



INTEGRATION MODULE FOR SOLARWINDS SERVICE DESK

For Keverion Runbook Studio and Azure Automation

User Guide

Version 1.2

December 2023

Microsoft
Azure

Certified

Kelverion Integration Module for SolarWinds Service Desk

Copyright © Kelverion Automation Ltd. All rights reserved.

Published: December 2023

Feedback

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

Contents

Installation and Configuration	4
System Requirements.....	4
Installing the Integration Module from PowerShell Gallery.....	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
Common Properties	8
Smart Discovery.....	9
Smart Parameters.....	9
Smart Filters	11
Retry Behavior	11
Additional Parameters.....	12
Custom Fields	12
Service Request Variables	13
Get-SolarWindsComment	14
Get-SolarWindsRecord.....	15
New-SolarWindsComment.....	16
New-SolarWindsRecord	17
Remove-SolarWindsRecord	18
Set-SolarWindsRecord	19

Installation and Configuration

The following sections outline how to deploy and configure the Keverion Integration Module for SolarWinds Service Desk.

System Requirements

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

- Keverion Runbook Studio 5.6.1
- Microsoft .NET Framework 4.7.2
- Azure Automation account
- SolarWinds Service Desk September 2023 (Professional or Enterprise)

Installing the Integration Module from PowerShell Gallery

The easiest way to install and deploy the Integration Module for Microsoft SQL Server 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 you the PowerShellGet module installed.
2. Start a PowerShell window as Administrator and run the command:
Install-Module -Name Keverion.SolarWinds.ServiceDesk -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.SolarWinds.ServiceDesk.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.SolarWinds.ServiceDesk.zip** file.
5. Copy the **Keverion.SolarWinds.ServiceDesk** 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.SolarWinds.ServiceDesk.zip** file that you downloaded.
- Click **OK**.

Licensing the Integration Module

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

Register an Integration Module license with Runbook Studio:

1. Open **Keverion 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 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 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.

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.

Working With Activities in Runbook Studio

The following sections describe how to configure the activities in the Keverion Integration Module for SolarWinds Service Desk in conjunction with Keverion Runbook Studio.

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.

The integration module includes the following activities:

Get-SolarWindsComment	Retrieves comment(s).
Get-SolarWindsRecord	Retrieves record(s).
New-SolarWindsComment	Creates a new comment.
New-SolarWindsRecord	Creates a new record.
Remove-SolarWindsRecord	Deletes a record.
Set-SolarWindsRecord	Modifies an existing record.

Smart Connections

In Keverion Runbook Studio you can configure one or more Smart Connections in order to establish reusable links between Runbook Studio and a specific SolarWinds Service Desk 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.

Note: It is highly recommended to use a dedicated SolarWinds Service Desk user for your smart connection, to ensure proper runbook operation during design and runtime.

Adding a Smart Connection to Keverion Runbook Studio:

1. Click **Connections** in the Runbook Studio toolbar.
2. In the **Smart Connections** dialog click **Add**.
3. In the **Name** box, enter a name for the configuration. This could be the name of the instance or a descriptive name to distinguish the type of configuration.
4. In the optional **Description** box, enter a description of the Smart Connection.
5. In the **Connection Type** box, select **Keverion.SolarWinds.ServiceDesk**.
6. In the **ServiceUrl** box, enter the URL to the SolarWinds Service Desk REST API service. Default value is **https://api.samanage.com**.
7. In the **Token** box, enter the user token that the integration module will use to connect to the SolarWinds Service Desk instance.

8. Click **OK**.

Global Connection Assets

The activities in the Kelverion Integration Module for SolarWinds Service Desk require connection information to connect to instances of SolarWinds Service Desk.

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.

Note: It is highly recommended to use a dedicated SolarWinds Service Desk user for your smart connection, to ensure proper runbook operation during design and runtime.

Adding a global connection asset to your Azure Automation Account:

2. In Kelverion Runbook Studio, click the **Azure** panel.
3. Select your Azure subscription.
4. Select your Automation account.
5. Select **Connections** and right click.
6. Select **Add New Connection**.
7. In the **Name** box, enter a name for the configuration. This could be the name of the instance or a descriptive name to distinguish the type of configuration.
8. In the optional **Description** box, enter a brief description describing the connection.
9. In the **Connection Type** box, select **Kelverion.SolarWinds.ServiceDesk**.
10. In the **ServiceUrl** box, enter the URL to the SolarWinds Service Desk REST API service. Default value is **https://api.samanage.com**.
11. In the **Token** box, enter the user token that the integration module will use to connect to the SolarWinds Service Desk instance.
12. Click **OK**.

Common Properties

All activities in the Kelverion Integration Module for SolarWinds Service Desk 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 great way to let everyone understand the function of the activity in the runbook.
Checkpoint	Indicates whether or not 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 Keverion Runbook Studio, you will notice that the activities in the Keverion Integration Module for SolarWinds Service Desk 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 SolarWinds Service Desk support interactive discovery of the SolarWinds Service Desk assets in your environments.

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

When connected to SolarWinds Service Desk, Runbook Studio will provide additional discovery options for some activities, **Record 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 SolarWinds Service Desk are determined by the Discovery options that you specify.

For example, when using the **Get-SolarWindsRecord** activity, the Discovery panel will contain options for selecting a SolarWinds Service Desk record type. Once you have selected a record type, Runbook Studio will provide you with parameters that coincide with the fields in the business object. If you select another business object, Runbook Studio will provide you with a different set of parameters.

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 SolarWinds Service Desk include a **Connection** parameter which is used to specify information that the activity will use to connect to SolarWinds Service Desk 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 SolarWinds Service Desk 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.

Activity output	<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>
Not configured	<p>Clears any value that was previously configured. You must configure all mandatory parameters.</p>
Certificate asset	<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>
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 • Timespan • Decimal • Double <p>When assigning a constant DateTime and Time values, 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 a System.String instance</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>

	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 of the activities in the Kolverion Integration Module for SolarWinds Service Desk include a **Filters** panel which lets you specify filters that can be used to retrieve specific issues in SolarWinds Service Desk.

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. Possible filter operators include: <ul style="list-style-type: none"> • Equals • Does not equal • Is less than • Is less than or equal to • Contains • Does not contain • Is greater than • Is greater than or equal to
Value	The data source used to retrieve the value 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 Kolverion Integration Module for SolarWinds Service Desk 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 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 SolarWinds Service Desk 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**

Custom Fields

The Kelverion Integration Module for SolarWinds Service Desk supports the following Service Desk custom field types. Note that Service Desk custom fields must be included in Service Desk custom forms to be available in the integration module activities.

Checkbox	Boolean type field.
-----------------	---------------------

Date	Date field.
Date and Time	DateTime field.
Dropdown	String type field with dropdown selection.
Multi picklist	Dropdown with multiple selection. Note: To specify multiple values, separate the values by the '#dlmtr#' string: Value1#dlmtr#Value2#dlmtr#Value3.
Text	String type field.
Text Area	Multiline string type field.
User	SolarWinds Service Desk user. Note: To specify a user, use the GroupUserId field returned by the Get-SolarWindsRecord activity, with record type User and record ID specified.

Service Request Variables

The Kelverion Integration Module for SolarWinds Service Desk supports the following service request variable types. Service request variables must first be defined in the Service Request Catalog specified when creating a new request.

Checkbox	Boolean type field.
Date	Date field.
Dropdown	String type field with dropdown selection.
Dependent Dropdown	Dropdown that depends on another dropdown selection.
Multi-picklist	Dropdown with multiple selection. Note: To specify multiple values, separate the values by the '#dlmtr#' string: Value1#dlmtr#Value2#dlmtr#Value3.
Text	String type field.
User	SolarWinds Service Desk user. Note: To specify a user, you can use the user email address, or you can use the GroupUserId field returned by the Get-SolarWindsRecord activity, with record type User and record ID specified.

Get-SolarWindsComment

The **Get-SolarWindsComment** activity retrieves comments for a specified record (incident or problem).

Discovery Options

This activity provides the following smart discovery options, which can be used to connect Runbook Studio to your SolarWinds Service Desk instance and discover the ITSM resources that are available.

Connection	The name of the Smart Connection used to connect Runbook Studio to SolarWinds Service Desk.
Record Type	The type of record for which comments will be retrieved.
Search By	The method used to retrieve the records. Select All Comments to retrieve all comments for specified record. Select Comment ID to get a specific comment.

Required Parameters

You must configure the following parameters.

Connection	Hashtable containing information used to connect to SolarWinds Service Desk.
RecordId	The ID of the record for which comments will be retrieved.
CommentId	The ID of a specific comment, when searching by comment ID.

Output

This activity returns **PSObject** objects that represent the comments that were retrieved.

Get-SolarWindsRecord

The **Get-SolarWindsRecord** activity retrieves SolarWinds Service Desk records using criteria that you specify.

Discovery Options

This activity provides the following smart discovery options, which can be used to connect Runbook Studio to your SolarWinds Service Desk instance and discover the ITSM resources that are available.

Connection	The name of the Smart Connection used to connect Runbook Studio to SolarWinds Service Desk.
Record Type	The type of record to retrieve.
Search By	The method used to retrieve the records. Select Filters to use filters, Record ID to get by record ID.

Required Parameters

You must configure the following parameters.

Connection	Hashtable containing information used to connect to SolarWinds Service Desk.
RecordId	The ID of the record to retrieve when searching by Record ID.

Optional Parameters

This activity provides the following optional parameters that may be configured as required.

Ascending	Indicates that records should be sorted in ascending order. This parameter is used only when Order By is specified.
Limit	Specifies the maximum number of records to be returned.
Order By	The field used to order the records by.

Filters

This activity provides filters based on the **Record Type** that is specified.

Output

This activity returns **PSObject** objects that represent the ITSM records that were retrieved. The **Record Type** you selected during discovery will determine the object's properties.

New-SolarWindsComment

The **New-SolarWindsComment** activity adds a new SolarWinds Service Desk comment for the specified record type.

Discovery Options

This activity provides the following smart discovery options, which can be used to connect Runbook Studio to your SolarWinds Service Desk instance and discover the ITSM resources that are available.

Connection	The name of the Smart Connection used to connect Runbook Studio to SolarWinds Service Desk.
Record Type	The type of record to which the comment will be added.

Required Parameters

You must configure the following parameters. The activity provides additional required parameters based on the **Record Type** that was selected.

Connection	Hashtable containing information used to connect to SolarWinds Service Desk.
Comment	The comment to be added.
RecordId	The ID of the record to which the comment will be added.

Output

This activity returns the unique **ID** of the comment comment that was created.

New-SolarWindsRecord

The **New-SolarWindsRecord** activity creates a new SolarWinds Service Desk record.

Discovery Options

This activity provides the following smart discovery options, which can be used to connect Runbook Studio to your SolarWinds Service Desk instance and discover the ITSM resources that are available.

Connection	The name of the Smart Connection used to connect Runbook Studio to SolarWinds Service Desk.
Record Type	The type of record to be created.

Required Parameters

You must configure the following parameters. The activity provides additional required parameters based on the **Record Type** that was selected.

Connection	Hashtable containing information used to connect to SolarWinds Service Desk.
-------------------	--

Optional Parameters

The activity will provide optional parameters based on the **Record Type** that was selected.

Output

This activity returns the **ID** of the ITSM record that was created.

Remove-SolarWindsRecord

The **Remove-SolarWindsRecord** activity removes a SolarWinds Service Desk record.

Discovery Options

This activity provides the following smart discovery options, which can be used to connect Runbook Studio to your SolarWinds Service Desk instance and discover the ITSM resources that are available.

Connection	The name of the Smart Connection used to connect Runbook Studio to SolarWinds Service Desk.
Record Type	The type of record to be removed.

Required Parameters

You must configure the following parameters.

Connection	Hashtable containing information used to connect to SolarWinds Service Desk.
RecordId	The ID of the record to remove.

Output

This activity returns the **ID** of the ITSM record that was removed.

Set-SolarWindsRecord

The **Set-SolarWindsRecord** activity modifies an existing record.

Discovery Options

This activity provides the following smart discovery options, which can be used to connect Runbook Studio to your SolarWinds Service Desk instance and discover the ITSM resources that are available.

Connection	The name of the Smart Connection used to connect Runbook Studio to SolarWinds Service Desk.
Record Type	The type of records to be modified.

Required Parameters

You must configure the following parameters.

Connection	Hashtable containing information used to connect to SolarWinds Service Desk.
RecordId	The record ID of the record to modify.

Optional Parameters

The activity will provide optional parameters based on the **Record Type** that was selected.

Output

This activity returns the **ID** of the ITSM record that was modified.