INTEGRATION MODULE FOR EASYVISTA SERVICE MANAGER

For Kelverion Runbook Studio and Azure Automation

User Guide

Version 1.4

Microsoft Azure

Certified

Kelverion Integration Module for EasyVista Service Manager

Published: December 2023

The Kelverion Integration Module for EasyVista Service Manager is Microsoft Azure Certified.

Contents

Getting Started

The following sections outline how to deploy and configure the Kelverion Integration Module for EasyVista Service Manager.

System Requirements

The following software must be installed and configured prior to deploying the integration modules. For more information about installing and configuring EasyVista, refer to the respective product documentation.

- Kelverion Runbook Studio 5.6
- Microsoft .NET Framework 4.7.2
- EasyVista Service Manager Cloud 2022.1.133.0.03

Installing the Integration Module

The easiest way to install and deploy the Integration Module for EasyVista Service Manager 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 Integration Module for EasyVista Service Manager 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 o 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.EasyVista -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 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 worker:

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

Licensing the Integration Module

Licenses for all Kelverion integration modules are managed and deployed using the **Kelverion Runbook Studio** and Azure Automation connection assets.

Register an integration module license with Runbook Studio:

1. Open **Kelverion Runbook Studio**.
2. On the **File** tab, click **About**.

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

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

Create a connection asset with a license key and upload it to 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.
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.

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 smart activities in the Kelverion Integration Module for EasyVista Service Manager.

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 to Azure Automation, Runbook Studio will automatically convert the dynamically generated parameters and filters of your smart activities into fully configured Azure Automation graphical runbook activities. Similarly, when generating PowerShell snippets, Runbook Studio will convert the dynamically generated parameters and filters in your smart activities into the appropriate PowerShell expression.

The integration module includes the following activities:

Add-EasyVistaAttachment	Attach a document to an incident/request
Close-EasyVistaRequest	Closes an existing incident/request
Complete-EasyVistaAction	Finish an ongoing action associated with an incident/request
Get-EasyVistaAction	Retrieve action records
Get-EasyVistaAttachment	Retrieve attachment records associated with an incident/request
Get-EasyVistaAttachmentContent	Download an attachment from an incident/request
Get-EasyVistaComment	Retrieve the comment from an incident/request or an action record
Get-EasyVistaDescription	Retrieve the description from an incident/request or an action record
Get-EasyVistaDepartment	Retrieve department records
Get-EasyVistaEmployee	Retrieve employee records
Get-EasyVistaLocation	Retrieve location records
Get-EasyVistaRequest	Retrieves incident/request records
Get-EasyVistaRequestCatalog	Retrieve request catalog records
New-EasyVistaAction	Create a new action record for an incident/request
New-EasyVistaRequest	Create a new incident/request

Resume-EasyVistaRequest	Resume a previously suspended incident/request
Set-EasyVistaAction	Update an existing action record.
Set-EasyVistaRequest	Update an existing incident/request
Suspend-EasyVistaRequest	Suspend an existing incident/request

Smart Connections

In Kelverion Runbook Studio you can configure one or more Smart Connections to establish reusable links between Runbook Studio and specific EasyVista instances. You can create as many Smart Connections as required.

Adding a Smart Connection to Kelverion 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 **Kelverion.EasyVista**.
5. In the **ServerUrl** box, type the URL of the EasyVista Service Manager server, in the form *https://<EasyVista Service Manager FQDN>*. For example: https://sandbox.easyvista.com.
6. In the **Account** box, type the EasyVista Service Manager account number.
7. In the **User** and **Password** boxes, type the credentials of a user with sufficient privileges for your EasyVista Service Manager integration scope. The integration module will use these credentials to connect to the EasyVista Service Manager server.
8. In the **UserDateFormat** box, type the date format corresponding to the configured user. For example: *mm/dd/yy*. For details, please see Working with Dates in EasyVista.
9. In the **Skip Certificate Validation** box, select whether you want to skip server certificate validation. ***Important:*** skipping server certificate validation should only be done in secure environments, when working with trusted servers and self-signed certificates.
10. In the **RequestTimeoutSeconds** box, type the number of seconds to wait for EasyVista to respond to requests before failing with a timeout exception.
11. Click **OK** to close the configuration dialog box, and then click **OK**.

Azure Automation Connection Assets

The activities in the Kelverion Integration Module for EasyVista Service Manager require connection information so that they can connect to instances of EasyVista from Azure Automation. The recommended way to pass connection information to your activities is to use **Azure Automation connection assets**.

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 an Azure Automation connection asset in Runbook Studio:

1. On the **Home** tab, click **Sign In**. The Sign In dialog appears.
2. Sign into your account.
3. In the **Active Azure Automation Account** box, select the account that you want to add the connection asset to.
4. Click **New Asset** and then click **Connection**. The New Connection dialog appears.
5. In the **Name** box, enter a name for the configuration.
6. In the **ServerUrl** box, type the URL of the EasyVista Service Manager server, in the form *https://<EasyVista Service Manager FQDN>*. For example: https://sandbox.easyvista.com.
7. In the **Account** box, type the EasyVista Service Manager account number.
8. In the **User** and **Password** boxes, type the credentials of a user with sufficient privileges for your EasyVista Service Manager integration scope. The integration module will use these credentials to connect to the EasyVista Service Manager server.
9. In the **UserDateFormat** box, type the date format corresponding to the configured user. For example: *mm/dd/yy*. For details, please see Working with Dates in EasyVista.
10. In the **Skip Certificate Validation** box, select whether you want to skip server certificate validation. ***Important:*** skipping server certificate validation should only be done in secure environments, when working with trusted servers and self-signed certificates.
11. In the **RequestTimeoutSeconds** box, type the number of seconds to wait for EasyVista to respond to requests before failing with a timeout exception.
12. Click **OK**.

Activity Properties

All activities in the Kelverion Integration Module for EasyVista Service Manager have the following properties:

Label	A unique label that identifies the activity in the runbook. Runbook Studio will provide a default name for each activity, but you can provide your own labels to make their role in the runbook more obvious.
Description	An optional description of the activity. Providing a description is a fantastic way to let everyone understand the function of the activity in the runbook.

Convert exceptions to errors	Indicates whether exceptions thrown by the activity should be caught and converted to non-terminating errors. This is useful when the activity has an outgoing error link that will be followed to handle the error. This property is only available for Graphical Runbooks.
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 EasyVista Service Manager include a **Discovery** panel instead of the **Parameter Sets** panel that is present for standard command activities. This is because the activities in the integration module support interactive discovery of the EasyVista assets in your environment.

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

When connected to EasyVista, 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

The **Parameters** panel in the Kelverion Integration Module for EasyVista Service Manager contains mandatory and optional parameters for each individual activity. **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 Kelverion Integration Module for EasyVista Service Manager include a **Connection** parameter which is used to specify information that the activity will use to connect to EasyVista Service Manager 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 EasyVista.

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	Specify the 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 Kelverion Integration Module for EasyVista Service Manger include a Filters panel which lets you specify server-side filters that can be used to reduce the record set when retrieving data from the EasyVista server.

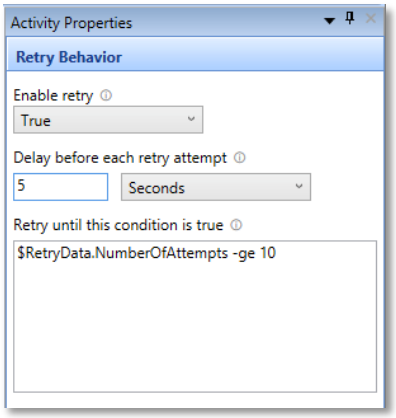
To add a filter to your activity, select the **Filters** panel and click **Add.** Filters have the following properties.

Filter	The name of the filter.
Operation	The operation 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 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 integration module 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.

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

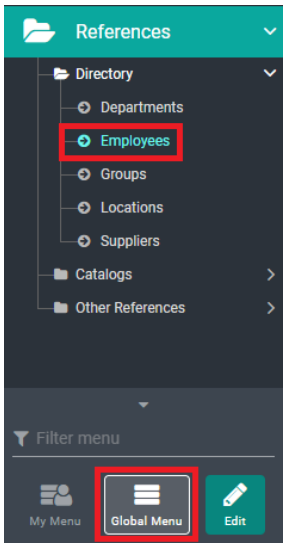
The activities in the Kelverion Integration Module for EasyVista Service Manager let you specify additional PowerShell parameters that you can use to control the behavior of the activity.

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

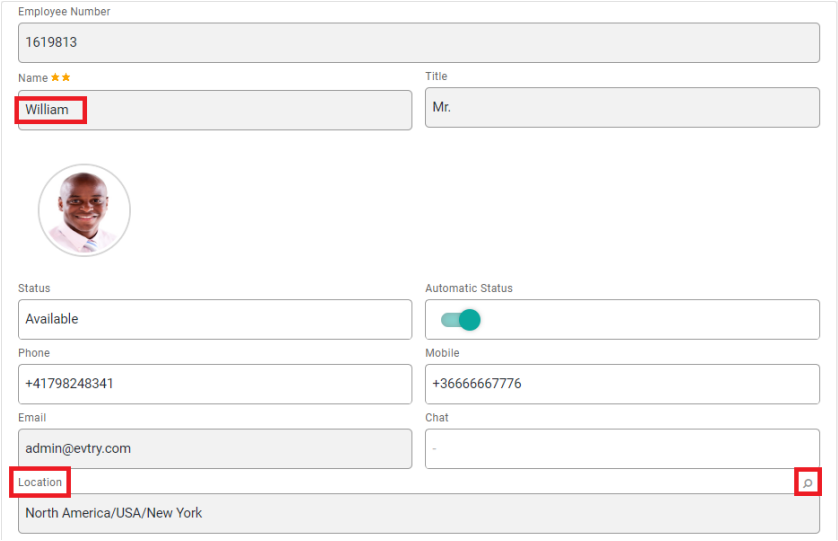
Working with Dates in EasyVista

When setting up Smart Connections and Global connection assets for the Integration Module for EasyVista, you must provide the date format for the user with which you will connecting. You can obtain the date format for users in the EasyVista Service Manager portal.

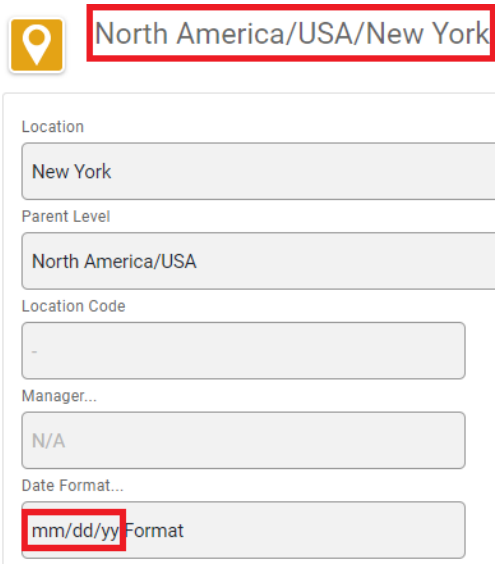
1. Make sure the **Global Menu** is selected. Under **References/Directory/Employees**, find the user that will be configured in the integration module to connect to EasyVista Service Manager.



2. In the Employee details screen, click on the magnifying glass icon to expand **Location** details:



3. You can find the **Date Format** box on the Location details screen:



4. Make note of the contents of the **Date Format** box, excluding the **Format** suffix. For example: *mm/dd/yy*. Use this format when creating new connections in Runbook Studio.

Additional Field Configuration

In addition to the typical configuration options that are usually provided by integration modules, the Integration Module for EasyVista Service Manager provides an XML field configuration file, which allows users to further customize activity field behavior, according to specific use-cases or environmental needs.

The configuration file can be used to:

- Map activity parameters to EasyVista Service Manager input/output fields.
- Add activity inputs/outputs.
- Remove activity inputs/outputs which are not required.
- Configure input/output data types.
- Configure browser lists for inputs/outputs.
- Configure filter support.
- Configure sorting support.

The configuration file is named **Kelverion.EasyVista.Configuration.xml** and the default version is installed in the same location as the integration module.

Important: When making changes to this configuration file, **we recommend that you make a backup**, should it be necessary to roll back your changes. Also, when deploying new versions of the integration module, we recommend that you **keep a copy of the modified version** so that you can restore your changes. This is particularly important when updating to new versions of the integration module, as these may overwrite your changes with the default version of the configuration file.

Configuration File Structure

The main purpose of the integration module configuration file is to provide information about the EasyVista Service Manager fields used by integration module activities. This includes information about inputs/outputs, data types, filtering, sorting, and more.

The configuration file is divided in two main sections under the root node:

```
<Kelverion.EasyVista>
  <Lists>...</Lists>
  <Activities>...</Activities>
</Kelverion.EasyVista>
```

Under the **Lists** element you will find all the lists used by activity browsers for fields which work with a pre-determined set of values. The location of the **Lists** element in the configuration file, must be prior to the **Activities** element.

Important: The lists included in the installed integration module configuration file are specific to the EasyVista Cloud instance. You may have to modify list item names/values according to your specific EasyVista environment instance.

The **Activities** element contains input and output information about each activity in the integration module. Depending on the type of activity, each **Activity** element will contain a collection of **Input** and **Output** elements, that you can use to specify the input parameters and outputs that are supported by the activity in your EasyVista environment. These input and output settings will apply to each instance of the activity in your runbooks.

Kelverion.EasyVista Element

The **Kelverion.EasyVista** element describes the configurations that will be used by the activities in the Integration Module for EasyVista Service Manager.

Parents

This element does not have any parents.

Children

The following table outlines the children of the **Kelverion.EasyVista** element.

Name	Occurrences	Description
Lists	1	Contains a list of **List** elements the define the options that are available for various EasyVista fields.
Activities	1	Contains the input and output definitions for each activity in the integration module

Attributes

This element does not have any attributes.

Lists Element

The **Lists** element contains the collection of lists with allowed values or options for the various fields in your EasyVista environment. The location of the **Lists** element, in the configuration file, must be prior to the **Activities** element.

Parent

The parent of the **Lists** element is the **Kelverion.EasyVista** element.

Children

The following table outlines the children of the **Lists** element.

Name	Occurrences	Description
List	[0, *]	Contains a collection of **item** elements that define the allowed values or options for a particular EasyVista field.

Attributes

This element does not have any attributes.

List Element

The **List** element contains information about the allowed values or options for a specific EasyVista field.

Parent

The parent of the **List** element is the **Lists** element.

Children

The following table outlines the children of the **List** element.

Name	Occurrences	Description
Item	[0, *]	Each **Item** defines a specific allowed value or option for an EasyVista field.

Attributes

The following table outlines the attributes of the **List** element.

Name	Type	Required	Description
Name	String	Yes	Unique name identifying the list. Used to refer to this list by activity input/output elements.

Item Element

The **Item** element describes a specific allowed value or option for an EasyVista field.

Parent

The parent of the **Item** element is the **List** element.

Children

This element does not have any children.

Attributes

The following table outlines the attributes of the **Item** element.

Name	Type	Required	Description
Value	string	Yes	The value that will be used for this item when communicating with EasyVista.
Label	string	No	The value that will be used for this item when it is displayed in Runbook Studio.

Activities Element

The **Activities** element contains collection of activity definitions used to configure the integration module activities in Runbook Studio. The **Activities** section must follow the **Lists** element in your configuration.

Parent

The parent of the **Activities** element is the **Kelverion.EasyVista** element.

Children

The following table outlines the children of the **Activities** element.

Name	Occurrences	Description
Activity	[0, *]	Describes the inputs and outputs of a particular integration module activity.

Attributes

This element does not have any attributes.

Activity Element

The **Activity** element describes the input and outputs that are supported by a specific integration module activity.

Parent

The parent of the **Activity** element is the **Activities** element.

Children

The following table outlines the children of the **Activity** element.

Name	Occurrences	Description
Input	[1, *]	Defines an activity input and how it correlates to a specific EasyVista field.

Name	Occurrences	Description
Output	[1, *]	Defines an activity output and how it correlates to a specific EasyVista field.

Attributes

The following table outlines the attributes of the **Activity** element.

Name	Type	Required	Description
Name	string	Yes	Specifies the name of the integration module activity that the definition is for.

Input Element

The **Input** element describes a specific activity input.

Parent

The parent of the **Input** element is the **Activity** element.

Children

The **Input** element does not have any children.

Attributes

The following table outlines the attributes of the **Input** element.

Name	Type	Required	Description
Name	String	Yes	Specifies the name of the EasyVista field that this input references.
Label	String	Yes	Specifies the label used to display this input in Runbook Studio.
Type	String	Yes	Specifies the input type. Valid options include: • string • bool • int32 • int64 • double • Date • DateTime
Requires	Boolean	No	Indicates whether the input is required. The default is **false**.
List	String	No	Specifies the name of the **List** that specifies the allowed values for this input.

Output Element

The **Output** element describes a specific activity output. Outputs are also used configure filters in Runbook Studio.

Parent

The parent of the **Output** element is the **Activity** element.

Children

The **Output** element does not have any children.

Attributes

The following table outlines the attributes of the **Output** element.

Name	Type	Required	Description
Name	String	Yes	Specifies the name of the EasyVista field that this output references.
Label	String	Yes	Specifies the label used to display this output in Runbook Studio.
Type	String	Yes	Specifies the output type. Valid options include: • string • bool • int32 • int64 • double • Date • DateTime
Requires	Boolean	No	Indicates whether the input is required. The default is **false**.
List	String	No	Specifies the name of the **List** that specifies the allowed value when configuring filters for this output.
FilterType	String	Yes	Specifies whether this output will support filtering. Valid values include: • None: the output does not support filtering • Sever: the output supports server-side filtering
CanSort	Boolean	No	Indicates whether this output supports server-side sorting. The default is **true**.
Prefix	String	No	Specifies a prefix used to qualify the output name in the data set returned from the server, or when building server-side filter expressions.

Add-EasyVistaAttachment

The **Add-EasyVistaAttachment** activity is used in a runbook to attach a document to a request or incident ticket.

Discovery Options

This command activity does not provide any discovery options.

Required Parameters

You must configure the following parameters.

Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Content	Specifies the attachment content that is to be uploaded.
File Name	Specifies the file name of the attachment.
Request Number	Identifies the incident/request to be closed.

Outputs

This activity outputs the request number of the incident/request ticket that the attachment was added to.

Close-EasyVistaRequest

The **Close-EasyVistaRequest** activity is used in a runbook to close an open request or incident ticket.

Discovery Options

Use the following options to connect to EasyVista and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to EasyVista Service Manager.

Required Parameters

You must configure the following parameters.

Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Request Number	Identifies the incident/request to be closed.

Optional Parameters

You can use the following parameters, as needed, to change how the activity runs.

Anticipated Close Date	Closing date of open actions associated with the request and the anticipated closure action. By default, the current date (now). **Note:** The closing date of the request is always the current date, and it cannot be modified.
Catalog GUID	Identifies the incident/request category. You can use this parameter to re-qualify the incident/request before it is closed.
Comment	Comment describing the closing of the incident/request.
Delete Actions	Specifies the measures to be taken for ongoing incident/request actions.
Status	Final incident/request status. Valid values include Archived, Cancelled, Closed, Reject Operation and Solved. The default is Closed.

Outputs

This activity outputs the request number of the incident/request ticket that was closed.

Complete-EasyVistaAction

The **Complete-EasyVistaAction** activity is used in a runbook to finish an ongoing request action.

Discovery Options

Use the following options to connect to EasyVista and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to EasyVista Service Manager.
Complete All Actions	Specifies whether all actions, or a specific action, on the specified request, will be completed.

Required Parameters

You must configure the following parameters.

Action ID	Identifies a specific action to be done, when completing a specific action.
Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Request Number	Identifies the incident/request containing the action to be completed.

Optional Parameters

You can use the following parameters, as needed, to change how the activity runs.

Available Field 1..6	Action available field.
Description	Specifies action description.
Done by Email	Specifies the email address of the user that completed the action.
Done by Name	Specifies the name of the user that completed the action.
Elapsed Time	Action Elapsed Time.
End Date	Action End Date.
Start Date	Action Start Date.

Outputs

This activity outputs the request number of the incident/ticket for which the action is being completed.

Get-EasyVistaAction

The **Get-EasyVistaAction** activity is used in a runbook to retrieve action records.

Important: When using System.DateTime filters, only the date portion of the DateTime value is considered by EasyVista Service Manager. In other words, the time portion of the filter value is disregarded.

Discovery Options

Use the following options to connect to EasyVista and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to EasyVista Service Manager.
Search By	Specifies how the activity will be retrieving records.

Required Parameters

You must configure the following parameters.

Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Action ID	Specifies the Action ID of an action record to be retrieved, when searching by Action ID.

Optional Parameters

You can use the following parameters, as needed, to change how the activity runs.

Descending	Specifies whether records should be returned in ascending or descending order. This parameter is available when searching by Filters. **Note:** This parameter is only considered if **Order By** is specified.
Record Limit	Specifies the maximum number of records to be returned by the activity. This parameter is available when searching by Filters.
Order By	Specifies the field that should be used to order the returned records. This parameter is available when searching by Filters.

Filters

This activity provides the following filters, that you can combine as needed, to select which action records to retrieve.

Action ID	Filter by Action ID
Action Type	Filter by Action Type

Action Type ID	Filter by Action Type ID
Application Date	Filter by Application Date
Asset ID	Filter by Asset ID.
Available Field 1.. 6	Filter by Available Field value 1 to 6.
Category Test ID	Filter by Category Test ID.
Contact ID	Filter by Contact ID.
Continuity Plan ID	Filter by Continuity Plan ID.
Contractual Cost	Filter by Contractual Cost.
Contractual Cost Currency ID	Filter by Contractual Cost Currency ID.
Create Date	Filter by Create Date.
Delay	Filter by Delay.
Elapsed Time	Filter by Elapsed Time.
End Date	Filter by End Date.
Estimated Net Charge	Filter by Estimated Net Charge.
Exit Value	Filter by Exit Value.
Expected Duration	Filter by Expected Duration.
Expected End Date	Filter by Expected End Date.
Expected Start Date	Filter by Expected Start Date.
History ID	Filter by History ID.
Known Problem ID	Filter by Known Problem ID.
Last Update Date	Filter by Last Update Date.
Location	Filter by Location.
Location Code	Filter by Location Code.
Location ID	Filter by Location ID.
Max Intervention Date	Filter by Max Intervention Date.
Max Resolution Date	Filter by Max Resolution Date.
Net Charge	Filter by Net Charge.
Origin Action ID	Filter by Origin Action ID.
Origin Tool ID	Filter by Origin Tool ID.
Parent Action ID	Filter by Parent Action ID.
Percent Complete	Filter by Percent Complete.

Person Cost Per Hour	Filter by Person Cost Per Hour.
Planned Budget	Filter by Planned Budget.
Priority ID	Filter by Priority ID.
Renewal Date	Filter by Renewal Date.
Request ID	Filter by Request ID.
Request Max Resolution Date	Filter by Request Max Resolution Date.
Request Number	Filter by Request Number.
Request Submit Date	Filter by Request Submit Date.
Stage ID	Filter by Stage ID.
Start Date	Filter by Start Date.
Status On Create	Filter on Status:
Status On Terminate	Filter by Status On Terminate.
Supplier ID	Filter by Supplier ID.
Support Person	Filter by Support Person.
Support Person Email	Filter by Support Person Email.
Support Person Group ID	Filter by Support Person Group ID.
Support Person ID	Filter by Support Person ID.
Support Staff ID	Filter by Support Staff ID.
Tax ID	Filter by Tax ID.
Time Cost	Filter by Time Cost.
Validator ID	Filter by Validator ID.
Workflow ID	Filter by Workflow ID.
Workflow Value	Filter by Workflow Value.

Outputs

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

Action ID	Identifies the action record.
Action Type	Action type name.
Action Type ID	Identifies the action type.
Application Date	Action Application Date.

Asset ID	Asset ID of asset associated with the action.
Available Field 1..6	Available Field value 1 to 6.
Category Test ID	Category test identifier.
Contact ID	Identifies the action contact.
Continuity Plan ID	Continuity plan identifier.
Contractual Cost	Contractual cost value.
Contractual Cost Currency ID	Contractual cost currency identifier.
Create Date	Action create date and time.
Delay	Action delay.
Elapsed Time	Action elapsed time.
End Date	Action end date.
Estimated Net Charge	Estimated net charge for the action.
Exit Value	Action exit value.
Expected Duration	Expected action duration.
Expected End Date	Expected action end date.
Expected Start Date	Expected action start date.
History ID	Action history identifier.
Known Problem ID	Identifies a known problem associated with the action.
Known Problem Path	Path of known problem associated with the action.
Last Update Date	Last time the action record was modified.
Location	Name of the location associated with the action.
Location Code	Code of the location associated with the action.
Location ID	Identifies the location associated with the action.
Location Path	Path of the location associated with the action.
Max Intervention Date	Latest action intervention date.
Max Resolution Date	Latest action resolution date.
Net Charge	Action net charge.
Origin Action ID	Identifies action origin.
Origin Tool ID	Identifies action origin tool.
Parent Action ID	Identifies the parent action.
Percent Complete	Percent completion of the action.

Person Cost Per Hour	Person cost per hour for the action.
Planned Budget	Action planned budget.
Priority ID	Action priority.
Renewal Date	Action renewal date.
Request ID	Identifies the incident/request associated with this action.
Request Max Resolution Date	Latest resolution date for the request associated with this action.
Request Number	Identifies the incident/request associated with this action.
Request Submit Date	Submit date for the incident/request associated with this action.
Stage ID	Action stage identifier.
Start Date	Action start date.
Status On Create	Request status when action was created.
Status On Terminate	Request status when action was ended.
Supplier ID	Action supplier.
Supplier Path	Action supplier path.
Support Person	Action support person.
Support Person Department Path	Department path for action support person.
Support Person Email	Action support person email.
Support Person Group ID	Action support person group identifier.
Support Person ID	Action support person identifier.
Support Person Location Path	Location of the support person.
Support Staff ID	Support staff identifier.
Tax ID	Tax identifier.
Time Cost	Time cost for the action.
Time Cost Currency ID	Time cost currency identifier.
Time Used To Complete	Time used to complete the action.
Validator ID	Validator identifier.
Workflow ID	Workflow identifier.
Workflow Value	Workflow value.

Get-EasyVistaAttachment

The **Get-EasyVistaAttachment** activity is used in a runbook to retrieve information about the attachments that are associated with a request or incident ticket.

Discovery Options

Use the following options to connect to EasyVista and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to EasyVista Service Manager.

Required Parameters

You must configure the following parameters.

Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Request Number	Specifies the request for which attachment information is to be retrieved.

Outputs

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

Attachment ID	Identifies an attachment associated with a request or incident.
File Name	File name of for the attachment

Get-EasyVistaAttachmentContent

The **Get-EasyVistaAttachmentContent** activity is used in a runbook to download the content of a specific attachment that is associated with a request or incident ticket.

Discovery Options

This command activity does not provide any discovery options.

Required Parameters

You must configure the following parameters.

Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Attachment ID	Identifies the attachment to be downloaded.

Outputs

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

Get-EasyVistaComment

The **Get-EasyVistaComment** activity is used in a runbook to retrieve the comment associated with an action or incident/request ticket.

Discovery Options

Use the following options to connect to EasyVista and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to EasyVista Service Manager.
Object Type	Specifies the type of object for which the comment will be retrieved.

Required Parameters

You must configure the following parameters.

Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Action ID	Identifies the action for which the comment will be retrieved, when retrieving an action comment.
Request Number	Identifies the incident/request for which the comment will be retrieved, when retrieving a request comment.

Outputs

This activity outputs the text of the incident/request ticket comment.

Get-EasyVistaDescription

The **Get-EasyVistaDescription** activity is used in a runbook to retrieve the description associated with an action or incident/request ticket.

Discovery Options

Use the following options to connect to EasyVista and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to EasyVista Service Manager.
Object Type	Specifies the type of object for which the description will be retrieved.

Required Parameters

You must configure the following parameters.

Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Action ID	Identifies the action for which the description will be retrieved, when retrieving an action description.
Request Number	Identifies the incident/request for which the description will be retrieved, when retrieving a request description.

Outputs

This activity outputs the text of the incident/request ticket description.

Get-EasyVistaDepartment

The **Get-EasyVistaDepartment** activity is used in a runbook to retrieve department records.

Discovery Options

Use the following options to connect to EasyVista and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to EasyVista Service Manager.
Search By	Specifies how the activity will be retrieving records.

Required Parameters

You must configure the following parameters.

Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Department ID	Specifies the ID of a department record to be retrieved, when searching by Department ID.

Optional Parameters

You can use the following parameters, as needed, to change how the activity runs.

Descending	Specifies whether records should be returned in ascending or descending order. This parameter is available when searching by Filters. **Note:** This parameter is only considered if **Order By** is specified.
Record Limit	Specifies the maximum number of records to be returned by the activity. This parameter is available when searching by Filters.
Order By	Specifies the field that should be used to order the returned records. This parameter is available when searching by Filters.

Filters

This activity provides the following filters, that you can combine as needed, to select which department records to retrieve.

Department	Filter by Department Name.
Department Code	Filter by Department Code.
Department ID	Filter by Department ID.

Outputs

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

Department	The name of the department
Department Code	The code that has been assigned to the department.
Department ID	The unique ID of the department
Department Path	The department path.

Get-EasyVistaEmployee

The **Get-EasyVistaEmployee** activity is used in a runbook to retrieve employee records.

Discovery Options

Use the following options to connect to EasyVista and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to EasyVista Service Manager.
Search By	Specifies how the activity will be retrieving records.

Required Parameters

You must configure the following parameters.

Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Employee ID	Specifies the ID of an employee record to be retrieved, when searching by Employee ID.

Optional Parameters

You can use the following parameters, as needed, to change how the activity runs.

Descending	Specifies whether records should be returned in ascending or descending order. This parameter is available when searching by Filters. **Note:** This parameter is only considered if **Order By** is specified**.**
Record Limit	Specifies the maximum number of records to be returned by the activity. This parameter is available when searching by Filters.

Order By	Specifies the field that should be used to order the returned records. This parameter is available when searching by Filters.

Filters

This activity provides the following filters, that you can combine as needed, to select which employee records to retrieve.

Cellular Number	Filter by Cellular Number.
Department	Filter by Department.
Department Code	Filter by Department Code.
Department ID	Filter by Department ID.
Email	Filter by Email.
Employee ID	Filter by Employee ID.
Identification	Filter by Identification.
Location	Filter by Location.
Location Code	Filter by Location Code.
Location ID	Filter by Location ID.
Login	Filter by Login.
Name	Filter by Name.
Phone Number	Filter by Phone Number.

Outputs

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

Cellular Number	Employee cellular number.
Department	Employee department name.
Department Code	Employee department code.
Department ID	Employee department ID.
Department Path	Employee department path.
Email	Employee email address.
Employee ID	Employee identifier.
Identification	Employee identifier.
Location	Employee location.

Location Code	Employee location code.
Location ID	Employee location ID.
Location Path	Employee location Path.
Login	Employee login name.
Name	Employee name.
Phone Number	Employee phone number.

Get-EasyVistaLocation

The **Get-EasyVistaLocation** activity is used in a runbook to retrieve and filter location records.

Discovery Options

Use the following options to connect to EasyVista and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to EasyVista Service Manager.
Search By	Specifies how the activity will be retrieving records.

Required Parameters

You must configure the following parameters.

Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Location ID	Specifies the ID of a location record to be retrieved, when searching by Location ID.

Optional Parameters

You can use the following parameters, as needed, to change how the activity runs.

Descending	Specifies whether records should be returned in ascending or descending order. This parameter is available when searching by Filters. **Note:** This parameter is only considered if **Order By** is specified
Record Limit	Specifies the maximum number of records to be returned by the activity. This parameter is available when searching by Filters.
Order By	Specifies the field that should be used to order the returned records. This parameter is available when searching by Filters.

Filters

This activity provides the following filters, that you can combine as needed, to select which location records to retrieve.

Location	Filter by Location Name.
Location Code	Filter by Location Code.
Location ID	Filter by Location Identifier.

Outputs

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

Location	Location name.
Location Code	Location code.
Location ID	Location identifier.
Location Path	Location path.

Get-EasyVistaRequest

The **Get-EasyVistaRequest** activity is used in a runbook to retrieve request or incident records.

Important: When using System.DateTime filters, only the date portion of the DateTime value is considered by EasyVista Service Manager. In other words, the time portion of the filter value is disregarded.

Discovery Options

Use the following options to connect to EasyVista and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to EasyVista Service Manager.
Search By	Specifies how the activity will be retrieving records.

Required Parameters

You must configure the following parameters.

Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Request Number	Specifies the Request Number of a request record to be retrieved, when searching by Request Number.
Request ID	Specifies the ID of a request record to be retrieved, when searching by Request ID.

Optional Parameters

You can use the following parameters, as needed, to change how the activity runs.

Descending	Specifies whether records should be returned in ascending or descending order. This parameter is available when searching by Filters. **Note:** This parameter is only considered if **Order By** is specified.
Record Limit	Specifies the maximum number of records to be returned by the activity. This parameter is available when searching by Filters.
Order By	Specifies the field that should be used to order the returned records. This parameter is available when searching by Filters.

Filters

This activity provides the following filters, that you can combine as needed, to select which incident/request ticket records to retrieve.

Asset ID	Filter by Asset ID.
Available Field 1 .. 6	Filter by Available Field 1 to 5 value.
Budget (effective)	Filter by Budget (effective) value.
Budget (planned)	Filter by Budget (planned) value.
Budget ID	Filter by Budget ID.
Can Be Duplicated	Filter by Can Be Duplicated value.
Catalog Code	Filter by Catalog Code.
Catalog ID	Filter by Catalog ID.
Catalog Name	Filter by Catalog Name.
CI ID	Filter by CI ID.
Cost Center ID	Filter by Cost Center ID.
Create Date	Filter by Create Date.
Delay (sec)	Filter by Delay seconds value.
Department	Filter by Department name.
Department Code	Filter by Department Code.
Department ID	Filter by Department ID.
Effective Change End	Filter by Effective Change End value.
Effective Change Start	Filter by Effective Change Start value.
End Date	Filter by End Date value.
Estimated Complete (%)	Filter by Estimated Complete percentage value.
Estimated Net Price.	Filter by Estimated Net Price.
Expected Date	Filter by Expected Date value.
Expected Duration (seconds)	Filter by Expected Duration seconds value.
Expected End Date	Filter by Expected End Date value.
Expected Start Date	Filter by Expected Start Date value.
External Reference	Filter by External Reference.
First Call Resolution	Filter by First Call Resolution value.
Impact	Filter by Impact. Valid values include: • 1 - High • 2 - Medium • 3 - Low

Is Major Incident	Filter by Is Major Incident value.
Is Template	Filter by Is Template value.
Known Problem ID	Filter by Known Problem ID.
Known Problem Number	Filter by Known Problem Number.
Known Problem Question	Filter by Known Problem Question.
Last Update Date	Filter by Last Update Date value.
Location	Filter by Location name.
Location Code	Filter by Location Code.
Location ID	Filter by Location ID.
Mark 1, 2	Filter by Mark 1, 2
Max Resolution Date	Filter by Max Resolution Date value.
Net Price	Filter by Net Price.
Not Deduced Call	Filter by Not Deduced Call value.
Order Net Price	Filter by Order Net Price.
Origin	Filter by Origin value.
Origin Tool ID	Filter by Origin Tool ID.
Parent Request ID	Filter by Parent Request ID.
Planned Change End	Filter by Planned Change End value.
Planned Change Start	Filter by Planned Change Start value.
PM Status ID	Filter by PM Status ID.
Project Name	Filter by Project Name.
Project Start Date	Filter by Project Start Date value.
Quantity	Filter by Quantity.
Recipient Email	Filter by Recipient Email.
Recipient ID	Filter by Recipient ID.
Recipient Name	Filter by Recipient Name.
Rental Net Price	Filter by Rental Net Price.
Requalification Processing	Filter by Requalification Processing**.**
Request ID	Filter by Request ID.
Request Number	Filter by Request Number.

Requested Change End	Filter by Requested Change End value.
Requested Change Start	Filter by Requested Change Start value.
Requestor Email	Filter by Requestor Email.
Requestor Feedback	Filter by Requestor Feedback.
Requestor ID	Filter by Requestor ID.
Requestor IP Address	Filter by Requestor IP Address.
Requestor Name	Filter by Requestor Name.
Requestor Phone	Filter by Requestor Phone.
Required Downtime	Filter by Required Downtime.
Risk Amount	Filter by Risk Amount.
Risk Level ID	Filter by Risk Level ID
Severity	Filter by Severity value. Value values include: • 1 - High • 2 - Medium • 3 - Low • Major Incident
Status	Filter by Status value.
Submit Date	Filter by Submit Date value.
Time Until Feedback (sec)	Filter by Time Until Feedback seconds.
Time Until Solved (sec)	Filter by Time Until Solved seconds.
Urgency	Filter by Urgency value. Valid values include: • 1 - High • 2 - Medium • 3 - Low
Validation Level	Filter by Validation Level.

Outputs

This activity outputs objects that represent the incident/request ticket records that were retrieved. Each object has the following properties.

Asset ID	Identifies an asset associated with the incident/request.
Available Field 1 .. 6	Request available field.
Budget (effective)	Request effective budget.

Budget ID	Budget identifier.
Budget (planned)	Request planned budget.
Can Be Duplicated	Specifies if request can be duplicated
Catalog ID	Request catalog identifier.
Catalog Code	Request catalog code.
Catalog Name	Request catalog name.
Catalog Path	Request catalog path.
CI ID	Identifies a configuration item associated with the incident/request.
Cost Center ID	Request cost center identifier.
Create Date	Date and time when record has been created.
Delay (sec)	Request delay, in seconds.
Department Code	Request department code.
Department ID	Request department ID.
Department Path	Request department path.
Effective Change End	Effective change request end.
Effective Change Start	Effective change request start.
End Date	Request end date and time.
Estimated Complete (%)	Estimated complete percentage of the request.
Estimated Net Price	Request estimated net price.
Expected Date	Request expected date.
Expected Duration (sec)	Request expected duration, in seconds.
Expected End Date	Request expected end date.
Expected Start Date	Request expected start date.
External Reference	Request external reference.
First Call Resolution	Specifies if request was resolved on first call.
Impact	Request impact.
Is Major Incident	Specifies if this is a major incident
Is Template	Specifies if this is a template record.
Known Problem ID	Identifies a known problem associated with the incident/request.

Known Problem Number	Identifies a known problem associated with the incident/request.
Known Problem Path	Path of a known problem associated with the incident/request.
Known Problem Question	Question for a known problem associated with the incident/request.
Last Update Date	Date and time when the record was last modified.
Location Code	Request location code.
Location ID	Request location ID.
Location Path	Request location path.
Mark 1, 2	Request mark.
Max Resolution Date	Latest date and time when issue can be resolved.
Net Price	Request net price.
Not Deduced Call	Not deduced call.
Order Net Price	Order net price.
Origin	Request origin.
Origin Tool ID	Origin tool identifier.
Parent Request ID	Identifies the parent request record.
Planned Change End	Date and time when change is planned to end.
Planned Change Start	Date and time when change is planned to start.
PM Status ID	PM status identifier.
Project Name	Project name.
Project Start Date	Project start date and time
Quantity	Quantity.
Recipient Email	Recipient email address.
Recipient ID	Recipient identifier.
Recipient Name	Recipient name.
Rental Net Price	Rental net price.
Requalification Processing	Requalification processing.
Request ID	Request record identifier.
Request Number	Request number.
Requested Change End	Date and time requested for change to end.
Requested Change Start	Date and time requested for change to start.

Requestor Email	Requestor email address.
Requestor Feedback	Indicates if record contains requestor feedback.
Requestor ID	Requestor identifier.
Requestor IP Address	Requestor IP address.
Requestor Name	Requestor name.
Requestor Phone	Requestor name.
Required Downtime	Required downtime.
Risk Amount	Risk amount.
Risk Level ID	Risk level identifier.
Severity	Request severity.
Status	Request status.
Submit Date	Date and time when the request was submitted.
Time Until Feedback (sec)	Duration until request received feedback, in seconds.
Time Until Solved (sec)	Duration until request was solved, in seconds.
Total Record Count	Total number of request records present on the server.
Urgency	Request urgency.
Validation Level	Validation level.

Get-EasyVistaRequestCatalog

The **Get-EasyVistaRequestCatalog** activity is used in a runbook to retrieve and filter request catalog records.

Discovery Options

Use the following options to connect to EasyVista and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to EasyVista Service Manager.
Search By	Specifies how the activity will be retrieving records.

Required Parameters

You must configure the following parameters.

Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Catalog ID	Specifies the ID of a request catalog record to be retrieved, when searching by Catalog ID.

Optional Parameters

You can use the following parameters, as needed, to change how the activity runs.

Descending	Specifies whether records should be returned in ascending or descending order. This parameter is available when searching by Filters. **Note:** This parameter is only considered if **Order By** is specified.
Record Limit	Specifies the maximum number of records to be returned by the activity. This parameter is available when searching by Filters.
Order By	Specifies the field that should be used to order the returned records. This parameter is available when searching by Filters.

Filters

This activity provides the following filters, that you can combine as needed, to select which request catalog records to retrieve.

Catalog Code	Filter by Catalog Code.
Catalog GUID	Filter by Catalog GUID.
Catalog ID	Filter by Catalog ID.

Default Urgency	Filter by Default Urgency value. Valid values include 1 – High, 2 – Medium and 3 – Low.
Impact	Filter by Impact value. 1 – High, 2 – Medium and 3 – Low.
Last Update Date	Filter by Last Update Date value.
Title	Filter by Title.

Outputs

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

Catalog Code	Request catalog code.
Catalog GUID	Request catalog identifier.
Catalog ID	Request catalog identifier.
Catalog Path	Request catalog path.
Default Urgency	Request catalog default urgency.
Impact	Request catalog impact.
Last Update Date	Last time the request catalog record was updated.
Title	Request catalog title.

New-EasyVistaAction

The **New-EasyVistaAction** activity is used in a runbook to create a new action for an existing incident/request ticket.

Important: When creating a new action, the support group must be specified (either the name, email, or ID) and the EasyVista user specified in the connection configuration must be part of this support group.

Discovery Options

Use the following options to connect to EasyVista and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to EasyVista Service Manager.
With Parent Action	Specifies whether the new action record should be created with a parent action or not.

Required Parameters

You must configure the following parameters.

Action Type	Specifies the type of action that is to be created.
Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Parent Action ID	Identifies the parent action for the new action, when creating an action with a parent.
Request Number	Specifies the incident/request record that the new action is to be associated with.

Optional Parameters

You can use the following parameters, as needed, to change how the activity runs.

Contact	Specifies the name of the contact for the new action.
Contact Email	Specifies the email of the contact for the new action.
Contact ID	Specifies the ID of the contact for the new action.
Description	Description for the new action.
Expected End Time	Expected action end time.
Expected Start Time	Expected action start time.

Max Intervention Time	The latest intervention time for the action.
Support Group	Specifies the name of the support group for the new action.
Support Group Email	Specifies the email of the support group for the new action.
Support Group ID	Specifies the ID of the support group for the new action.

Outputs

This activity outputs the ID of the action that was created.

New-EasyVistaRequest

The **New-EasyVistaRequest** activity is used in a runbook to create a new incident/request ticket.

Discovery Options

Use the following options to connect to EasyVista and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to EasyVista Service Manager.
Create With	Specifies whether the new request record should be created with Catalog Request Code or Catalog Request GUID.

Required Parameters

You must configure the following parameters.

Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Catalog Code	Identifies the catalog category for the new incident/request, when creating with Catalog Code.
Catalog GUID	Identifies the catalog category for the new incident/request, when creating with Catalog GUID.

Optional Parameters

You can use the following parameters, as needed, to change how the activity runs.

Asset ID	ID of an asset to be associated with the request. **Note:** It is recommended to specify either the *Asset ID* or the *Asset Tag*
Asset Tag	Asset Tag of an asset to be associated with the request. **Note:** It is recommended to specify either the *Asset ID* or the *Asset Tag*
CI ID	ID of a configuration item to be associated with the request. **Note:** It is recommended to specify either the *CI ID* or the *CI Number*
CI Number	Configuration item number of a CI to be associated with the request. **Note:** It is recommended to specify either the *CI ID* or the *CI Number*
Department Code	Department to be associated with the request.
External Reference	Request external reference.
Location Code	Location to be associated with the request.
Origin	Request origin.

Parent Request Number	Request number of parent request to be associated with this request.
Recipient Email	Recipient email address.
Recipient Name	Recipient name.
Requestor Email	Requestor email address.
Requestor Name	Requestor email address.
Severity	Request severity. Valid values include 1 - High, 2 - Medium and 3 – Low.
Submit Date	Request submit date and time.
Urgency	Request urgency. Valid values include 1 - High, 2 - Medium and 3 – Low.

Outputs

This activity outputs the request number of the incident/request ticket that was created.

Resume-EasyVistaRequest

The **Resume-EasyVistaRequest** activity is used in a runbook to resume a previously suspended request or incident ticket.

Discovery Options

Use the following options to connect to EasyVista and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to EasyVista Service Manager.

Required Parameters

You must configure the following parameters.

Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Request Number	Identifies the incident/request to be resumed.

Optional Parameters

You can use the following parameters, as needed, to change how the activity runs.

Comment	Comment detailing resume information.

Outputs

This activity outputs the request number of the incident/request ticket that was resumed.

Set-EasyVistaAction

The **Set-EasyVistaAction** activity is used in a runbook to update an existing action record.

Discovery Options

Use the following options to connect to EasyVista and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to EasyVista Service Manager.

Required Parameters

You must configure the following parameters.

Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Action ID	Identifies the action record to be updated.

Optional Parameters

You can use the following parameters, as needed, to change how the activity runs.

Application Date	Action application date.
Asset ID	Asset ID for the action.
Available Field 1..6	Action available field 1 to 6.
Category Test ID	Action category test identifier.
Comment	Action comment.
Contact ID	Action contact identifier.
Continuity Plan ID	Action continuity plan identifier.
Contractual Cost	Action contractual cost.
Contractual Cost Currency ID	Action contractual cost currency identifier.
Delay	Action delay.
Description	Action description.
Support Person ID	Action support person ID.
Elapsed Time	Action elapsed time.
End Date	Action end date.
Estimated Net Charge	Action estimated net charge.

Exit Value	Action exit value.
Expected Duration	Action expected duration.
Expected End Date	Action expected end.
Expected Start Date	Action expected start.
Group ID	Action support person Group ID.
Known Problem ID	Action known problem identifier.
Location ID	Action location.
Max Intervention Date	Latest intervention date for the action.
Max Resolution Date	Latest resolution date for the action.
Net Charge	Action net charge.
Origin Action ID	Action origin identifier.
Origin Tool ID	Action origin tool identifier.
Parent Action ID	Action parent identifier.
Percent Complete	Action completion percentage.
Priority ID	Action priority identifier.
Renewal Date	Action renewal date.
Stage ID	Action stage identifier.
Start Date	Action start.
Status ID on Create	Action status when created.
Status ID on Terminate	Action status when terminated.
Supplier ID	Action supplier identifier.
Support Staff ID	Action support staff identifier.
Task Person Cost per Hour	Action task person cost per hour.
Task Planned Budget	Action task planned budget value.
Tax ID	Action tax identifier.
Time Cost	Action time cost value.
Time Cost Currency ID	Action time cost currency identifier.
Time Used to Complete	Time used to complete the action.
Validator ID	Action validator identifier.
Workflow ID	Action workflow identifier.
Workflow Value	Action workflow value

Outputs

This activity outputs the ID of action that was modified.

Set-EasyVistaRequest

The **Set-EasyVistaRequest** activity is used in a runbook to update an existing request or incident.

Discovery Options

Use the following options to connect to EasyVista and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to EasyVista Service Manager.

Required Parameters

You must configure the following parameters.

Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Request Number	Identifies the incident/request record to be updated.

Optional Parameters

You can use the following parameters, as needed, to change how the activity runs.

Asset ID	ID of an asset to be associated with the request. **Note:** It is recommended to specify either the *Asset ID* or the *Asset Serial Number* or the *Asset Tag*
Asset Serial Number	Serial number of an asset to be associated with the request. **Note:** It is recommended to specify either the *Asset ID* or the *Asset Serial Number* or the *Asset Tag*
Asset Tag	Asset tag of an asset to be associated with the request. **Note:** It is recommended to specify either the *Asset ID* or the *Asset Serial Number* or the *Asset Tag*
Available Field 1 .. 6	Request available field.
Budget (planned)	Planned budget for the request.
Budget Effective	Effective budget for the request.
Budget ID	Budget identifier.
Can Be Duplicated	Specifies if request can be duplicated.
CI Name	Configuration item name of a CI to be associated with the request. **Note:** It is recommended to specify either the *CI Name* or the *CI Number* or the *CI Serial Number.*

CI Number	Configuration item number of a CI to be associated with the request. **Note:** It is recommended to specify either the *CI Name* or the *CI Number* or the *CI Serial Number.*
CI Serial Number	Configuration item serial number of a CI to be associated with the request. **Note:** It is recommended to specify either the *CI Name* or the *CI Number* or the *CI Serial Number.*
Comment	Request comment.
Cost Center ID	Cost center identifier.
Delay (sec)	Request delay in seconds.
Description	Request description.
Effective Change End	Effective change request end.
Effective Change Start	Effective change request start.
End Date	Request end date.
Estimated Complete (%)	Estimated complete percentage of the request.
Estimated Net Price	Request estimated net price.
Expected Date	Request expected date.
Expected Duration (sec)	Request expected duration in seconds.
Expected End Date	Request expected end date.
Expected Start Date	Request expected start date.
External Reference	Request external reference.
First Call Resolution	Specifies if request was resolved on first call.
Impact	Request impact.
Is Major Incident	Specifies if this is a major incident
Is Template	Specifies if this is a template record.
Known Problem ID	Request known problem ID.
Mark 1, 2	Request mark.
Max Resolution Date	Latest date and time when issue can be resolved.
Net Price	Request net price.
Not Deduced Call	Not deduced call.
Order Net Price	Order net price.
Origin	Request origin.

Origin Tool ID	Origin tool identifier.
Planned Change End	Date and time when change is planned to end.
Planned Change Start	Date and time when change is planned to start.
PM Status ID	PM status identifier.
Project Name	Project name.
Project Start Date	Project start date and time
Quantity	Quantity.
Rental Net Price	Rental net price.
Requalification Processing	Requalification processing.
Requested Change End	Date and time requested for change to end.
Requested Change Start	Date and time requested for change to start.
Requestor Feedback	Indicates if record contains requestor feedback.
Requestor IP Address	Requestor IP address.
Requestor Phone	Requestor phone number.
Required Downtime	Required downtime.
Risk Amount	Risk amount.
Risk Level ID	Risk level identifier.
Submit Date	Request submit date.
Time Until Feedback (sec)	Duration until request received feedback, in seconds.
Time Until Solved (sec)	Duration until request was solved, in seconds.
Validation Level	Validation level.

Outputs

This activity outputs the request number of the incident/request ticket that was modified.

Suspend-EasyVistaRequest

The **Suspend-EasyVistaRequest** activity is used in a runbook to suspend (place on hold) a request or incident ticket.

Discovery Options

Use the following options to connect to EasyVista and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to EasyVista Service Manager.

Required Parameters

You must configure the following parameters.

Connection	A hash table containing connection information. This is typically obtained using a connection asset data source or Get-AutomationConnection activity.
Request Number	Identifies the incident/request to be suspended.

Optional Parameters

You can use the following parameters, as needed, to change how the activity runs.

Comment	Comment detailing suspend information.

Outputs

This activity outputs the request number of the incident/request ticket that was suspended.