

INTEGRATION MODULE FOR HALOITSM

For Kelverion Runbook Studio and Azure Automation

User Guide

Version 1.2



Kelverion Integration Module for HaloITSM

Copyright © 2026 Kelverion Inc. All rights reserved.

Published: November 2026

Feedback

Send suggestions and comments about this document to 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
Integrating with HaloITSM 7
Create a HaloITSM API Application 7
Smart Connections 8
Global Connection Assets 8
Activity Properties 10
Smart Discovery 10
Smart Parameters 10
Smart Filters 12
Retry Behavior 12
Additional Parameters 13

Activity Reference 14
Add-HaloItsmAttachment 14
Get-HaloItsmAgent 15
Get-HaloItsmAsset 16
Get-HaloItsmAttachment 17
Get-HaloItsmAttachmentContent 18
Get-HaloItsmClient 19
Get-HaloItsmItem 20
Get-HaloItsmSite 21
Get-HaloItsmSupplier 22
Get-HaloItsmTeam 23
Get-HaloItsmTicket 24
Get-HaloItsmUser 26
New-HaloItsmTicket 28
Remove-HaloItsmAttachment 29
Remove-HaloItsmTicket 30
Set-HaloItsmTicket 31

Notes 32

Getting Started

The following sections outline how to deploy and configure the Kelverion Integration Module for HaloITSM.

System Requirements

The Integration Module for Kelverion HaloITSM requires the following software to be installed and configured prior to implementing the integration. For more information on installing Kelverion Runbook Studio, please refer to the Kelverion Runbook Studio User Guide.

- Kelverion Runbook Studio 5.8
- Microsoft .NET Framework 4.7.2

Deploying the Integration Module

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

You must install and deploy the Integration Module to each Azure Automation Account and hybrid runbook 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 Kelverion Integration Module for HaloITSM 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 runbook worker:

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

```
Install-Module -Name Kelverion.HaloItsm -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 too.
5. Click **OK**.

Manual Installation

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

The download package from Kelverion 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.

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

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

1. Copy the **Kelverion.AutomationPortal.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 **Kelverion.HaloItsm.zip** file.
5. Copy the **Kelverion.HaloItsm** folder to a location in the `%PsModulePath%` path.

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.
4. Click **Add a module** at the top of the list.
5. In the **Upload File** box, select the **Kelverion.HaloItsm.zip** file that you downloaded.
6. Click **OK**.

Note: Importing the module may take several minutes.

Licensing the Integration Module

Licenses for Kelverion Integration Modules are managed and deployed using the *Kelverion Runbook Studio* and *Automation Connection Assets*.

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

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**.

Create a Connection Asset with a license key and upload it to Azure:

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 fields provided.
8. Click **OK**.

Update all Connection Assets license keys and upload them to Azure:

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 and then and then and then and then click **Update License Keys**. A summary is displayed.

Working with Activities in Runbook Studio

The following sections outline the common configuration options that are available to you when working with the activities in the Kelverion Integration Module for HaloITSM.

The integration module includes the following activities:

Add-HaloItsmAttachment	Add an attachment to HaloITSM ticket.
Get-HaloItsmAgent	Gets HaloITSM agents.
Get-HaloItsmAsset	Gets HaloITSM assets.
Get-HaloItsmAttachment	Gets HaloITSM attachments.
Get-HaloItsmAttachmentContent	Downloads the content of a HaloITSM attachment
Get-HaloItsmItem	Gets HaloITSM items.
Get-HaloItsmSite	Gets HaloITSM sites.
Get-HaloItsmSupplier	Gets HaloITSM suppliers.
Get-HaloItsmTicket	Gets HaloITSM tickets.
Get-HaloItsmUser	Gets HaloITSM users.
New-HaloItsmTicket	Creates a new HaloITSM ticket.
Remove-HaloItsmAttachment	Removes a HaloITSM attachment.
Remove-HaloItsmTicket	Removes a HaloITSM ticket.
Set-HaloItsmTicket	Update an existing HaloITSM ticket.

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

When you publish your runbooks from Kelverion 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.

Integrating with HaloITSM

The following sections outline how to configure Runbook Studio and Azure Automation to integrate with HaloITSM.

Create a HaloITSM API Application

Before you can configure Runbook Studio smart connection and Azure Automation connection assets, you must first create obtain a unique Client ID and Client Secret from HaloITSM.

1. Login to your HaloITSM portal as an administrator.
2. On the navigation bar, go to **Configuration > Integrations > HaloITSM API**

3. Click **View Applications**.
4. Click **+ New**.
5. In the **Application Name** box, enter a name for the application (e.g., Kelverion)
6. For **Authentication Method**, select **Client ID and Secret (Services)**.
7. For **Login Type**, select **Agent.**
8. For **Agent to log in as**, select an agent with the required privileges.
9. Copy the **Client ID** and **Client Secret** values for later use.
10. Click **Save.**

Smart Connections

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

Add a Smart Connection in Kelverion Runbook Studio:

1. Click **Connections** in the Runbook Studio toolbar.
2. In the **Smart Connections** dialog click **Add.**
3. In the **Name** box, type a name for the configuration. This could be the name of the instance or a descriptive name to distinguish the type of configuration.
4. For **Connection Type**, select **Kelverion.HaloItsm**.
5. In the **ResourceUrl** box, enter the URL of the HaloITSM REST API (e.g., https://contoso.haloitsm.com/api).
6. In the **AuthorizationUrl** box, enter the URL of the Halo authorization server (e.g., https://contoso.haloitsm.com/auth)
7. In the **ClientId** box, enter your applications client ID.
8. In the **ClientSecret** box, enter your applications client secret.
9. Click **OK** to close the configuration dialog box, and then click **OK**.

Global Connection Assets

The activities in the Kelverion Integration Module for HaloITSM require connection information to connect to instances of Automation Portal as well as the Kelverion Management server.

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. In Kelverion Runbook Studio, click the **Azure** panel.

2. Select your Azure subscription.
3. Select your Automation account.
4. Select **Connections** and right click.
5. Select **Add New Connection.**
6. In the **Name** box, type a name for the configuration. This could be the name of the instance or a descriptive name to distinguish the type of configuration.
7. In the **Connection Type** box, select **Kelverion.HaloItsm**.
8. In the **ResourceUrl** box, enter the URL of the HaloITSM REST API.
9. In the **AuthorizationUrl** box, enter the URL of the Halo authorization server.
10. In the **ClientId** box, enter your applications client ID.
11. In the **ClientSecret** box, enter your applications client secret.
12. Click **OK** to close the New Connection dialog box.

Activity Properties

All activities in the Kelverion Integration Module for HaloITSM have the following properties:

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 fantastic 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 *Add-AzureRMAccount* in case the runbook is suspended and restarts from this checkpoint on a different worker.

Smart Discovery

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

All activities in the Kelverion Integration Module for HaloITSM have a **Connection** option on the **Discovery** panel which lets you specify how Runbook Studio should connect to Automation Portal.

When connected to Automation Portal, Runbook Studio will offer more discovery options. Once you have filled in the discovery options Runbook Studio will provide more parameters and for some activities, filters which can be used to configure the activity.

Smart Parameters

Unlike standard command activities, whose parameters are decided by the Parameter Set that is selected, the parameters in the Kelverion Integration Module Kelverion Automation Portal are decided by the Discovery options that you specify.

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 Kelverion Integration Module for HaloITSM include a **Connection** parameter which is used to specify information that the activity will use to connect to Automation Portal when it is executed as part of a runbook running on a hybrid runbook 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 Automation Portal to offer design-time configuration options.

Several factors decide 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.

Activity output	**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.** **Available when the activity is linked to a source activity.**
Not configured	Clears any value that was previously configured. You must configure all mandatory parameters.
Certificate asset	Specify the name of the global certificate asset that will be used to provide a value for the parameter. 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.
Credential asset	Specify the name of the global credential asset that will be used to provide a value for the parameter. 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 credentials that are available.
Constant	Specify a constant value to assign to the parameter. Available for parameters that have the following data types: • String • DateTime • Timespan • Decimal • Double When assigning a constant DateTime and Time values, Runbook Studio assumes the value is in UTC.
Connection asset	Specify the name of the global connection asset that will be used to provide a value for the parameter. 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.
Empty string	An empty string will be assigned to the parameter. Available when the parameter type is *System.String*

Null	A null ($null) value will be assigned to the parameter. Available when the parameter type is a reference type.
PowerShell expression	Specify a *simple* PowerShell expression whose output will be assigned to the parameter. You can use variables in the expression to access the output of an activity or a runbook parameter.
Runbook input	Specify the name of the runbook input parameter whose value will be assigned to the parameter. Available when the runbook has one or more input parameters.
Variable asset	Specify the name of the global variable asset that will be used to provide a value for the parameter. 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.

Smart Filters

Some activities in the Kelverion Integration Module for HaloITSM include a **Filters** panel which lets you specify filters that can be used to retrieve specific records in Automation Portal.

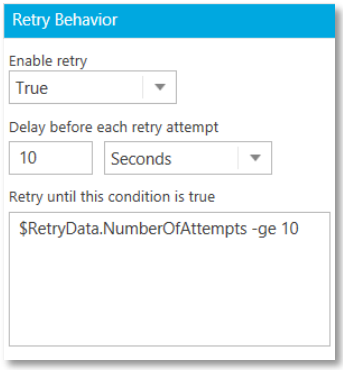
To add a filter to your activity, select the **Filters** panel and click **Add.** Filters have the following properties.

Filter	The name of the filter.
Operation	The operation is used to evaluate the filter. Different operators will be provided based on the filter that is selected. Filter operators include: • Equals • Is less than • Is greater than • Contains • Starts with
Value	The data source used to retrieve the value used to evaluate the filter. 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.

Retry Behavior

The activities in the Kelverion Integration Module Kelverion Automation Portal 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, which are prone to error 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, at once after it is completed.



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.

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

The following are some examples of activity retry conditions.

```
# Run the activity exactly five 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 Kelverion Integration Module for HaloITSM let you specify other 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`**

Activity Reference

Add-HaloItsmAttachment

The **Add-HaloItsmAttachment** activity is used in a runbook to upload an attachment to a ticket in HaloITSM.

Discovery Options

This activity includes the following smart discovery options:

Connection	Select the name of the Smart Connection used to connect Runbook Studio to HaloITSM.

Parameters

This activity provides the following parameters.

Connection	Specifies a hashtable holding information used to connect to HaloITSM. This parameter is mandatory.
Content	Specifies the content to upload. This parameter is mandatory.
ContentType	Specifies the MIME type of the attachment.
FileName	Specifies the name to assign to the attachment. This parameter is mandatory.
Note	Specifies more information.
TicketId	Specifies the ID of the ticket record that the attachment is for. This parameter is mandatory.

Output

This activity returns the unique ID of the attachment record that was created.

Get-HaloItsmAgent

The **Get-HaloItsmAgent** activity is used in a runbook to query and retrieve agent records from HaloITSM.

Discovery Options

This activity includes the following smart discovery options:

Connection	Select the name of the Smart Connection used to connect Runbook Studio to HaloITSM.
Search By	Select whether you want to retrieve agents by their **Agent ID** or by using **Filters**.

Parameters

This activity includes the following parameters when searching by **Agent ID**. Both parameters are mandatory.

AgentId	Specifies the IDs of the agent(s) to be returned.
Connection	Specifies a hashtable holding information used to connect to HaloITSM.

This activity includes the following parameters when searching by **Filters**.

ClientId	Filter by agents who have access to the client with the specified ID.
Connection	Specifies a hashtable holding information used to connect to HaloITSM. This parameter is mandatory.
DepartmentId	Filter by agents that belong to the department with the specified ID.
Search	Search for agents whose name, email address or telephone number are like the specified text.
IncludeInactive	Indicates whether to include inactive agents in the response.
TeamId	Filter by agents that belong to the team with the specified ID.

Output

This activity returns objects that have information about HaloITSM agents.

Get-HaloItsmAsset

The **Get-HaloItsmAsset** activity is used in a runbook to query and retrieve Asset records from HaloITSM.

Discovery Options

This activity includes the following smart discovery options:

Connection	Select the name of the Smart Connection used to connect Runbook Studio to HaloITSM.
Search By	Select whether you want to retrieve agents by their **Asset ID** or by using **Filters**.

Parameters

This activity includes the following parameters when searching by **Asset ID**. Both parameters are mandatory.

AssetId	Specifies the IDs of the asset records to retrieve. Included when searching by **Asset ID**.
Connection	Specifies a hashtable holding information used to connect to HaloITSM.

This activity includes the following parameters when searching by **Filters**.

ClientId	Filter by assets belonging to the client with the specified ID.
Connection	Specifies a hashtable holding information used to connect to HaloITSM. This parameter is mandatory.
ContractId	Filter by assets belonging to the contract with the specified ID.
Descending	Indicates whether to return asset records in descending order.
Search	Search for assets whose tag, key field, asset type, or ownser is like the specified text.
IncludeInactive	Indicates whether to return inactive agent asset records.
Limit	Specifies the maximum number of assets to return. The default is fifty.
OrderBy	Specifies the property used to order the returned assets.
SiteId	Filter by assets belonging to the site with the specified ID.
TicketId	Filter by assets belonging to the ticket with the specified ID.
Username	Filter by assets belonging to the user with the specified username.

Output

This activity returns objects that have information about HaloITSM assets.

Get-HaloItsmAttachment

The **Get-HaloItsmAttachment** activity is used in a runbook to query and retrieve attachment records from HaloITSM.

Discovery Options

This activity includes the following smart discovery options:

Connection	Select the name of the Smart Connection used to connect Runbook Studio to HaloITSM.

Parameters

This activity includes the following mandatory parameters.

Connection	Specifies a hashtable holding information used to connect to HaloITSM.
TicketId	Specifies the ID of the ticket to return attachment records for.

Output

This activity returns objects that have information about HaloITSM attachment records.

Get-HaloItsmAttachmentContent

The **Get-HaloItsmAttachmentContent** activity is used in a runbook to download the actual file contents of an attachment record from HaloITSM. This is useful when you need to process or store files that were uploaded to HaloITSM.

Discovery Options

This activity includes the following smart discovery options:

Connection	Select the name of the Smart Connection used to connect Runbook Studio to HaloITSM.

Parameters

This activity includes the following mandatory parameters.

AttachmentId	Specifies the ID of the attachment to download.
Connection	Specifies a hashtable holding information used to connect to HaloITSM.

Output

This activity outputs the downloaded attachment content as an array of bytes.

Get-HaloItsmClient

The **Get-HaloItsmClient** activity in a runbook is used to query and retrieve client records from HaloITSM. Clients include organizations, customers, and prospects.

Discovery Options

This activity includes the following smart discovery options:

Connection	Select the name of the Smart Connection used to connect Runbook Studio to HaloITSM.
Searby By	Select whether you want to retrieve clients by their **Client ID** or with **Filters**.

Parameters

This activity includes the following mandatory parameters when searching by **Client ID**.

ClientId	Specifies the IDs of the clients records to return. Included when searching by **Client ID.**
Connection	Specifies a hashtable holding information used to connect to HaloITSM.

This activity includes the following parameters when searching by **Filters**.

Connection	Specifies a hashtable holding information used to connect to HaloITSM. This parameter is mandatory.
Descending	Indicates whether to return the records in descending order. The default is ascending order.
IncludeInactive	Specifies whether the return inactive clients. The default is false.
Limit	The maximum number of records to get. The default is fifty.
OrderBy	Specifies the name of the property used to order the returned records.
Search	Search for clients whose name is like the specified text.
TopLevelId	Filter by clients whose parent is the client with the specified ID.

Output

This activity returns objects that have information about HaloITSM client.

Get-HaloItsmItem

The **Get-HaloItsmItem** activity is used in a runbook to query and retrieve item records from HaloITSM.

Discovery Options

This activity includes the following smart discovery options:

Connection	Select the name of the Smart Connection used to connect Runbook Studio to HaloITSM.
Search By	Indicates whether you want to query and return agents by their **Item ID** or by using one or more **Filters**.

Parameters

This activity includes the following mandatory parameters when searching by **Item ID**.

Connection	Specifies a hashtable holding information used to connect to HaloITSM.
ItemId	Specifies the IDs of the item records to return. Included when searching by Item ID.

This activity includes the following parameters when searching by **Filters**.

Connection	Specifies a hashtable holding information used to connect to HaloITSM. This parameter is mandatory.
Descending	Specifies whether to return the records in descending order. The default is false.
Search	Search for items whose name, internal reference, external reference, manufacturer SKU or description is like the specified text.
IncludeInactive	Specifies whether the return inactive items. The default is false.
Limit	Specifies the maximum number of item records to return. The default is fifty.
OrderBy	Specifies the name of the property that should be used to order the returned records.
SupplierId	Filter by items belonging to the supplier with the specified ID.

Output

This activity returns objects that have information about HaloITSM items.

Get-HaloItsmSite

The **Get-HaloItsmSite** activity is used in a runbook to query and retrieve Site records from HaloITSM.

Discovery Options

This activity includes the following smart discovery options:

Connection	Select the name of the Smart Connection used to connect Runbook Studio to HaloITSM.
Search By	Select whether you want to retrieve agents by their **Site ID** or by using **Filters**.

Parameters

This activity includes the following mandatory parameters when searching by **Site ID**.

Connection	Specifies a hashtable holding information used to connect to HaloITSM.
SiteId	Specifies the IDs of the site records to return.

This activity includes the following parameters when searching by **Filters**.

ClientId	Filter by sites that belong to the client with the specified ID.
Connection	Specifies a hashtable holding information used to connect to HaloITSM. This parameter is mandatory.
Descending	Indicates whether to return the records in descending order. The default is ascending order.
IncludeInactive	Specifies whether to return inactive sites. The default is false.
Limit	The maximum number of records to get. The default is fifty.
OrderBy	Specifies the name of the property used to order the returned records.
Search	Search for sites whose name is like the specified text.

Output

This activity returns objects that are HaloITSM sites.

Get-HaloItsmSupplier

The **Get-HaloItsmSupplier** activity is used in a runbook to query and retrieve Supplier records from HaloITSM.

Discovery Options

This activity includes the following smart discovery options:

Connection	Select the name of the Smart Connection used to connect Runbook Studio to HaloITSM.
Search By	Select whether you want to retrieve agents by their **Supplier ID** or by using **Filters**.

Parameters

This activity includes the following mandatory parameters when searching by **Supplier ID**.

Connection	Specifies a hashtable holding information used to connect to HaloITSM.
SupplierId	Specifies the IDs of the supplier records to return.

This activity includes the following parameters when searching by **Filters**.

Connection	Specifies a hashtable holding information used to connect to HaloITSM. This parameter is mandatory.
Descending	Indicates whether to return the records in descending order. The default is ascending order.
IncludeInactive	Specifies whether to include inactive supplier records. The default is false.
Limit	Specifies the maximum number of supplier records to return. The default is fifty;
Limit	The maximum number of records to get. The default is fifty.
OrderBy	Specifies the name of the property used to order the returned records.
Search	Search for suppliers whose name contains the specified text.

Output

This activity returns objects that have information about HaloITSM suppliers.

Get-HaloItsmTeam

The **Get-HaloItsmTeam** activity is used in a runbook to query and retrieve Team records from HaloITSM.

Discovery Options

This activity includes the following smart discovery options:

Connection	Select the name of the Smart Connection used to connect Runbook Studio to HaloITSM.
Search By	Select whether you want to retrieve agents by their **Team ID** or by using **Filters**.

Parameters

This activity includes the following mandatory parameters when searching by **Team ID**.

Connection	Specifies a hashtable holding information used to connect to HaloITSM
TeamId	Specifies the IDs of the team records to return. Included when search by **Team ID**.

This activity includes the following parameters when searching by **Filters**.

Connection	Specifies a hashtable holding information used to connect to HaloITSM. This parameter is mandatory.
DepartmentId	Filter by teams that belong to the department with the specified ID.
IncludeInactive	Specifies whether to include inactive records. The default is false.

Output

This activity returns objects that have information about HaloITSM teams.

Get-HaloItsmTicket

The **Get-HaloItsmTicket** activity is used in a runbook to query and retrieve Ticket records from HaloITSM.

Discovery Options

This activity includes the following smart discovery options:

Connection	Select the name of the Smart Connection used to connect Runbook Studio to HaloITSM.
ITIL Ticket Type	Select the ITIL ticket type. Options include Incident, Change Request, Service Request, Problem and Advice/Other.
Search By	Indicates whether you want to query and return tickets by their Ticket ID or by using Filters.
Ticket Type	Select the ticket type. Included options depend on the ITIL Ticket Type that was selected. For example, if Incident is selected then the included options are Incident, Major Incident, Technical Task, Business Task, and Microsoft Teams Task.

Parameters

This activity includes the following mandatory parameters when searching by **Ticket ID**.

Connection	Specifies a hashtable holding information used to connect to HaloITSM.
TicketId	Specifies the ID of the ticket record(s) to return.

This activity includes the following parameters when searching by **Filters**.

ClientId	Filter by tickets belonging to the client with the specified ID.
Connection	Specifies a hashtable holding information used to connect to HaloITSM. This parameter is mandatory.
IncludeInactive	Specifies whether to include inactive tickets. The default is false.
OrganizationId	Filter by tickets belonging to the organization with the specified ID.
SiteId	Filter by tickets belonging to the site with the specified ID.
TeamId	Filter by tickets belonging to the team with the specified ID.
UserId	Filter by tickets belonging to the user with the specified ID.
StatusId	Filter by tickets having the status with the specified ID

Filters

This activity includes the following filter options when searching by **Filters**.

ClientName	Filter tickets by client name
DateAssigned	Filter tickets by the date that they were assigned
DateColosed	Filter tickets by the date that they were closed
DateReported	Filter tickets by the date they were reported
Details	Filter tickets by their details
IsFlagged	Filter tickets by whether they have been flagged.
SiteName	Filter tickets by site name
Summary	Filter tickets by their summary
TeamN*ame*	Filter tickets by team name
Username	Filter tickets by username.

Output

This activity returns objects that have information about HaloITSM tickets. Object properties are decided by the ticket's ticket type.

Get-HaloItsmUser

The **Get-HaloItsmAsset** activity is used in a runbook to query and retrieve User records from HaloITSM.

Discovery Options

This activity includes the following smart discovery options:

Connection	Select the name of the Smart Connection used to connect Runbook Studio to HaloITSM.
Search By	Select whether you want to retrieve agents by their **User ID** or by using **Filters**.

Parameters

This activity requires the following parameters.

Connection	Specifies a hashtable holding information used to connect to HaloITSM.
UserId	Specifies the IDs of the user record to return.

This activity includes the following parameters when searching by **Filters**.

ClientId	Filter by users that are associated with the client with the specified ID.
Connection	Specifies a hashtable holding information used to connect to HaloITSM. This parameter is mandatory.
DepartmentId	Filter by users belonging to the department with the specified ID.
Descending	Indicates whether to return the records in descending order. The default is ascending order.
IncludeInactive	Specifies whether to include inactive tickets. The default is false.
Limit	The maximum number of records to get. The default is fifty.
OrderBy	Specifies the name of the property used to order the returned records.
SiteId	Filter by users belonging to the site with the specified ID.

Filters

This activity includes the following filter options when searching by **Filters**.

ClientName	Filter by users that satisfy the specified client name criteria.
Email	Filter by users that satisfy the specified email address criteria.
FirstName	Filter by users that satisfy the specified first name criteria.
LastName	Filter by users that satisfy the specified last name criteria.
Name	Filter by users that satisfy the specified name criteria.

SiteName	Filter by users that satisfy the specified site name criteria.
Username	Filter by users that satisfy the specified last username criteria.

Output

This activity returns objects that have information about HaloITSM users.

New-HaloItsmTicket

The **Get-HaloItsmAsset** activity is typically used in a runbook to query and retrieve Asset records from HaloITSM.

Discovery Options

This activity includes the following smart discovery options:

Connection	Select the name of the Smart Connection used to connect Runbook Studio to HaloITSM.
ITIL Ticket Type	Select the ITIL ticket type. Options include Incident, Change Request, Service Request, Problem and Advice/Other.
Ticket Type	Select the ticket type. Included options depend on the ITIL Ticket Type that was selected. For example, if Incident is selected then the included options are Incident, Major Incident, Technical Task, Business Task, and Microsoft Teams Task.

Parameters

This activity includes the following mandatory parameters. Other activity parameters, which may be mandatory, are included during discovery based on the selected ticket type and how your HaloITSM instance is configured.

Connection	Specifies a hashtable holding information used to connect to HaloITSM.

Output

This activity returns an object that stands for the ticket that was created. The object's properties are decided by the **Ticket Type** that was selected.

Remove-HaloItsmAttachment

The **Remove-HaloItsmAttachment** activity is used in a runbook to remove an attachment record from HaloITSM.

Discovery Options

This activity includes the following smart discovery options:

Connection	Select the name of the Smart Connection used to connect Runbook Studio to HaloITSM.

Parameters

This activity provides the following parameters.

AttachmentId	Specifies the ID of the attachment record(s) to remove. This parameter is mandatory.
Connection	Specifies a hashtable holding information used to connect to HaloITSM. This parameter is mandatory.

Output

This activity does not return any output.

Remove-HaloItsmTicket

The **Remove-HaloItsmTicket** activity is used in a runbook to remove a ticket record from HaloITSM.

Discovery Options

This activity includes the following smart discovery options:

Connection	Select the name of the Smart Connection used to connect Runbook Studio to HaloITSM.

Parameters

This activity provides the following parameters.

Connection	Specifies a hashtable holding information used to connect to HaloITSM. This parameter is mandatory.
Reason	Indicates why the ticket is being removed. This parameter is optional.
TicketId	Specifies the ID of the ticket record(s) to remove. This parameter is mandatory.

Output

This activity does not return any output.

Set-HaloItsmTicket

The **Get-HaloItsmAsset** activity is typically used in a runbook to change a ticket in HaloITSM.

Discovery Options

This activity includes the following smart discovery options:

Connection	Select the name of the Smart Connection used to connect Runbook Studio to HaloITSM.
ITIL Ticket Type	Select the ITIL ticket type. Options include **Incident**, **Change Request**, **Service Request**, **Problem** and **Advice/Other**.
Search By	Select whether you want to retrieve agents by their **Agent ID** or by using **Filters**.
Ticket Type	Select the ticket type. Included options depend on the ITIL Ticket Type that was selected. For example, if Incident is selected then the included options are Incident, Major Incident, Technical Task, Business Task, and Microsoft Teams Task.

Parameters

This activity requires the following parameters Additional parameters will be provided based on the **Ticket Type** that was selected.

Connection	Specifies a hashtable holding information used to connect to HaloITSM. This parameter is mandatory.
TicketId	Specifies the ID of the ticket(s) to update. This parameter is mandatory.

Output

This activity returns an object that represents the ticket that was updated. The object's properties are decided by the **Ticket Type** that was selected.

Notes