or equal to</li><li>• Contains</li><li>• Does not contain</li><li>• Is greater than</li><li>• Is greater than or equal to</li></ul>
<b>Value</b>	<p>The data source used to retrieve the value to use to evaluate the filter.</p> <p>The value used to evaluate the filter will be obtained. For more information on data sources, please refer to the Parameters section for more information on configuring data sources.</p>

## Retry Behavior

The activities in the Kelverion Integration Module for Azure DevOps can be configured to run multiple times until a condition, which you specify, is satisfied. You can use the retry behavior options to configure activities that should run multiple times, that are error prone or may need more than one attempt for success.

When you enable retry for an activity, you can configure the runbook to wait a specified number of minutes or seconds before running the activity again. If no delay is specified the runbook will run the activity again, immediately after it completes.



The retry condition lets you specify a PowerShell expression that the runbook will evaluate after each time the activity runs. If the result of the expression is true the activity does not run again, and the runbook moves on to the next child activity in the runbook.

When defining the retry conditions for your activity you can take advantage of a global variable called **\$RetryData**. Specific information about the last time the activity ran can be accessed using the following properties.

Property	Description
<b>NumberOfAttempts</b>	Number of times that the activity has ran
<b>Output</b>	Output that was generated by the activity the last time that it ran
<b>TotalDuration</b>	Time elapsed since the activity was started
<b>StartedAt</b>	Time in UTC when the activity was first started

The following are some examples of activity retry conditions:

```
# Run the activity exactly 5 times
$RetryData.NumberOfAttempts -eq 5

# Run the activity until it produces some output
$RetryData.Output.Count -ge 1

# Run the activity until at least 2 minutes has elapsed
$RetryData.TotalDuration.TotalMinutes -ge 2
```

## Additional Parameters

The activities in the Keverion Integration Module for Azure DevOps let you specify additional PowerShell parameters that you can use to control the behavior of the activity.

For example, to output detailed information about the operation performed by an activity you would specify **-Verbose:\$True**

# Add-DevOpsAttachmentContent

---

The **Add-DevOpsAttachmentContent** activity adds an attachment to a work item.

## *Discovery Options*

This activity provides the following smart discovery options:

<b>Connection</b>	The name of the Smart Connection used to connect Runbook Studio to Azure DevOps.
<b>Collection</b>	The name of the collection. <b>Note:</b> Only applies to TFS.

## *Required Parameters*

This activity requires the following parameters:

<b>Connection</b>	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
<b>Work Item ID</b>	The ID of the work item to add the attachment to.
<b>File Name</b>	File name of attachment.
<b>Content</b>	Byte array of file content.

## *Optional Parameters*

This activity provides the following optional properties:

<b>Comment</b>	Comment to add to the attachment.
----------------	-----------------------------------

## *Output*

This activity returns the ID of the attachment that was added to the issue.

# Remove-DevOpsWorkItem

---

The **Remove-DevOpsWorkItem** activity removes a work item.

## *Discovery Options*

This activity provides the following smart discovery options:

<b>Connection</b>	The name of the Smart Connection used to connect Runbook Studio to Azure DevOps.
<b>Collection</b>	The name of the collection. <b>Note:</b> Only applies to TFS.

## *Required Parameters*

This activity requires the following parameters:

<b>Connection</b>	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
<b>Work Item ID</b>	The ID of the issue to delete.

## *Optional Parameters*

This activity does not have any optional parameters.

# Get-DevOpsAttachmentContent

---

The **Get-DevOpsAttachmentContent** activity gets the content of the specified attachment.

## *Discovery Options*

This activity provides the following smart discovery options:

<b>Connection</b>	The name of the Smart Connection used to connect Runbook Studio to Azure DevOps.
<b>Collection</b>	The name of the collection. <b>Note:</b> Only applies to TFS.

## *Required Parameters*

This activity requires the following parameters:

<b>Connection</b>	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
<b>Attachment ID</b>	Unique Id of the attachment to get.

## *Optional Parameters*

This activity does not have any optional parameters.

## *Output*

This activity returns the content of the attachment as a byte array.



# Get-DevOpsAttachmentInfo

The **Get-DevOpsAttachmentInfo** activity gets information about the attachments associated with a work item.

## Discovery Options

This activity provides the following smart discovery options:

<b>Connection</b>	The name of the Smart Connection used to connect Runbook Studio to Azure DevOps.
<b>Collection</b>	The name of the collection. <b>Note:</b> Only applies to TFS.

## Required Parameters

This activity requires the following parameters:

<b>Connection</b>	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
<b>Work Item ID</b>	Unique Id of the issue.

## Filters

This activity provides the following smart filters:

<b>Attachment ID</b>	The unique ID of the attachment
<b>Created Date</b>	The date/time that attachment was created
<b>Modified Date</b>	The date/time the attachment was last modified
<b>Content Length</b>	The number of bytes in the attachment
<b>Comment</b>	The comment given to the attachment
<b>Name</b>	The name of the attached file

## Output

This activity returns objects that represent issue attachments. Each attachment object has the following properties:

<b>AttachmentId</b>	The unique ID of the attachment
<b>CreatedDate</b>	Resource The date/time that attachment was created
<b>ModifiedDate</b>	Resource The date/time the attachment was last modified
<b>ContentLength</b>	The number of bytes in the attachment
<b>Comment</b>	Attachment comment
<b>Name</b>	Attachment name

# Get-DevOpsWorkItem

---

The **Get-DevOpsWorkItem** activity gets work items.

## *Discovery Options*

This activity provides the following smart discovery options:

<b>Connection</b>	The name of the Smart Connection used to connect Runbook Studio to Azure DevOps.
<b>Collection</b>	The name of the collection. <b>Note:</b> Only applies to TFS.
<b>Search By</b>	Search by ID, Filters or WIQL.

## *Required Parameters*

This activity requires the following parameters:

<b>Connection</b>	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
-------------------	--

## *Optional Parameters*

This activity provides the following optional properties:

<b>Query</b>	When Search By is set to WIQL you can specify a WIQL query instead of using filters.
<b>Work Item ID</b>	A unique work item ID

## *Filters*

When searching by *Filters*, this activity will provide filter options based on the fields that have been configured in DevOps/TFS environment that Runbook Studio is connected to.

## *Output*

This activity returns objects that represent the work items that were retrieved. The object properties are determined by the fields that have been configured in DevOps/TFS that Runbook Studio is connected to.

# New-DevOpsWorkItem

---

The **New-DevOpsWorkItem** activity creates a new work item.

## *Discovery Options*

This activity provides the following smart discovery options:

<b>Connection</b>	The name of the Smart Connection used to connect Runbook Studio to Azure DevOps.
<b>Collection</b>	The name of the collection. <b>Note:</b> Only applies to TFS.
<b>Project</b>	The name of the project to create the work item in.
<b>Work Item Type</b>	The type of work item to create.

## *Required Parameters*

This activity requires the following parameters:

**Note:** The required fields are dynamic and change depending on Project and Issue Type selected.

<b>Connection</b>	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
<b>Title</b>	The title of work item.

## *Optional Parameters*

The optional fields are dynamic and change depending on Project and Issue Type selected.

## *Output*

This activity outputs the unique id of the work item that was added.

# Remove-DevOpsAttachment

---

The **Remove-DevOpsAttachment** activity removes an attachment from a work item.

## *Discovery Options*

This activity provides the following smart discovery options:

<b>Connection</b>	The name of the Smart Connection used to connect Runbook Studio to Azure DevOps.
<b>Collection</b>	The name of the collection. <b>Note:</b> Only applies to TFS.

## *Required Parameters*

This activity requires the following parameters:

<b>Connection</b>	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
<b>Attachment ID</b>	The Attachment ID of the attachment to remove.
<b>Work Item ID</b>	The Work Item Id of the work item containing the attachment.

## *Optional Parameters*

This activity does not have any optional parameters.

# Set-DevOpsWorkItem

---

The **Set-DevOpsWorkItem** activity updates an existing work item.

## *Discovery Options*

This activity provides the following smart discovery options:

<b>Connection</b>	The name of the Smart Connection used to connect Runbook Studio to Azure DevOps.
<b>Collection</b>	The name of the collection. <b>Note:</b> Only applies to TFS.
<b>Project</b>	The name of the project to update the work item in.
<b>Work Item Type</b>	The type of work item to update.

## *Required Parameters*

This activity requires the following parameters:

<b>Connection</b>	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
<b>Work Item ID</b>	The id of the work item to update.

## *Optional Parameters*

The optional fields are dynamic and change depending on fields available in DevOps/TFS. Not all listed fields will be valid for the specified Issue and it is important to verify you are specifying valid fields. If an invalid field is specified, the activity will fail with an error and provide a list of valid fields.