



INTEGRATION MODULE FOR FRESHSERVICE

For Keverion Runbook Studio and Azure Automation

Release Notes

Version 2.0

January 2024

Microsoft
Azure

Certified

Kelverion Integration Module for Freshservice

Copyright © Kelverion Automation Limited. All rights reserved.

Published: January 2024

The Kelverion Integration Module for Freshservice is Microsoft Azure Certified

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	6
Smart Connections.....	7
Global Connection Assets.....	7
Obtaining a Freshservice API Key.....	7
Freshservice API Rate Limit	9
Working With Activities in Runbook Studio	10
Activity Properties	10
Discovery	11
Parameters	11
Retry Behavior	13
Additional Parameters.....	13
Add-FreshserviceTicketNote	15
Get-FreshserviceAsset.....	16
Get-FreshserviceAgent.....	17
Get-FreshserviceDepartment.....	19
Get-FreshserviceRequester	20
Get-FreshserviceTicket.....	22
Get-FreshserviceTicketNote.....	24
New-FreshserviceTicket	25
Remove-FreshserviceTicket	27
Set-FreshserviceTicket	28

Getting Started

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

System Requirements

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

- Keverion Runbook Studio 5.6
- Freshservice cloud-based subscription with access to Freshservice REST API
- Microsoft .NET Framework 4.6.2

Deploying the Integration Module

The easiest way to install and deploy the Integration Module for Freshservice 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 Microsoft SQL Server 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:

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

```
Install-Module -Name Keverion.Freshservice -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:

1. Copy the **Keverion.Freshservice.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.Freshservice.zip** file.
5. Copy the **Keverion.Freshservice** 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.Freshservice.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 *Keverion Runbook Studio* and *Automation Connection Assets*.

Register an Integration Module license with Runbook Studio:

1. Open **Keverion Runbook Studio**.
2. In 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 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 that 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 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 click **Update License Keys**. A summary is displayed.

Smart Connections

The activities in the Keverion Integration Module for Freshservice require connection information to connect to instances of Freshservice.

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 the name for the connection.
4. In the **Connection Type** box, select **Keverion.Freshservice**.
5. In the **ServerUrl** box, enter the URL to your Freshservice site (e.g., <https://dev.freshservice.com>)
6. In the **ApiKey** box, enter the API key for the specified user. For details on how to obtain the API key, please see [Obtaining User API Key](#).
7. Click **OK** to close the New Connection dialog box.

Global Connection Assets

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.

Adding a global connection asset to your Azure Automation Account:

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 the name for the connection.
4. In the **Connection Type** box, select **Keverion.Freshservice**.
5. In the **ServerUrl** box, enter the URL to your Freshservice site (e.g., <https://dev.freshservice.com>)
6. In the **ApiKey** box, enter the API key for the specified user. For details on how to obtain the API key, please see [Obtaining User API Key](#).
7. Click **OK** to close the New Connection dialog box.

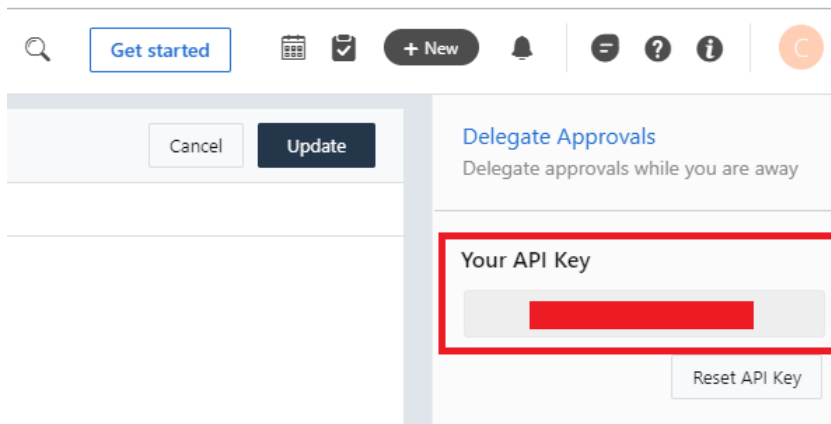
Obtaining a Freshservice API Key

Using API Keys are the recommended way to connect to Freshservice.

To obtain your user API key:

1. Login into your Freshservice Support Portal.
2. Click on your profile picture in the top right corner.
3. Go to Profile settings Page.

4. Your API key will be available below the change password section to your right.



Freshservice API Rate Limit

When using the Freshservice REST API, the number of API calls per hour is restricted to one thousand. The Keverion Integration Module is subject to this restriction, and you will have to take this limitation into account when designing your runbooks. Integration module activities will start failing after the number API requests per hour has reached the limit. The error message will indicate how long you must wait until the Freshservice REST service is accepting requests again:

“You have exceeded the limit of requests per hour. Retry after 3123 seconds.”

Working With Activities in Runbook Studio

The Keverion Integration Module for Freshservice provides the following activities.

Activity	Description
Add-FreshserviceTicketNote	Add a note to an existing ticket.
Get-FreshserviceAgent	Retrieves and filters agent users.
Get-FreshserviceAsset	Retrieves and filters assets.
Get-FreshserviceDepartment	Retrieves and filters departments.
Get-FreshserviceRequester	Retrieves and filters requester users.
Get-FreshserviceTicket	Retrieves and filters tickets.
Get-FreshserviceTicketNote	Retrieves ticket notes.
New-FreshserviceTicket	Creates a new ticket.
Remove-FreshserviceTicket	Removes a ticket.
Set-FreshserviceTicket	Updates an existing ticket.

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 Freshservice.

Activity Properties

All activities in the Keverion Integration Module for Freshservice 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 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 <i>Add-AzureRMAccount</i> in case the runbook is suspended and restarts from this checkpoint on a different worker.

Discovery

When designing runbooks in Keverion Runbook Studio, you will notice that the activities in the Keverion Integration Module for Freshservice include a **Discovery** panel instead of the **Parameter Sets** panel that is present for standard command activities.

All activities in the Keverion Integration Module for Freshservice have a **Connection** option on the **Discovery** panel, which lets you specify a smart connection, so that Runbook Studio can connect to Freshservice and retrieve object and field information.

Parameters

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

For example, when using the **Get-FreshserviceTicket** activity, the Discovery panel will contain **Search By** options for specifying how tickets will be retrieved. If you select the **By ID**, for example, Runbook studio will provide a parameter for specifying a ticket ID. If you select **By Filters**, it will provide a parameter for specifying Freshservice filter name.

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.

All activities in the Keverion Integration Module for Freshservice include a **Connection** parameter. This should not be confused with the similarly named Connection property on the Discovery panel. The former specifies runtime connection information, which the activity uses when it executes as part of a runbook, in Azure or on the Hybrid Worker. The latter specifies design-time connection information, so that Runbook Studio can access Freshservice at design-time and retrieve object and field information.

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.

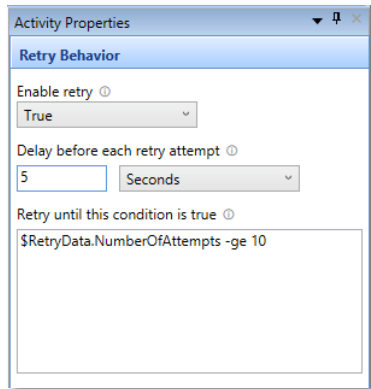
Runbook studio supports the following data sources.

Data Source	Description
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

	<p>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>
Credential asset	<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 Account on the toolbar, the data source will provide the names of the credentials that are available.</p>
Constant	<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"> • String • DateTime • Boolean • Char • Byte • SByte • Int16 • Int32 • Int64 • UInt16 • UInt32 • UInt64 • Decimal • Double • Float • SwitchParameter <p>When assigning a constant DateTime value, Runbook Studio assumes the value is in UTC.</p>
Connection asset	<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>
Empty string	<p>An empty string will be assigned to the parameter. Available when the parameter is type <i>System.String</i></p>
Null	<p>A null (\$null) value will be assigned to the parameter. Available when the parameter type is a reference type.</p>
PowerShell expression	<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>
Runbook input	<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>
Variable asset	<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>

Retry Behavior

The activities in the Kelverion Integration Module for Freshservice can be configured to run multiple times until a particular condition, which you specify, is satisfied. You can use the retry behavior options to configure activities that should run multiple times, which 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 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.

Property	Description
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 Freshservice 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-FreshserviceTicketNote

The **Add-FreshserviceTicketNote** activity can be used in a runbook to add a note to an existing Freshservice ticket.

Discovery Parameters

You can use the following discovery options to connect to Freshservice and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to Freshservice.
With Attachment	Specifies if the note should include an attachment.

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
Body	The HTML contained in the note.
TicketId	The ID of the ticket that the note will be added to.

Optional Parameters

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

AttachmentContent	The Attachment to be added with the ticket note.
AttachmentFileName	The name of the Attachment file, when adding an attachment with the note.
UserId	The User ID of the requester.
BodyText	The body text for the note.
Private	Specifies if the note is private or public.
Incoming	Specifies if the note should appear to come from outside (i.e., not from the web portal).

Outputs

The activity returns the ID for the new ticket note.

Get-FreshserviceAsset

The **Get-FreshserviceAsset** activity can be used in a runbook to retrieve and filter assets.

Discovery Parameters

You can use the following discovery options to connect to Freshservice and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to Freshservice.
Search By	Specifies how assets will be retrieved and filtered. <ul style="list-style-type: none">• All Assets: retrieve all assets.• Display ID: retrieve an asset by Display ID.• Asset Name: retrieve an asset by name.

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
DisplayId	Specifies the Display ID for an asset to be retrieved.
Name	Specifies the name for asset to be retrieved.

Outputs

This activity returns objects that contain information about an asset. Each object has the following properties.

Tag	The asset's Tag.
AssetType	The assets Type.
Description	The asset's description.
DisplayId	The asset's display ID.
ID	The asset's unique ID.
Name	The asset's name.

Get-FreshserviceAgent

The **Get-FreshserviceAgent** activity can be used in a runbook to retrieve and filter agent users.

Discovery Parameters

You can use the following discovery options to connect to Freshservice and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to Freshservice.
Search By	Specifies how agents will be retrieved and filtered. <ul style="list-style-type: none">• All Agents: retrieve all agents.• Email: retrieve an agent by email.• Agent ID: retrieve an agent by ID.• Phone: retrieve an agent by phone.• State: retrieve agents by state.

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
Email	Specifies the email for an agent to be retrieved.
AgentId	Specifies the ID for an agent to be retrieved.
Phone	Specifies the work phone for an agent to be retrieved.
State	Specifies the state for agents to be retrieved.

Outputs

This activity returns objects that contain information about an agent. Each object has the following properties.

Active	Indicates if the user is active.
Address	The agent's physical address.
AgentId	The agent's unique identifier.
Created	The date/time that the agent was created.
Email	The agent's email address.
ExternalId	The agent's external identifier.
FirstName	The agent's first name.
JobTitle	The agent's job title.

Language	The agent's preferred language.
LastName	The agent's last name.
MobilePhone	The agent's mobile phone number.
Occasional	Indicates whether an occasional agent.
TimeZone	The agent's time zone.
Updated	The date/time that the agent was last updated.
WorkPhone	The agent's work phone number.

Get-FreshserviceDepartment

The **Get-FreshserviceDepartment** activity can be used in a runbook to retrieve and filter departments.

Discovery Parameters

You can use the following discovery options to connect to Freshservice and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to Freshservice.
Search By	Specifies how agents will be retrieved and filtered. <ul style="list-style-type: none">• All Departments: retrieve all departments.• Department ID: retrieve a department by Department ID.• Department Name: retrieve a department by name.

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
DepartmentId	Specifies the Department ID for a department to be retrieved.
Name	Specifies the name for department to be retrieved.

Outputs

This activity returns objects that contain information about a department. Each object has the following properties.

Description	The department's description.
ID	The department's unique ID.
Name	The department's name.

Get-FreshserviceRequester

The **Get-FreshserviceRequester** activity can be used in a runbook to retrieve and filter requester users.

Discovery Parameters

You can use the following discovery options to connect to Freshservice and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to Freshservice.
Search By	Specifies how agents will be retrieved and filtered. <ul style="list-style-type: none">• All Requesters: retrieve all requesters.• Email: retrieve a requester by email.• Requester ID: retrieve a requester by ID.• Mobile Phon: retrieve a requester by mobile phone.• Phone: retrieve a requester by phone.• State: retrieve requesters by state.

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
Email	Specifies the email for a Requester to be retrieved.
RequesterId	Specifies the ID for a Requester to be retrieved.
MobilePhone	Specifies the mobile phone for a Requester to be retrieved.
Phone	Specifies the phone for a Requester to be retrieved.
State	Specifies the state for Requesters to be retrieved.

Outputs

This activity returns objects that contain information about a requestor. Each object has the following properties.

Active	Indicates if the user is active.
Address	The requestor's physical address.
Created	The date and time that the requestor was created.
Email	The requestor's email address.
ExternalId	The requestor's external identifier.

FirstName	The agent's first name.
JobTitle	The requestor's job title.
Language	The requestor's preferred language.
LastName	The agent's last name.
MobilePhone	The requestor's mobile phone number.
TimeZone	The requestor's time zone.
Updated	The date and time that the requestor was last updated.
RequesterId	The requestor's User ID.
WorkPhone	The requestor's work phone number.

Get-FreshserviceTicket

The **Get-FreshserviceTicket** activity can be used in a runbook to retrieve and filter tickets.

Discovery Parameters

You can use the following discovery options to connect to Freshservice and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to Freshservice.
Search By	Specifies how tickets will be retrieved and filtered. <ul style="list-style-type: none">• All Tickets: retrieves all tickets.• Freshservice Filter: filter tickets by using a pre-defined filter.• Requester Email: filter tickets by requester email.• Requester ID: filter tickets by requester ID.• Ticket ID: retrieve a specific ticket.
Workspace	Specifies the workspace for which tickets should be retrieved. If (None) is specified, the primary workspace is used.

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
Filter	Specifies a pre-defined Freshservice filter to be used when retrieving tickets. Available values are: <ul style="list-style-type: none">• Deleted• New & My Open Tickets• Spam• Watching
Email	Specifies a requester for which tickets are to be retrieved.
RequesterId	Specifies a requester for which tickets are to be retrieved.
TicketId	Specifies a Freshservice account specific ticket identifier. The activity will retrieve the specified ticket.

Optional Parameters

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

IncludeAssets	When retrieving a ticket by ID, specifies that ticket assets should also be returned.
----------------------	---

Outputs

This activity returns objects that contain information about a ticket. Each object has the following properties.

Assets	The assets associated with the ticket.
Category	The ticket category.
Created	The date and time that the ticket was created.
Deleted	True if the ticket is deleted/trashed.
DepartmentId	The department ID for the ticket.
Description	The HTML content of the ticket.
DescriptionText	The ticket description.
DueBy	The date and time that the ticket is due.
GroupId	The Group ID assigned to the ticket.
ID	The ticket's unique ID.
Impact	The impact of the ticket.
IsEscalated	Indicates whether the ticket has been escalated.
ItemCategory	The ticket item category.
Priority	The priority of the ticket.
RequesterId	The User ID of the ticket requester.
ResponderId	The User ID of the ticket responder (Agent).
Source	The channel through which the ticket was created.
Spam	Indicates whether the ticket is spam.
Status	The status of the ticket.
SubCategory	The ticket sub-category.
Subject	The ticket subject.
TicketType	The Type of ticket.
Updated	The date and time that the ticket record was last modified.
Urgency	The Urgency of the ticket.
WorkspaceId	The workspace ID of the ticket.

Get-FreshserviceTicketNote

The **Get-FreshserviceTicketNote** activity can be used in a runbook to retrieve notes associated with a specific ticket.

Discovery Parameters

You can use the following discovery options to connect to Freshservice and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to Freshservice.
-------------------	---

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
TicketId	Identifies the ticket for which notes are to be retrieved.

Outputs

This activity returns objects that contain information about a ticket note. Each object has the following properties.

ID	The ticket note's unique ID
NoteText	The note's text
NoteTextHtml	The note's HTML text
Private	Indicates whether the note is private.
Incoming	Indicates whether the note should appear as created from the outside.
UserId	The ID of the user who created the note.
NotifyEmails	Email addresses of agents/users who need to be notified about this note.

New-FreshserviceTicket

The **New-FreshserviceTicket** activity can be used in a runbook to create a new Freshservice ticket.

Discovery Parameters

You can use the following discovery options to connect to Freshservice and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to Freshservice.
Workspace	The Workspace where the ticket should be created. If (None) is specified, the primary workspace is used.

Required Parameters

You must configure the following parameters. If you have specified any required custom parameters, these must also be configured.

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
Description	The Ticket description.
RequesterId	The requestor's User ID. Use the UserId value returned from the Get-FreshserviceRequester activity.
Status	The status of the ticket. <ul style="list-style-type: none">• Closed• Open• Pending• Resolved
Subject	The Ticket subject.

Optional Parameters

You can use the following optional parameters to control the behavior of the activity, as necessary. If you have specified any optional custom parameters, these will also be provided.

DepartmentId	The department ID for the ticket.
DescriptionHTML	The HTML content of the ticket. Description and Description HTML should not be passed together.
GroupId	The agent Group to which the ticket is assigned.
Impact	The Impact of the ticket. Affects ticket Priority when Priority Matrix is enabled.
IsEscalated	Specifies whether the ticket is escalated.
ItemCategory	The Ticket item category.

Priority	The Priority of the ticket. Determined by Impact and Urgency when Priority Matrix is enabled.
ResponderId	The agent's User ID. Use the UserId value returned from the Get-The FreshserviceAgent activity.
Source	The channel through which the ticket is created.
Spam	Specifies whether the ticket is spam.
SubCategory	The ticket sub-category.
TicketType	The type of ticket.
Urgency	The urgency of the ticket. Affects ticket Priority when Priority Matrix is enabled.

Outputs

The activity returns the Ticket Display ID for the newly created ticket.

Remove-FreshserviceTicket

The **Remove-FreshserviceTicket** activity can be used in a runbook to remove an existing ticket.

Discovery Parameters

You can use the following discovery options to connect to Freshservice and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to Freshservice.
-------------------	---

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
TicketId	The ticket ID of the ticket to be removed.

Outputs

The activity returns the Ticket Display ID for the deleted ticket. The ID is no longer valid.

Set-FreshserviceTicket

The **Set-FreshserviceTicket** activity can be used in a runbook to update an existing Freshservice ticket. The following tables list the parameters for this activity.

Discovery Parameters

You can use the following discovery options to connect to Freshservice and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to Freshservice.
Workspace	The Workspace of the ticket. If (None) is specified than the primary workspace is used.

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
TicketId	The ID of the ticket to be updated.

Optional Parameters

You can use the following optional parameters to control the behavior of the activity, as necessary. If you have specified any optional custom parameters, these will also be provided.

AssetDisplayId	The Display ID of an asset to be associated with the ticket.
Category	The ticket category.
DepartmentId	The department ID for the ticket.
GroupId	The agent group to which the ticket is assigned.
Impact	The impact of the ticket. Affects ticket Priority when Priority Matrix is enabled.
IsEscalated	Specifies whether to escalate the ticket.
ItemCategory	The ticket's item category.
Priority	The priority of the ticket. Determined by Impact and Urgency when Priority Matrix is enabled.
RequesterId	The User ID of the Requester. Use the UserId value returned from the Get-FreshserviceRequester activity.
ResponderId	The User ID of the Agent that responded to the ticket. Use the UserId value returned from the Get-FreshserviceAgent activity.
Source	The channel through which the ticket is created.

Spam	Specifies whether the ticket is spam.
Status	The status of the ticket.
SubCategory	The ticket's sub-category.
Subject	The subject of the ticket.
TicketType	The type of ticket.
Urgency	The urgency of the ticket. Affects ticket Priority when Priority Matrix is enabled.

Outputs

The activity returns the Ticket Display ID for the updated ticket.