INTEGRATION MODULE FOR SERVICENOW

For Kelverion Runbook Studio and Azure Automation

User Guide

Version 3.5

Microsoft Azure

Certified

Kelverion Integration Module for ServiceNow

Published: April 2024

The Kelverion Integration Module for ServiceNow is Microsoft Azure Certified.

Feedback

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

Contents

Getting Started

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

System Requirements

The Integration Module for ServiceNow requires the following software to be installed and configured prior to implementing the integration.

- Kelverion Runbook Studio 5.6
- Microsoft .NET Framework 4.7.2

The integration module Supports the following versions of ServiceNow:

- Washington
- Vancouver
- Utah

Important: The Kelverion Integration Module for ServiceNow requires the user, that is used to connect to ServiceNow, be configured to use the **English** language.

Changes Affecting Backwards Compatibility

Starting with version 3.0 connection asset and Smart Connection fields have been updated to support OAuth 2.0 authentication. As a result, **connection assets are not backwards compatible with previous versions**.

Before upgrading the module in Azure Automation, you must first remove all Kelverion.ServiceNow connection assets as well as the Kelverion.ServiceNow connection type. For more information, please refer to Appendix A. After you have upgraded the module, you can recreate your connection assets.

Important: Before removing connection assets, you should document the connection asset name(s) and field values so that they can be recreated. Using the same connection asset names, will save you from having to update your runbooks.

You must also update your Smart Connections in Runbook studio as some configuration property names have changed.

Deploying the Integration Module

The easiest way to install and deploy the Integration Module for ServiceNow 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 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 ServiceNow from the PowerShell Gallery and install it on your local computer. You can also deploy the module directly from the PowerShell Gallery to any of your Azure Automation Accounts.

Install the Integration Module on your local computer or hybrid worker:

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

```
Install-Module -Name Kelverion.ServiceNow -Scope AllUsers
```

Upload the Integration Module to an Azure Automation Account:

1. Go to the PowerShell Gallery.
2. On the **Azure Automation** tab, click **Deploy to Azure Automation**.
3. Select the **Automation Account** that you want to deploy the module to and 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.

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

1. Copy the **Kelverion.Servicenow.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.Servicenow.zip** file.
5. Copy the **Kelverion.ServiceNow** 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 **Kelverion.Servicenow.zip** file that you downloaded.
6. Click **OK**. Importing the module may take several minutes.

Licensing the Integration Module

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

Register an Integration Module license with Runbook Studio:

1. Open **Kelverion 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 Automation:

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 asset 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 click **Update License Keys**. A summary is displayed.

Preparing to Connect to ServiceNow

The following sections outline how to configure your ServiceNow instance to work with the activities in the Kelverion Integration Module for ServiceNow.

Security Configuration

To integrate successfully with ServiceNow, the Kelverion Integration Module for ServiceNow requires access to the tables that will be targeted by your runbooks as well as several system tables. System table access is used to retrieve information, such as table and field descriptions, that are used to provide a rich runbook authoring experience.

For non-admin users, access to ServiceNow is configured using user roles and access control rules (ACLs). The following sections outline the process of setting up a dedicated user role and ACLs to enable the Integration Module for ServiceNow to integrate with your ServiceNow environment.

1. Create a dedicated Role for Automation.
2. Create system table Access Control List (ACL) rules.
3. Create a dedicated User for Automation.
4. Enable API Endpoints.
5. Create ACL rules to support your runbooks.

Step 1: Create a Dedicated Role for Automation

Roles are used to control access to features and capabilities in ServiceNow and it is strongly recommended that you create a dedicated role for the integration module and your runbooks to access your ServiceNow environment.

1. Navigate to **User Administration > Roles** and create a new record.
2. In the **Name** field, type a name for the role, such as *auto_admin*.
3. In the **Application** field, select *Global.*
4. Disable the **Elevated privilege** option.
5. In the **Description** field, enter a description of the role.
6. Click **Submit.**

To enable access to the tables that will be targeted by your runbooks it may be necessary to add existing roles, such as the ITIL role, to the role that you created for the integration module.

1. Navigate to **User Administration > Roles** and open the role that you created for the integration module.
2. Click **Edit** in the **Contains Roles** list.
3. Use the slush-bucket to add one or more roles. Note, you must add the ***snc_platform_rest_api*** role.
4. Click **Save**.

The user that you define in Runbook Studio to connect to ServiceNow must use this role and the Access Control List Rules that you will assign to it in the next step.

Step 2: Create System Table ACL Rules

To provide a rich authoring experience, the Integration Module for ServiceNow must be able to retrieve system information from your ServiceNow environment. To enable access for non-admin users it is necessary to create Access Control List (ACL) rules.

The integration module requires ***Table and Field ACLs*** for the following ServiceNow System tables:

- Choice [sys_choice]
- Table [sys_db_object]
- View Table [sys_db_view_table]
- Database View [sys_db_view]
- Dictionary Entry [sys_dictionary]
- Field Label [sys_documentation]
- Field Class [sys_glide_object]
- *Field Map [sys_transform_entry]
- Journal Entry [sys_journal_field]
- *Table Transform Map [sys_transform_map]

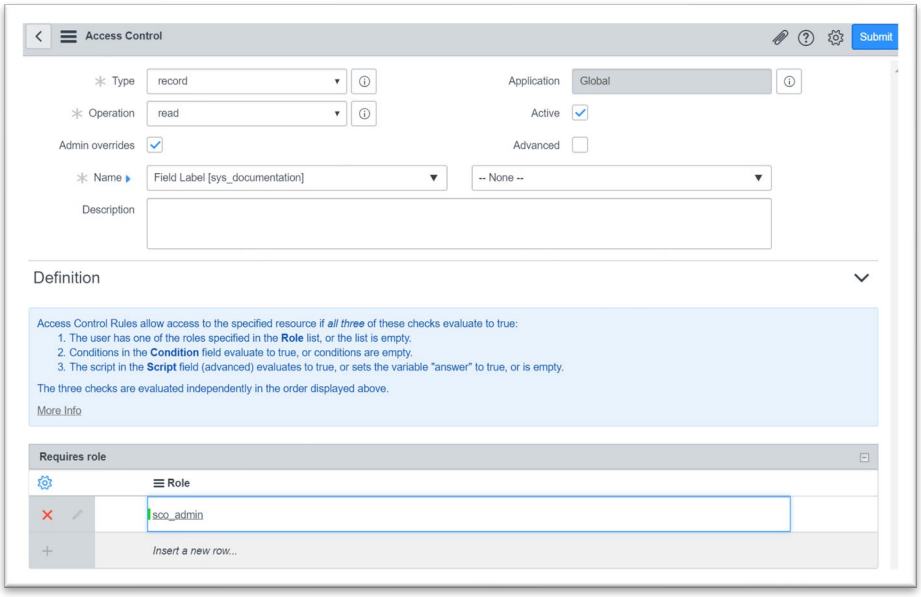
* Only required if you are using the **Import-ServiceNowRecord** activity.

First, you need to create ACLs to let the integration pack access the preceding ServiceNow system tables.

For each table in the preceding list of system tables:

1. Elevate privileges to the *security_admin* role.
2. Navigate to **System Security > Access Control (ACL).**
3. Click **New**.
4. In the **Type** field, select *record*.
5. In the **Operation** field, select *read*.
6. Leave the **Active** and **Admin overrides** fields enabled.
7. In the **Name** field, select that system table that is being secured and in the adjacent field select *None.*
8. Optionally, in the **Description** field, enter a description of the rule.
9. In the **Requires role** section type or enter the role that your created for your integration module in the previous section.
10. Click **Submit**.

When you are finished, the form should look something like this:

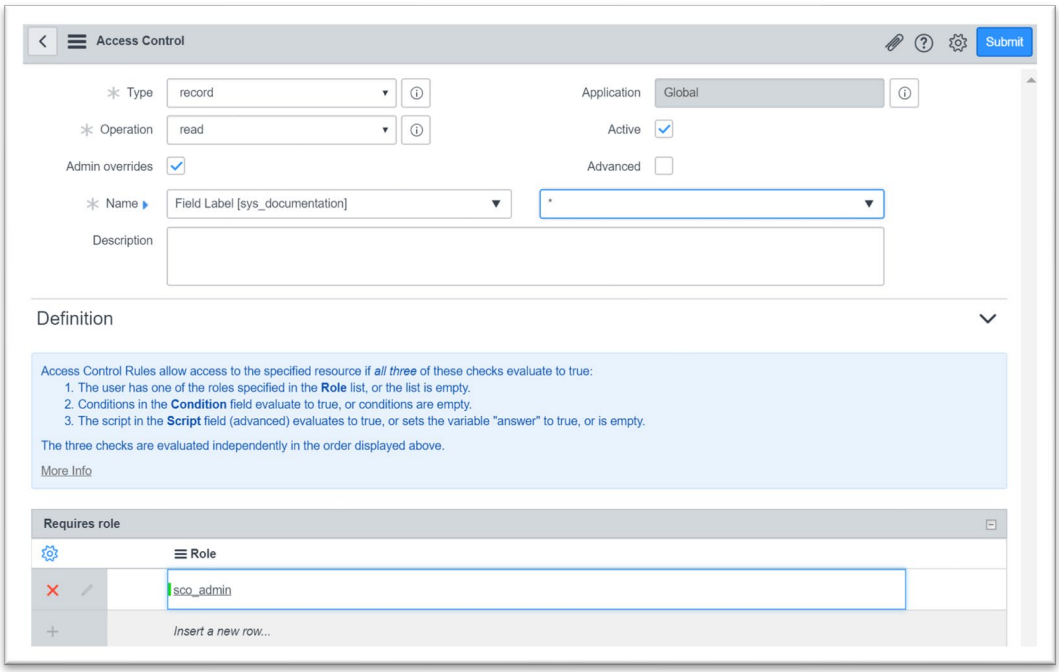


Next, you need to create ACLs to let the integration pack access the **fields** in the preceding ServiceNow system tables.

For each table in the preceding list of system tables, create a read ACL to enable the integration module to access the table.

1. Elevate privileges to the *security_admin* role.
2. Navigate to **System Security > Access Control (ACL).**
3. Click **New**.
4. In the **Type** field, select *record*.
5. In the **Operation** field, select *read*.
6. Leave the **Active** and **Admin overrides** fields enabled.
7. In the **Name** field, select that system table that is being secured and in the adjacent field select *asterisk (*).*
8. Optionally, in the **Description** field, enter a description of the rule.
9. In the **Requires role** section type or enter the role that your created for your integration module in the previous section.
10. Click **Submit**.

When you are finished, the form should look something like this:



Step 3: Create a Dedicated User for Automation

It is recommended that you create a dedicated user to enable the Integration Module for ServiceNow to access your ServiceNow environment. Using a dedicated user will provide the ability to more easily configure the ACL rules and other options and roles that are required by the Integration Module for ServiceNow and the runbooks that your author in Runbook Studio.

1. Navigate to **User Administration > Users**.
2. Click **New**.
3. In the **User ID** field, type a unique identifier.
4. In the **First name** and **Last name** fields type appropriate values.
5. In the **Password** field, type a password.
6. Disable the **Password needs reset** option.
7. Optionally, enable the Web service access only option.
8. In the **Time zone** field, select an appropriate time zone. See the next section regarding working with date and time values in ServiceNow and Runbook Studio.
9. If language settings are enabled, in the **Language** box, select **English**.
10. Fill in additional information, as needed.
11. Click **Submit**.

Assign roles to the new user:

1. Navigate to **User Administration > Users** and open the user that you created in the previous steps.
2. In the **Roles** related list, click **Edit.**
3. In the **Collection** list, select the dedicated Automation role that you created in step 1.
4. Optionally, add additional roles as necessary to support your runbooks.
5. Click **Save**.

Step 4: Enable API Endpoints

The Integration Pack requires that **REST_Endpoint ACLs** be enabled for the following APIs.

- Table API
- Attachment API
- Import Set API

For each API in the preceding list, enable the ACL entry:

1. Elevate privileges to the *security_admin* role.
2. Navigate to **System Security > Access Control (ACL).**
3. Search for the API name.
4. Open the API entry.
5. Select the Active check box.
6. Click **Update.**

Step 5: Create ACL Rules to Support Your Runbooks

It may also be necessary to create additional ACL rules to enable your runbooks to access other tables in your ServiceNow environment. Alternatively, you may be able to assign additional roles to the user that you are using to connect to ServiceNow from Runbook Studio to enable the access that your runbooks require.

Connecting to ServiceNow using OAuth 2.0

The Integration Module for ServiceNow supports basic and OAuth 2.0 authentication. For most scenarios, basic authentication using a username and password is recommended; however, OAuth is provided in those cases where basic authentication has been disabled. For more information, see OAuth 2.0 in the online ServiceNow documentation.

Setup OAuth

To use OAuth 2.0 to connect to ServiceNow, you must first setup and activate the OAuth plugin and create an OAuth application endpoint for the integration module to access the instance.

To Activate OAuth 2.0 in your ServiceNow instance:

1. Login to your ServiceNow instance using a user with the **Admin** role.
2. Navigate to **System Applications** > **All Available Applications** > **All**.

3. Find the plugin titled “OAuth 2.0” using the filter criteria and search bar. You can search for the plugin by its **name** or **ID**. If you cannot find a plugin, you might have to request it from *ServiceNow* personnel. For more information, see Request a plugin.
4. Click **Install**, and then in the Activate Plugin dialog box, click **Activate**.

To Create a Client Endpoint:

1. Login to your ServiceNow instance using a user with the **Admin** role.
2. Navigate to **System OAuth > Application Registry.**
3. Click **New**.
4. Click **Create an OAuth API endpoint for external clients.**
5. In the **Name** box, enter a unique name for the endpoint such as **Kelverion Automation**.
6. In the **Client Secret** box, enter a secret that both the instance and module can use to authorize communication. Leave empty to auto-generate a client secret.
7. In the **Refresh Token Lifespan** field, enter a lifespan for the refresh token. We recommend using a value sufficient to support your automation environment. For example, enter the value 86,400,000 to indicate that the token will expire in 1,000 days.
8. Click **Submit**.

Alternatively, you can use a third-party OAuth provider that provides the authorization to access your instance. See Use a third-party OAuth provider in the online ServiceNow documentation.

Refresh Tokens

Refresh tokens are credentials used by the integration module to obtain access tokens to access your ServiceNow instance. Refresh tokens have a limited lifespan; however, you have the option to override the default lifespan when you setup and configure the application endpoint. To minimize the risk of refresh tokens expiring unexpectedly, we recommend using a lifespan of at least 365 days.

The Integration Module for ServiceNow includes a **Get-ServiceNowRefreshToken** cmdlet to assist you with obtaining refresh tokens for your Runbook Studio smart connections and Azure Automation connection assets. The following PowerShell snippet demonstrates how you can use the activity to request a refresh token.

To create a Refresh Token using PowerShell:

```
$authUrl = 'https://<your instance>.service-now.com/oauth_token.do'
$clientId = 'your client ID'
$clientSecret = 'your client secret'
$credential = Get-Credential

Get-ServiceNowRefreshToken
    -AuthorizationUrl $authUrl
    -ClientId $clientId
    -ClientSecret $clientSecret
    -Credential $credential
```

Working with Date/Time Values

Due to the way that the ServiceNow web service API handles date/time values, the following information should be considered when working the date/time values.

Date/Time Parameters

When configuring runbook activities, it is assumed that the values for all date/time fields are relative to the time zone of the ServiceNow user that owns the record. If an Owner is not explicitly defined, ServiceNow will assume that the owner is the ServiceNow user that was specified in the connection that was used to connect to ServiceNow.

When using strings to define date/time values they should use the ISO 8601 date/time format, regardless of the date and time formats that have been assigned to the ServiceNow user that owns the record. The ISO 8601 date/time format is **yyyy-MM-ddTHH:mm:sszzz**, for example 2012-12-14T15:00:00Z.

To minimize problems with date/time values we suggest that you create a dedicated user in ServiceNow that can be used by your ServiceNow runbooks. This user should be assigned an agreed upon time zone that is appropriate for your runbooks. Choosing GMT as your time-zone should help you avoid most of the confusion associated with time zones, especially when you consider that the date/time editor in Kelverion Runbook Studio assumes dates are in Universal Coordinated Time (which is equivalent to GMT)

TimeSpan Parameters

When specifying values for Parameters that are based on System.TimeSpan we recommend using the PowerShell Expression data source and provide an expression that uses the New-TimeSpan cmdlet. For example, *New-TimeSpan –Days 1 –Hours 10 –Minutes 15*. When providing values for fields that represent time, the number of days in the System.TimeSpan object must be zero.

Date/Time Output

Date/time values retrieved from ServiceNow are published as System.DateTime objects using the time zone of the ServiceNow user that owns the record. If an owner is not explicitly defined, ServiceNow will assume that the owner is the ServiceNow user that was specified in the connection that was used to connect to ServiceNow.

Working with Duration Fields

When providing values for input properties and filters associated with ServiceNow duration fields, you can use a .NET Framework TimeSpan object or a string with the format [days] [hours]:[minutes]:[seconds]. For example:

Input	**Result**
6	6 00:00:00
6:12	06:12:00
6:12:14	6:12:14
10 6:12:14	10 6:12:14

Duration fields are published as a TimeSpan object. ***Note:*** When the duration is greater than 60 seconds, ServiceNow excludes the seconds from the result. For example, a duration of **6:12:30** will be published as **6:12:00**.

ServiceNow Table Cache

To provide users with the ability to connect to any ServiceNow table, the Kelverion Integration Module for ServiceNow builds a cache of ServiceNow table information as required. The process of building a table cache can be time consuming, and this explains the delay that you may experience when connecting to a new table for the first time.

The table cache provides a significant performance benefit, however it can become outdated when changes are made your ServiceNow system, such as adding fields to a ServiceNow table. When this happens, you will have to remove the cached files so that they can be rebuilt.

To remove the ServiceNow Web Reference cache:

1. Open Windows Explorer.
2. Select *C:\ProgramData\Kelverion\ServiceNow.*
3. Delete the folder that matches your ServiceNow host.
4. Repeat for each host system.

Working with Activities in Runbook Studio

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

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.

The integration module includes the following activities:

Add-ServiceNowAttachmentContent	Upload an attachment to a ServiceNow record
Get-ServiceNowAttachmentContent	Download the content of an attachment
Get-ServiceNowAttachmentInfo	Get details of the attachments for a ServiceNow record
Get-ServiceNowRecord	Get details of ServiceNow records
Get-ServiceNowRecordCount	Count ServiceNow records
Get-ServiceNowRefreshToken	Get a refresh token for OAuth 2.0 authentication
Import-ServiceNowRecord	Create or update a ServiceNow record using an import set
New-ServiceNowRecord	Create a new ServiceNow record
Remove-ServiceNowRecord	Remote a ServiceNow record
Set-ServiceNowRecord	Update an existing ServiceNow record

Smart Connections

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

Add a Smart Connection using basic authentication:

1. Open Runbook Studio.
2. On the **Home** tab, click **Smart Connections**.
3. In the Smart Connections dialog, click **Add a connection**. In the Add Smart Connection dialog box, you must configure the following properties.
4. In the **Name** box, enter a name for the connection.
5. In the **Connection Type** box, select *Kelverion.ServiceNow**.***

6. In the **ServiceNowUrl** box, type the URL of the ServiceNow instance you want to connect with.
7. In the **Username** box, type the username of the user that you have configured for use by the integration module.
8. In the **Password** box, type the password of the user that you have configured for use by the integration module.
9. In the **DateFormat** box, type the date format for the user that you configured for use by the integration module. The default is *yyyy-MM-dd*.
10. In the **TimeFormat** box, type the time format for the user that you configured for use by the integration module. The default is *HH:MM:ss*.
11. Click **OK**, and then click **OK** again.

Add a Smart Connection using OAuth2.0:

1. Open Runbook Studio.
2. On the **Home** tab, click **Smart Connections**.
3. In the Smart Connections dialog box, click **Add a connection**.
4. In the **Name** box, enter a name for the connection. In the Add Smart Connection dialog box, you must configure the following properties.
5. In the **Connection Type** box, select *Kelverion.ServiceNow.*
6. In the **ServiceNowUrl** box, type the URL of the ServiceNow instance you want to connect with.
7. In the **AuthorizationUrl** box, type the URL of your authorization server. For example, *https//test123.service-now.com/oath_token.do*.
8. In the **ClientId** box, type the Client ID of the application endpoint that you created.
9. In the **ClientSecret** box, type the Client Secret of the application endpoint that you created.
10. In the **RefreshToken** box, type the Refresh Token that you generated.
11. In the **DateFormat** box, type the date format for the user that you configured for use by the integration module. The default is *yyyy-MM-dd*.
12. In the **TimeFormat** box, type the time format for the user that you configured for use by the integration module. The default is *HH:MM:ss*.
13. Click **OK**, and then click **OK** again.

Global Connection Assets

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

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 using basic authentication:

1. Open Runbook Studio.
2. On the **Home** tab, click **Sign In**.
3. In the Sign In dialog box, sign into your account.
4. In the **Active Azure Automation Account** box, select the account that you want to add the connection asset to.
5. Click **New Asset** and then click **Connection**. In the New Connection dialog box, you must configure the following properties.
6. In the **Name** box, type the name for the configuration.
7. In the **Connection Type** box, select *Kelverion.ServiceNow.*
8. In the **ServiceNowUrl** box, type the URL of the ServiceNow instance you want to connect with.
9. In the **Username** box, type the username of the user that you have configured for use by the integration module.
10. In the **Password** box, type the password of the user that you have configured for use by the integration module.
11. In the **DateFormat** box, type the date format for the user that you configured for use by the integration module. The default is *yyyy-MM-dd*.
12. In the **TimeFormat** box, type the time format for the user that you configured for use by the integration module. The default is *HH:MM:ss*.
13. Click **OK**.

Add a global connection asset using OAuth 2.0:

1. Open Runbook Studio.
2. On the **Home** tab, click **Sign In**.
3. In the Sign In dialog box, sign into your account.
4. In the **Active Azure Automation Account** box, select the account that you want to add the connection asset to.
5. Click **New Asset** and then click **Connection**. In the New Connection dialog box, you must configure the following properties.
6. In the **Name** box, type the name for the configuration.
7. In the **Connection Type** box, select *Kelverion.ServiceNow.*
8. In the **ServiceNowUrl** box, type the URL of the ServiceNow instance you want to connect with.
9. In the **AuthorizationUrl** box, type the URL of your authorization server. For example, *https://test123.service-now.com/oauth_token.do*.
10. In the **ClientId** box, type the Client ID of the application endpoint that you created.
11. In the **ClientSecret** box, type the Client Secret of the application endpoint that you created.

12. In the **RefreshToken** box, type the Refresh Token that you generated.
13. In the **DateFormat** box, type the date format for the user that you configured for use by the integration module. The default is *yyyy-MM-dd*.
14. In the **TimeFormat** box, type the time format for the user that you configured for use by the integration module. The default is *HH:MM:ss*.
15. Click **OK**.

Activity Properties

All activities in Kelverion Integration Module for ServiceNow 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 *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 ServiceNow 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 ServiceNow support interactive discovery of the database assets in your on-premises and cloud environments.

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

When connected to ServiceNow, Runbook Studio will provide additional discovery options, such as Database and Table Name, which can be used to specify the database 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 Kelverion Integration Module for ServiceNow are determined by the Discovery options that you specify.

For example, when using the **New-ServiceNowRecord** activity, the Discovery panel will contain options for selecting Database Name and Table Name. Once you have selected a table, Runbook

Studio will provide you with parameters that coincide with the columns in the table schema. If you select another table, Runbook Studio will provide you with a different set of parameters automatically.

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

In addition, all activities in the Kelverion Integration Module for ServiceNow include a **Connection** parameter which is used to specify information that the activity will use to connect to ServiceNow when it is executed as part of a runbook running in Azure. 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 ServiceNow 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 source types:

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 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 • Boolean • Char • Byte • SByte • Int16 • Int32 • Int64 • UInt16 • UInt32 • Uint64 • Decimal • Double • Float • SwitchParameter When assigning a constant DateTime value, 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 is type *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 of the activities in the Integration Module include a Filters panel which lets you specify filters that can be used to retrieve or modify specific rows in a database table.

If you do not provide any filters, then the activity will retrieve or modify all records in the target table.

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

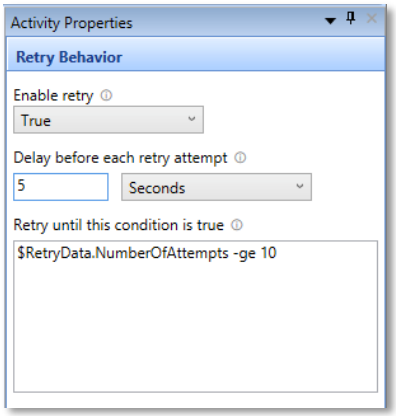
Property	Description
Filter	The name of the filter.

Operation	The operation used to evaluate the filter. Different operators will be provided based on the filter that is selected. Filter operators include: • Equals • Does not equal • Is less than • Is less than or equal to • Is greater than • Is greater than or equal to • Contains • Does not contain • Matches • Does not match • Starts with • Ends with
Value	The data source used to retrieve the value to use 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. If you want to filter for a NULL database value, use a PowerShell Expression data source and enter the expression **$null.**

Retry Behavior

The activities in the Kelverion Integration Module for ServiceNow 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, that are error prone or may need more than one attempt for success.

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



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

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

Property	Description
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 ServiceNow 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-ServiceNowAttachmentContent

The **Add-ServiceNowAttachmentContent** activity is used in a runbook to upload an attachment to a ServiceNow record.

Discovery Parameters

You can use the following options to connect to ServiceNow and configure the smart activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to ServiceNow
Table Name	The name of the ServiceNow table

Required Parameters

You must configure the following parameters.

Content	A byte array containing the attachment data to be uploaded. The data specified for content must match the proper **Content Type** and extension of the **File Name.**
Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
Content Type	The attachment's content-type. Content Type must match the data specified for **Content** and the extension of the **File Name.** For example, *test/plain**,*** *Image/jpeg* or *application/json*.
File Name	The attachment's file name, including file extension. The extension of the **File Nam**e must match the proper **Content Type** and the data specified for **Content**. See **Content Type** examples above. Ex: IssueScreenShot.png
Sys ID	The Sys ID of the record the attachment will be associated with.

Optional Parameters

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

Encryption Context	The Sys ID of a ServiceNow encryption context record. Specify this parameter to allow only users with the specified encryption context to access the attachment. If you do not specify this parameter, the attached file is not encrypted with any encryption context.

Outputs

This activity outputs the Sys ID of the ServiceNow record that was created.

Get-ServiceNowAttachmentContent

The **Get-ServiceNowAttachmentContent** activity is used in a runbook to download a ServiceNow 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.
Sys ID	The Sys ID of attachment to be downloaded.

Outputs

This activity outputs the content of the attachment that was downloaded as a byte array.

Get-ServiceNowAttachmentInfo

The **Get-ServiceNowAttachmentInfo** activity is used in a runbook to retrieve information about attachments associated with a ServiceNow record.

Discovery Parameters

You can use the following options to connect to ServiceNow and configure the smart activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to ServiceNow.
Table Name	The name of a ServiceNow table.

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.
Sys ID	Sys ID of the ServiceNow record.

Output

This activity outputs objects that represent the attachment records that were retrieved. Each object has the following properties.

Compressed	Indicates whether the attachment is compressed.
CompressedSize	The compressed file size.
ContentType	The attachment's content type.
CreatedBy	The name of the user that created the attachment record.
CreatedOn	The date and time that the attachment record was created.
FileName	The attachment's file name.
Size	The attachment size, in bytes
SysId	The Sys ID of the attachment record.
TableName	The name of the table that contains the parent record.
RecordSysId	The Sys ID of the parent record.
UpdatedBy	The name of the user that updated the attachment record.
UpdatedOn	The date and time that the attachment record was updated.

Get-ServiceNowRecord

The **Get-ServiceNowRecord** activity is used in a runbook to select records from a ServiceNow table using filter criteria that you specify.

Discovery Parameters

You can use the following options to connect to ServiceNow and configure the smart activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to ServiceNow.
Table Name	The name of a ServiceNow table.
Search By	Specifies the method used to retrieve records. Options include: • None (retrieve all records up to the specified or default limit) • Filters • Query • Sys ID

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.
Sys ID	The Sys ID of the ServiceNow record to retrieve. Available when **Search By** is set to *Sys ID*.
Query	The ServiceNow encoded query used to filter the records to retrieve. Available when **Search By** is set to *Query*.

Optional Parameters

You can use the following parameters, as needed, to alter the behavior of the activity.

Ascending	Indicates whether to order the results in ascending order. The alternative is descending order. Only used when the **Order By** property is defined.
Order By	Indicates the column that should be used to order the results.
Record Limit	The maximum records to retrieve.
Record Offset	The number or records to skip.

Filters

When the **Search By** parameter is set to *Filters*, the activity will provide filters that you can use to filter which records to retrieve.

Note: When filtering with a reference field, you must use the Sys ID of the referenced object and not its display name.

Output

This activity outputs objects that represent the records that were retrieved. Each object has properties that correspond to the fields in the ServiceNow table that was selected.

Get-ServiceNowRecordCount

The **Get-ServiceNowRecordCount** activity is used in a runbook to get the number of records in a ServiceNow table. Optionally, you can use an encoded query string or filters to select the number or records that match certain criteria.

Discovery Parameters

You can use the following options to connect to ServiceNow and configure the smart activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to ServiceNow.
Table Name	The name of a ServiceNow table.
Search By	Specifies the method used to retrieve records. Options include: • None (retrieve all records up to the specified or default limit) • Filters • Query

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.
Query	The ServiceNow encoded query used to filter the records to count. Available when **Search By** is set to *Query*.

Filters

When the **Search By** parameter is set to *Filters*, the activity will provide filters that you can use to filter which records to count.

Output

This activity outputs the number of records that were found that matched the specified filter criteria.

Get-ServiceNowRefreshToken

The **Get-ServiceNowRefreshToken** activity is used to obtain refresh tokens for OAuth 2.0 based authentication. This activity does not support discovery.

Parameter Sets

```
Get-ServiceNowRefreshToken
   -AuthorizationUrl [<String>]
   -ClientId [<String>]
   -ClientSecret [<String>]
   [-Credentials [<PSCredential>]]
```

Parameters

The activity has the following parameters.

AuthorizationUrl	Specifies the URL of the server used for OAuth authentication.
ClientId	Specifies the unique ID of the ServiceNow application endpoint.
ClientSecret	Specifies the shared secret used to authorize communication between ServiceNow and the integration module.
Credential	Specifies the credential used to authorize the request for a refresh token.

Input

This activity does not access input from the pipeline.

Output

This activity generates an OAuth 2.0 refresh token. This token is used by the integration module to access the ServiceNow instance.

Remarks

Typically, refresh tokens for Smart connections and connection assets will be created outside of Runbook Studio before you start building runbooks to integrate and automate ServiceNow. We recommend creating a PowerShell script that you can execute whenever you need to obtain a new refresh token.

Import-ServiceNowRecord

The **Import-ServiceNowRecord** activity is used in a runbook to insert or update into ServiceNow using an import set.

Additional required and optional parameters are generated based on the columns that are associated with the table that was selected.

Discovery Parameters

You can use the following options to connect to ServiceNow and configure the smart activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to ServiceNow.
Table Name	The name of the ServiceNow import set table

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.

Output

This activity outputs an object that represents the results of the import. The object has the following properties.

Table	The name of the import set table
SysId	The unique identifier of the record that was created
DisplayName	The name of the identifier field in the table that is targeted by the import set
DisplayValue	The value or the identifier in the table that is targeted by the import set
Status	The status of the import set activity
StatusMessage	A description of the import set activity

New-ServiceNowRecord

The **New-ServiceNowRecord** activity is used in a runbook to insert a record into a ServiceNow table.

Additional required and optional parameters are generated based on the columns that are associated with the table that was selected.

Discovery Parameters

You can use the following options to connect to ServiceNow and configure the smart activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to ServiceNow
Table Name	The name of the ServiceNow table

Required Parameters

If the table that you select has any required fields, then the activity will add parameters that correspond to these fields, and they must be configured. You must also configure the following parameters.

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

Optional Parameters

If the table that you select has any optional fields, then the activity will add parameters that correspond to these fields and you can configure them, as needed, to initialize the new record.

Output

This activity outputs the Sys ID of the ServiceNow record that was created.

Remove-ServiceNowRecord

The **Remove-ServiceNowRecord** activity is used in a runbook to delete a ServiceNow record.

Discovery Parameters

You can use the following options to connect to ServiceNow and configure the smart activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to ServiceNow
Table Name	The name of the ServiceNow table

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.
Sys ID	The Sys ID of record that you want to delete.

Output

This activity outputs the Sys ID of the ServiceNow record that was deleted.

Set-ServiceNowRecord

The **Set-ServiceNowRecord** activity is used in a runbook to update a record into a ServiceNow table.

Discovery Parameters

You can use the following options to connect to ServiceNow and configure the smart activity.

Table Name	The name of the ServiceNow table

Required Parameters

You must configure the following parameters.

Connection	Azure Connection Asset name

Optional Parameters

The activity will add parameters that correspond to the fields in the table that you selected, and you can use them to update the ServiceNow record you specified.

Output

This activity outputs the Sys ID of the ServiceNow record that was updated.

Appendix A: Removing Connection Type/Assets

Azure Automation prevents you from uploading a newer version of an integration module when changes have been made to the module's connection type definition (the import will fail with an error). To resolve this issue, you must remove all Kelverion.ServiceNow connection assets as well as the Kelverion.ServiceNow connection type. Connection assets can be removed in the Azure Automation portal, but you must use PowerShell to remove the connection type.

Important: Before removing connection assets, you should document the connection asset name(s) and field values so that they can be recreated. Using the same connection asset names, will save you from having to update your runbooks.

The following PowerShell script can be used to remove the Kelverion.ServiceNow connection type and connection assets from an Azure Automation account.

```
param(
    [Parameter(Mandatory = $true)]
    [ValidateNotNullOrEmpty()]
    [string] $AutomationAccountName,

    [Parameter(Mandatory = $true)]
    [ValidateNotNullOrEmpty()]
    [string] $ResourceGroupName,

    [Parameter(Mandatory = $true)]
    [ValidateNotNullOrEmpty()]
    [string] $SubscriptionId
)

Connect-AzAccount
Set-AzContext -Subscription $SubscriptionId

$commonParams = @{
    ResourceGroupName = $ResourceGroupName
    AutomationAccountName = $AutomationAccountName
}

#Find and remove all connections for the specified connection type name
$connections = Get-AzAutomationConnection @commonParams `
    -ConnectionTypeName Kelverion.ServiceNow

foreach ($connection in $connections) {
    Remove-AzAutomationConnection @commonParams `
    -Name Kelverion.ServiceNow
}

# Remove the connection type
Remove-AzAutomationConnectionType @commonParams `
    -Name Kelverion.ServiceNow `
    -Force
```