

May 18, 2020

# Automation Anywhere Version A2019

# Legal Notices

© 2020 Automation Anywhere, Inc. All Rights Reserved.

See the list of Automation Anywhere trademarks at <https://www.automationanywhere.com/trademark>.

All other customer or partner trademarks or registered trademarks are owned by those companies.

The information contained in this documentation is proprietary and confidential. Your use of this information and Automation Anywhere Software products is subject to the terms and conditions of the applicable End-User License Agreement and/or Nondisclosure Agreement and the proprietary and restricted rights notices included therein.

You may print, copy, and use the information contained in this documentation for the internal needs of your user base only. Unless otherwise agreed to by Automation Anywhere and you in writing, you may not otherwise distribute this documentation or the information contained here outside of your organization without obtaining Automation Anywhere's prior written consent for each such distribution.

Examples and graphics are provided only as reference information and might not match your site.

# Content

Build.....	4
Bot editor for creating bots.....	4
Actions palette content for bot creation.....	5
Universal Recorder overview.....	223
Recording tasks from remote applications using AISense.....	227
Working with bots.....	230
Keyboard shortcuts.....	238
Variables overview.....	239
Bot dependencies.....	251
Attended automation using Enterprise A2019.....	253
Using interactive forms.....	254
Adding a trigger to run a bot.....	257
Get started building bots.....	261
Build a Go be Great bot.....	262
Build a basic bot that uses a desktop application.....	263
Extract data from a web table and save it to a file.....	265
Enter data into a web form from a worksheet.....	266
Perform an action based on cell value.....	267
Use Python script to join a list.....	268
Build advanced bots and packages.....	270
Package Software Development Kit (SDK).....	270
Build a bot using REST Web Services and JavaScript actions.....	304
Post to Salesforce through custom app with OAuth 2.0.....	308
Use Python to build a bot to parse JSON response.....	312
Use JavaScript to build a bot to take user input.....	315
Bot developer recommendations.....	317
Bot Store submissions checklist.....	317
Building reusable bots.....	319
Building reusable packages.....	322
Troubleshooting and debugging.....	324
Troubleshoot bot run issue.....	324
Debugger features.....	324
Bot agent log files.....	325

# Build

---

This collection of topics will get you started automating tasks and processes as quickly as possible.

- [Bot editor for creating bots](#)  
The Bot editor enables users to create and edit bots from any device, and from anywhere the user can access a web browser.
- [Attended automation using Enterprise A2019](#)  
Robotic Process Automation (RPA) enables users to automate repetitive business processes with bots. The collaboration between humans and software bots is called attended automation.
- [Get started building bots](#)  
Create and edit bots in the cloud-based Bot editor. Use the examples below to become familiar with features and learn to build bots in Enterprise A2019.
- [Build advanced bots and packages](#)  
Learn how to build action packages and advanced bots that include custom features such as scripting, and API calls. Find recommendations on bot and action package design and reusability.
- [Bot developer recommendations](#)  
Automation Anywhere provides a flexible platform for bot and package development. The information in this topic provide guidelines and recommendations on how to structure and develop robust and reusable bots and packages.
- [Troubleshooting and debugging](#)  
Troubleshooting and debugging information.

Related reference

[Actions palette content for bot creation](#)

[Variables overview](#)

Related information

[Training - Hello A2019 Bot: Getting Started with Building Bots](#)

## Bot editor for creating bots

The Bot editor enables users to create and edit bots from any device, and from anywhere the user can access a web browser.

### Overview

Because Enterprise A2019 is web-based, there is no separate client to download. Access the Bot editor by logging in to Enterprise A2019, navigating to BOTS > My bots, and then opening an existing bot or by creating a new one.

The Bot editor is a web-based environment and includes the following features:

- Universal Recorder to simplify capturing processes
- Three view options for bot creation:
  - Flow: Graphical representation of the process (default).
  - List: Sequential entries for each action.
  - Dual: Split screen of the Flow and List views.
- Powerful bot code management
- Python and JavaScript actions support inline scripting without linking, with drag-and-drop integration

- Rich variable passing, with no cross-language mapping required

## Accessing the Bot editor

To access the Bot editor, you must start creating or modifying a bot.

Go to My Bots in the Enterprise Control Room

- To create a bot, click the Create TaskBot icon on the top-right of the page.
- To modify a bot, click the bot from the list.

Watch the following video on how to get started with the Bot editor:

Get started with the Bot editor

- [Actions palette content for bot creation](#)  
Use actions to build an automation workflow and to instruct a bot what to do. In Enterprise A2019, the available actions are located in the Actions palette in the Bot editor.
- [Universal Recorder overview](#)  
Use the Universal Recorder to record interactions, such as click, read (data extraction), and write (data entry) with user interface (UI) objects on the desktop, taskbar, or in an application or browser window.
- [Recording tasks from remote applications using AISense](#)  
AISense is the artificial intelligence (AI) powered capability of Enterprise A2019 that helps you identify objects from an image or an application with a complex user interface (UI) and make automation in all environments faster and more accurate.
- [Working with bots](#)  
Create, record, run, and edit automated processes or bots.
- [Keyboard shortcuts](#)  
List of keyboard shortcuts supported.
- [Variables overview](#)  
Enterprise A2019 offers a variety of variables, each designed to hold specific types of data and is intended for specific use. Use the topics below to learn more about each variable and how to use them.
- [Bot dependencies](#)  
Bots dependencies are files and other bots that are required to run that bot successfully.

Related tasks

[Run bot now](#)

Related reference

[Editing bots](#)

## Actions palette content for bot creation

Use actions to build an automation workflow and to instruct a bot what to do. In Enterprise A2019, the available actions are located in the Actions palette in the Bot editor.

## Actions, packages, and dependencies

Actions are grouped into packages. For example, the Excel advanced package contains Excel-related actions, such as Open workbook, Go to cell, and Delete cell, that you can insert in to the Bot editor to automate a spreadsheet process.

Learn more about managing packages in [Managing packages](#). Developers can learn more about creating packages in [Package Software Development Kit \(SDK\)](#).

After an action from a package is used in a bot, that package becomes a dependent file of that bot. Learn more about managing bots and package dependencies in [Bot dependencies](#).

## Working with actions

Actions in the Bot editor are configurable. Double-click the action to see the fields and features that you can configure. Double-click or drag an action to insert it into the automation workflow.

Mouse over the vertical ellipsis at the top right of an action icon to access the following features:

### Copy action

Duplicate the action.

### Cut action

Copy the action and remove it from the bot code.

### Paste after action

Paste the copied action below the selected action.

Note: This option only appears when there is an action in the clipboard.

### Delete action

Remove the action from the bot code.

### Disable action

At run time, bot ignores the action until you enable it.

### Enable breakpoint

Pauses the bot for debugging purposes.

[Debugger features](#)

You can edit multiple actions at a time, using the toolbar at the top of the Bot editor.

### Copy items

Copy actions to a clipboard so they can be duplicated within the bot.

### Cut items

Copy the actions and remove them from the bot code.

### Paste items

Paste one or more actions after the highlighted action. If you have not highlighted an action, the actions are appended to the end of the automation sequence.

Note: This icon is only enabled when there are actions in the clipboard.

### Copy to shared clipboard

Copy actions, triggers, and metadata to a clipboard that is shared between bots.

### Paste from shared clipboard

Paste actions, triggers, and metadata from another bot.

Note: This icon is only enabled when there are actions in the shared clipboard.

## Resources

To learn more, see [Training - Install and upgrade actions without reinstallation](#). This course introduces you to packages, benefits, and installation.

Note: You must log in with a registered A-People Community [account](#) to access course.

Watch the following video on how to use actions in Enterprise A2019:

## Using actions

- [Analyze package](#)  
Use the actions in the Analyze package to specify the actions and variables to use in the Bot Insight dashboard and widgets. The Analyze package enables you to perform transactional analytics for the data that is logged by the variables when the bot runs.
- [App Integration package](#)  
Use the Capture text of window action in the App Integration package to extract text from a window and save it to a string variable.
- [Application package](#)  
Use the Open Program/File action in the Application package to launch an application or a file. This action supports .exe, .bat, script files, or shortcut paths.
- [AWS Comprehend NLP package](#)  
The AWS Comprehend NLP package contains actions that enable you to connect to and consume the Amazon Comprehend API to identify the language, sentiment, key phrases, and entities.
- [Boolean package](#)  
The Boolean package contains actions that enable you to do various operations on Boolean values.
- [Browser package](#)  
The Browser package contains actions that enable you to download files, find broken links, and launch a website. This package supports Internet Explorer, and Google Chrome browsers.
- [Bot migration package](#)  
The Bot migration package contains the Migrate bot action that enables you to migrate a bot (TaskBots and MetaBots) from 11.x to A2019.
- [Clipboard package](#)  
The Clipboard package contains actions that enable you to store a value in a string variable, use a stored value in another application, and clear values from the clipboard.
- [Comment package](#)  
Use the Comment package to insert a user-specified comment into your bot logic.
- [CSV/TXT package](#)  
The CSV/TXT package contains actions that enable you to open a CSV or text file, read data from that file, and assign the data to a Table variable. This package supports files encoded in ANSI, Unicode, UTF-8, or Win1251, and can process up to one million records.
- [Database package](#)  
Databases support internal operations of an enterprise by storing a variety of data, such as sales transactions, product catalogs, inventories, and customer profiles. Use the Database package to connect to a database, begin a transaction, and manipulate the stored data by retrieving, inserting, updating, deleting, and exporting it to a CSV file.
- [Datetime package](#)  
A datetime value consists of a date, time, and time zone. Automation Anywhere Enterprise stores datetime values in a Datetime variable. The Datetime package contains actions that enable you to perform various operations on datetime values. You can use these actions to manipulate and compare values in the Datetime variables.
- [Delay package](#)  
Use the Delay package to add a timed delay to the logic.
- [Dictionary package](#)  
The Dictionary package contains actions that enable you to do various operations on dictionary-type values.
- [DLL package](#)  
A dynamic-link library (DLL) file contains a shared library of functions that can be used by Windows programs. The DLL package uses a .dll file as reference and call functions from the bot.
- [Email package](#)  
The Email package contains actions to automate email-related tasks through Exchange Web Services (EWS), Microsoft Outlook, and other email servers. You can use these actions for sending, receiving, and modifying messages, folders, and the status of messages.

- [Error handler package](#)  
The Error handler package contains actions that enable you to easily handle exceptions that a bot encounters and transfers control to the other actions within that bot.
- [Excel basic package](#)  
The Excel basic package contains actions that enable you to automate many of the repetitive tasks in XLSX workbooks. You use these actions when Microsoft Excel is not available on the device that you want to use to automate Microsoft Excel-related tasks.
- [Excel advanced package](#)  
The Excel advanced package contains actions that enable you to automate many of the repetitive tasks when working with Microsoft Excel spreadsheets.
- [File package](#)  
The File package contains actions that enable you to automate various file-related operations such as creating, opening, copying, deleting, and renaming a file.
- [Folder package](#)  
The Folder package contains actions that enable you to automate folder-related operations.
- [FTP / SFTP package](#)  
Use the FTP / SFTP package to automate FTP / SFTP operations.
- [Fuzzy match package](#)  
Use the Fuzzy match action to compare the values of two strings or files for similarity. This action returns a decimal value; the closer the value to 1.0, the greater the similarity between the two strings.
- [G-Suite Apps package](#)  
The G-Suite Apps package contains the OAuth action, which enables you to authorize and connect to the G-Suite server. With this package, you only have to provide your credentials once.
- [Google Calendar package](#)  
The Google Calendar package contains actions that enable you to automate creating and deleting events.
- [Google Drive package](#)  
The Google Drive package contains actions that enable you to automate tasks related to files and folders.
- [Google Sheets package](#)  
The Google Sheets package contains actions that enable you to automate tasks involving cells, columns, rows, and sheets.
- [IBM Watson Authentication package](#)  
The IBM Watson Authentication package contains actions that enable you to authenticate the API token and location URL for each service, while connecting to and disconnecting from your IBM Cloud account. With this package, you only have to provide your credentials once.
- [IBM Watson Speech to Text package](#)  
This package supports the following audio file formats: flac, mpeg, mp3, ogg, pcm, wav, and webm. The following languages are supported: Arabic, Brazilian Portuguese, Chinese (Mandarin), English (United Kingdom and United States), French, German, Japanese, Korean, Spanish (Argentinian, Castilian, Chilean, Colombian, Mexican, and Peruvian).
- [If package](#)  
Use the actions in the If package to control the sequence of execution based on one or more conditions of a task.
- [Image Recognition package](#)  
The Image Recognition package contains actions that enable you to search for a user interface (UI) element in an application based on an image to automate a task in that application.
- [Interactive forms package](#)  
The interactive forms package contains actions that handle exceptions encountered by a bot. All the actions performed by users on the interactive forms can be monitored to execute logic using subtasks.
- [IQ Bot package](#)  
The IQ Bot package enables you to upload and download documents from an IQ Bot server.
- [JavaScript package](#)  
The JavaScript package contains actions to run a JavaScript from a bot.



- [Simulate keystrokes package](#)  
Use the Simulate keystrokes package to simulate keystrokes in Chinese (simplified and traditional), English, French, German, Japanese, Korean, Italian, or Spanish characters.
- [List package](#)  
The List package contains actions that enable you to perform various operations on a variable of the list data type.
- [Log To File package](#)  
Use the Log To File package to create a log file with data.
- [Loop package](#)  
Use the Loop package to run a sequence of actions repeatedly for a specific number of times or until a specific condition is met.
- [Message box package](#)  
Use the Message box action from the Message box package to insert a message box that shows a message when the task runs. For example, you can insert a Message Box action to follow a web form so that the action displays the message: `Web Form Filled and Complete`.
- [Microsoft LUIS NLP package](#)  
The Microsoft LUIS NLP package contains actions that enable you to connect to and consume the Microsoft Cognitive Services Text Analytics API to identify the language, sentiment, key phrases, and entities. This package supports the following languages: English, Chinese (Simplified), French, German, and Spanish.
- [Mouse package](#)  
Use the Mouse package to simulate mouse actions.
- [Number package](#)  
The Number package contains actions that enable you to perform various operations on a number variable. A number variable holds numeric values, including integers and decimals. It holds values from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807, and up to 15 decimal digits.
- [OCR package](#)  
The OCR package contains actions that enable you to extract text from images or applications.
- [Office 365 Calendar package](#)  
The Office 365 Calendar package contains actions that enable you to automate meeting-related tasks in Office 365 Calendar.
- [Office 365 Excel package](#)  
The Office 365 Excel package contains actions that enable you to automate tasks in the online version of Microsoft Excel.
- [Office 365 One Drive package](#)  
The One Drive package contains actions that enable you to automate many of the repetitive tasks in Microsoft cloud storage.
- [Play Sound package](#)  
The Play Sound package contains actions that enable you to play a beep sound and media file before or after an action is executed in a bot.
- [PGP package](#)  
Use the PGP (Pretty Good Privacy) package to automatically encrypt and decrypt files for security.
- [Printer package](#)  
Use the actions in the Printer package to automate retrieving and setting the default printer.
- [Prompt package](#)  
Use the Prompt package to accept an input value, a yes/no response, or to open a file or folder.
- [Python Script package](#)  
The Python Script package contains actions that enable Python Script functions in a task.
- [REST Web Service package](#)  
The REST Web Service methods (actions) work similarly to a representational state transfer (REST) client, using the REST request-response model.
- [SAP package](#)  
The SAP package contains actions to automate tasks and processes on a SAP application.

- [Screen package](#)  
Use the Screen package to automate the process of capturing screenshots. Using the actions in this package, you can capture an area of an application window, the entire computer screen, or an active open window and save it in a specified location in an image format.
- [SNMP package](#)  
The SNMP package allows you to automate network management tasks, such as retrieving and modifying data, and sending notification messages.
- [SOAP web service package](#)  
Use the SOAP web service action from the SOAP web service package to access and exchange information between two systems in XML format.
- [Step package](#)  
The Step package groups various actions together and runs them in a specific order. You can provide a relevant name for a step to identify the operation performed by the actions included in that step.
- [String package](#)  
Use the String package to perform various operations such as comparing two strings, retrieving the string length, or converting a string to uppercase or lowercase.
- [System package](#)  
Use the actions in the System package to automate locking, logging off, restarting, and shutting down the computer. Use these actions at the end of a task.
- [Task Bot package](#)  
Use the Run, Pause, and Stop actions in the Task Bot package to manage running one or more child bots from a parent bot or with a third-party software using an API.
- [Terminal Emulator package](#)  
The Terminal Emulator package contains actions that enable you to connect to and automate tasks on another machine. Use these actions to access and control operations on a remote machine. For example, you can run applications and access files on a different operating system.
- [Trigger loop package](#)  
The Trigger loop package enables you to run a series of actions when a trigger event occurs. You can insert multiple trigger loops within a bot or nest one trigger loop within another trigger loop.
- [VBScript package](#)  
The VBScript package contains actions that enable VBScript functions in a task.
- [Wait package](#)  
Use the actions in the Wait package to add a condition to wait for an application screen to change, or a separate window to open or close before proceeding to the next action.
- [Window package](#)  
Use the Window package to automate tasks relating to the window.
- [Workload package](#)  
The Workload package enables you to insert work items in a queue for workload automation. It also enables data chaining between multiple queues. You can orchestrate multiple bots, and enable optimal device utilization through the queueing mechanism of workload management.
- [XML package](#)  
Extensible Markup Language (XML) is a markup language designed to store and transport data. Use the actions in the XML package to automate the processing of XML data generated from web services and cloud computing applications.

Related reference

[Bot editor for creating bots](#)

## Analyze package

Use the actions in the Analyze package to specify the actions and variables to use in the Bot Insight dashboard and widgets. The Analyze package enables you to perform transactional analytics for the data that is logged by the variables when the bot runs.

### Actions in the Analyze package

The Analyze package includes the following actions:

Action	Description
Close	<p>Closes the transaction.</p> <ul style="list-style-type: none"> <li>In the Transaction name field, enter the transaction name that you provided in the Open action.</li> <li>In the Dictionary variable, select the following options: <ul style="list-style-type: none"> <li>All user-defined variables: Include all the string, numeric, and date time user-defined variables from the bot to populate the Bot Insight dashboards.</li> <li>Manually select which variables to include in the Bot Insight dashboards.</li> </ul> </li> </ul>
Open	<p>Opens the transaction. Insert the actions that you want to include for analysis between the Open and Close actions. In the Transaction name field, enter the transaction name.</p>

## App Integration package

Use the Capture text of window action in the App Integration package to extract text from a window and save it to a string variable.

The App Integration package supports the following technologies:

- UNIX shells, such as Cygwin, PowerShell, and GIT shell
- Windows applications, such as Calculator, Command Prompt, and Notepad

### Building bots with the Capture text of window action

1. Open the application or file from which you want to capture the text using the Application > Open program/file action.

#### Application package

2. Extract the text using the App integration > Capture text of window action.
3. Use any of the following actions from the String package to manipulate the captured text:
  - Extract text: Extracts text from the source string using logical operators.
  - Split: Splits the string into multiple strings and stores the output in a list variable.
  - Trim: Trims blanks and white spaces from a string.

## String package

### Application package

Use the Open Program/File action in the Application package to launch an application or a file. This action supports .exe, .bat, script files, or shortcut paths.

### Actions in the Application package

The Application package includes the following action:

Action	Description
Open program/file	<p>Opens an application or file.</p> <ul style="list-style-type: none"> <li>Enter the name of the application or file, or insert a variable. For example, chrome.exe. Note: You can also enter the full file path in this field.</li> <li>Optional: If you did not enter the full file path in the previous field, enter the Start in path.  For example, C:\Program Files (x86)\Google\Chrome\Application\</li> <li>Optional: In the Parameters field, further specify the program to open.  For example, if you are using the command line, enter the arguments in this field.</li> </ul>

### Conditions in the Application package

Use the following conditions inside an If loop to control the flow of execution in an automation task:

Condition	Description
File	<ul style="list-style-type: none"> <li>File date</li> </ul> <p>Use this condition to verify the date and time that the specified file was created or modified. Specify a date range using the Is within last, Is between, or Is before options. For the Is within last option, specify the number of days or time (in hours, minutes, and seconds). Enter the amount of time (in seconds) to wait for this condition to be true.</p>

Condition	Description
	<ul style="list-style-type: none"> <li>File exists and File does not exist</li> </ul> <p>Use these conditions to execute an action based on whether a file exists. For example, if a data file exists, format the file and upload it to a database.</p> <ul style="list-style-type: none"> <li>File size</li> </ul> <p>This condition verifies if the specified file is larger, smaller, not the same, or the same as the size you specify.</p> <p>Enter the amount of time (in seconds) to wait for this condition to be true.</p> <p>See <a href="#">If package</a>.</p>

## AWS Comprehend NLP package

The AWS Comprehend NLP package contains actions that enable you to connect to and consume the Amazon Comprehend API to identify the language, sentiment, key phrases, and entities.

## Before you start

You require the following information for an existing AWS account to use these actions:

- Access key and Secret key: Credentials that authenticate Automation Anywhere Enterprise with your AWS account.

See [Access Keys \(Access Key ID and Secret Access Key\)](#).

- Region: Specifies the AWS service endpoint.

See [AWS Service Endpoints](#).

## Actions in the AWS Comprehend NLP package

Action	Description
Detect language	Identifies the language of the provided content and returns it in ISO 639-1 language code. The output is stored in a string variable. This action supports over 100 languages. For the full list, see <a href="#">Detect the Dominant Language</a> .
Get key phrases	Identifies the main points and returns a list of key phrases. For example, if the input text is about a basketball game, this action returns the names of teams, the name of the venue, and the final score.

Action	Description
Get named entities	<p>Identifies the entities in the provided content such as people, places, organizations, date/time, quantities, branded products, and book titles.</p> <p>The output is stored in a dictionary variable, where each name is a key, and the corresponding entity is the value.</p>
Get sentiment	<p>Analyzes the provided content and returns the overall sentiment and scores for all possible sentiments. An example output of the dictionary values:</p> <pre>POSITIVE {Positive: 0.66238534, Negative: 0.0013064129, Neutral: 0.33621928, Mixed: 8.892125E-5}</pre> <p>The output is stored in a dictionary variable containing two keys and their corresponding values: <code>sentiment</code> and <code>score</code>.</p>

## Boolean package

The Boolean package contains actions that enable you to do various operations on Boolean values.

## Actions in the Boolean package

The actions in the Boolean package accept a variable as an input and assign the output to a variable. These actions enable you to compare two Boolean values, convert a Boolean value to a string or numeric value, and convert a string value to a Boolean value.

The Boolean package includes the following actions:

Action	Description
Assign	<p>Assigns a constant value (True or False) or a user-defined value to a Boolean value.</p> <ul style="list-style-type: none"> <li>Select the source Boolean variable or value.</li> <li>Select the variable to use to store the output from the Destination Boolean variable list.</li> </ul> <p>The output is stored in a Boolean variable.</p>
Compare to	<p>Compares two Boolean values and assigns the output to a numeric variable.</p> <ul style="list-style-type: none"> <li>Select the Boolean variables to compare from the Select the first Boolean variable and Select the second Boolean variable lists.</li> <li>Select the variable to use to store the output from the Assign the output to number variable list.</li> </ul> <p>The output is stored in a number variable.</p>

Action	Description
Equal to	<p>Verifies whether the two Boolean values are equal or not, and assigns the output to a Boolean variable.</p> <ul style="list-style-type: none"> <li>Select the variables that contain the Boolean values to verify from the Select the first Boolean variable and Select the second Boolean variable lists.</li> <li>Select the variable to use to store the output from the Assign the output to number variable list.</li> </ul> <p>The output is stored in a number variable.</p>
Invert	<p>Converts a Boolean value to the opposite value (True to False and False to True), and assigns the output to a variable.</p> <ul style="list-style-type: none"> <li>Select the Boolean value to convert. Choose from False, True, or a Variable.</li> <li>Select the numeric variable to use to store the converted value from the Assign the output to variable list.</li> </ul>
To number	<p>Converts a Boolean value to a numeric value. This action converts True to 1 and False to 0.</p> <ul style="list-style-type: none"> <li>Select the Boolean variable to convert from the Select Boolean variable list.</li> <li>Select the numeric variable to use to store the converted value from the Assign the output to variable list.</li> </ul>
To string	<p>Converts a Boolean value to a string value.</p> <ul style="list-style-type: none"> <li>Select the Boolean variable to convert from the Select Boolean variable list.</li> <li>Select the string variable to use to store the converted value from the Select the string variable to store the result list.</li> </ul>

## Compare results for the Compare to action

The following table illustrates how two Boolean values are compared using the Compare to action and their output:

Boolean value 1	Boolean value 2	Compare result
True	True	0
True	False	1
False	True	-1
False	False	0

## Compare results for the Equal to action

The following table illustrates how two Boolean values are compared using the Equal to action and their output:

Boolean value 1	Boolean value 2	Equal result
True	True	True
True	False	False
False	True	False
False	False	True

## Browser package

The Browser package contains actions that enable you to download files, find broken links, and launch a website. This package supports Internet Explorer, and Google Chrome browsers.

## Actions in the Browser package

The Browser package includes the following actions:

Action	Description
Download files	<p>Downloads and saves files from URLs.</p> <ul style="list-style-type: none"> <li>Specify the URL of the file you want to download.</li> <li>In the Save to location field, enter the location where you want to save the file.</li> </ul>
Find broken links	See <a href="#">Using Find broken links action</a> .
Launch website	<p>Launches the browser.</p> <ul style="list-style-type: none"> <li>In the URL field, specify the website you want to open.</li> <li>Select your browser from Default Browser, Internet Explorer, or Google Chrome.</li> </ul> <p>Note: Ensure that Internet Explorer or Google Chrome is set as the default browser. If any browser that is not listed is set as the default browser for the device, the bot might encounter an error. When using commands such as OCR and Image Recognition, the browser must finish rendering before the commands execute.</p>

## Using Find broken links action

The Find broken links action enables you to find links that are not working on a specific page or an entire website.



## Procedure

Follow these steps to find broken links:

1. In the Actions palette, double-click or drag the Find broken links action from the Browser package.
2. In the Page or URL field, enter the URL of the page or website.
3. In the Scope option, choose either check only this page or check the whole site.
4. In the Save list to location field, specify the location of the CSV file.
5. Select the Append to already existing csv file check box if you want to append the data to an existing CSV file.
6. Select an option from the Encoding list to specify the encoding that is applied on the file.
7. In the Number of parallel threads field, enter the number of processes you want to run simultaneously.  
A higher number of parallel threads results in faster execution.  
Note: The maximum value you can provide in the field is 99.
8. In the Time out field, specify the maximum time the system must wait to receive a response from each URL link.
9. Click Apply.

## Bot migration package

The Bot migration package contains the Migrate bot action that enables you to migrate a bot (TaskBots and MetaBots) from 11.x to A2019.

[View certified 11.x versions](#)

## Action in the Bot migration package

The Bot migration package includes the following action:

Action	Description
Migrate bot	<p>Migrates the version 11.x bot file to A2019 format and uploads the migrated file to the specified location in your private repository with the same name as .atmx and .mbot file. This action only migrates the bot you specify, but does not migrate its dependencies. Dependencies are the bots and other files that are required to run the bot. You need to migrate the dependent bots separately and upload other files manually to Enterprise Control Room.</p> <ul style="list-style-type: none"> <li>• Use the Control Room file, Desktop file, or Variable tab to specify the location of the bot you want to migrate.</li> <li>• In the Output folder path field, specify the location where you want to save the bot file that is migrated to A2019 format.</li> </ul> <p>The system does not upload the bot if it fails during the migration process. The system creates an XML report at the same location that provides information that helps you to troubleshoot if the system encounters an error during migrating the bot file.</p> <ul style="list-style-type: none"> <li>• Select the Overwrite the file if exists check box to overwrite an existing bot file.</li> </ul>

Action	Description
	Note: You can only migrate bot (.atmx) files; migration of MetaBot (.mbot) files is not supported.

## Clipboard package

The Clipboard package contains actions that enable you to store a value in a string variable, use a stored value in another application, and clear values from the clipboard.

### Actions in the Clipboard package

The Clipboard package includes the following actions:

Action	Description
Clear	Clears the clipboard. This action will remove any value that is stored in the clipboard.
Copy from	Copies the value that is stored in the clipboard and uses it in a String variable that you select from the Assign the output to variable list.
Copy to	Stores a value in the clipboard. You can either enter the value or specify the variable that contains the value in the Value field.

## Comment package

Use the Comment package to insert a user-specified comment into your bot logic.

### Action in the Comment package

The Comment package includes the following action:

Action	Description
Comment	Inserts a comment. Note: A comment is saved and displayed as a single line. A comment with multiple lines is displayed as a single line, followed by an ellipsis when the comment is saved. Comments are ignored when the bot runs.

## CSV/TXT package

The CSV/TXT package contains actions that enable you to open a CSV or text file, read data from that file, and assign the data to a Table variable. This package supports files encoded in ANSI, Unicode, UTF-8, or Win1251, and can process up to one million records.

Perform the following actions within the CSV/TXT package as part of using the set of available actions:

1. Open the file to be used in the automation. See [Using the Open action for CSV/TXT file](#).
2. Use the Read action to retrieve values from a CSV or TXT file and store them in a Table variable. See [Using Read action](#).  
Note: You can use this action in a Loop action to read data from each row in the file and assign it to a record variable. See [Using Read action in loop](#).
3. After you have automated the CSV/TXT-related tasks, close the file using the Close action. Enter the session name that was used to open the file with the Open action.

Related reference

[Loop package](#)

[User-defined variables](#)

## Using the Open action for CSV/TXT file

This action enables you to specify the delimiter used in the file, whether to trim the spaces, and the encoding applied on the file.

To open a CSV or text file, do the following:

### Procedure

1. Double-click or drag the Open action from the CSV/TXT node in the Actions palette.
2. Select any of the following options to specify the location of the CSV or text file to open:
  - From 'My Bots': Enables you to select a file that is available in a folder within Automation Anywhere Enterprise.
  - From local device: Enables you to select a file that is available on your device.
  - Select an existing file variable: Enables you to specify the file variable that contains the location of the file to open.
3. Select the Contains header check box if the file contains a header row and you want to retrieve values from that row.
4. Select any of the following options to specify the Delimiter used in the file:
  - Comma
  - Tab
  - Regional list separator: Enables you to specify the delimiter as defined in the regional settings. Automation Anywhere Enterprise supports the default regional list separators of English, German, French, Italian, and Spanish.

For example, semi-colon (;) will be used as a list separator if the regional settings are configured to German.

You can configure the regional settings with the globe icon at the top-right of the page.

- Newline
  - Other: Enables you to specify a delimiter other than the options listed above.
5. Select the Trim leading spaces and Trim trailing spaces check boxes to trim additional leading and trailing spaces from the data.
  6. Select an option from the Encoding list to specify the encoding that is applied on the file.
  7. Click Apply.

## Next steps

Use the Read action to retrieve the table values and assign them to a variable in order to perform operations with the data. Do one of the following:

- [Using Read action in loop](#)
- [Using Read action](#)

## Using Read action

Use the Read action to retrieve values from a CSV or TXT file and insert them into a Table variable in order to perform operations with the data values.

To retrieve values from a CSV/TXT file, do the following:

### Procedure

1. Double-click or drag the Read action from the CSV/TXT node in the Actions palette.
2. Enter the name of the session that you have used to open the CSV or text file in the Open action.
3. Select a Table variable from the Assign value to the variable list.  
Create a variable if it does not already exist.
4. Click Apply.

## Next steps

Perform operations with the values in the Table variable using the [Data Table package](#).

## Using Read action in loop

Use the Read action within a Loop action to read the data of each row available in a CSV or text file.

To use a Read action in a Loop action, do the following:

### Procedure

1. Double-click or drag the Loop action from the Loop package in the Actions palette.
2. Select the For each row in CSV/TXT option from the Iterator list.
3. Enter the name of the session that you have used to open the CSV or text file in the Open action.
4. Select a Record variable from the Assign the current row to this variable list.
5. Click Apply.

## Database package

Databases support internal operations of an enterprise by storing a variety of data, such as sales transactions, product catalogs, inventories, and customer profiles. Use the Database package to connect to a database, begin a transaction, and manipulate the stored data by retrieving, inserting, updating, deleting, and exporting it to a CSV file.

## Before you start

Perform the following actions within the Database package as part of using the set of available actions:

1. Establish a connection with the database server using the Connect action.

See [Using Connect action for database](#).

2. Choose from the following:

- Use the Read from action to retrieve records from the database. [Using Read from action](#)
- If you are automating a task that involves making changes to the database, insert the Begin database transaction action, followed by the actions that automate the changes.

This action ensures that all records are updated or deleted in their entirety, and prevents accidental updates or deletions of incomplete data if the bot encounters an error during run time.

3. If you used the Begin database transaction action in this session, insert the End database transaction to commit the changes.
4. Every set of database actions ends with the Disconnect action to terminate the connection to the database server.

## Actions in the Database package

The Database package includes the following actions:

Action	Description
Begin database transaction	<p>Starts a database transaction.</p> <p>The actions that you insert between the Begin database transaction and End database transaction actions are treated as a single unit. The bot must run all of the actions successfully in order to update the database. This prevents a partial entry in the event that one of the actions fail.</p> <p>For example, in double-entry accounting every debit requires the recording of a credit. If a company receives \$5000 of products, the accountant must debit \$5000 to inventory and credit \$5000 to accounts payable.</p> <p>Insert the actions that record these entries between the Begin database transaction and End database transaction actions to ensure that either both entries are recorded or neither is recorded in the database.</p> <p>In the Session name field, enter the name of the session you used to connect to the database server in the Connect action.</p>
Connect	See <a href="#">Using Connect action for database</a> .

Action	Description
Disconnect	Disconnects from a database. In the Session name field, enter the name of the session you used to connect to the database server in the Connect action.
End database transaction	Commits all the database operations that were performed with the actions that followed the Begin database transaction action, under the condition that the bot successfully ran those actions. In the Session name field, enter the name of the session you used to connect to the database server in the Connect action.
Insert/Update/Delete	<p>Executes an INSERT, UPDATE, or DELETE statement from the database.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the name of the session you used to connect to the database server in the Connect action.</li> <li>In the Statement field, enter the SQL statement to insert, update, or delete the records. <ul style="list-style-type: none"> <li>Use an INSERT statement to create new records in a table: <pre>INSERT INTO table_name (column1, column2, column3, ...) VALUES (value1, value2, value3, ...);</pre> </li> <li>Use an UPDATE statement to modify a record: <pre>UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition;</pre> </li> <li>Use a DELETE statement to remove a record: <pre>DELETE FROM table_name WHERE condition;</pre> </li> </ul> </li> <li>In the Timeout for the query in seconds field, specify the time within which the statement execution should stop, even if the execution is not completed.</li> </ul>
Manage stored procedure	<p>Creates, updates, and deletes a stored procedure within the specified database. A stored procedure is SQL code saved to the database, enabling you to run it repeatedly.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the name of the session you used to connect to the database server in the Connect action.</li> <li>In the Enter full command field, enter the command to create, update, or delete the stored procedure. You can specify input and output parameters for the command; you provide values or variables holding the values in the Run stored procedure action. Recommended: Declare a delimiter and use it to close the SQL statement. MySQL example: In this example, the bot checks if there is a procedure in the database named <code>sum_of_two</code>. If not, the bot creates a procedure that</li> </ul>

Action	Description
	<p>accepts two input parameters (num1 and num2), adds them together, and produces the sum as the output parameter:</p> <pre> DROP PROCEDURE IF EXISTS sum_of_two; DELIMITER \$\$ CREATE PROCEDURE sum_of_two(IN num1 INT,IN num2 INT,OUT sot INT) BEGIN SET sot := num1 + num2; END \$\$ </pre> <ul style="list-style-type: none"> <li>In the Timeout for the query in seconds field, specify the time within which the statement execution should stop, even if the execution is not completed.</li> </ul>
Read from	See <a href="#">Using Read from action</a> .
Run stored procedure	See <a href="#">Using Run stored procedure action</a> .

Related reference

[Loop package](#)

[Variables overview](#)

## Using Connect action for database

Use the Connect action to establish a connection with the database server that you want to use to automate database-related tasks. This must be the first action you use to automate a database-related task.

Specify the details of a database server and associate it with a session name. Use the session name provided in this action in the other actions so that you do not have to provide the details of the database server in those actions.

### Procedure

To establish a connection with a database server, follow these steps:

1. Double-click or drag the Connect action from the Database package in the Actions palette.
2. Enter a unique name for the session in the Session name field.
3. Select the Default or User defined connection option.
  - If you select the Default option, you can directly enter the connection string for any of the supported database types.

To ensure a more secure automation, use a Credential Vault variable for the connection string.

- If you select the User defined option, select the database type from the available options and complete the following fields based on the selection:

Database Type	Options
Microsoft SQL Server, PostgreSQL	<ul style="list-style-type: none"> <li>Server name: Enter the name of the database server you want to connect to.</li> <li>Database name: Enter the database name.</li> <li>Username: Enter the username you want to use to access the database server. To ensure a secure user name, select a Credential Vault variable. Otherwise, enter a value.</li> <li>Password: Enter the password for the username you have provided. To ensure a secure password, select a Credential Vault variable. Otherwise, enter a value.</li> <li>Instance name: Enter a name for this connection instance.</li> </ul> <a href="#">Connect to a Microsoft SQL Server with Windows authentication</a>
MySQL	Enter the same options as in the Microsoft SQL Server database type. Also, enter the port number. The default port number is 3306.
Microsoft Access, SQLite	Select the database file path from: <ul style="list-style-type: none"> <li>My bots folder</li> <li>local device</li> <li>existing file variable</li> </ul>
Oracle	<ul style="list-style-type: none"> <li>Server name: Enter the name of the Oracle server you want to connect to.</li> <li>Oracle system id (sid): Enter the system ID.</li> <li>Username: Enter the username you want to use to access the Oracle server. To ensure a secure user name, select a Credential Vault variable. Otherwise, enter a value.</li> <li>Password: Enter the password for the username you have provided. To ensure a secure password, select a Credential Vault variable. Otherwise, enter a value.</li> <li>Port: Enter the port number. The default port number is 1521.</li> </ul>

- Select a driver file from the My bots folder, the local device, or a file variable.
- Click Apply.
- Click Save.

## Next steps

Choose from the following:

- Use the Read from action to retrieve records from the database. [Using Read from action](#)
- If you are automating a task that involves making changes to the database, insert the Begin database transaction action.



This action ensures that all records are updated or deleted in their entirety, and prevents accidental updates or deletions of incomplete data if the bot encounters an error during run time. [Database package](#)

## Connect to a Microsoft SQL Server with Windows authentication

Configure your device and the Connect action from the Database package to automate connecting to a Microsoft SQL Server with Windows NT authentication.

### Procedure

To use Windows NT authentication for connecting to the Microsoft SQL Server, follow these steps:

1. Download the latest JDBC driver from Microsoft.  
<https://docs.microsoft.com/en-us/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server?view=sql-server-ver15>
2. Unzip the package and upload the mssql-jdbc-7.2.2.jre8.jar file to the Enterprise Control Room repository.  
how to upload files
3. Copy the auth\x64\sqljdbc\_auth.dll from the zip package to the following file paths:
  - C:\Windows\System32
  - C:\Program Files\Automation Anywhere\Bot Agent\jre\binThis enables Windows authentication, because the JDBC driver cannot perform that by default.
4. Select the Default connection option.
5. Enter the connection string: jdbc:sqlserver://localhost;databaseName=Test;integratedSecurity=true;.
6. Select the Use Specific database option.
7. Click Browse to select the drive file you uploaded in step two.
8. Click Apply.
9. Click Save.

### Next steps

Choose from the following:

- Use the Read from action to retrieve records from the database. [Using Read from action](#)
- If you are automating a task that involves making changes to the database, insert the Begin database transaction action.

This action ensures that all records are updated or deleted in their entirety, and prevents accidental updates or deletions of incomplete data if the bot encounters an error during run time. [Database package](#)

## Using Run stored procedure action

Stored procedures are a set of SQL statements that are created and stored in the database. These SQL statements might be complex and have to run multiple times. Use the Run stored procedure action to execute existing stored procedures.

## Procedure

To automate a task of executing a stored procedure, follow these steps:

1. Enter the name of the session you used to connect to the database server in the Connect action.  
You do not have to provide the details of the database server here because you have already associated those details with the session name when using the Connect action.
2. Enter the name of the stored procedure.
3. Enter the input parameter value if the stored procedure accepts parameters.  
For example, if running the example function from the Manage stored procedure action, provide the following two numeric values for the `sum_of_two` function to add:

```
num1=5, num2=10
```

4. In the Output parameter value field, provide a variable to hold the output if the stored procedure returns a value.  
Following the example above, when the bot runs, the variable you assign in this field holds the numeric value 15.
5. Enter the maximum number of records to retrieve.  
You can limit the results of the execution.
6. Optional: Enter a timeout value.  
When the specified time expires, the statement execution stops even if the execution is not completed.
7. Select the Export data to CSV option to save the retrieved data.
  - a) Select the file path from the My bots folder, the local device, or an existing file variable.
  - b) Select the CSV file encoding to be either ANSI, UNICODE, or UTF8.
  - c) Select whether to export the CSV file with or without the column headers.

With column headers

CustomerName	City
Manny	Pittsburgh
Kate	Los Angeles
John	Boston

Without column headers

Manny	Pittsburgh
Kate	Los Angeles
John	Boston

- d) Specify whether to overwrite the file or append the data to the existing file if a CSV file with the same name exists.
8. Click Apply.
  9. Click Save to save the automation.

# Using Read from action

Use the Read from action to retrieve records from the database and save the retrieved data in a CSV file. This action enables you to retrieve up to one million records from the database.

## Procedure

To automate the task of selecting and saving a set of records, follow these steps:

1. Enter the name of the session you used to connect to the database server in the Connect action.  
You do not have to provide the details of the database server here because you have already associated those details with the session name when using the Connect action.
2. Enter the SELECT statement to specify the column and table names.  
This field supports SQL syntax. For example, `SELECT CustomerName, City FROM Customers`
3. Enter the maximum number of records to retrieve.
4. Optional: Enter a timeout value.  
When the specified time expires, the statement execution stops even if the execution is not completed.
5. Select the Export data to CSV option to save the retrieved data.
  - a) Select the file path from the My bots folder, the local device, or an existing file variable.
  - b) Select the CSV file encoding to be either ANSI, UNICODE, or UTF8.
  - c) Select whether to export the CSV file with or without the column headers.

With column headers

CustomerName	City
Manny	Pittsburgh
Kate	Los Angeles
John	Boston

Without column headers

Manny	Pittsburgh
Kate	Los Angeles
John	Boston

- d) Specify whether to overwrite the file or append the data to the existing file if a CSV file with the same name exists.
6. Click Apply.
7. Click Save.

## Data Table package

The Data Table package contains actions that enable you to perform various operations on the values of table variables. Use these actions to join or merge content, search for specific values, insert rows and columns, remove duplicate rows, and write values to a file.

## Actions in the Data Table package

The Data Table package includes the following actions:

Action	Description
Assign	<p>Assigns values to a table variable. You can use this action to manually enter the table data.</p> <ul style="list-style-type: none"> <li>• Select whether to create a table with or without values. <ul style="list-style-type: none"> <li>• If you select to create a table with values, either manually enter the values or select a source table variable.</li> <li>• If you select to create a table without values, specify the number of rows and columns.</li> </ul> </li> <li>• Select a table variable to hold the output. You can either select the source table variable to overwrite the values, or select a different table variable.</li> </ul>
Clear content	<p>Clears the contents of the specified table variable. This action clears the content until the execution of the bot is completed. After the execution of the bot is completed, the original values of the table variables are restored. It enables you to reuse an existing table variable to store values that are used only during the execution of the bot.</p> <p>Note: The data type of the values must be same as the data type of the columns in the table variable.</p>
Delete column	<p>Deletes a specific column.</p> <ul style="list-style-type: none"> <li>• Select the Table variable name from the drop-down list.</li> <li>• Specify the column name or column index to delete.</li> </ul> <p>Note: The index count starts from 0.</p>
Delete row	<p>Deletes a specific row.</p> <ul style="list-style-type: none"> <li>• Select the Table variable name from the drop-down list.</li> <li>• Specify the row name or row index to delete.</li> </ul> <p>Note: The index count starts from 0.</p>
Get number of columns	Retrieves the number of columns and assigns it to a Number variable. You have the option to select the non-empty columns or include all the columns in the selection.
Get number of rows	Retrieves the number of rows and assigns it to a Number variable. You have the option to select the non-empty rows or include all the rows in the selection.
Insert column	<p>Inserts a column from one table to another table.</p> <p>For destination table:</p> <ul style="list-style-type: none"> <li>• Select the destination table variable form the Enter into data table list.</li> <li>• Select where to insert the column: first index, last index, or enter the column index number.</li> </ul> <p>Note: The index count starts from 0.</p> <p>For source table:</p>

Action	Description
	<ul style="list-style-type: none"> <li>Select the source table variable from the Insert from table list.</li> <li>Enter the name or index number of the column to insert.</li> </ul>
Insert row	<p>Inserts a row into the table.</p> <ul style="list-style-type: none"> <li>Select the Table variable name from the drop-down list.</li> <li>Select where to insert the row: first index, last index, or enter the row index number. Note: The index count starts from 0.</li> </ul>
Join	<p>See <a href="#">Using Join action</a>.</p> <p>Use the <a href="#">Join type output examples</a> to help you decide whether to use the Join or Merge action.</p>
Merge	<p>See <a href="#">Using Merge action</a>.</p> <p>Use the <a href="#">Merge output example</a> to help you decide whether to use the Join or Merge action.</p>
Remove duplicate rows	Deletes duplicate rows (if any exist). Select the Table variable name from the drop-down list.
Search for a value	<p>Searches for a specific value in the table, returns the row and column numbers in which the value occurs, and assigns the row and column numbers to a List variable. For example, if you search for the value <code>abc</code> that is available at the fourth row and third column of a table variable, the action returns 3, 2 as output. As the index number for the row and column starts with zero, the values in the output indicates the fourth row and third column.</p> <ul style="list-style-type: none"> <li>Select the Table variable name from the drop-down list.</li> <li>Enter the value to search for.</li> <li>Select Match case to ensure the case matches the value searched.</li> </ul>
Set cell value	<p>Updates the value of a specific cell.</p> <ul style="list-style-type: none"> <li>Select the Table variable name from the drop-down list.</li> <li>Enter the row index. For example, to set a value to the first row, enter 0</li> <li>Select whether to specify the column by Name or Index. <ul style="list-style-type: none"> <li>If you select to specify by name, enter the column name. Capitalization is not important.</li> <li>If you select to specify by index, enter the column index. For example, to set a value to the first column, enter 0</li> </ul> </li> <li>Enter the value to set.</li> </ul>

Action	Description
Sort	<p>Sorts the table data by a column. This action enables you to sort numeric and text data.</p> <ul style="list-style-type: none"> <li>• Select the Table variable name from the drop-down list.</li> <li>• Specify the column name or column index to sort by.</li> <li>• Select the order in which to sort: ascending or descending.</li> </ul>
Write to file	See <a href="#">Using Write to file action</a> .

## Example

[Extract data from a web table and save it to a file](#)

## Using Join action

Use the Join action to combine content from two Table variables. This action enables you to combine content based on the shared values of a specific column in the tables and store the content into a third Table variable or one of the two source tables.

To join the content from two Table variables, do the following:

### Procedure

1. Double-click or drag Data table > Join.
2. Select the first Table variable to use from the Enter first data table name list.
3. Enter the name of the column from the first Table variable that contains the data to join.
4. Select the second Table variable to use from the Enter second data table name list.
5. Enter the name of the column from the second Table variable that contains the data to join.
6. Select an option to specify the type of join. For more information, see [Join type output examples](#).
  - inner join: Returns only the records that have matching values in the selected columns in both tables.
  - left outer join: Returns all records from the first table, and the matched records from the second table.
  - right outer join: Returns all records from the second table, and the matched records from the first table.
  - full outer join: Returns all records when there is a match in either left or right table.
7. Select the table variable to store the combined values from the Enter name of data table in which to join list.
8. In the Assign value to variable list, select a string variable.

## Join type output examples

Select a join type option based on your desired output. Use the examples below to guide your selection.

### Inner join

Returns only the records that have matching values in the selected columns in both tables.

For example, if you have a table of employees and their departments, and a table of employees and their pay rates, this option will return a table of the employees that exist in both tables, and their departments and pay rates.

Table 1: Employees and Departments

Employee	Department
John	101
Jill	102
Mike	103
Betty	104
Cindy	105

Table 2: Employees and Pay rates

Employee	Pay rate
John	50
Betty	50
Mike	40
Jill	35
Dan	45

Table 3: Employees, Departments, and Pay rates

Employee	Department	Pay rate
John	101	50
Jill	102	35
Mike	103	40
Betty	104	50

## Left outer join

Returns all records from the first table, and the matched records from the second table.

Using the example Tables 1 and 2, this option returns the following table:

Table 3: Employees, Departments, and Pay rates

Employee	Department	Pay rate
John	101	50
Jill	102	35
Mike	103	40
Betty	104	50

Employee	Department	Pay rate
Cindy	105	

## Right outer join

Returns all records from the second table, and the matched records from the first table.

This option returns the following table:

Table 3: Employees, Departments, and Pay rates

Employee	Department	Pay rate
John	101	50
Betty	104	50
Mike	103	40
Jill	102	35
Dan		45

## Full outer join

Returns all records when there is a match in either left or right table.

This option returns the following table:

Table 3: Employees, Departments, and Pay rates

Employee	Department	Pay rate
John	101	50
Jill	102	35
Mike	103	40
Betty	104	50
Cindy	105	
Dan		45

## Using Merge action

Use the Merge action to append the content from the second Table variable to the first Table variable, and store the merged content into a third Table variable or one of the two source Table variables.

Use this action when the two source tables contain identical column headers. To combine Table variables containing different column headers, use the Join action. To merge the content, do the following:



## Procedure

1. Double-click or drag Data table > Merge.
2. Select the first Table variable you want to use from the Enter first data table name list.
3. Select the second Table variable you want to use from the Enter second data table name list.
4. Select the Table variable that you want to use to store the merged data from the Enter name of data table in which to merge list.
5. Click Apply.

For more information, see [Merge output example](#)

## Merge output example

The Merge action appends the content from one Table variable to another Table variable, and stores the merged content into a third Table variable or one of the two source Table variables. This action is best used when the two source Table variables contain identical column headers.

For example, if you have a table of employees hired in September and a table of employees that were hired in October, this option will return a table of all the employees, with the October hires appended below the September hires.

Table 1: Employees hired in September

Employee	Department	Pay rate
John	101	50
Jill	102	35

Table 2: Employees hired in October

Employee	Department	Pay rate
Mike	103	40
Betty	104	50

Table 3: All employees

Employee	Department	Pay rate
John	101	50
Jill	102	35
Mike	103	40
Betty	104	50

# Using Write to file action

Use the Write to file action to write the data from a Table type variable to a CSV or TXT file.

To write data into a file, do the following:

## Procedure

1. Double-click or drag Data table > Write to file.
2. Select the Table variable that contains the data to write from the Data table name list.
3. Specify the location of the file in which to write the data in the Enter file name field.
4. Select the Create folders/files if it doesn't exist check box to create the file or folder that you specified in the Enter file name field.
5. Select an option to specify what to do when writing data in an existing file:
  - Append to the existing file
  - Override existing file
6. Select an option from the Row delimiter list to specify the delimiter to use for rows.
7. Select an option from the Column delimiter list to specify the delimiter to use for columns.
8. Select an option from the Encoding list to specify the encoding to apply on the file.
9. In the Assign value to variable list, select a string variable.

# Extract data from a web table and save it to a file

Build a bot to open a browser window to the NASDAQ website, extract the data from a table, and write it to a CSV file on your desktop. This example uses actions from the Browser, Data Table, Recorder, and Window packages.

To extract data from a table, do the following steps:

## Procedure

1. Open a new bot:
  - a) From the Automation Anywhere Enterprise web interface, select Bots > My bots.
  - b) Click Create TaskBot.
  - c) Enter a bot name.
  - d) Accept the default folder location \Bots\.

To change where your bot is stored, click Choose and follow the prompts.

  - e) Click Create and Edit.
2. Open a browser window to the web page from which you will extract the table.
  - a) Double-click or drag the Browser > Launch website action.
  - b) In the URL field, enter  
`https://old.nasdaq.com/`
  - c) Specify the Internet Explorer browser.

Note: It is recommended to use Internet Explorer because it reliably launches the website in a new window, even if there is already an open window. Other browsers might launch the website in a new tab if there is an open window.

  - d) Click Apply.
  - e) Click Save.

- f) Click Run.  
The bot opens the window.
- 3. Specify the table.
  - a) Double-click or drag the Recorder > Capture action.
  - b) Click the Window tab and select the Daily Stock Market Overview window from the drop-down list.
  - c) Click Capture object.  
The Daily Stock Market Overview window activates.
  - d) Hover over the table below the Stock Market Overview heading.  
An orange box appears, surrounding the table.
  - e) Click the table.  
The Object Processing message box appears.
  - f) Return to the Enterprise Control Room.
  - g) In the Object properties table, verify the Control Type is TABLE.  
If it is not, click Recapture object.
  - h) From the Action drop-down list, select Get table.
  - i) In the Assign output to variable field, create the `tNasdaqTable`.
  - j) Click Apply.

The Daily Stock Market Overview window is saved as the variable `window-1`.

- 4. Specify the file where to save the data.
  - a) Double-click or drag the Data Table > Write to file action.
  - b) From the Data table name list, select `tNasdaqTable`.
  - c) Provide a file path to create a CSV file.  
For example, `C:\Users\<username>\Desktop\NasdaqTable.csv`.
  - d) Select the Create folders/files if it doesn't exist option.
  - e) Select to overwrite the existing file.
  - f) Click Apply.
- 5. Close the Daily Stock Market Overview window.
  - a) Double-click or drag the Window > Close action.
  - b) Select the Variable tab and insert `window-1`.
  - c) Click Apply.
- 6. Click Save.
- 7. Click Run.

The bot creates a CSV file on the desktop with data on seven indexes, their values, and net change.

## Datetime package

A datetime value consists of a date, time, and time zone. Automation Anywhere Enterprise stores datetime values in a Datetime variable. The Datetime package contains actions that enable you to perform various operations on datetime values. You can use these actions to manipulate and compare values in the Datetime variables.

## Actions in the Datetime package

The actions in the Datetime package accept a variable as an input and assign the output to a variable. These actions enable you to compare two Datetime values, add to or subtract from a Datetime value, and convert a Datetime value to a string value.

The Datetime package includes the following actions:

Action	Description
Add	See <a href="#">Using the Add action.</a>
Assign	See <a href="#">Using the Assign action.</a>
Is after	<p>Compares two Datetime variables and verifies if the value of the source variable is after the value available in the comparison variable, and stores the output to a Boolean variable.</p> <ul style="list-style-type: none"> <li>• Select the source variable from the Source date and time variable list, and select the variable you want to compare with from the Date and time variable to be compared to list.</li> <li>• Select the Datetime variable from the Assign the output to a variable list to specify the variable you want to use to assign the output.</li> </ul> <p>For example, if the value in the source variable is after the value in the comparison variable, the system stores 'True' as the output in the Boolean variable. If the value in the source variable is not after the value in the comparison variable, the system stores 'False' as the output in the Boolean variable.</p>
Is before	<p>Compares two Datetime variables and verifies if the value of source variable is before the value available in the comparison variable, and stores the output to a Boolean variable.</p> <ul style="list-style-type: none"> <li>• Select the source variable from the Source date and time variable list, and select the variable you want to compare with from the Date and time variable to be compared to list.</li> <li>• Select the Datetime variable from the Assign the output to a variable list to specify the variable you want to use to assign the output.</li> </ul> <p>For example, if the value in the source variable is before the value in the comparison variable, the system stores 'True' as the output in the Boolean variable. If the value in the source variable is not before the value in the comparison variable, the system stores 'False' as the output in the Boolean variable.</p>
Is equal	<p>Compares two Datetime variables and verifies if the value of the source variable is equal to the value available in the comparison variable, and stores the output to a Boolean variable.</p> <ul style="list-style-type: none"> <li>• Select the source variable from the Source date and time variable list, and select the variable you want to compare with from the Date and time variable to be compared to list.</li> <li>• Select the Datetime variable from the Assign the output to a variable list to specify the variable you want to use to assign the output.</li> </ul> <p>For example, if the value in the source variable is equal to the value in the comparison variable, the system stores 'True' as the output in the Boolean variable. If the value in the source variable is not equal to the value in the comparison variable, the system stores 'False' as the output in the Boolean variable.</p>
Subtract	See <a href="#">Using the Subtract action.</a>
To string	See <a href="#">Using the To string action.</a>

## Using the Add action

Use the Add action to increase the value in the Datetime variable by a specified time value and unit. For example, you can use this action to increase the Datetime variable value by three hours or by three days.

### Procedure

To add a value to a Datetime variable, do the following:

1. Double-click or drag the Add action from the Datetime node in the Actions palette.
2. Select an option from the Source date and time variable list to specify the variable that contains the value to which you want to add the time unit.
3. Enter the value you want to add in the Time value to add field.
4. Select an option from the Time unit to add list to specify the time unit you want to add.  
Choose from the following time unit options:
  - Milliseconds
  - Seconds
  - Minutes
  - Hours
  - Days
  - Weeks
  - Months
  - Years
5. Select the Datetime variable from the Assign the output to a variable list to specify the variable to which you want to assign the output.
6. Click Apply.

### Next steps

To see the output value, convert the value in the Datetime variable to a String variable, then print the value with the Message Box action. For more information, see [Using the To string action](#) and [Using the Message box action](#).

## Using the Assign action

Use the Assign action to assign one or more String variables, a user-entered value, or an existing Datetime variable into a Datetime variable.

### Procedure

To assign values to a Datetime variable, do the following:

1. Double-click or drag the To string action from the Datetime node in the Actions palette.
2. Select either the Enter the date time or Variable option.
  - If you have selected the Enter the date time option, do the following:
    - a) Enter the date time values or select a String variable.

Note: To enter multiple variables, separate them with single quotation marks. For example, to input the variables date1 and time1, enter 'date1"time1'.

b) Select the date time format.

Choose from the prebuilt formats or provide a custom format. For more information about the prebuilt formats, see [Date time formats](#).

- If you have selected the Variable option, select a variable from the Source date time variable list.
3. Select the variable to use to store the output from the Destination Datetime variable list.
  4. Click Apply.

## Using the Subtract action

Use the Subtract action to decrease the value in the Datetime variable by a specified time value and unit. For example, you can use this action to decrease the Datetime variable value by three hours or by three days.

### Procedure

To subtract a value from a Datetime variable, do the following:

1. Double-click or drag the Subtract action from the Datetime node in the Actions palette.
2. Select an option from the Source date and time variable list to specify the variable that contains the value from which you want to subtract the time unit.
3. Enter the value you want to subtract in the Time value to add field.
4. Select an option from the Time unit to add list to specify the time unit you want to subtract.

Choose from the following time unit options:

- Milliseconds
  - Seconds
  - Minutes
  - Hours
  - Days
  - Weeks
  - Months
  - Years
5. Select the Datetime variable from the Assign the output to a variable list to specify the variable to which you want to assign the output.
  6. Click Apply.

### Next steps

To see the output value, convert the value in the Datetime variable to a String variable, then print the value with the Message Box action. For more information, see [Using the To string action](#) and [Using the Message box action](#).

## Using the To string action

Use the To string action to convert a datetime value to a string value. This action enables you to select a predefined format or specify a custom format for the output value.

You can select a predefined format for datetime or provide a custom format based on your requirements. See [Date time formats](#).

To convert a datetime value to string, do the following:

## Procedure

1. Double-click or drag the To string action from the Datetime node in the Actions palette.
2. Select an option from the Source date and time variable list to specify the variable that contains the datetime value that you want to convert to a string value.
3. Select an option from the Formats list to specify the predefined format in which you want to store the string value.
4. Select the Custom format to specify a custom format based on your requirement.
5. Select the string variable that you want to use to store the converted value from the Assign the output to a variable list.
6. Click Apply.

## Date time formats

Automation Anywhere Enterprise provides various predefined formats and also allows you to specify a custom format when you convert a datetime value to a string value.

The To string action enables you to convert a datetime value to a string value. This action also enables you to select a predefined datetime format or specify a custom format.

### Predefined datetime format

When you convert a datetime value to a string value, the predefined datetime formats are available in the Formats list. The following predefined datetime formats are available:

- **BASIC\_ISO\_DATE**: Converts the datetime value to a `yyyymmdd` string value.
- **ISO\_LOCAL\_DATE**: Converts the datetime value to a `yyyy-mm-dd` string value.
- **ISO\_OFFSET\_DATE**: Converts the datetime value to a `yyyy-mm-dd+offset value` string value. The offset value indicates the difference between the local time and UTC in hours and minutes.
- **ISO\_DATE**: Converts the datetime value to a `yyyy-mm-dd+offset value` or `yyyy-mm-dd` string value. The system adds an offset value if it is available.
- **ISO\_LOCAL\_TIME**: Converts the datetime value to a `hh-mm-ss` string value.
- **ISO\_OFFSET\_TIME**: Converts the datetime value to a `hh-mm-ss+offset value` string value. The offset value indicates the difference between the local time and UTC in hours and minutes.
- **ISO\_TIME**: Converts the datetime value to a `hh:mm`, `hh:mm:ss`, or `hh:mm:ss+offset value` string value. The offset value indicates the difference between the local time and UTC in hours and minutes.
- **ISO\_LOCAL\_DATE\_TIME**: Converts the datetime value to a `yyyy-mm-ddThh:mm:ss` string value.
- **ISO\_OFFSET\_DATE\_TIME**: Converts the datetime value to a `yyyy-mm-ddThh:mm:ss+offset value` string value. The offset value indicates the difference between the local time and UTC in hours and minutes.
- **ISO\_ZONED\_DATE\_TIME**: Converts the datetime value to a `yyyy-mm-ddThh:mm:ss+offset value[zone id]` string value. The offset value indicates the difference between the local time and UTC in hours and minutes. The 'zone id' in the format represents the time zone.

- **ISO\_DATE\_TIME**: Converts the datetime value to a `yyyy-mm-ddThh:mm:ss`, `yyyy-mm-ddThh:mm:ss+offset value`, or `yyyy-mm-ddThh:mm:ss+offset value[zone id]` string value.
- **ISO\_ORIGINAL\_DATE**: Converts the datetime value to a `yyyy-[day of the year]` string value. For example, if the datetime value is 'January 25, 2018', it is formatted as '2018-25' where '25' indicates the day of the year.
- **ISO\_WEEK\_DATE**: Converts the datetime value to a `yyyy-[week of the year]-[day of the week]` string value. For example, if the datetime value is 'January 25, 2018', it is formatted as '2018-W4-4' where 'W4' indicates that it is the fourth week of the year and '4' indicates the day of that week.
- **ISO\_INSTANT**: Converts the datetime value to a `yyyy-mm-ddThh:mm:ss+offset valueZ` string value.
- **RFC\_1123\_DATE\_TIME**: Converts the datetime value to a `[day of the week], [day of the month] [month in 'MMM' format] [year in 'YYYY' format] hh:mm:ss [offset id]` string value. For example, if the datetime value is '2018-01-25 10:15', it is formatted as 'Thu, 25 December 2018 10:15:00GMT'. The 'GMT' indicates that there is no offset available.

## Custom datetime format

Automation Anywhere Enterprise offers predefined conventions that you can use to specify a custom datetime format. The following conventions are available:

- 'y': Enables you to specify a year as 'yy' or 'yyyy'.
- 'M': Enables you to specify a month as 'MM'.
- 'd': Enables you to specify a day as 'dd'.
- 'h' or 'H': Enables you to specify an hour as 'hh' or 'HH'.
- 'm': Enables you to specify a minute in an hour as 'mm'.
- 's': Enables you to specify a second in a minute as 'ss'.
- 'S': Enables you to specify a millisecond as 'SSS'.
- 'z': Enables you to specify a time zone such as 'Pacific Standard Time', 'Indian Standard Time', and so on.
- 'Z': Enables you to specify an offset time for a time zone such as '+0550' for 'Indian Standard Time'.

You can use the above convention to specify the custom format you want to use for converting a datetime value. Below are some of the examples that use the above convention:

Format	Sample output
yyyy-MM-dd	2018-12-25
dd-MM-yyyy	25-12-2018
yyyy-MM-dd HH:mm:ss	2018-25-12 23:50:55
hh:mm:ss.SSS	22:15:35.889
yyyy-MM-dd HH:mm:ss.SSS	2018-12-25 23:50:55.999
yyyy-MM-dd HH:mm:ss.SSS Z	2018-12-25 23:50:55.999 +0530

## Delay package

Use the Delay package to add a timed delay to the logic.



## Action in the Delay package

The Delay package includes the following action:

Action	Description
Delay	<p>Adds a timed delay.</p> <ul style="list-style-type: none"> <li>Specify delay for a specific period of time or a randomized period of time.</li> <li>Select the time unit in seconds or milliseconds.</li> <li>Use random delay when bots run simultaneously on several machines.</li> <li>Use random delay to reduce the amount of requests made to a database or the FTP server.</li> </ul>

Watch the following video on how to use the Delay actions:

Using Delay actions

## Dictionary package

The Dictionary package contains actions that enable you to do various operations on dictionary-type values.

### Dictionary data type

A dictionary is a collection of key-value pairs, in which each key has a value. It is similar to an entry in a dictionary where each word has a corresponding definition or explanation. The key is similar to the word and the definition or explanation is similar to its value. The properties of dictionary entries are as follows:

- Keys are unique within each dictionary variable. If you try to duplicate a key, you will overwrite its value.
- Key names cannot be changed.
- Keys are case-sensitive and cannot contain a dollar sign (\$).
- Keys do not have leading or trailing whitespaces.

Each key-value pair in a dictionary is separated by a colon. The key-value pairs are separated from each other by commas. An example of a dictionary is as follows:

```
invoice = {'CustomerID': 9876, 'Location': 'XYZ', 'Amount': 12,34,567}
```

In the example, 'CustomerID', 'Location', and 'Amount' are keys and the corresponding value assigned to each key is separated by a colon.

### Working with variables of dictionary data type

A dictionary variable can hold values of boolean, number, or string data subtype. When initializing a dictionary variable, you can select the Any data subtype in order to hold any of the three data subtypes. You can manually enter values by creating a new variable or selecting an existing one from the Variables menu, and then clicking Add.

Use a dictionary variable to hold [email](#) or [PDF](#) properties.

## Actions in the Dictionary package

The following actions are available:

Action	Description
Assign	<p>Assigns the value of the source dictionary to the destination dictionary variable.</p> <p>Select the source dictionary variable from the drop-down list. Select a variable or create one to hold the output.</p>
Get	<p>Verifies if a key exists in a dictionary variable and returns the corresponding value.</p> <ul style="list-style-type: none"> <li>• Select the dictionary variable from the drop-down list.</li> <li>• Enter the key name.</li> <li>• Select a variable or create one to hold the output.</li> </ul>
Put	<p>Assigns a value to a key in the dictionary. If the key is already associated with a value, that value is reassigned to a variable.</p> <ul style="list-style-type: none"> <li>• Select the dictionary variable from the drop-down list.</li> <li>• Enter the key name.</li> <li>• Select the new value to associate with the key.</li> <li>• Select the variable to hold the previous value.</li> </ul>
Remove	<p>Removes a value from the specified key. The removed value is assigned to a variable selected from the Assign removed item to variable list.</p> <p>Removes a value from the specified key and optionally assigns it to a variable selected from the Assign removed item to variable list.</p> <ul style="list-style-type: none"> <li>• Select the dictionary variable from the drop-down list.</li> <li>• Enter the key name.</li> <li>• Select the variable to hold the removed value.</li> </ul>
Size	<p>Retrieves the number of entries in the dictionary specified in the Dictionary variable field and assigns the output to a number variable selected from the Assign the output to variable list.</p>

## DLL package

A dynamic-link library (DLL) file contains a shared library of functions that can be used by Windows programs. The DLL package uses a .dll file as reference and call functions from the bot.

## Before you start

Perform the following actions within the DLL package as part of using the set of available actions:

1. Use the Open action to specify the location of the .dll file you want to use and session name. Use this same session name for the other actions. The .dll referenced is automatically added as a bot dependency.
  2. Use the Run function action to run a function from the .dll and save its output value as a variable.
  3. After running the required functions, close the .dll reference. It is important to close the .dll reference to free the memory of the operating system.
- Note: Windows and appropriate versions of Microsoft .NET Framework are available on the devices running the DLLs.

Note: There might be instances where a DLL refers another DLL. In such cases, you must upload the another DLL to the Cloud Control Room and add it as bot dependency manually. The bot encounters an if the supporting DLL is not added as a bot dependency.

## Actions in the DLL package

The DLL package includes the following actions:

Action	Description
Close	Closes the current session. In the Session name field, enter the name of the session.
Open	<p>Opens the reference file.</p> <ul style="list-style-type: none"> <li>• In the Session name field, enter the name of the session.</li> <li>• In the File path field, specify the location of the path.</li> </ul> <p>Note: The DLL package supports only DLLs written in C#.</p>
Run function	See <a href="#">Using Run function action</a> .

- [Using Run function action](#)  
Use the Run function action to run a function within the .dll file and specify the parameters to pass a function and store the output as a variable.

## Using Run function action

Use the Run function action to run a function within the .dll file and specify the parameters to pass a function and store the output as a variable.

Use a dictionary variable to pass the parameters to a function. The dictionary key contains the parameter name to pass to a function.

## Procedure

Follow these steps to run a function:

1. In the Actions palette, double-click or drag the Run function action from the DLL package.
2. In the Session name field, enter the name of the session.
3. In the Enter the namespace field, specify the namespace.  
You can specify any of the namespaces defined in the DLL file.  
Note: Ensure that the value you have provided in the field is correct.  
Important: This field is case-sensitive.
4. In the Enter the class name field, specify the name of the class.  
You can specify the name of any classes available in the namespace.  
Important: This field is case-sensitive.
5. In the Enter the name of function to be executed field, specify the name of the function to execute.  
Ensure that the name you have provided is correct and available in the class specified in the previous step.  
Important: This field is case-sensitive.
6. Optional: In the Parameters to the function list, select the dictionary variable that contains all the parameters needed by the function.  
Note: Use the dictionary subtype Any to enable the bot to pass parameters of different data types to the function including boolean, numeric, and string.  
Use name of the parameter as a dictionary key and its value as a dictionary value. For example, you have a function ReturnSum that returns sum of two numeric values and want to pass Param1 and Param2 as its parameters. In the dictionary variable, you must specify Param1 and Param2 as dictionary key and their values as the dictionary values.
7. Optional: In the Assign output to variable list, select a variable to use to store the output of the function.  
You can use dictionary, string, numeric, boolean, or datetime variable types to store the output based on the type of output the function returns.
8. Click Apply.
9. Click Save.

## Email package

The Email package contains actions to automate email-related tasks through Exchange Web Services (EWS), Microsoft Outlook, and other email servers. You can use these actions for sending, receiving, and modifying messages, folders, and the status of messages.

The Email package supports EWS, POP3, SMTP, and IMAP protocols and enables you to perform the following tasks:

- Manage and organize email messages and folders.
- Download attachments from emails to specific folders on devices.
- Extract data from emails to variables, to use as extracted data in other applications. For example, you can extract data from a sender's email and store it in a Microsoft Excel spreadsheet.

Note: If you want to automate an email-related task on a device using Microsoft Outlook, ensure that the Outlook application is open.

## Before you start

Perform the following actions within the Email package as part of using the set of available actions:

1. Establish a connection with an email server using the Connect action.

While establishing the connection, specify the details and session name of the email server. Use this same session name for the other actions.

You do not have to establish a connection for the Forward, Reply, and Send actions because you will add the sender and email server details when you use these actions.

2. Use the actions to automate a task.
3. After you have automated all the email-related tasks, terminate the connection to the mail server using the Disconnect action.

Important: You must enable POP settings to automate an email-related task on the Gmail server using the POP3 protocol.

See [Read Gmail messages using POP](#).

## Actions in the Email package

Note: Some of the actions must be used within a Loop action. See [Using Email action in loop](#).

The Email package includes the following actions:

Action	Description
Change status	<p>Changes the statuses of emails to read or unread. Use this action within a Loop action.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the name of the session you used to connect to the email server in the Connect action.</li> <li>Select the Read or Unread option to change the status of the email.</li> </ul>
Check if folder exists	<p>Checks whether a folder with the name you specified exists in the email server.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the name of the session you used to connect to the email server in the Connect action.</li> <li>In the Folder name field, enter the name of the folder you want to check. Note: This field is not case sensitive.</li> <li>In the Select the destination boolean variable list, select a Boolean variable to store the output of the existing folder. The output is either True (file exists) or False (does not exist).</li> </ul>
Connect	See <a href="#">Using the Connect action</a> .
Delete	Deletes the most recent email from the inbox. Use this action within a Loop action. In the Session name field, enter the name of the session you used to connect to the email server in the Connect action.
Delete all	<p>Deletes read, unread, or all emails from the mail server.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the name of the session you used to connect to the email server in the Connect action.</li> <li>Select the All, Read, or Unread option to specify the type of email you want to delete. Note: For an email server using the POP3 protocol, this action deletes all the email messages.</li> </ul>

Action	Description
Disconnect	Terminates the connection established with the email server. In the Session name field, enter the name of the session you used to connect to the email server in the Connect action.
Forward	See <a href="#">Using the Forward action</a> .
Move all	<p>Moves an email from one folder to another in the email server.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the name of the session you used to connect to the email server in the Connect action.</li> <li>In the Destination folder path in mailbox field, provide the path to the location where you want to move the email.</li> <li>Select which emails to move: All, Read, or Unread.</li> <li>In the From a specific folder field, specify the folder from which to move the emails. For example, <code>Inbox/project1</code>.</li> </ul> <p>Note: For an email server using the POP3 protocol, you can retrieve only from the Inbox.</p> <ul style="list-style-type: none"> <li>Optional: Specify any of the following conditions: <ul style="list-style-type: none"> <li>When subject contains: Enter keywords separated with a semi-colon.</li> <li>From specific senders: Enter senders' email addresses separated with a semi-colon.</li> <li>When received date is on or after or When received date is before: Select a datetime variable from the list.</li> </ul> </li> </ul>
Reply	See <a href="#">Using Reply action</a> .
Save all attachments	<p>Saves attachments from all emails on the email server to a specified folder. Note: If connected to an Outlook server, images embedded in the emails are downloaded along with attachments.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the name of the session you used to connect to the email server in the Connect action.</li> <li>Select which emails to save: All, Read, or Unread.</li> <li>In the Save attachments to folder field, specify the location to save the attachments. Note: This field is not case sensitive.</li> <li>Overwrite file(s): select this option to overwrite an existing file with the same name. If this option is not selected, the downloaded attachment is saved with the same filename suffixed with a numeric value; it does not overwrite the existing file.</li> </ul>
Save attachments	<p>Saves attachments from one email to a specified folder. Use this action within a Loop action. Note: If connected to an Outlook server, images embedded in the emails are downloaded along with attachments.</p> <ul style="list-style-type: none"> <li>In the Save attachments to folder field, specify the location to save the attachments. Note: This field is not case sensitive.</li> </ul>

Action	Description
	<ul style="list-style-type: none"> <li>Overwrite file(s): select this option to overwrite an existing file with the same name. If this option is not selected, the downloaded attachment is saved with the same filename suffixed with a numeric value; it does not overwrite the existing file.</li> </ul>
Save email	<p>Saves an email message as an EML file to a folder. Use this action within a Loop action.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the name of the session you used to connect to the email server in the Connect action.</li> <li>In the Save emails to folder field, specify the location to save the email. Note: This field is not case sensitive.</li> <li>Overwrite file(s): select this option to overwrite the existing file with the same name. If this option is not selected, the downloaded attachment is saved with the same filename suffixed with a numeric value; it does not overwrite the existing file.</li> </ul>
Send	See <a href="#">Using the Send action</a> .

Related reference

[Loop package](#)

[Variables overview](#)

## Using the Connect action

Use the Connect action to establish a connection with an email server. This is the first action you must use to automate an email-related task.

This action enables you to specify the details and session name of the email server. Use this same session name for the other actions.

## Procedure

To establish a connection with an email server, follow these steps:

1. In the Actions palette, double-click or drag the Connect action from the Email package.
2. In the Session name field, enter the name of the session you used to connect to the email server in the Connect action.
3. In the Outlook, Email server, or EWS option, specify whether you want to establish a connection with Microsoft Outlook or a mail server.
  - If you have selected the Outlook option, you do not need to provide any additional details.
  - If you have selected the Email server option, complete the following fields:  
Note: For information about the host and port to be used for the various mail servers, see [Configuring email server](#).
    - Host: Enter the name of the mail server you want to connect.  
Note: This field is not case sensitive.
    - Port: Enter the port you want to use to establish the connection.

- Username: Click Credential to use a value available in the Credential Vault or String to manually enter a username. [Credentials and lockers](#)
- Password: Click Credential to use a value available in the Credential Vault or String to manually enter a password. [Credentials and lockers](#)
- Use secure connection (SSL/TLS): Select this option if you want to use a secure connection with the mail server.
- Protocol: Select the IMAP or POP3 option to specify the protocol used for the mail server.
- If you have selected the EWS option, complete the following fields:
  - Username: Enter the username you want to use to access the mail server.

For example, john.smith@myCompanyName.com

Choose the username as a Credential or String.

- Password: Enter the password for the username you have provided.

Choose the username as a Credential or String.

- Optional: Domain name:
  - If you are an Office 365 customer and you leave this field blank, Automation Anywhere Enterprise uses smtp.office365.com to connect to the server.
  - If you are an Office 365 customer and you have entered a domain name in the username field, you must enter smtp.office365.com into this field.
  - If you are not an Office 365 customer, enter your company domain name. Otherwise, Automation Anywhere Enterprise uses the domain name you provided in the Username field.
- Exchange Version: Select the version your organization is using.

4. Click Apply.

5. Click Save.

## Configuring email server

The details regarding the mail server you provide when establishing a connection varies based on the email server you are connecting to.

The following table provides information about the host name and the port you must use when establishing a connection to the specified mail server:

Server	For reading or fetching emails	For sending emails
Microsoft Outlook/Microsoft Office 365	Host: outlook.office365.com Port: 993	Host: outlook.office365.com, Port: 993 Host:smtp.office365.com Port: 587
Gmail	Host: imap.gmail.com Port: 993	Host: smtp.gmail.com Port: 587



Yahoo	Host: imap.mail.yahoo.com Port: 993	Host: smtp.mail.yahoo.com Port: 465 or 587
MSN	Host: imap-mail.outlook.com Port: 993 For POP3: Host: pop-mail.outlook.com Port: 993	Host: smtp-mail.outlook.com Port: 465 or 587

## Additional configuration for Gmail and Yahoo

If you are using a Gmail or Yahoo account to automate an email-related task, ensure that the option to allow access for less secured apps is enabled. This is required to enable a TaskBot to send email using a Gmail or Yahoo account.

To enable access for less secure apps:

1. Log in to your Gmail or Yahoo account.
2. Go to account settings.
3. Enable the option to allow less secure apps.

## Using the Forward action

Use the Forward action to forward emails and attachments to one or more recipients. To forward emails as plain text or HTML through Exchange Web Services (EWS), Microsoft Outlook, or other email servers, use this action within a Loop action.

### Procedure

To forward emails, follow these steps:

1. In the Actions palette, double-click or drag the Forward action from the Email package.
2. In the To address field and the optional Cc and Bcc fields, enter the email addresses of the recipients.  
Note: The To address, Cc, and Bcc fields are not case sensitive.
3. Optional: In Attachment, select the attachment from a location:
  - Control Room file: Enables you to select an attachment that is available in a folder.
  - Desktop profile: Enables you to select an attachment that is available on your device.

To attach multiple files, separate each file path with a semi-colon.

- Variable: Enables you to specify the file variable that contains the location of the attachment.
4. Optional: Select the Validate if attachment is missing check box to verify that you have attached a file and the attached file exists.

Option	Result
The Validate if attachment is missing check box is selected	If a file is not attached, the email is not sent and the bot encounters an error.
The Validate if attachment is missing check box is not selected	The email is sent even if a file is not attached.

5. In the Plain text or HTML options, select the format that you want to forward emails.
6. Optional: Enter the content you want to send along with the email in the Message field.  
The email will be appended to the message you have specified.
7. Optional: Select the Include Go Green message at the end of the email check box.  
The Go Green message appends the following text to the bottom of the email body: `Please consider the environment before printing. Let's Go Green!`
8. Select the Email server, EWS or Outlook option from the Send email via list to specify whether to send the emails using Microsoft Outlook or a mail server.
  - If you have selected the Outlook option, you do not need to provide any additional details.
  - If you have selected the Email server option, complete the following fields:  
Note: For information about the host and port to be used for the various mail servers, see [Configuring email server](#).
    - Email server host: Enter the host you want to connect.  
Note: If you use the host Outlook.office365.com, there is a limit of 30 messages sent per minute and 10,000 recipients per day.
    - Email server port: Enter the port you want to use to establish the connection.
    - Use secure connection (SSL/TLS): Select this option if you want to use a secure connection with the mail server.
    - My server requires authentication: Select this option if the server requires credentials for access.
      - Username: Enter the username you want to use to access the mail server.  
  
Choose the username as a Credential or String.
      - Password: Enter the password for the username you have provided.  
  
Choose the password as a Credential or String.
  - If you have selected the EWS option, complete the following fields:
    - Username: Enter the username you want to use to access the mail server.  
  
For example, john.smith@myCompanyName.com  
  
Choose the username as a Credential or String.
    - Password: Enter the password for the username you have provided.  
  
Choose the username as a Credential or String.
    - Optional: Domain name:
      - If you are an Office 365 customer and you leave this field blank, Automation Anywhere Enterprise uses smtp.office365.com to connect to the server.
      - If you are an Office 365 customer and you have entered a domain name in the username field, you must enter smtp.office365.com into this field.
      - If you are not an Office 365 customer, enter your company domain name. Otherwise, Automation Anywhere Enterprise uses the domain name you provided in the Username field.

- Exchange Version: Select the version your organization is using.
9. Click Apply.
  10. Click Save.

## Using Reply action

Use the Reply action to send a response to an email with the same subject.

To send a reply to emails as plain text or HTML through Exchange Web Services (EWS), Microsoft Outlook, or other email servers, use this action within a Loop action. This action does not include any files attached to the original email.

## Procedure

Follow these to reply to an email:

1. In the Actions palette, double-click or drag the Reply action from the Email package.
2. In the Cc and Bcc fields, enter the email address of additional recipients.  
Note: The Cc and Bcc fields are not case sensitive.
3. Optional: In Attachment, select the attachment from a location:
  - Control Room file: Enables you to select an attachment that is available in a folder.
  - Desktop profile: Enables you to select an attachment that is available on your device.

To attach multiple files, separate each file path with a semi-colon.

- Variable: Enables you to specify the file variable that contains the location of the attachment.
4. Optional: Select the Validate if attachment is missing check box to verify that you have attached a file and the attached file exists.

Option	Result
The Validate if attachment is missing check box is selected	If a file is not attached, the email is not sent and the bot encounters an error.
The Validate if attachment is missing check box is not selected	The email is sent even if a file is not attached.

5. In the Plain text or HTML options, select the format that you want to forward emails.
6. Optional: Enter the content you want to send along with the email in the Message field.  
The email will be appended to the message you have specified.
7. Optional: Select the Include Go Green message at the end of the email check box.  
The Go Green message appends the following text to the bottom of the email body: Please consider the environment before printing. Let's Go Green!
8. Select the Email server, EWS or Outlook option from the Send email via list to specify whether to send the emails using Microsoft Outlook or a mail server.
  - If you have selected the Outlook option, you do not need to provide any additional details.
  - If you have selected the Email server option, complete the following fields:  
Note: For information about the host and port to be used for the various mail servers, see [Configuring email server](#).
    - Email server host: Enter the host you want to connect.

Note: If you use the host Outlook.office365.com, there is a limit of 30 messages sent per minute and 10,000 recipients per day.

- Email server port: Enter the port you want to use to establish the connection.
- Use secure connection (SSL/TLS): Select this option if you want to use a secure connection with the mail server.
- My server requires authentication: Select this option if the server requires credentials for access.
  - Username: Enter the username you want to use to access the mail server.

Choose the username as a Credential or String.

- Password: Enter the password for the username you have provided.

Choose the password as a Credential or String.

- If you have selected the EWS option, complete the following fields:
  - Username: Enter the username you want to use to access the mail server.

For example, john.smith@myCompanyName.com

Choose the username as a Credential or String.

- Password: Enter the password for the username you have provided.

Choose the username as a Credential or String.

- Optional: Domain name:
  - If you are an Office 365 customer and you leave this field blank, Automation Anywhere Enterprise uses smtp.office365.com to connect to the server.
  - If you are an Office 365 customer and you have entered a domain name in the username field, you must enter smtp.office365.com into this field.
  - If you are not an Office 365 customer, enter your company domain name. Otherwise, Automation Anywhere Enterprise uses the domain name you provided in the Username field.
- Exchange Version: Select the version your organization is using.

9. Click Apply.

10. Click Save.

## Using the Send action

Use the Send action to send an email. This action enables you to send an email to one or more recipients from Microsoft Outlook or a mail server, attach files, and select to send the email in plain text or HTML format.

The Send action enables you to attach multiple files of various formats to an email. There is no restriction on the type of file you can attach or its size. However, restrictions implemented by the mail server you are using to send an email apply. For example, a task will fail if the files you have attached to the email you are sending do not adhere to the restrictions applied by the mail server.

To send an email from a Gmail or Yahoo account, enable access to a less secure app for that account. See [Configuring email server](#).

## Procedure

1. In the Actions palette, double-click or drag the Send action from the Email package.
2. In the To address field and the optional Cc and Bcc fields, enter the email addresses of the recipients.  
Note: The To address, Cc, and Bcc fields are not case sensitive.
3. In the Subject field, enter the subject for the email.
4. Optional: In Attachment, select the attachment from a location:
  - Control Room file: Enables you to select an attachment that is available in a folder.
  - Desktop profile: Enables you to select an attachment that is available on your device.

To attach multiple files, separate each file path with a semi-colon.

- Variable: Enables you to specify the file variable that contains the location of the attachment.
5. Optional: Select the Validate if attachment is missing check box to verify that you have attached a file and the attached file exists.

Option	Result
The Validate if attachment is missing check box is selected	If a file is not attached, the email is not sent and the bot encounters an error.
The Validate if attachment is missing check box is not selected	The email is sent even if a file is not attached.

6. In the Plain text or HTML options, select the format that you want to forward emails.
7. Optional: Enter the content you want to send along with the email in the Message field.  
The email will be appended to the message you have specified.
8. Optional: Select the Include Go Green message at the end of the email check box.  
The Go Green message appends the following text to the bottom of the email body: Please consider the environment before printing. Let's Go Green!
9. Select the Email server, EWS or Outlook option from the Send email via list to specify whether to send the emails using Microsoft Outlook or a mail server.
  - If you have selected the Outlook option, you do not need to provide any additional details.
  - If you have selected the Email server option, complete the following fields:  
Note: For information about the host and port to be used for the various mail servers, see [Configuring email server](#).
    - Email server host: Enter the host you want to connect.  
Note: If you use the host Outlook.office365.com, there is a limit of 30 messages sent per minute and 10,000 recipients per day.
    - Email server port: Enter the port you want to use to establish the connection.
    - Use secure connection (SSL/TLS): Select this option if you want to use a secure connection with the mail server.
    - My server requires authentication: Select this option if the server requires credentials for access.
      - Username: Enter the username you want to use to access the mail server.

Choose the username as a Credential or String.

  - Password: Enter the password for the username you have provided.

Choose the password as a Credential or String.

  - If you have selected the EWS option, complete the following fields:
    - Username: Enter the username you want to use to access the mail server.

For example, john.smith@myCompanyName.com

Choose the username as a Credential or String.

- Password: Enter the password for the username you have provided.

Choose the username as a Credential or String.

- Optional: Domain name:
  - If you are an Office 365 customer and you leave this field blank, Automation Anywhere Enterprise uses smtp.office365.com to connect to the server.
  - If you are an Office 365 customer and you have entered a domain name in the username field, you must enter smtp.office365.com into this field.
  - If you are not an Office 365 customer, enter your company domain name. Otherwise, Automation Anywhere Enterprise uses the domain name you provided in the Username field.
- Exchange Version: Select the version your organization is using.

10. Click Apply.

11. Click Save.

## Using Email action in loop

You must use certain Email actions within a Loop action.

### Procedure

To use an Email action within a loop, follow these steps:

1. Double-click or drag the Loop action from the Loop package in the Actions palette.
2. Select the For each mail in mail box option from the Iterator list.
3. In the Session name field, enter the name of the session you used to connect to the email server in the Connect action.
4. In the ALL, READ, or UNREAD options, specify the type of email to retrieve from the email server.  
Note: For an email server that uses the POP3 protocol, all emails are retrieved.
5. Optional: Specify the folder from which you want to retrieve the emails.
  - Microsoft Outlook and Yahoo: You can specify the name of the folder.

For example, Inbox, Sent, Inbox/Sales, Inbox/IT and so on. The Sales and IT folders in this example are user-created folders and not available by default.

- Gmail: To retrieve emails from the default folders, you must use [Gmail]/FOLDERNAME, except for the Inbox folder.

For example, [Gmail]/Draft, [Gmail]/Important, [Gmail]/Trash, and so on. To retrieve email from the folders you have created and the Inbox folder, you must specify the folder names as they are. For example, if you have created folders called Bank and Sports in your Gmail, specify Bank and Sports without the [Gmail] prefix to retrieve emails from these folders.

6. Optional: In the When subject contains field, specify a value to retrieve emails containing the value you specified in their subject.
7. Optional: In the From specific senders field, specify the email addresses to retrieve the emails.

8. Optional: In the When received date is on or after list, select an option to retrieve specific emails on or after a certain date.
9. Optional: In the When received date is on or before list, select an option to retrieve specific emails on or before a certain date.
10. In the Plain text or HTML options, select the format that you want to forward emails.
11. Optional: In the Assign the current value to variable list, select or create a dictionary variable.

The dictionary variable stores the properties of each email. See [Using dictionary variable for email properties](#).

12. Click Apply.

Insert a Message Box action into the Loop container to print the subject of each email. Use the following message body, substituting the generic variable name for the one you used in Step 10: `$dictionaryVar{emailSubject}`  
\$

## Using dictionary variable for email properties

When you automate an email-related task, Automation Anywhere Enterprise retrieves various properties of an email and stores the values of these properties in a dictionary variable. These properties are stored in a dictionary variable when you use an email action within a loop action.

Some of the email properties that Automation Anywhere Enterprise retrieves are its subject, recipients, senders, message, and received date and time. The email properties are stored in a dictionary variable within the following dictionary keys:

- emailSubject
- emailFrom
- emailTo
- emailCc
- emailBcc
- emailMessage
- emailReceivedTime
- emailReceivedDate

The system automatically associates the properties of an email with the appropriate dictionary keys. You can use the values stored in these dictionary keys in another task, store them in a database, or for any other purpose. These keys are available in the system and you must use them as specified above. For example, if you have created a dictionary variable 'Test' and want to display the subject and sender of an email in a message prompt, you must enter `Test{emailSubject}` and `Test{emailFrom}` in the appropriate fields.

If you press F2 to use a variable in a field, you must select `Test` from the Choose a variable list and enter `emailSubject` in the Dictionary key field in the Insert a variable dialog box to store the subject of an email in that field. Similarly, you can enter `emailFrom` in the Dictionary key field to use the email address of the sender of an email in a field.

The dictionary key variable is required for all email-related tasks you automate from Microsoft Outlook and any mail server if you want to use the email properties. You can use the dictionary key variable for the following Email actions:

- Change status
- Delete
- Forward

- Reply
- Save attachments
- Save email

Note: The time required to save an email varies based on the size of the attachments available in that email.

## Error handler package

The Error handler package contains actions that enable you to easily handle exceptions that a bot encounters and transfers control to the other actions within that bot.

The Error handler actions enable you to separate the actions that you want to use to perform a task from the actions that you want to use to handle an exception. Handling exceptions ensures that a bot completes a task when it encounters an error.

## Actions in the Error handler package

The Error handler package includes the following actions:

Action	Description
Try	Enables you to specify a sequence of actions that might encounter an error. If the bot encounters an error from running the actions inside of the Try action, it attempts the actions inside of the Catch action.
Catch	<p>Enables you to specify an alternative sequence of actions to run, and an optional contingency, if the bot encounters an error while running any of the actions in the Try action.</p> <p>You can assign the exception message or line number to a variable, then insert the variable in a Log to file, Message box, or Email &gt; Send action. You can also use the actions from the Screen package to capture a screen shot of window or application when the error occurs.</p> <p>You can use the Catch action multiple times in a bot to handle various exceptions. This enables you to run a different set of actions based on the exception encountered.</p>
Finally	Enables you to specify the actions that you want to run regardless if the bot encounters an error or not.
Throw	Displays a custom message if the bot encounters an error. Specify the message you want to display in the Please enter exception message field.

## Using the actions in the Error handler package

In this example, you have a bot that reads data from a Microsoft Excel file and stores it in a database. The bot might encounter an error if the required file is not available or while updating a table in the database. Use the following methods to handle the errors:

- Exception 1:



- The Microsoft Excel spreadsheet from which you want to extract data is not available.
- How to handle: Use another file that contains the same data.
- Exception 2:
  - The table that you want to use to store the data is not available in the database.
  - How to handle: Display a message that the required table is not available.

Based on this example, perform the following to handle the mentioned exceptions:

1. Insert all the actions that are for reading the data from the Microsoft Excel spreadsheet and storing the extracted data into a table in the database within the Try action.
2. Insert the actions to run for the following exceptions within the Catch action:
  - a) Exception 1: Insert the actions to use the alternate file that contains the same data.
  - b) Exception 2: Insert the Message box action to display a relevant message.
3. Insert the Database > Disconnect action to terminate the connection with the database within the Finally action.

Regardless of whether the bot encounters an error or not, it must terminate the connection it established with the database.

Watch the following video on how to use the Error handler package:

Using Error handler package

## Excel basic package

The Excel basic package contains actions that enable you to automate many of the repetitive tasks in XLSX workbooks. You use these actions when Microsoft Excel is not available on the device that you want to use to automate Microsoft Excel-related tasks.

Note: This package supports files up to 7 MB in size. The XLS and CSV formats are not supported and you cannot set a value in the XFD column of a spreadsheet.

## Choosing the Excel package in Enterprise A2019

Enterprise A2019 includes packages to support three types of Microsoft Excel usage. For optimal results, use the package that corresponds to the type of Excel that is available on the device you are running bots on.

- No Excel installed: If you do not have Microsoft Excel installed on the device on which you are running bots to automate Excel-related processes, use the Excel basic package.
- Desktop Excel installed: If you have a desktop version of Microsoft Excel installed on your computer, use the Excel advanced package in your bots.
- Online Office 365 Excel only: If you are using Microsoft Excel 365 on a web browser, use the Office 365 Excel package for automating tasks related to Excel.

## Before you start

Perform the following actions within the Excel basic package as part of using the set of available actions:

1. Open the Microsoft Excel spreadsheet that you want to read data from using the Open action.

You must associate the details of the file you want to use with a session name, and use the session name in the other actions in the Excel basic package, so that you do not have to provide the details of the file in those actions again. See [Using Open action](#).

2. Use the different actions available in the Excel basic package to automate the Microsoft Excel-related tasks.
3. After you have automated all the Microsoft Excel-related tasks, close the spreadsheet using the Close action.

## Actions in the Excel basic package

The Excel basic package includes the following actions:

Action	Description
Switch to sheet	<p>Switches to another sheet in a Microsoft Excel file.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Select the Sheet by Index (numerical value) or Sheet by Name option to specify how to activate the sheet.</li> </ul>
Close	<p>Closes the current workbook and provides an option to Save changes when closing the file. Enter the name of the session used to open the current workbook with the Open action.</p> <p>Note: This action terminates the process for the session.</p>
Delete cell	<p>Deletes the Active cell or a Specific cell from the current worksheet or a CSV file.</p> <ul style="list-style-type: none"> <li>Select the Shift cells left or Shift cells up option to specify whether to shift the cell one position to the left or up after the cell is deleted.</li> <li>Select the Entire row or Entire column option to specify whether to delete the entire row or column of the cell.</li> </ul>
Find	See <a href="#">Using Find action</a> .
Get multiple cells	<p>Retrieves the values from cells in a Microsoft Excel spreadsheet and stores them in a table variable.</p> <p>Note: This action only recognizes values of string and number data types.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Select the Multiple cells or All cells option to specify whether to retrieve the values from a range of cells or all the cells.</li> </ul>
Get single cell	<p>Retrieves the values from a single cell in a Microsoft Excel spreadsheet or a CSV file and stores them in a string variable.</p> <p>Note: This action only recognizes values of string and number data types.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Select the Active cell or Specific cell option to specify whether to retrieve the values from the active cell or a specific cell.</li> </ul>

Action	Description
	You can use the output value to <a href="#">Perform an action based on cell value</a> .
Go to cell	Moves the cursor to a specific cell in a Microsoft Excel spreadsheet or a CSV file. <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• In the Active cell list, select the specific cell to which the cursor should be moved.</li> </ul>
Open	Reads data from a Microsoft Excel spreadsheet. See <a href="#">Using Open action</a> . Note: This action does not open a Microsoft Excel spreadsheet. It only enables a task to read data from the spreadsheet.
Replace	See <a href="#">Using Replace action</a> .
Save workbook	Saves a Microsoft Excel spreadsheet to a specified location.
Set cell	Sets the value you have specified in a cell in a Microsoft Excel spreadsheet. Select the Active cell or Specific cell option to specify whether to set the value in the active cell or a specific cell.

## Excel advanced package

The Excel advanced package contains actions that enable you to automate many of the repetitive tasks when working with Microsoft Excel spreadsheets.

You can automate tasks related to the workbook, worksheet, rows, columns, and cell operations. You can use these actions when Microsoft Excel is installed on the device that you want to use to automate the Microsoft Excel-related tasks.

Actions in the Excel advanced package support Microsoft Office 2000 through Microsoft Office 2016 and .xls, .xlsx, .xlsm, .xltm, .xltx, .xlsb, and .xlsm formats. You can also use some of the actions with the .csv format. See [CSV operations](#).

See [Example: Migrate data from Excel to a database](#).

## Choosing the Excel package in Enterprise A2019

Enterprise A2019 includes packages to support three types of Microsoft Excel usage. For optimal results, use the package that corresponds to the type of Excel that is available on the device you are running bots on.

- No Excel installed: If you do not have Microsoft Excel installed on the device on which you are running bots to automate Excel-related processes, use the Excel basic package.
- Desktop Excel installed: If you have a desktop version of Microsoft Excel installed on your computer, use the Excel advanced package in your bots.
- Online Office 365 Excel only: If you are using Microsoft Excel 365 on a web browser, use the Office 365 Excel package for automating tasks related to Excel.

## Before you start

Perform the following actions within the Excel advanced package as part of using the set of available actions:

1. Open the Microsoft Excel spreadsheet that you want to use to automate the Microsoft Excel-related tasks. Use the Open action to open the spreadsheet. See [Using Open action](#).
2. Use the different actions available in the Excel advanced package to automate the Microsoft Excel-related tasks.
3. After you have automated all the Microsoft Excel-related tasks, close the spreadsheet using the Close action.

## Actions in the Excel advanced package

The actions in the Excel advanced package enable you to perform the following operations:

Action	Description
Cell operations	Perform operations such as extracting data from cells, deleting values from a cell, moving a cursor to a specific cell, and finding and replacing content. See <a href="#">Cell operations</a> .
Row and column operations	Perform operations such as inserting new rows and columns, and hiding and unhiding rows and columns. See <a href="#">Row and column operations</a> .
Table operations	Perform operations such as sorting and filtering data in a table, inserting and deleting columns, and getting the table range. See <a href="#">Table operations</a> .
Workbook operations	Perform operations such as opening a workbook, appending data, protecting and unprotecting a workbook, and converting a Microsoft Excel workbook to a PDF file. See <a href="#">Workbook operations</a> .
Worksheet operations	Perform operations such as creating and deleting a worksheet, appending data, and hiding and unhiding a worksheet. See <a href="#">Worksheet operations</a> .

Watch the following video on how to use the Excel advanced package:

Using Excel advanced package

Related tasks

[Using Find action](#)

## Workbook operations

A workbook is a file that contains one or more worksheets. The Excel advanced package contains various actions that you can use to automate workbook-related tasks.

## Workbook actions in the Excel advanced package

The Excel advanced package includes the following actions:

Action	Description
Append workbook	<p>Adds all the worksheets from the specified workbook to the end of the currently open workbook.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Specify the workbook from which you want to append the worksheets in the Append from workbook field.</li> <li>• If the workbook is protected, select the Password is required check box and enter the password in the To open field.</li> </ul>
Close	<p>Closes the current workbook and provides an option to Save changes when closing the file.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> </ul>
Convert excel to PDF	<p>Converts the entire workbook, specific worksheets in a workbook, or a CSV file to a PDF file.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Specify whether you want to convert the Entire excel file, Active sheet, or Specific sheet to a PDF file.</li> <li>• Enter a name for the PDF file in the Select PDF file name field.</li> <li>• Specify the location where you want to save the file in the Select PDF storage location field.</li> </ul>
Create workbook	<p>Creates a Microsoft Excel workbook or a CSV file.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Specify the location where you want to save the workbook in the File path field.</li> <li>• Specify the name of the worksheet and passwords to open and edit the workbook.</li> </ul>
Open	See <a href="#">Using Open action</a> .
Protect workbook	<p>Protects the workbook and its structure using a password. Protecting a workbook prevents other users from opening the workbook without the password, and protecting the structure of a workbook prevents other users from adding, moving, deleting, hiding, and renaming worksheets within that workbook.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the Protect workbook and Protect workbook structure check boxes and provide a password in the respective fields.</li> </ul>

Action	Description
Save workbook	Saves the current workbook. Enter the name of the session used to open the current workbook with the Open action.
Unprotect workbook	<p>Unprotects a workbook and its structure. Unprotecting a workbook removes the restriction applied on opening the workbook, and unprotecting the structure of a workbook removes the restriction applied on modifying the structure of the workbook. Unprotecting allows other users to open the workbook, add, move, delete, hide, and rename worksheets within the workbook.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the Unprotect workbook and Unprotect workbook structure check boxes and provide a password to unprotect the workbook structure.</li> </ul>

## Using Open action

Use the Open action to open a Microsoft Excel spreadsheet or a CSV file. This action enables you to specify whether to open the spreadsheet in read-only mode or read-write mode, a password to open the spreadsheet, and so on.

To open a Microsoft Excel spreadsheet, do the following:

### Procedure

1. Double-click or drag the Open action from the Excel node in the Actions palette.
2. Specify a name for the session in the Session name field.
3. Select from where you want to open the Microsoft Excel spreadsheet:
  - From 'My bots': Enables you to open a Microsoft Excel spreadsheet from an existing bot.
  - From local device: Enables you to open a Microsoft Excel spreadsheet from a local device.
  - Select an existing file variable: Enables you to open a Microsoft Excel spreadsheet using a file variable.
4. Select the Specific sheet name option and specify the name of the sheet to activate when the Microsoft Excel spreadsheet opens.
5. Select Read-only mode or Read-write mode to open the Microsoft Excel spreadsheet in read-only or edit mode respectively.
6. Select the Password is required check box if a password is required to open or edit the Microsoft Excel spreadsheet.
7. Optional: In the User password or Owner password field, enter a password to restrict access to the encrypted PDF file.
  - User password: Allow users to perform specific operations on the encrypted PDF file.
  - Owner password: Allow users to use a password to open the file.
8. Select the Sheet contains a header check box if the Microsoft Excel spreadsheet contains a header row.
9. Select the Load Add-ins check box if you want to load the add-ins available in the Microsoft Excel spreadsheet.
10. Click Apply.

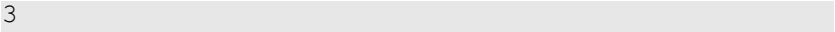
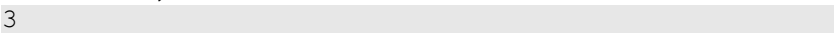
# Worksheet operations

The Excel advanced package contains various actions that you can use to automate worksheet-related tasks.

## Worksheet actions in the Excel advanced package

The Excel advanced package includes the following actions:

Action	Description
Access password protected worksheet	<p>Accesses a password-protected worksheet in the current workbook.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Enter the password to access the worksheet.</li> </ul>
Append worksheet	See <a href="#">Using Append worksheet action</a> .
Create worksheet	<p>Creates a worksheet in the current workbook.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Specify either an index number in the Sheet by Index field or a name in the Sheet by Name field for the worksheet. The index number is the number assigned to a worksheet. For example, if you want to perform an operation on the worksheet at the third position in the workbook, enter <code>3</code> in the field.</li> </ul>
Delete worksheet	<p>Deletes a spreadsheet from the current workbook.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Specify either an index number in the Sheet by Index field or a name in the Sheet by Name field for the worksheet. The index number is the number assigned to a worksheet. For example, if you want to perform an operation on the worksheet at the third position in the workbook, enter <code>3</code> in the field.</li> </ul> <p>Note: You can delete a worksheet only if the workbook contains more than one worksheet.</p>
Get current worksheet name	<p>Gets the name of the current worksheet and assigns it to a string variable.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Select a string variable that you want to use to store the name of the worksheet from the Assign the output to variable list.</li> </ul>
Get worksheet as data table	Gets data from a worksheet and saves it in a table variable.

Action	Description
	<ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Specify either an index number in the Sheet by Index field or a name in the Sheet by Name field for the worksheet. The index number is the number assigned to a worksheet. For example, if you want to perform an operation on the worksheet at the third position in the workbook, enter  in the field.</li> <li>Select a table variable that you want to use to store the data from the worksheet from the Assign value to the variable list.</li> </ul>
Get worksheet names	<p>Gets the names of all the worksheets and assigns them to a list variable of string data type.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Select a string variable that you want to use to store the name of the worksheet from the Assign the output to variable list.</li> </ul>
Hide worksheet	<p>Hides a worksheet from the current workbook.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Specify the name of the worksheet to hide in the Enter worksheet name to hide field.</li> </ul> <p>Note: You can hide a worksheet only if the workbook contains more than one worksheet.</p>
Password protect worksheet	<p>Protects a worksheet with a password. You can also specify the operations to restrict on the worksheet.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Specify the password you want to use to protect the worksheet and select the check boxes for the operation that you want to restrict on the worksheet. For example, select the Delete row and Delete column check boxes to restrict a user from deleting a row or column from the worksheet.</li> </ul>
Rename worksheet	<p>Renames a worksheet in the current workbook.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Specify the index number or name of the worksheet to rename in the Sheet by Index or Sheet by Name field.</li> </ul> <p>The index number is the number assigned to a worksheet. For example, if you want to perform an operation on the worksheet at the third position in the workbook, enter  in the field.</p>



Action	Description
	<p>Note: You cannot rename the worksheet if a worksheet with the same name already exists in the workbook.</p> <ul style="list-style-type: none"> <li>• Enter the new worksheet name that is under 31 characters.</li> </ul>
Retrieve sheets count	<p>Gets the number of sheets available in the current workbook and stores it in a number variable.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the appropriate option to specify whether to include the hidden worksheet or not and assign the count to a variable.</li> </ul>
Run macro	<p>Runs a macro in a worksheet.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Specify the name of the macro you want to run and its arguments.</li> </ul>
Switch to sheet	<p>Activates a particular sheet in a Microsoft Excel file.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Specify whether to activate the Sheet by Index (numerical value) or Sheet by Name.</li> </ul>
Unhide all worksheets	<p>Unhides all worksheets in the current workbook. Enter the name of the session used to open the current workbook with the Open action.</p>
Unhide worksheet	<p>Unhides a specific worksheet in the current workbook.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Enter the name of the worksheet you want to unhide.</li> </ul>
Write data table to worksheet	<p>Writes data available in a data table variable in a worksheet.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Specify the data table variable that contains the data you want to write in a worksheet.</li> <li>• Specify whether you want to write data in the currently Active worksheet or Specific worksheet.</li> <li>• Specify the address of the cell that you want to use as the starting point of the data in the Specify the first cell field.</li> </ul>

## Using Append worksheet action

Use the Append worksheet action to append a worksheet from another workbook to the current workbook.

Note: If the current workbook already has a worksheet with the same name, you must rename the worksheet being appended.

To append a worksheet, do the following:

### Procedure

1. Double-click or drag the Append worksheet action from the Excel node in the Actions palette.
2. Specify the name of the session in which you want to perform the action in the Session name field.
3. Select an option to specify the location of the spreadsheet from which you want to append the worksheet:
  - From 'My bots': Enables you to open a Microsoft Excel spreadsheet from an existing bot.
  - From local device: Enables you to open a Microsoft Excel spreadsheet from a local device.
  - Select an existing file variable: Enables you to open a Microsoft Excel spreadsheet using a file variable.
4. Select the Password is required option if the Microsoft Excel spreadsheet requires a password to open it.
5. Optional: In the User password or Owner password field, enter a password to restrict access to the encrypted PDF file.
  - User password: Allow users to perform specific operations on the encrypted PDF file.
  - Owner password: Allow users to use a password to open the file.
6. Select the Enter worksheet name or Enter worksheet index option to specify the name or number of the worksheet that you want to append.
7. Click Apply.

## Row and column operations

The Excel advanced package contains various actions that you can use to automate tasks related to the row and column operations in a Microsoft Excel spreadsheet.

### Row and column actions in the Excel advanced package

The Excel advanced package includes the following actions:

Action	Description
Get Column	<p>Retrieves the column that contains the specific cell and stores it to a string variable. This action supports .xlsx and .xlsm files.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the From active cell or From specific cell option to specify the cell location.</li> <li>• Select a string variable from the Save the column name to variable list.</li> </ul>
Get Row	<p>Retrieves the row that contains the specific cell and stores it to a string variable. This action supports .xlsx and .xlsm files.</p>

Action	Description
	<ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the From active cell or From specific cell option to specify the cell location.</li> <li>• Select a string variable from the Save the row name to variable list.</li> </ul>
Hide row(s)/column(s) in selection	<p>Hides rows or columns in the current worksheet.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the appropriate option to specify whether to hide one or more rows and columns.</li> </ul>
Insert/Delete Row(s)/Column(s)	See <a href="#">Using Insert or Delete actions for rows and columns.</a>
Read column	<p>Extracts data from a column and stores it in a list variable of string data type.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the From active cell or From specific cell option to specify the starting point. You can also select the Read full column option to extract data for the entire column.</li> <li>• Select a list variable of string data type that you want to use to store the extracted values.</li> </ul>
Read row	<p>Extracts data from a row and stores it in a list variable of string data type.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the From active cell or From specific cell option to specify the starting point. You can also select the Read full row option to extract data for the entire row.</li> <li>• Select a list variable of string data type that you want to use to store the extracted values.</li> </ul>
Remove blank rows	Removes blank rows from the current worksheet. You can specify the range from which you want to delete the blank rows.
Select cell(s)/row(s)/column(s)	See <a href="#">Using Select action for cells, rows, and columns.</a>
Unhides row(s)/column(s) in selection	<p>Unhides the hidden rows or columns in the current worksheet.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the appropriate option to specify whether you want to unhide a row or column and which row or column to unhide.</li> </ul>

## Using Insert or Delete actions for rows and columns

Use the Insert or Delete actions to create or remove rows or columns from the current worksheet or CSV file.

To insert or delete rows or columns in a worksheet, do the following:

### Procedure

1. Double-click or drag the Insert or Delete action from the Excel node in the Actions palette.
2. Specify the name of the session in which you want to perform the action in the Session name field.
3. Select the Row operations if you want to insert or delete rows from the spreadsheet.
  - a) Select the Insert Row at option to insert a row and specify the location where you want to insert the row in the field.
  - b) Select the Delete Row(s) at option to delete rows. You must specify the row number you want to delete in the field. For example, if you want to delete the tenth row in the worksheet, you must enter 10 in the field. If you want to delete the first five rows, you must enter 1 : 5 in the field.
4. Select the Column operations if you want to insert or delete columns from the spreadsheet.
  - a) Select the Insert Column at option to insert a column and specify the location where you want to insert the column in the field.
  - b) Select the Delete Column(s) at option to delete columns. You must specify the address of the column you want to delete in the field. For example, if you want to delete column 'D' in the worksheet, you must enter D in the field. If you want to delete the first five columns, you must enter A : E in the field.
5. Click Apply.

## Using Select action for cells, rows, and columns

Use the Select action to select cells, rows, or columns.

To select cells, rows, or columns do the following:

### Procedure

1. Double-click or drag the Select cell, row, or column action from the Excel node in the Actions palette.
2. Specify the name of the session in which you want to perform the action in the Session name field.
3. Select an option from the Select list to specify whether you want to select a cell, row, or column.
  - a) If you have selected the Cell option, select any of the following options:
    - b) Active cell: Enables you to select the active cell from the worksheet.
    - c) Specific cell: Enables you to select the cell you have specified in the field.
    - d) Cell range: Enables you to select all the cells in the range you have specified in the field.
    - e) All cells in the sheet: Enables you to select all the cells in the worksheet.
  - f) If you have selected the Column option, select any of the following options:
    - g) Column of active cell: Selects the column of the current active cell in the worksheet.
    - h) Specific column: Enables you to select the column you have specified in the field.
    - i) Column range: Enables you to select all the columns in the range you have specified in the field.
  - j) If you have selected the Row option, select any of the following options:
    - k) Row of active cell: Selects the row of the current active cell in the worksheet.

- l) Specific row: Enables you to select the row you have specified in the field.
  - m) Row range: Enables you to select all the rows in the range you have specified in the field.
4. Click Apply.

## Cell operations

The Excel advanced package contains various actions that you can use to automate tasks related to cell operations.

### Cell actions in the Excel advanced package

The Excel advanced package includes the following actions:

Action	Description
Delete cells	<p>Deletes the Active cell or a Specific cell from the current worksheet or a CSV file. After deleting the cell, you can:</p> <ul style="list-style-type: none"> <li>• Shift cells left: Deletes the specified cell and shifts the cell one position to the left.</li> <li>• Shift cells up: Deletes the specified cell and shifts the cell one position up.</li> <li>• Entire row: Deletes the entire row that contains the cell you have specified to delete.</li> <li>• Entire column: Deletes the entire column that contains the cell you have specified to delete.</li> </ul>
Find next empty cell	See <a href="#">Using Find next empty cell action.</a>
Find	See <a href="#">Using Find action.</a>
Get cell address	<p>Retrieves the location of the active cell and stores it to a string variable. This action supports .xlsx and .xlsm files.</p> <p>See <a href="#">Using the Get cell address action</a></p>
Get cell color	See <a href="#">Using Get cell color action.</a>
Get multiple cells	<p>Retrieves the values from the cells in a Microsoft Excel spreadsheet and stores them in a table variable.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the Multiple cells option to retrieve values from a range of cells, or select All cells to retrieve values from all the cells.</li> </ul>
Get number of rows	See <a href="#">Using Get number of rows action.</a>
Get single cell	Retrieves the values from a single cell in a Microsoft Excel spreadsheet or a CSV file and stores them in a string variable.

Action	Description
	<ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the Active cell option to retrieve the value from the active cell, or select Specific cell to retrieve the value from a specific cell in a Microsoft Excel spreadsheet.</li> </ul> <p>You can use the output value to <a href="#">Perform an action based on cell value</a>.</p>
Go to cell	<p>Moves the cursor to a specific cell in a Microsoft Excel spreadsheet or a CSV file.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the option to specify the cell to which the cursor should be moved, or select an option from the Active cell list.</li> </ul>
Go to next empty cell	<p>Finds the next empty cell in the current worksheet. You can specify whether to find the empty cell toward the left, right, up, or down.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the Active cell or Specified cell option to specify the cell from which to start searching for the empty cell. If you have selected the Specified cell option, specify the address of the cell in the field.</li> <li>• Select the left, right, up, or down option to specify the direction in which to search for the next empty cell.</li> </ul>
Read cell formula	<p>Gets the formula available in the Active cell or Specified cell and assigns the output to a string variable. This action returns a blank value if the specified cell does not contain a formula.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the Active cell or Specified cell option to specify the cell from which to read the formula. If you have selected the Specified cell option, specify the address of the cell in the field.</li> <li>• Select a string variable to store the name of the formula from the Assign the output to variable list.</li> </ul>
Replace	See <a href="#">Using the Replace action</a> .
Set cell	<p>Sets a value in the Active cell or Specific cell in a Microsoft Excel spreadsheet or a CSV file. You can also use this action to set a formula.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the Active cell or Specified cell option to specify the cell in which to set the value. If you have selected the Specified cell option, specify the address of the cell in the field.</li> <li>• Enter the value to set in the Cell value field.</li> </ul>

Action	Description
Set cell formula	<p>Sets a formula in the active cell or a specific cell in a Microsoft Excel spreadsheet or a CSV file. To generate a random number, use the Number &gt; Random action. See <a href="#">Random action</a>.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the Set formula for active cell or Set formula for specified cell option to specify the cell in which to set the formula. If you have selected the Set formula for specified cell option, specify the address of the cell in the field.</li> <li>• Enter the formula to set in the Enter formula for specific cell field.</li> </ul>

## Using Find action

Use the Find action to find a particular string in a Microsoft Excel spreadsheet or a CSV file.

To find a value in a Microsoft Excel spreadsheet, do the following:

### Procedure

1. Double-click or drag the Find action from the Excel node in the Actions palette.
2. Specify the name of the session in which you want to perform the action in the Session name field.
3. Select an option from the From list to specify a starting point of the cell range for search:
  - Beginning: Starts the search from the beginning of the spreadsheet.
  - End: Starts the search from the end of the spreadsheet.
  - Active cell: Starts the search from the active cell in the spreadsheet.
  - Specific cell: Enables you to specify the address of the cell from where you want to start the search.
4. Select an option from the Till list to specify an end point of the cell range for search:
  - Beginning: Ends the search at the beginning of the spreadsheet.
  - End: Ends the search at the end of the spreadsheet.
  - Active cell: Ends the search at the active cell in the spreadsheet.
  - Specific cell: Enables you to specify the address of the cell where you want to end the search.
5. Specify the string you want to search for in the Find field.
6. Select from the following search options:
  - By rows: Enables you to search by rows.
  - By columns: Enables you to search by columns.
  - Match by case: Only performs this action on cells that contain a string that matches the uppercase and lowercase characters of the string you specified in the Find field.
  - Match entire cell contents: Enables you to find only those cells that contain the entire string you have specified in the Find field.
7. Select the list variable of string data type that you want to use to store the output from the Assign cell addresses variable list.
8. In the Assign value to variable list, select a string variable.
9. Click Apply.

## Using the Replace action

Use the Replace action to find cells that contain a specific string and replace it with another string. This action supports .xlsx, .xlsm, and .csv files.

### Procedure

1. Double-click or drag the Replace action from the Excel advanced node in the Actions palette.
2. Specify the name of the session in which you want to perform the action in the Session name field.
3. Select an option from the From list to specify a starting point of the cell range for search:
  - Beginning: Starts the search from the beginning of the spreadsheet.
  - End: Starts the search from the end of the spreadsheet.
  - Active cell: Starts the search from the active cell in the spreadsheet.
  - Specific cell: Enables you to specify the address of the cell from where you want to start the search.
4. Select an option from the Till list to specify an end point of the cell range for search:
  - Beginning: Ends the search at the beginning of the spreadsheet.
  - End: Ends the search at the end of the spreadsheet.
  - Active cell: Ends the search at the active cell in the spreadsheet.
  - Specific cell: Enables you to specify the address of the cell where you want to end the search.
5. Specify the string you want to search for in the Find field.
6. Select from the following search options:
  - By rows: Enables you to search by rows.
  - By columns: Enables you to search by columns.
  - Match by case: Only performs this action on cells that contain a string that matches the uppercase and lowercase characters of the string you specified in the Find field.
  - Match entire cell contents: Enables you to find only those cells that contain the entire string you have specified in the Find field.
7. Select Replace with and specify the replacement string.
8. Select Replace with and specify the replacement string, or leave the field blank to replace the matching cells with an empty character.
9. Click Apply.
10. Click Save.

## Using Find next empty cell action

Use the Find next empty cell action to find the next empty cell in the current worksheet.

To find the next empty cell, do the following:

### Procedure

1. Double-click or drag the Find next empty cell action from the Excel node in the Actions palette.
2. Specify the name of the session in which you want to perform the action in the Session name field.
3. Select the row or column option from the Traverse by section to specify whether you want to find the empty cell in a row or column.
4. Select the active cell or specific cell option from the Start from section to specify the point from where you want to start the search.



5. Select the string variable you want to use to store the address of the empty cell from the Assign the output to variable list.
6. Click Apply.

## Using Get cell color action

Use the Get cell color action to get the color of the background or text in a cell. This action retrieves the color of a cell as RGB values. For example, if the background or text in a cell is of red color, the value retrieved is 255,0,0.

To get the color of the background or text in a cell, do the following:

### Procedure

1. Double-click or drag the Get cell color action from the Excel node in the Actions palette.
2. Specify the name of the session in which you want to perform the action in the Session name field.
3. Select the Background color option to get the background color of the cell or the Text color option to get the color of the text.
4. Select the Active cell option to get the color from the current active cell or the Specific cell option to get the color from the address of the cell you have specified.
5. Select a variable from the Assign the output to variable list to assign the address of the empty cell to the variable you have selected from the list.
6. Select the string variable you want to use to store the address of the empty cell from the Assign the output to variable list.
7. Click Apply.

## Using the Get cell address action

Use the Get cell address action to retrieve the location of the active cell and store it to a string variable. This action supports .xlsx and .xlsm files.

### Prerequisites

Open a worksheet using the Open action. You must select the Contains header option when configuring the Open action in order to use this action to retrieve the cell address based on the column title name.

### Procedure

1. Double-click or drag the Get cell address action from the Excel advanced node in the Actions palette.
2. Specify the name of the session in which you want to perform the action in the Session name field.
3. Select whether to retrieve the cell address from the Active cell or Based on header:

Option	Steps
Active cell	If you select this option, the bot will retrieve the cell address of the currently active cell. Skip to Step 5.

Option	Steps
Based on header	<p>If you select this option, the bot will retrieve the cell address of the cell specified by the column title name and cell position. Complete the following fields:</p> <p>a) Enter the column title: If you selected the Sheet contains header option when you opened this sheet using the Open action, enter the column name. For example, <u>First Name</u></p> <p>.</p> <p>If you did not select that option, enter the default column name. For example, <u>A</u></p> <p>.</p> <p>Note: This field is not case-sensitive.</p> <p>b) Enter cell position from column title: Enter the number of cells between the header cell and the cell from which you want to retrieve the address. Enter up to three digits.</p> <p>For example, if the header cell is located at B1 and you specify 2 as the cell position, the action will return B3.</p>

4. Select whether to retrieve the cell address from the Active cell or a Specific cell.

- If you have selected the Active cell option, the bot will retrieve the cell address of the currently active cell. Skip to Step 5.
- If you have selected the Specific cell option, the bot will retrieve the cell address of the cell specified by the column title name and cell position. Complete the following fields:

- Enter the column title: Enter the column name. For example,

Name

.

Note: This field is not case-sensitive.

- Enter cell position from column title: Enter the number of cells between the header cell and the cell from which you want to retrieve the address. Enter up to three digits.

For example, if the header cell is located at B1 and you specify 2 as the cell position, the action will return B3.

5. In the Save active or user specified cell address in local variable field, create or insert an existing string variable.

6. Click Apply.

7. Click Save.

## Next steps

Next, you can insert the string variable containing the cell address into the Specific cell field of actions related to cell operations. See [Cell operations](#).

## Using Get number of rows action

Use the Get number of rows action to get the number of rows that contain data.

To get the number of rows that are not empty or contain data, do the following:

### Procedure

1. Double-click or drag the Get number of rows action from the Excel node in the Actions palette.
2. Specify the name of the session in which you want to perform the action in the Session name field.
3. Select the Index option to specify the number of the worksheet or the Name option to specify the name of the worksheet from which you want to get the number or rows.
4. Select the Non-empty rows option to get the number of rows that are not empty or the Total rows with data option to get the number of rows that contain data.
5. Select the number variable you want to use to store the output from the Assign to variable list.
6. Click Apply.

## Perform an action based on cell value

In this example, you build a bot that prints a message based on whether a cell has a value or is empty. Use the actions from the Excel basic or Excel advanced, If, and Message Box packages.

### Prerequisites

Before building this bot, save an empty Excel worksheet to the desktop.

Because this bot does not create a new Excel worksheet, you can use either the Excel basic or Excel advanced packages.

Note: All of the actions must be from the same package.

### Procedure

1. Open a new bot.
  - a) From Automation Anywhere web interface, select Bots > My bots.
  - b) Click Create TaskBot.
  - c) Enter a bot name and click Create and Edit.
2. Use the Open action from the Excel basic or the Excel advanced package to open the Excel sheet.
  - a) Double-click or drag the Open action.
  - b) Enter a session name.
  - c) Click Browse to provide the file path to the empty Excel worksheet on the desktop.

- d) Click Apply.
3. Use the Get single cell action to assign the value of a cell to a string variable.
  - a) Double-click or drag the Get single cell action from the same package that you used for the Open action.
  - b) Provide the session name that you used in the Open action.
  - c) Select the Active cell option.
  - d) In the Store cell contents to field, create the variable `Output`.
  - e) Click Apply.
4. Use the If action to configure the conditional statement.
  - a) Double-click or drag the If action.
  - b) Select String from the Condition drop-down list.
  - c) In the Source value field, insert the variable `Output`.
  - d) Select Equals to as the Operator.
  - e) Leave the Target value field empty.
  - f) Click Apply.
5. Insert a Message box into the If container.
  - a) Drag the Message box action.
  - b) In the Enter the message to display field, enter  
`Cell is empty`
  - c) Click Apply.
6. Use the Else and Message box actions to configure the alternative sequence of actions.
  - a) Drag the Else action next to the If action.
  - b) Drag the Message box action into the Else container.
  - c) In the Enter the message to display field, enter  
`Cell is not empty`
  - d) Click Apply.
7. Click Save.
8. Run the bot.  
As the bot runs, the message box appears with the message `Cell is empty`.
9. Enter a value into the cell located at A1 in the Excel sheet and save the sheet.
10. Run the bot.  
As the bot runs, the message box appears with the message `Cell is not empty`.

## Table operations

The Excel advanced package contains various actions that you can use to automate tasks related to table operations in a Microsoft Excel spreadsheet.

### Table actions in the Excel advanced package

Note: Ensure that a table is available in the workbook. A worksheet that contains data in various rows and columns is not considered as a table.

The Excel advanced package includes the following actions:

Action	Description
Delete table column	Deletes a column in a table.

Action	Description
	<ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Specify the Table name from which you want to delete a column.</li> <li>• Select the Name or Position to specify the name or position of the column to delete.</li> </ul>
Filter table	See <a href="#">Using Filter table action</a> .
Get table range	<p>Gets the range of a table available in a worksheet and stores the output in a string variable.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Specify the Table name for which you want to get the range.</li> <li>• Select the options to specify whether you want to include a header and pivot table in the range and a variable to store the output.</li> </ul>
Insert table column	<p>Inserts a column in a table.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Specify the Table name in which you want to insert a column.</li> <li>• Specify the name of the column in the Column name field and the position where you want to insert the column in the Column position field.</li> </ul>
Sort table	<p>Sorts the data in a column of a table. This action enables you to sort numeric and text data.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Specify the Table name for which you want to sort the data.</li> <li>• Select the Column name or Column position to specify the name or position of the column.</li> <li>• Select an appropriate option from the Number or Text list to specify the sort order.</li> </ul>

## Using Filter table action

Use the Filter table action to filter data from a column in a table.

To filter data in a table, do the following:

### Procedure

1. Double-click or drag the Filter table action from the Excel node in the Actions palette.

2. Specify the name of the session in which you want to perform the action in the Session name field.
3. Specify the name of the table from which you want to filter data in the Table name field.
4. Select the Column name to specify the name of the column or the Column position to specify the position of the column that contains the data you want to filter.
5. Select the Number option if the column you have specified contains number data.
  - a) Select an option from the list to specify the operator you want to use to filter the data.  
The following options are available:
    - b) Equals: Filters the data that is equal to the value you have specified.
    - c) Does not equal: Filters the data that is not equal to the value you have specified.
    - d) Greater than: Filters the data that is greater than the value you have specified.
    - e) Greater than or equal to: Filters the data that is greater than or equal to the value you have specified.
    - f) Less than: Filters the data that is less than the value you have specified.
    - g) Less than or equal to: Filters the data that is less than or equal to the value you have specified.
    - h) Between: Filters the data that is between the two values you have specified.

Note: Apart from the Between option, you do not have to provide a value in the second field. If you have provided a value in the second field, the system ignores that value when filtering the data.
6. Select the Text option if the column you have specified contains textual data.
  - a) Select an option from the list to specify the operator you want to use to filter the data.  
The following options are available:
    - b) Equals: Filters the data that is equal to the value you have specified.
    - c) Does not equal: Filters the data that is not equal to the value you have specified.
    - d) Begins with: Filters the data that begins with the value you have specified.
    - e) Ends with: Filters the data that ends with the value you have specified.
    - f) Contains: Filters the data that contains the value you have specified.
    - g) Does not contain: Filters the data that does not contain the value you have specified.
7. Click Apply.

## CSV operations

You can use some of the actions available in the Excel advanced package to perform various operations on a CSV file.

### CSV actions in the Excel advanced package

The following table lists the actions that you can use with a CSV file:

Supported	Not Supported
Open	Find next empty cell
Set cell	Get cell colour
Go to cell	Go to next empty cell
Insert/Delete row column	Run macro
Insert/Delete row column	Unhide all worksheets
Get cell	Hide row/column in selection
Find/Replace	Unhide row/column in selection
Convert excel to pdf	Access password protected worksheet

Delete cells	Protect/Unprotect workbook
Set cell formula	Table related commands
Create workbook	Create worksheet

You can use the Loop action to retrieve data from each row in a CSV file. You can also use any of the above action that supports the use of a CSV file within a Loop action.

Note: All the other actions available in the Excel advanced package that are not listed in the above table cannot be used with a CSV file.

## Example: Migrate data from Excel to a database

Transfer data from a Microsoft Excel spreadsheet to a database using the Database, Excel, and Loop actions.

### Procedure

1. Open the spreadsheet.
  - a) Drag Excel > Open.
  - b) Select the spreadsheet filepath.
  - c) If the workbook contains multiple sheets, select the Specific Sheet Name check box and enter the sheet name.
  - d) Select the option for Open in Edit mode.
  - e) If the sheet contains a header row, select the Contains Header check box to exclude the first row from the automation.
  - f) Click Save.
2. Get data from the sheet.
  - a) Drag Excel > Get Multiple Cells.
  - b) Enter the cell range. For example  
B2:D10
  - c) Click Save.

When the task runs, this action stores the cell values in the system variable excelColumn.
3. Connect to the database.
  - a) Drag Database > Connect.
  - b) Enter the connection string.
  - c) Click Save.
4. Insert the Microsoft Excel data into the database.
  - a) Drag Loop.
  - b) Select the Excel > For each row in worksheet iterator.
  - c) Drag Database > Insert/Update/Delete into the Loop container.
  - d) Enter an insert SQL statement.  
For example:  
INSERT INTO OrderMaster value ("Excel  
Column (1) \$", \$Excel Column (2) \$", \$Excel  
Column (3) \$")
  - e) Click Save.
5. Disconnect from the database.

- a) Drag Database > Disconnect.
- b) Click Save.
- 6. Close the spreadsheet.
  - a) Drag Excel > Close Spreadsheet.
  - b) Click Save.

## File package

The File package contains actions that enable you to automate various file-related operations such as creating, opening, copying, deleting, and renaming a file.

Use the actions available in the File package either individually to perform an action one time or in a Loop action to perform that action for each file available in a folder.

## Actions in the File package

Note: You can use the Zip action available in the Folder package to compress a file. See [Using Zip action](#).

The File package includes the following actions:

Action	Description
Assign	Assigns a file from the Enterprise Control Room, Desktop, or an existing File variable to a user-defined File variable.
Copy	See <a href="#">Using Copy action for file</a> .
Create	<p>Creates a new file.</p> <ul style="list-style-type: none"> <li>In the File field, specify the name including the location to store the file.</li> <li>In the Overwrite an existing file option, select an existing file with the same name to overwrite it. If existing files are not overwritten, the Enterprise Control Room appends the name of the new file with a numeric value.</li> </ul>
Delete	See <a href="#">Using Delete action for file</a> .
Download CR file	<p>Downloads a file from the Enterprise Control Room to a specified location on the device.</p> <p>Note: You cannot use this action to download a bot or a file from the My Scripts folder.</p> <ul style="list-style-type: none"> <li>In the Select a Control Room file field, specify the file to download.</li> <li>In the Save CR file to location field, specify the path where you want to save the file. You can insert a variable that holds the entire file path, or part of the file path. For example, C:\user\My Docs\test.csv or C:\user\My Docs\\${fileName}\$</li> <li>Note: If the folder structure does not already exist, the bot creates the folders as needed.</li> <li>Select True or False, or insert a Boolean variable to specify whether to overwrite the existing file.</li> </ul>



Action	Description
Open	Opens an existing file. In the File field, specify the name including the location to store the file.
Print	See <a href="#">Using Print action for file.</a>
Print multiple files	See <a href="#">Using Print multiple files action.</a>
Rename	See <a href="#">Using Rename action for file.</a>

Related reference

[If package](#)

[Loop package](#)

[Variables overview](#)

## Using Copy action for file

Use the Copy action to copy an existing file. This action enables you to copy a file based on its size and the date it was created or modified.

### Procedure

To copy an existing file, follow these steps:

1. In the Actions palette, double-click or drag the Copy action from the File package.
2. In the Source file field, specify the name and location of the file.
3. In the Destination file/folder field, specify the name of the file and location.
4. Select the Overwrite existing files check box to overwrite an existing file with the same name at the specified location.  
If this option is not selected, the system appends the name of the copied file with a numeric value. For example, if you have copied a file named June\_Quarter\_report.pdf and a file with the same name exists in the location where you want to save the copied file, the system saves the copied file as June\_Quarter\_report\_(1).pdf. The numeric value is incremented each time you copy a file with the same name when the overwrite check box is not selected.
5. Select the Size check box to copy a file based on its size.
  - a) Select any of the following options from the list:
    - b) Atleast: Copies a file only if the file size is more than the size you have specified.
    - c) Atmost: Copies a file only if the file size is less than the size you have specified.
    - d) Exact: Copies a file only if the file size is the same as the size you have specified.
  - e) Specify a value in the Size field.
6. Select the Date check box to copy a file based on the date it was created or modified.
  - a) Select any of the following options from the list:
    - b) Created: Enables you to copy a file based on the date it was created.
    - c) Modified: Enables you to copy a file based on the date it was modified.
  - d) Select the Is within last option and specify a value in the field.  
This option enables you to perform the operation on the file if it was created or modified within the number of days you have specified. For example, if you specify  
7  
in the field, the system performs the operation on the file if it was created or modified in the last 7 days.
  - e) Select the Is between option to specify a period.

This option enables you to specify a Start date and an End date for the period. For example, if you specify 01/01/19

as the start date and

01/31/19

as the end date of the period, the system performs the operation on the file if it was created or modified between this period.

Note: The Start date and End date are included in the period.

f) Select the Is before option and specify a value in the field.

This option enables you to perform the operation on the file if it was created or modified on or before the date you have specified.

7. Click Apply.

8. Click Save.

## Using Delete action for file

Use the Delete action to delete a file. This action enables you to delete a file based on its size and the date it was created or modified.

### Procedure

Follow these steps to delete a file:

1. In the Actions palette, double-click or drag the Delete action from the File package.
2. In the File field, specify the name and location of the file.
3. Select the Size check box to delete a file based on its size.
  - a) Select any of the following options from the list:
    - b) Atleast: Deletes a file only if the file size is more than the size you have specified.
    - c) Atmost: Deletes a file only if the file size is less than the size you have specified.
    - d) Exact: Deletes a file only if the file size is the same as the size you have specified.
  - e) Specify a value in the Size field.
4. Select the Date check box to delete a file based on the date it was created or modified.
  - a) Select any of the following options from the list:
    - b) Created: Enables you to delete a file based on the date it was created.
    - c) Modified: Enables you to delete a file based on the date it was modified.
  - d) Select the Is within last option and specify a value in the field.
 

This option enables you to perform the operation on the file if it was created or modified within the number of days you have specified. For example, if you specify

7

in the field, the system performs the operation on the file if it was created or modified in the last 7 days.
  - e) Select the Is between option to specify a period.
 

This option enables you to specify a Start date and an End date for the period. For example, if you specify

01/01/19

as the start date and

01/31/19

as the end date of the period, the system performs the operation on the file if it was created or modified between this period.

Note: The Start date and End date are included in the period.
  - f) Select the Is before option and specify a value in the field.

This option enables you to perform the operation on the file if it was created or modified on or before the date you have specified.

5. Click Apply.
6. Click Save.

## Using Rename action for file

Use the Rename action to rename a file. This action enables you to rename a file based on its size and the date it was created or modified.

### Procedure

To rename a file, follow these steps:

1. In the Actions palette, double-click or drag the Rename action from the File package.
2. In the File field, specify the name and location of the file you want to rename.
3. In the New file name field, specify a new name for the file.
4. Select the Size check box to rename a file based on its size.
  - a) Select any of the following options from the list:
    - b) Atleast: Renames a file only if the file size is more than the size you have specified.
    - c) Atmost: Renames a file only if the file size is less than the size you have specified.
    - d) Exact: Renames a file only if the file size is the same as the size you have specified.
  - e) Specify a value in the Size field.
5. Select the Date check box to rename a file based on the date it was created or modified.
  - a) Select any of the following options from the list:
    - b) Created: Enables you to rename a file based on the date it was created.
    - c) Modified: Enables you to rename a file based on the date it was modified.
  - d) Select the Is within last option and specify a value in the field.  
 This option enables you to perform the operation on the file if it was created or modified within the number of days you have specified. For example, if you specify  
 7  
 in the field, the system performs the operation on the file if it was created or modified in the last 7 days.
  - e) Select the Is between option to specify a period.  
 This option enables you to specify a Start date and an End date for the period. For example, if you specify  
 01/01/19  
 as the start date and  
 01/31/19  
 as the end date of the period, the system performs the operation on the file if it was created or modified between this period.  
 Note: The Start date and End date are included in the period.
  - f) Select the Is before option and specify a value in the field.  
 This option enables you to perform the operation on the file if it was created or modified on or before the date you have specified.
6. Click Apply.
7. Click Save.

## Using Print action for file

Use the Print action to print a file. This action enables you to print a file based on its size and the date it was created or modified.

### Procedure

Follow these steps to print a file:

1. In the Actions palette, double-click or drag the Print action from the File package.
2. In the File field, specify the name and location of the file.
3. Select the Size check box to print a file based on its size.
  - a) Select any of the following options from the list:
    - b) Atleast: Prints a file only if the file size is more than the size you have specified.
    - c) Atmost: Prints a file only if the file size is less than the size you have specified.
    - d) Exact: Prints a file only if the file size is the same as the size you have specified.
  - e) Specify a value in the Size field.
4. Select the Date check box to print a file based on the date it was created or modified.
  - a) Select any of the following options from the list:
    - b) Created: Enables you to print a file based on the date it was created.
    - c) Modified: Enables you to print a file based on the date it was modified.
  - d) Select the Is within last option and specify a value in the field.  
 This option enables you to perform the operation on the file if it was created or modified within the number of days you have specified. For example, if you specify  
 7  
 in the field, the system performs the operation on the file if it was created or modified in the last 7 days.
  - e) Select the Is between option to specify a period.  
 This option enables you to specify a Start date and an End date for the period. For example, if you specify  
 01/01/19  
 as the start date and  
 01/31/19  
 as the end date of the period, the system performs the operation on the file if it was created or modified between this period.  
 Note: The Start date and End date are included in the period.
  - f) Select the Is before option and specify a value in the field.  
 This option enables you to perform the operation on the file if it was created or modified on or before the date you have specified.
5. Click Apply.
6. Click Save.

## Using Print multiple files action

Use Print multiple files to print multiple files based on the size and the date they were created or modified.

### Procedure

Follow these steps to print multiple files:

1. In the Actions palette, double-click or drag the Print multiple files action from the File package.
2. In the Folder field, specify the name and location of the folder.
3. Optional: In the File type field, specify the file type of your files.
4. Select the Include subfolders check box to include all the subfolders in your selected folder.
5. Select the Size check box to print multiple files based on their size.
  - a) Select any of the following options from the list:
    - b) Atleast: Prints multiple files only if the file size is more than the size you have specified.
    - c) Atmost: Prints multiple files only if the file size is less than the size you have specified.
    - d) Exact: Prints multiple files only if the file size is the same as the size you have specified.
  - e) Specify a value in the Size field.
6. Select the Date check box to print multiple files based on the date they were created or modified.
  - a) Select any of the following options from the list:
    - b) Created: Enables you to print multiple files based on the date they were created.
    - c) Modified: Enables you to print multiple files based on the date they were modified.
  - d) Select the Is within last option and specify a value in the field.  
 This option enables you to perform the operation on the files if they were created or modified within the number of days you have specified. For example, if you specify  
 7  
 in the field, the system performs the operation on the files if they were created or modified in the last 7 days.
  - e) Select the Is between option to specify a period.  
 This option enables you to specify a Start date and an End date for the period. For example, if you specify  
 01/01/19  
 as the start date and  
 01/31/19  
 as the end date of the period, the system performs the operation on the files if they were created or modified between this period.  
 Note: The Start date and End date are included in the period.
  - f) Select the Is before option and specify a value in the field.  
 This option enables you to perform the operation on the files if they were created or modified on or before the date you have specified.
7. Click Apply.
8. Click Save.

## Folder package

The Folder package contains actions that enable you to automate folder-related operations.

## Actions in the Folder package

The Folder package includes the following actions:

Action	Description
Copy	See <a href="#">Using Copy action</a> .
Create	Creates a new folder. <ul style="list-style-type: none"> <li>• In the Folder field, specify the name and location.                Note: Folder names cannot include system actions or device references such as AUX, CON, LPT, NUL, and PRN.</li> </ul>

Action	Description
	<ul style="list-style-type: none"> <li>Select the Overwrite an existing folder check box to overwrite an existing folder.</li> </ul> <p>Note: If this option is not selected, the system appends the name of the new folder with a numeric value.</p>
Delete	See <a href="#">Using Delete action</a> .
Open	<p>Opens a folder at a specific location.</p> <p>In the Folder field, specify the name and location.</p>
Rename	See <a href="#">Using Rename action</a> .
Unzip	See <a href="#">Using Unzip action</a> .
Zip	See <a href="#">Using Zip action</a> .

## Using Copy action

Use the Copy action to copy an existing folder.

### Procedure

1. In the Actions palette, double-click or drag the Copy action from the Folder package.
2. In the Source folder field, specify the name and location.
3. In the Destination folder field, specify the folder name and location to save the copied folder.
4. Select the Overwrite existing files/folders check box to overwrite existing folders with the same name.  
 Note: If this option is not selected, the system appends a numeric value at the end of the file name. For example, if you save the June\_Quarter\_report file in a location that has a file with the same name, the system saves the file as June\_Quarter\_report\_(1).pdf. The numeric value is incremented each time you save the file as long as the option is selected.
5. Select the Size check box to copy a folder based on its size.
  - a) Select any of the following options from the list:
    - b) Atleast: Copies a folder only if the folder size is more than the size you have specified.
    - c) Atmost: Copies a folder only if the folder size is less than the size you have specified.
    - d) Exact: Copies a folder only if the folder size is the same as the size you have specified.
  - e) In the Size field, specify the folder size.
6. Select the Date check box to copy a folder based on the date.
  - a) Select any of the following options from the list:
    - b) Created: Enables you to copy a folder based on the date it was created.
    - c) Modified: Enables you to copy a folder based on the date it was modified.
  - d) In the Is within last, specify the value.  
 This option enables you to perform the operation on the folder if it was created or modified within the last number of days you have specified. For example, if you specify  
 7  
 in the field, the system performs the operation on the folder if it was created or modified in the last 7 days.
  - e) In the Is between, specify the period.

This option enables you to specify a Start date and an End date of the period. For example, if you specify  
01/01/19

and

01/31/19

as the start date and end date of the period, the system performs the operation on the folder if it was created or modified between this period.

Note: The Start date and End date are included in the period.

f) In the Is before, specify the value.

This option enables you to perform the operation on the folder if it was created or modified on or before the date you have specified.

7. Click Apply.

8. Click Save.

## Using Delete action

Use the Delete action to delete a folder.

### Procedure

1. In the Actions palette, double-click or drag the Delete action from the Folder package.

2. In the Folder field, specify the name and location.

3. Select the Size check box to delete a folder based on its size.

a) Select any of the following options from the list:

- b) Atleast: Deletes a folder only if the folder size is more than the size you have specified.
- c) Atmost: Deletes a folder only if the folder size is less than the size you have specified.
- d) Exact: Deletes a folder only if the folder size is the same as the size you have specified.

e) In the Size field, specify the folder size.

4. Select the Date check box to delete a folder based on the date it was created or modified.

a) Select any of the following options from the list:

- b) Created: Enables you to delete a folder based on the date it was created.
- c) Modified: Enables you to delete a folder based on the date it was modified.

d) In the Is within last, specify the value.

This option enables you to perform the operation on the folder if it was created or modified within the last number of days you have specified. For example, if you specify

7

in the field, the system performs the operation on the folder if it was created or modified in the last 7 days.

e) In the Is between, specify the period.

This option enables you to specify a Start date and an End date of the period. For example, if you specify

01/01/19

and

01/31/19

as the start date and end date of the period, the system performs the operation on the folder if it was created or modified between this period.

Note: The Start date and End date are included in the period.

f) In the Is before, specify the value.

This option enables you to perform the operation on the folder if it was created or modified on or before the date you have specified.

5. Click Apply.

6. Click Save.

## Using Rename action

Use the Rename action to rename a folder.

### Procedure

1. In the Actions palette, double-click or drag the Rename action from the Folder package.
2. In the Folder field, specify the name and location.
3. In the New folder name field, specify a new name.
4. Select the Size check box to rename a folder based on its size.
  - a) Select any of the following options from the list:
    - b) Atleast: Renames a folder only if the folder size is more than the size you have specified.
    - c) Atmost: Renames a folder only if the folder size is less than the size you have specified.
    - d) Exact: Renames a folder only if the folder size is the same as the size you have specified.
  - e) In the Size field, specify the folder size.
5. Select the Date check box to rename a folder based on the date it was created or modified.
  - a) Select any of the following options from the list:
    - b) Created: Enables you to rename a folder based on the date it was created.
    - c) Modified: Enables you to rename a folder based on the date it was modified.
  - d) In the Is within last, specify the value.  
 This option enables you to perform the operation on the folder if it was created or modified within the last number of days you have specified. For example, if you specify  
 7  
 in the field, the system performs the operation on the folder if it was created or modified in the last 7 days.
  - e) In the Is between, specify the period.  
 This option enables you to specify a Start date and an End date of the period. For example, if you specify  
 01/01/19  
 and  
 01/31/19  
 as the start date and end date of the period, the system performs the operation on the folder if it was created or modified between this period.  
 Note: The Start date and End date are included in the period.
  - f) In the Is before, specify the value.  
 This option enables you to perform the operation on the folder if it was created or modified on or before the date you have specified.
6. Click Apply.
7. Click Save.

## Using Unzip action

Use the Unzip action to extract compressed files and folders from a zip file to a specific location.

### Procedure

1. In the Actions palette, double-click or drag the Unzip action from the Folder package.
2. In the Zip file name with full path field, specify the name and location.



3. In the Extract to path field, specify the location.
4. Select the Replace existing file check box to overwrite the file.
5. Optional: In the Password to access zip file field, select either Credential to specify a stored password, or select String to enter a password manually.
6. Click Apply.
7. Click Save.

## Using Zip action

Use the Zip action to compress files and folders into a zip file.

### Procedure

1. In the Actions palette, double-click or drag the Zip action from the Folder package.
2. In the Specify file(s)/folder(s) to compress field, specify the location.
3. Optional: In the Specify file type(s) to compress field, specify the extension.  
For example, you can specify .txt and .png to compress only text files and PNG images. The system compresses the entire folder if you do not specify any file type. You can also specify the name of a file to compress a specific file.
4. In the Specify destination filename and location field, specify the zip file name and location.
5. Select the Update only if newer check box to compress the files only if one or more files were updated after the last compression.  
This option is useful when you are repeatedly compressing the same set of files and storing the output zip file with the same name at the same location. For example, you compress five PDF files from the Reports folder on a monthly basis and save the output zip file with the Monthly Report name in the D: drive. When this option is selected, the system compresses the files only if one or more PDF files are updated after the last compression.
6. Select the Delete original files check box if you want to delete the original files after they are included in the zip file.
7. In the Compression list, specify the speed for compression.
8. Optional: In the Password protection field, select Credential to specify a stored password or select String to enter a password manually.
9. Click Apply.
10. Click Save.

### FTP / SFTP package

Use the FTP / SFTP package to automate FTP / SFTP operations.

An FTP/SFTP server hosts the files to share. The client accesses, downloads, or uploads files to the server. The transfer of data between the client and server is done using a TCP/IP network, which is the standard protocol of communication over the internet.

- The following are some forms of authentication for an FTP server:
  - User credentials: Requires an FTP username and password.
  - Anonymous: This form of authentication is enabled on sites where files are available for public access and the users need not identify themselves to the server.
  - Key-based: SFTP authentication is usually done with a private and public key. The key pair is automatically generated by the computer. The private key is kept with the SFTP client and the

corresponding public key with the SFTP server. When establishing a connection, the client shares the private key to be matched with the corresponding public key on the server.

- FTP can run in active or passive transfer mode.
  - In the active mode, the client informs the server about the port used for listening and starts listening for incoming data connections from the server.
  - In the passive mode, the client receives a server IP address and server port number from the server. The client opens a data connection to the server IP address and server port number that are received. Most organizations prefer the passive mode because it involves less or no alterations to the firewall settings.
- FTP supports binary and ASCII file transfer types:
  - Use the binary option when transferring executable files.
  - Use the ASCII option when transferring text files.

## Before you start

Perform the following actions within the FTP / SFTP package.

1. Establish a connection with the FTP/SFTP server using the Connect action. When establishing a connection, associate the FTP/SFTP server details with a session name. Use the same session name for all corresponding actions.
2. Use the FTP/SFTP actions to automate a task. The actions enable you to perform following tasks:
  - Upload, download, delete, or rename files.
  - Upload, download, create, or delete folders.
  - Navigate to the parent folder or a specific folder.
3. After you have automated the tasks related to FTP / SFTP, terminate the connection to the server using the Disconnect action.

Note: SFTP with SSH1 protocol is no longer supported by SFTP.

## Actions in the FTP / SFTP package

The FTP / SFTP package includes the following actions:

Action	Description
Change folder	<p>Navigates to either the parent folder or another specific folder on an FTP/SFTP server.</p> <ul style="list-style-type: none"> <li>• Enter the session name.</li> </ul> <p>Use the same session name that you have provided when establishing a connection with the FTP/SFTP server using the Connect action.</p> <ul style="list-style-type: none"> <li>• Select either Go to parent folder or Specific folder in Navigate to options.</li> </ul> <p>If the Specific folder option is selected, enter the complete folder path.</p>
Connect	See <a href="#">Using Connect action for FTP/SFTP</a> .
Create folder	<p>Creates a folder in an existing folder on the FTP/SFTP server.</p> <ul style="list-style-type: none"> <li>• Enter the session name.</li> </ul>

Action	Description
	<p>Use the same session name that you have provided when establishing a connection with the FTP/SFTP server using the Connect action.</p> <ul style="list-style-type: none"> <li>Specify the folder name in the Remote folder field with the path to create a folder on the FTP server.</li> </ul> <p>Note: The system encounters an error if a folder with the same name exists in the parent folder.</p>
Delete files	<p>Deletes files from an FTP/SFTP folder.</p> <ul style="list-style-type: none"> <li>Enter the session name.</li> </ul> <p>Use the same session name that you have provided when establishing a connection with the FTP/SFTP server using the Connect action.</p> <ul style="list-style-type: none"> <li>Specify file names in the Remote files field along with the complete path, separated by a semicolon.</li> </ul>
Delete folder	<p>Deletes a folder (including all the subfolders and files within it) from an FTP/SFTP server.</p> <ul style="list-style-type: none"> <li>Enter the session name.</li> </ul> <p>Use the same session name that you have provided when establishing a connection with the FTP/SFTP server using the Connect action.</p> <ul style="list-style-type: none"> <li>Specify the folder name in the Remote folder field.</li> </ul>
Disconnect	<p>Terminates the connection to the FTP/SFTP server.</p> <p>Enter the session name – Use the same session name that you have provided when establishing a connection with the FTP/SFTP server using the Connect action.</p>
Get files	<p>Downloads files from a remote FTP/SFTP folder to a specific folder on the client machine.</p> <ul style="list-style-type: none"> <li>Enter the session name.</li> </ul> <p>Use the same session name that you have provided when establishing a connection with the FTP/SFTP server using the Connect action.</p> <ul style="list-style-type: none"> <li>Select files on the remote FTP server that you want to download on the local system.</li> <li>In the Local folder field, enter the name of the folder where you want to download files from the FTP/SFTP server.</li> <li>Select the Transfer Type as Binary or ASCII.</li> <li>You can also specify folders based on the date they are created or modified.</li> </ul>

Action	Description
Get folders	<p>Downloads a folder from an FTP/SFTP server to a client machine.</p> <ul style="list-style-type: none"> <li>• Enter the session name.</li> </ul> <p>Use the same session name that you have provided when establishing a connection with the FTP/SFTP server using the Connect action.</p> <ul style="list-style-type: none"> <li>• Select folders on the remote FTP server that you want to download on the local system.</li> </ul> <p>Specify filters using wildcard characters to download only specific types of files.</p> <ul style="list-style-type: none"> <li>• In the Local folder field, enter the name of the folder where you want to download files from the FTP/SFTP server.</li> <li>• You can also specify folders based on the date they are created or modified.</li> </ul>
Put files	<p>Uploads one or more files from the client machine to the FTP/SFTP server.</p> <ul style="list-style-type: none"> <li>• Enter the session name. Use the same session name that you have provided when establishing a connection with the FTP/SFTP server using the Connect action.</li> <li>• Enter all the file names along with their location, separated by a semicolon.</li> <li>• Select the Transfer type to be either Binary or ASCII</li> </ul>
Put folders	<p>Uploads a folder from the client machine to the FTP/SFTP server.</p> <ul style="list-style-type: none"> <li>• Enter the session name. Use the same session name that you have provided when establishing a connection with the FTP/SFTP server using the Connect action.</li> <li>• Enter the folder name in the Local folder field which you want to upload on FTP/SFTP server.</li> <li>• Specify filters using wild card characters to restrict uploads to specific types of files.</li> </ul>
Rename files	<p>Renames files in an FTP/SFTP folder.</p> <ul style="list-style-type: none"> <li>• Enter the session name.</li> </ul> <p>Use the same session name that you have provided when establishing a connection with the FTP/SFTP server using the Connect action.</p> <ul style="list-style-type: none"> <li>• In the Remote file field, enter the name of the file that you want to rename.</li> <li>• Enter the new name in the New remote file field.</li> </ul>

- [Using Connect action for FTP/SFTP](#)

Use the Connect action to establish a connection with the FTP/SFTP server that you want to use to automate tasks. This must be the first action you use to automate an FTP/SFTP-related task.

## Using Connect action for FTP/SFTP

Use the Connect action to establish a connection with the FTP/SFTP server that you want to use to automate tasks. This must be the first action you use to automate an FTP/SFTP-related task.

Specify details of the FTP/SFTP server and associate it with a session name. Use the session name provided in this action in the other actions so that you do not have to provide details of the server in those actions.

### Procedure

To establish a connection with an FTP/SFTP server, perform these steps:

1. Double-click or drag the Connect action from the FTP/SFTP node in the Actions palette.
2. Enter a name for the session in the Session name field.
3. Enter the FTP/SFTP server name in the Server name field.
4. Specify the port number.
5. Select one of the following options to specify the server type:

Option	Steps
FTP	<p>Authentication type: You can either choose to authenticate the user using User credentials or log in as Anonymous.</p> <p>If you choose User credentials, complete these fields:</p> <p>Username: Select Credential for enhanced security or String:</p> <ul style="list-style-type: none"> <li>• Credential: Select the Credential Vault variable to insert the Username and Password.</li> <li>• String: Enter the value directly in the field.</li> </ul> <p>Transfer Mode: Select the transfer mode to be Active or Passive.</p> <p>Transfer type: Select the type to be either Binary or ASCII for transferring files between the server and client.</p> <p>Optional: Enter the default directory path on the FTP server.</p>
FTP Secure	<p>Authentication type: You can either choose to authenticate the user using User credentials or log in as Anonymous. If you choose User credentials, complete these fields:</p> <p>Username: Select Credential for enhanced security or String:</p> <ul style="list-style-type: none"> <li>• Credential: Select the Credential Vault variable to insert the Username and Password.</li> <li>• String: Enter the value directly in the field.</li> </ul>
SSH FTP	<p>Select the Private Key File from Control Room file, Desktop file, or Variable.</p> <p>Note: Ensure that the SFTP server has a corresponding public key file.</p>

Option	Steps
	<p>You can either choose to authenticate the user using User credentials or log in as Anonymous. If you choose User credentials, complete these fields:</p> <ul style="list-style-type: none"> <li>• With username: To enter the Username, select Credential or String option.</li> </ul> <p>Credential: Select the Credential Vault variable to insert the Username.</p> <p>String: Enter the value directly in the field.</p> <ul style="list-style-type: none"> <li>• With username-password: To enter the Username and Password, select either the Credential or String option.</li> </ul> <p>If you choose to authenticate using the user credentials, insert a Credential Vault variable for enhanced security or enter the value directly in the string field.</p>

6. Select Reconnect if connection fails if you want to automatically reconnect:

Note: If the connection fails due to incorrect credentials, the action will not attempt to reconnect.

- a) Specify the number of attempts in the Attempts field.
- b) Select the time lapse in the Time between attempts field.

7. Click Apply, and then click Save.

## Fuzzy match package

Use the Fuzzy match action to compare the values of two strings or files for similarity. This action returns a decimal value; the closer the value to 1.0, the greater the similarity between the two strings.

Use this action to automate the process of evaluating strings of data for similarity. For example, you have an automation sequence in which a bot extracts data from invoices, searches a database for the company record, and updates the record with data from the invoice. Use the Fuzzy match action to handle two possible scenarios:

- A mistake occurs at the extraction step where a letter is incorrectly extracted. Instead of Apple, the bot extracts App1e, with a numerical one instead of the letter l.
- There is a small variation between the company name on the invoice and in the database. The invoice contains the company name Apple Inc, but the database has a record for the company name Apple.

## G-Suite Apps package

The G-Suite Apps package contains the OAuth action, which enables you to authorize and connect to the G-Suite server. With this package, you only have to provide your credentials once.

Use the OAuth action to establish a connection with the G-Suite server using the client ID, redirect URI, and client secret. Insert this action at the beginning of automation sequences that use Google packages so that you have to input the credentials only once. All of the fields in this action accept a credential from the Credential Vault or a user-input value.

To use this action, provide the following:

- Client ID: Identifies the client application.
- Redirect URI: Identifies the application that receives the data from Google.
- Client Secret: Client access token.

To retrieve your credentials, see [Obtain OAuth Credentials](#).

## Google Calendar package

The Google Calendar package contains actions that enable you to automate creating and deleting events.

### Before you start

Perform the following actions within the Google packages as part of using the set of available actions:

1. Use the OAuth action from the G-Suite Apps package to establish a connection to the G-Suite server. For more information on establishing a connection, see the [G-Suite Apps package](#).
2. Use a combination of actions available in the Google packages to automate your tasks.

Note: You can automate tasks using actions from different packages with the same connection.

## Actions in the Google Calendar package

The Google Calendar package includes the following actions:

Action	Description
Create event	See <a href="#">Using the Create new calendar event action</a> .
Delete event	<p>Removes the event from the calendar.</p> <ul style="list-style-type: none"><li>• Enter the same session name that you provided in the OAuth action.</li><li>• Enter the Event Id.</li><li>• Optional: Assign the event to a String variable.</li></ul>

- [Using the Create new calendar event action](#)

Use the Create new calendar event action to specify event attendees, duration, location, recurrence, and title. During run time, this action triggers an email notification to meeting attendees.

## Using the Create new calendar event action

Use the Create new calendar event action to specify event attendees, duration, location, recurrence, and title. During run time, this action triggers an email notification to meeting attendees.

### Procedure

To create a new calendar event, do the following:

1. Enter the session name you used to connect to the G-Suite server in the OAuth action.  
See [G-Suite Apps package](#).
2. Enter the event title.
3. Optional: Enter the location.
4. Optional: Enter the attendees' email addresses, separated with commas.
5. Enter the start date.  
Use the format yyyy-MM-dd.
6. Enter the end date.  
Use the format yyyy-MM-dd.
7. Select the All Day or Specify Time option from the Event Time option.
  - If you select the All Day option, you do not need to provide any additional details.
  - If you select the Specify Time option, complete the following fields:
    - Specify the Start Time using the HH:mm:ss format.
    - Specify the End Time using the HH:mm:ss format.
8. Select the Use System Timezone or Specify Timezone option from the Timezone option.
  - If you select the Use System Timezone option, you do not need to provide any additional details.
  - If you select the Specify Timezone option, complete the following fields:
    - Specify the Start Timezone.
    - Specify the End Timezone.
9. Optional: Mark the Recurring option to make this event repeat.
10. Select a Visibility option from the drop-down list.  
Select from Default, Public, or Private.
11. Optional: Enter an event description.
12. Optional: Select a string variable from the drop-down list to store the id of the created event.
13. Click Apply.

## Google Drive package

The Google Drive package contains actions that enable you to automate tasks related to files and folders.

## Before you start

Perform the following actions within the Google packages as part of using the set of available actions:

1. Use the OAuth action from the G-Suite Apps package to establish a connection to the G-Suite server. For more information on establishing a connection, see the [G-Suite Apps package](#).
2. Use a combination of actions available in the Google packages to automate your tasks.  
Note: You can automate tasks using actions from different packages with the same connection.

## Actions in the Google Drive package

The Google Drive package includes the following actions:

Action	Description
Copy file	<p>Copies a file from one folder to another in the Google Drive</p> <ul style="list-style-type: none"> <li>• Enter the same session name that you provided in the OAuth action.</li> <li>• Select whether to identify the source file by file path and name, or by ID.</li> </ul>



Action	Description
	<p>Note: The ID is the value in the URL after the last forward slash. For example, if the URL is <code>https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9</code>, the ID is <code>12dlAwvrEA4JeLysfFky9</code>.</p> <ul style="list-style-type: none"> <li>• Select whether to identify the destination folder by directory path and name, or by ID.</li> <li>• Optional: Mark the Rename file option and enter the new file name.</li> </ul>
Create file permission	<p>Creates a new permission for a file.</p> <ul style="list-style-type: none"> <li>• Enter the same session name that you provided in the OAuth action.</li> <li>• Select whether to identify the file by file path and name, or by ID. Note: The ID is the value in the URL after the last forward slash. For example, if the URL is <code>https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9</code>, the ID is <code>12dlAwvrEA4JeLysfFky9</code>.</li> <li>• Select a role. For more information, see <a href="#">Google Drive Roles</a>.</li> <li>• Select the grantee type. Choose from: <ul style="list-style-type: none"> <li>• User</li> <li>• Group</li> <li>• Domain</li> <li>• Anyone</li> </ul> </li> <li>• If you select User or Group, enter the email address. If you select domain, enter the domain.</li> <li>• Optional: Select or create a variable to hold the permission ID.</li> </ul>
Delete file	<p>Deletes a file on your Google Drive.</p> <ul style="list-style-type: none"> <li>• Enter the same session name that you provided in the OAuth action.</li> <li>• Select whether to identify the file by file path and name, or by ID. Note: The ID is the value in the URL after the last forward slash. For example, if the URL is <code>https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9</code>, the ID is <code>12dlAwvrEA4JeLysfFky9</code>.</li> <li>• Optional: Select or create a boolean variable to hold the file delete status.</li> </ul>
Delete file permission	<p>Deletes a file permission for a file.</p> <ul style="list-style-type: none"> <li>• Enter the same session name that you provided in the OAuth action.</li> <li>• Select whether to identify the file by file path and name, or by ID. Note: The ID is the value in the URL after the last forward slash. For example, if the URL is <code>https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9</code>, the ID is <code>12dlAwvrEA4JeLysfFky9</code>.</li> <li>• In the Permission id field, enter the permission ID.</li> <li>• Optional: Select or create a variable to hold the permission ID.</li> </ul>
Download file	<p>Downloads a file from your Google Drive to a specific location on your desktop.</p> <ul style="list-style-type: none"> <li>• Enter the same session name that you provided in the OAuth action.</li> <li>• Select whether to identify the file by file path and name, or by ID. Note: The ID is the value in the URL after the last forward slash. For example, if the URL is <code>https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9</code>, the ID is <code>12dlAwvrEA4JeLysfFky9</code>.</li> <li>• Specify the download file path. For example, <code>C:/Users/Downloads</code>.</li> </ul>

Action	Description
	<ul style="list-style-type: none"> <li>Optional: Mark the Overwrite existing file option to overwrite the file of the same name.</li> <li>Optional: Mark the Rename file option to enter a new name for the downloaded file.</li> <li>Optional: Select or create a variable to hold the ID of the downloaded file.</li> </ul>
Find file/folder	<p>Finds files or folders in a specific directory in your Google Drive.</p> <ul style="list-style-type: none"> <li>Enter the same session name that you provided in the OAuth action.</li> <li>Select whether to search for files or folders.</li> <li>Enter the source folder to search in. For example, /Home/Accounts/.</li> <li>Specify whether to return exact matches or</li> <li>Enter the file or folder name to search for.</li> <li>Select or create a variable to hold the list of files or folders.</li> </ul>
Get file information	<p>Gets file information of a file.</p> <ul style="list-style-type: none"> <li>Enter the same session name that you provided in the OAuth action.</li> <li>Select whether to identify the file by file path and name, or by ID. Note: The ID is the value in the URL after the last forward slash. For example, if the URL is <code>https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9, the 12dlAwvrEA4JeLysfFky9.</code></li> <li>Select or create a dictionary variable to hold the file information which is stored in two keys. The <code>name</code> key holds the file name and the <code>extension</code> key holds the file extension.</li> </ul>
Get file permission	<p>Retrieves file information for specific file or folder from Google Drive.</p> <ul style="list-style-type: none"> <li>Enter the same session name that you provided in the OAuth action.</li> <li>Select whether to identify the file by file path and name, or by ID. Note: The ID is the value in the URL after the last forward slash. For example, if the URL is <code>https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9, the 12dlAwvrEA4JeLysfFky9.</code></li> <li>Select or create a list variable to hold the file permissions.</li> </ul>
Move file	<p>Moves a file from one folder to another.</p> <ul style="list-style-type: none"> <li>Enter the same session name that you provided in the OAuth action.</li> <li>Select whether to identify the file by file path and name, or by ID. Note: The ID is the value in the URL after the last forward slash. For example, if the URL is <code>https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9, the 12dlAwvrEA4JeLysfFky9.</code></li> <li>Select whether to identify the destination folder by directory path and name, or by ID. Note: The ID is the value in the URL after the last forward slash. For example, if the URL is <code>https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9, the 12dlAwvrEA4JeLysfFky9.</code></li> </ul>
Open file	<p>Opens a file from the specified directory in Google Drive.</p> <ul style="list-style-type: none"> <li>Enter the same session name that you provided in the OAuth action.</li> <li>Select whether to identify the file by file path and name, or by ID.</li> </ul>

Action	Description
	<p>Note: The ID is the value in the URL after the last forward slash. For example, if the URL is <a href="https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9">https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9</a>, the ID is 12dlAwvrEA4JeLysfFky9.</p>
Open folder	<p>Opens a folder from the specified directory in Google Drive.</p> <ul style="list-style-type: none"> <li>• Enter the same session name that you provided in the OAuth action.</li> <li>• Select whether to identify the folder by directory path and name, or by ID.</li> </ul> <p>Note: The ID is the value in the URL after the last forward slash. For example, if the URL is <a href="https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9">https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9</a>, the ID is 12dlAwvrEA4JeLysfFky9.</p>
Rename file	<p>Renames a file in the Google Drive.</p> <ul style="list-style-type: none"> <li>• Enter the same session name that you provided in the OAuth action.</li> <li>• Select whether to identify the file by file path and name, or by ID.</li> </ul> <p>Note: The ID is the value in the URL after the last forward slash. For example, if the URL is <a href="https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9">https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9</a>, the ID is 12dlAwvrEA4JeLysfFky9.</p> <ul style="list-style-type: none"> <li>• Enter the new file name.</li> </ul>
Rename folder	<p>Renames a folder in a specific directory in Google Drive.</p> <ul style="list-style-type: none"> <li>• Enter the same session name that you provided in the OAuth action.</li> <li>• Select whether to identify the folder by directory path and name, or by ID.</li> </ul> <p>Note: The ID is the value in the URL after the last forward slash. For example, if the URL is <a href="https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9">https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9</a>, the ID is 12dlAwvrEA4JeLysfFky9.</p> <ul style="list-style-type: none"> <li>• Enter the new folder name.</li> </ul>
Upload file	<p>Uploads a file from the desktop to your Google Drive.</p> <ul style="list-style-type: none"> <li>• Enter the same session name that you provided in the OAuth action.</li> <li>• Enter the file path and name.</li> <li>• Select whether to identify the folder where to upload the file by directory path and name, or by ID.</li> </ul> <p>Note: The ID is the value in the URL after the last forward slash. For example, if the URL is <a href="https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9">https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9</a>, the ID is 12dlAwvrEA4JeLysfFky9.</p> <ul style="list-style-type: none"> <li>• Optional: Mark the Overwrite existing file option to overwrite the file of the same name.</li> <li>• Optional: Select or create a variable to hold the ID of the uploaded file.</li> </ul>
Update file permission	<p>Update the permissions for a file.</p> <ul style="list-style-type: none"> <li>• Enter the same session name that you provided in the OAuth action.</li> <li>• Select whether to identify the file by file path and name, or by ID.</li> </ul> <p>Note: The ID is the value in the URL after the last forward slash. For example, if the URL is <a href="https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9">https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9</a>, the ID is 12dlAwvrEA4JeLysfFky9.</p>

Action	Description
	<ul style="list-style-type: none"> <li>In the Permission id field, enter your user ID.</li> <li>Select a role. For more information, see <a href="#">Google Drive Roles</a>.</li> <li>Optional: Select or create a variable to hold the permission ID.</li> </ul>

## Google Sheets package

The Google Sheets package contains actions that enable you to automate tasks involving cells, columns, rows, and sheets.

### Before you start

Perform the following actions within the Google packages as part of using the set of available actions:

1. Use the OAuth action from the G-Suite Apps package to establish a connection to the G-Suite server. For more information on establishing a connection, see the [G-Suite Apps package](#).
2. Use the Open spreadsheet action from the Google Sheets package to select a workbook, or the Create spreadsheet action to create a new workbook.
3. Optional: If the workbook contains more than one sheet, use the Activate sheet action to specify which sheet to use.
4. Use the Close action to exit from the workbook.

## Actions in the Google Sheets package

The actions in the Google Sheets package enable you to perform the following operations:

Action	Description
Activate sheet	<p>Activates a specific sheet in the open spreadsheet.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current spreadsheet with the Open spreadsheet action.</li> <li>Specify whether to activate the sheet by index or name.</li> </ul>
Create spreadsheet	<p>Creates a new spreadsheet in your Google Drive.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current spreadsheet with the Open spreadsheet action.</li> <li>Optional: Enter the file path. If you do not enter a value, the spreadsheet will be created under my-drive.</li> <li>Enter the spreadsheet name.</li> <li>Select a variable to hold the ID of the created spreadsheet.</li> </ul>
Close	Saves and closes the current spreadsheet. Enter the name of the session used to open the current spreadsheet with the Open spreadsheet action.
Create worksheet	Creates a new sheet in the open spreadsheet.

Action	Description
	<ul style="list-style-type: none"> <li>Enter the name of the session used to open the current spreadsheet with the Open spreadsheet action.</li> <li>Specify whether to create the sheet by index or name</li> </ul>
Delete cells	<p>Deletes cells within a worksheet.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current spreadsheet with the Open spreadsheet action.</li> <li>Select either the Active cell or Specific cell option. If you have selected the Specific cell, enter the cell address.</li> <li>Select from the following delete options: <ul style="list-style-type: none"> <li>Shift cells left</li> <li>Shift cells up</li> <li>Entire row</li> <li>Entire column</li> </ul> </li> </ul>
Delete worksheet	<p>Deletes a sheet from the current spreadsheet.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current spreadsheet with the Open spreadsheet action.</li> <li>Specify whether to delete the sheet by index or name.</li> </ul>
Find cells which contains search key	See <a href="#">Using the Find action</a>
Get multiple cells	<p>Retrieves the values from the cells in a Google sheet and stores them in a table variable.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current spreadsheet with the Open spreadsheet action.</li> <li>Select the Multiple cells option to retrieve values from a range of cells, or select All cells to retrieve values from all the cells.</li> <li>Select or create a table variable to hold the output.</li> </ul>
Get single cell	<p>Retrieves the value from the specific cell in a Google sheet and stores it in a string variable.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current spreadsheet with the Open spreadsheet action.</li> <li>Select either the Active cell or Specific cell option. If you have selected the Specific cell, enter the cell address.</li> <li>Select or create a string variable to hold the output.</li> </ul>
Go to cell	<p>Goes to the specified cell.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current spreadsheet with the Open spreadsheet action.</li> <li>Select either the Specific cell or Active cell option.</li> </ul>

Action	Description
	<ul style="list-style-type: none"> <li>If you have selected the Specific cell option, enter the cell address. For example, <code>B4</code>.</li> <li>If you have selected the Active cell option, select the direction in which to move from the active cell.</li> </ul>
Open spreadsheet	<p>Opens an existing spreadsheet.</p> <ul style="list-style-type: none"> <li>Enter a session name. Use this session name in subsequent actions to associate them with this spreadsheet.</li> <li>Select whether to open the spreadsheet by name, URL, or spreadsheet ID: <ul style="list-style-type: none"> <li>Name: Enter the file name.</li> <li>URL: Enter the entire URL.</li> <li>Spreadsheet ID: The ID is the value in the URL after the last forward slash. For example, if the URL is <code>https://docs.google.com/spreadsheets/d/12dlAwvrEA4JeLysfFky9</code>, the ID is <code>12dlAwvrEA4JeLysfFky9</code>.</li> </ul> </li> <li>If the spreadsheet contains multiple sheets, select the Specific sheet name option and enter the sheet name.</li> </ul>
Read column	<p>Retrieves data from a column and stores it in a list variable.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current spreadsheet with the Open spreadsheet action.</li> <li>In the Cell name field, specify the cell location from which to read the values. For example, enter <code>A5</code> to retrieve all the values located in the first column, below the fifth row.</li> <li>You can also select the Read full column option to extract values from the entire column.</li> <li>Select or create a list variable to hold the output.</li> </ul>
Read row	<p>Retrieves data from a row and stores it in a list variable.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current spreadsheet with the Open spreadsheet action.</li> <li>In the Cell name field, specify the cell location from which to read the values. For example, enter <code>D5</code> to retrieve the values located in the fifth row and right of the fourth column.</li> <li>You can also select the Read full row option to extract values from the entire row.</li> <li>Select or create a list variable to hold the output.</li> </ul>
Set cell	<p>Sets the value of a specific cell.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current spreadsheet with the Open spreadsheet action.</li> <li>Enter the address of the cell. For example <code>A5</code>.</li> <li>Enter the value to set.</li> </ul>

Action	Description
	<ul style="list-style-type: none"> <li>Select or create a string variable to hold the value of the cell.</li> </ul>

- [Using the Find action](#)

Use the Find cells which contain search key action to find a particular string in a Google sheet.

## Using the Find action

Use the Find cells which contain search key action to find a particular string in a Google sheet.

### Procedure

To find a value in a Google sheet, do the following:

1. Double-click or drag the Find cells which contain search key action from the Google Sheets node in the Actions palette.
2. Enter the same session name that you provided in the OAuth action.
3. Select an option from the From list to specify a starting point of the cell range for search:
  - Beginning: Starts the search from the beginning of the spreadsheet.
  - End: Starts the search from the end of the spreadsheet.
  - Active cell: Starts the search from the active cell in the spreadsheet.
  - Specific cell: Enables you to specify the address of the cell from where you want to start the search.
4. Select an option from the Till list to specify an end point of the cell range for search:
  - Beginning: Ends the search at the beginning of the spreadsheet.
  - End: Ends the search at the end of the spreadsheet.
  - Active cell: Ends the search at the active cell in the spreadsheet.
  - Specific cell: Enables you to specify the address of the cell where you want to end the search.
5. Select from the following search options:
  - By rows: Enables you to search by rows.
  - By columns: Enables you to search by columns.
  - Match by case: Only performs this action on cells that contain a string that matches the uppercase and lowercase characters of the string you specified in the Find field.
  - Match entire cell contents: Enables you to find only those cells that contain the entire string you have specified in the Find field.
6. Enter the string you want to search for in the Find field.
7. Optional: Mark the Replace option and enter the string to replace the matching values.
8. Select the list variable of string data type that you want to use to store the output from the Assign cell addresses variable list.
9. Click Apply.

### IBM Watson Authentication package

The IBM Watson Authentication package contains actions that enable you to authenticate the API token and location URL for each service, while connecting to and disconnecting from your IBM Cloud account. With this package, you only have to provide your credentials once.

## Before you start

Each service requires a unique API key and location URL. Log in to your IBM Cloud account to create an instance of the service to obtain the API Key and location URL. Automation Anywhere Version A2019 currently includes [Speech to Text](#).

## Actions in the IBM Watson Authentication package

Action	Description
Connect	Establishes a connection with the IBM Cloud account. Insert this action at the beginning of automation sequences that use IBM Watson packages so that you have to input the credentials only once. To use this action, select the services to connect with, then provide the API Key and URL for each service you want to connect.
Disconnect	Terminates the connection with the IBM Cloud account. Insert this action at the end of automation sequences that use IBM Watson packages.

## IBM Watson Speech to Text package

This package supports the following audio file formats: flac, mpeg, mp3, ogg, pcm, wav, and webm. The following languages are supported: Arabic, Brazilian Portuguese, Chinese (Mandarin), English (United Kingdom and United States), French, German, Japanese, Korean, Spanish (Argentinian, Castilian, Chilean, Colombian, Mexican, and Peruvian).

Feature	Description
Detect speakers	<p>Identifies the individuals in a conversation between multiple people.</p> <ul style="list-style-type: none"> <li>• Supports English, Japanese, and Spanish.</li> <li>• Use for conversation between two people; maximum six people.</li> <li>• For best results, use an audio file at least a minute long.</li> </ul> <p>The output contains the words spoken by each speaker and the timestamp.</p>
Keyword spotting	Detects specific strings in the transcript. The output contains the timestamp(s) for each keyword and a confidence score.
Smart formatting	<p>Converts the following types of strings into more conventional representations to make the transcript easier to read:</p> <ul style="list-style-type: none"> <li>• Dates</li> <li>• Times</li> <li>• Series of digits and numbers</li> <li>• Phone numbers</li> <li>• Currency values</li> <li>• Email and web addresses</li> </ul> <p>For examples, see <a href="#">Smart formatting results</a>. This feature supports English, Japanese, and Spanish.</p>
Profanity filter	Obscures profanity by replacing it with asterisks in the transcript.



## If package

Use the actions in the If package to control the sequence of execution based on one or more conditions of a task.

Use the actions in the If package to check if an application is running, a folder or file exists, a variable matches the specified value, an application window exists, or a machine or server is running, before executing a set of actions.

## Actions in the If package

The If package includes the following actions:

Action	Description
If	<p>Specifies a condition and holds a sequence of actions to run if the condition is true. See <a href="#">Using If action</a>. You can configure multiple conditions within a single If action.</p> <ol style="list-style-type: none"> <li>1. Click Add condition.</li> <li>2. Select either of the following options: <ul style="list-style-type: none"> <li>• And: Both of the conditions must be met for the actions to run.</li> <li>• Or: Either of the conditions must be met for the actions to run.</li> </ul> </li> <li>3. Select the condition from the drop-down list.</li> </ol>
Else if (optional)	<p>Specifies an alternative condition to test if the condition specified in the If action is false. If this alternative condition is true, the bot runs the sequence of actions contained within the Else if action. You can configure multiple conditions within a single Else if action.</p> <ol style="list-style-type: none"> <li>1. Click Add condition.</li> <li>2. Select either of the following options: <ul style="list-style-type: none"> <li>• And: Both of the conditions must be met for the actions to run.</li> <li>• Or: Either of the conditions must be met for the actions to run.</li> </ul> </li> <li>3. Select the condition from the drop-down list.</li> </ol>
Else	<p>Specifies the alternative sequence of actions if the condition specified in the If action and Else if action (if used) is false.</p>

## Conditions in the If action

Use the following conditions in the If action to control the flow of execution in an automation task:

Condition	Description
Application	<p>Use the Application is not running or Application is running condition to execute actions based on whether an application is running or not running.</p> <p>Enter the application path or specify the path using a variable along with the amount of time to wait (in seconds) for the condition to be true.</p>

Condition	Description
Boolean	<p>Use the Boolean variable condition to execute actions based on whether a Boolean variable contains the value True or False.</p> <p>You can also use this condition to compare the values of two Boolean variables by selecting the second variable from the Insert a Variable window.</p>
Datetime	Use the Datetime variable condition to execute actions based on whether the value of the source datetime variable is Equal to or Not Equal to, is Greater than or Equal to, or is Lesser than or Equal to the value of the target datetime variable.
Dictionary	Use this condition to execute actions based on whether the specified key exists in the dictionary.
File	<p>Use the following conditions to execute the action:</p> <ul style="list-style-type: none"> <li>File date <p>Use this condition to verify the date and time that the specified file was created or modified. Specify a date range using the Is within last, Is between, or Is before options. For the Is within last option, specify the number of days or time (in hours, minutes, and seconds). Enter the amount of time (in seconds) to wait for this condition to be true.</p> </li> <li>File exists and File does not exist <p>Use these conditions to execute an action based on whether a file exists. For example, if a data file exists, format the file and upload it to a database.</p> </li> <li>File size <p>This condition verifies if the specified file is larger, smaller, not the same, or the same as the size you specify.</p> <p>Enter the amount of time (in seconds) to wait for this condition to be true.</p> </li> </ul>
Folder	Use the Folder exists or Folder does not exist condition to execute an action based on whether a folder exists.
Image Recognition	<p>Use these conditions to verify whether:</p> <ul style="list-style-type: none"> <li>Image file is found in the Image file or not.</li> <li>Image file is found in the Window or not.</li> <li>Window is found in the Image file or not.</li> <li>Window is found in the Window or not.</li> </ul>
JavaScript	Use the Script is successful or Script is unsuccessful condition to execute actions based on the status of the specified JavaScript. Select the file that contains the script and optionally specify the parameters by selecting a list variable.
Legacy automation	Use the Legacy automation conditions to execute actions on bots migrated from Version 11.3. Use the following conditions to verify the following:

Condition	Description
	<ul style="list-style-type: none"> <li>Whether Web control exists or not.</li> <li>Whether Web control is active or not.</li> <li>Whether Window control is active or not.</li> </ul>
List	Use the List variable condition to execute actions based on whether the specified list variable contains a particular value. The value can be of Number, String, or Boolean data type.
Number	Use the Number variable condition to execute actions based on whether the specified number variable is Equal to or Not Equal to, or is Greater than or Equal to, or is Lesser than or Equal to a particular value.
Ping	Use the Ping is successful or Ping is unsuccessful condition to verify if a machine or server is running, and execute actions based on the result. Enter the amount of time (in seconds) to wait for the condition to be true.
Recorder	Use this condition to detect an Object in a window. Select a window or variable to capture the object. Enter the amount of time (in seconds) to wait for this condition to be true.
String	<p>Use the String variable condition to execute actions based on whether the specified source string value is Equal to or Not equal to, or Includes or Does not include the target value.</p> <p>You can select the Case sensitive option to only execute actions if the two strings have matching uppercase and lowercase letters.</p> <p>To create a condition based on whether a string is empty or not empty, compare the source value to an empty target field using the Equal to operator. See <a href="#">Perform an action based on cell value</a>.</p>
Task Bot	Use the Task successful or Task unsuccessful condition to execute actions based on the status of the specified Task Bot.
VBScript	Use the Script is successful or Script is unsuccessful condition to execute actions based on the status of the specified Visual Basic script. Select the file that contains the script and optionally specify the parameters by selecting a variable.
Window	Use the Window exists or Window does not exist condition to verify if a specific application window is open by entering the Window title or using a variable. Enter the amount of time (in seconds) to wait for the condition to be true.

## Example

### [Build a basic bot that uses a desktop application](#)

An example of how to build a basic TaskBot that uses the If package.

- [Using If action](#)  
Use the If Else If and If Else actions to change the flow of execution in an automation task based on certain conditions.

- [Perform an action based on cell value](#)

In this example, you build a bot that prints a message based on whether a cell has a value or is empty. Use the actions from the Excel basic or Excel advanced, If, and Message Box packages.

## Using If action

Use the If Else If and If Else actions to change the flow of execution in an automation task based on certain conditions.

To use the actions in the If package, do the following:

### Procedure

1. Double-click or drag the If action from the If package in the Actions palette.
2. Select the required condition from the Condition list.  
See [If package](#) for a list of available conditions.
3. Drag the actions to be executed if the condition is satisfied within the If condition.
4. Double-click or drag the Else If action from the If package in the Actions palette to include alternative actions to run if the condition for the If action is false, and if the condition for the Else If action is true.
5. Drag the actions to be executed if the condition is satisfied within the Else If condition.
6. Double-click or drag the Else action from the If package in the Actions palette to include actions to run if the conditions for the If and Else If actions are false.
7. Drag the actions to be executed if the condition is satisfied within the Else condition.
8. Click Apply.

## Perform an action based on cell value

In this example, you build a bot that prints a message based on whether a cell has a value or is empty. Use the actions from the Excel basic or Excel advanced, If, and Message Box packages.

### Prerequisites

Before building this bot, save an empty Excel worksheet to the desktop.

Because this bot does not create a new Excel worksheet, you can use either the Excel basic or Excel advanced packages.

Note: All of the actions must be from the same package.

### Procedure

1. Open a new bot.
  - a) From Automation Anywhere web interface, select Bots > My bots.
  - b) Click Create TaskBot.
  - c) Enter a bot name and click Create and Edit.
2. Use the Open action from the Excel basic or the Excel advanced package to open the Excel sheet.
  - a) Double-click or drag the Open action.

- b) Enter a session name.
  - c) Click Browse to provide the file path to the empty Excel worksheet on the desktop.
  - d) Click Apply.
3. Use the Get single cell action to assign the value of a cell to a string variable.
  - a) Double-click or drag the Get single cell action from the same package that you used for the Open action.
  - b) Provide the session name that you used in the Open action.
  - c) Select the Active cell option.
  - d) In the Store cell contents to field, create the variable `Output`.
  - e) Click Apply.
4. Use the If action to configure the conditional statement.
  - a) Double-click or drag the If action.
  - b) Select String from the Condition drop-down list.
  - c) In the Source value field, insert the variable `Output`.
  - d) Select Equals to as the Operator.
  - e) Leave the Target value field empty.
  - f) Click Apply.
5. Insert a Message box into the If container.
  - a) Drag the Message box action.
  - b) In the Enter the message to display field, enter  
`Cell is empty`
  - c) Click Apply.
6. Use the Else and Message box actions to configure the alternative sequence of actions.
  - a) Drag the Else action next to the If action.
  - b) Drag the Message box action into the Else container.
  - c) In the Enter the message to display field, enter  
`Cell is not empty`
  - d) Click Apply.
7. Click Save.
8. Run the bot.  
As the bot runs, the message box appears with the message `Cell is empty`.
9. Enter a value into the cell located at A1 in the Excel sheet and save the sheet.
10. Run the bot.  
As the bot runs, the message box appears with the message `Cell is not empty`.

## Image Recognition package

The Image Recognition package contains actions that enable you to search for a user interface (UI) element in an application based on an image to automate a task in that application.

## Actions in the Image Recognition package

Use these actions to automate a task when it is not possible to capture UI elements of applications that are:

- Exposed over Citrix
- Accessed using the Remote Desktop Protocol (RDP)
- Developed using legacy technology

Image recognition is also useful when object-based recognition does not work or is unreliable. For example, you can use an image to search for the Close button in an application and perform a left-click operation.

The Image Recognition package contains the following actions:

Action	Description
Find image in window	See <a href="#">Using Find image in window action</a> .
Find window in window	See <a href="#">Using the Find window in window action</a> .

## Secure Recording

When secure recording mode is enabled, bots do not capture object images or values. This ensures that sensitive data is not stored in the bots.

When you record a task in secure recording mode, the Preview window temporarily shows the captured area. This image is deleted after you click Apply and navigate away from the action editor window.

A user with admin privileges must enable this setting. See [Settings](#).

## Using Find image in window action

Use the Find image in window action to search for a UI element in an application window (haystack) using a target image (needle). The target image is an existing image that you can use to search for a UI element.

### Procedure

To find an image inside an application window, follow these steps:

1. Double-click or drag the Find image in window action from the Image Recognition package in the Actions palette.
2. Specify the window (haystack) in which you want to find the image:  
Choose from the Window or Variable tab.
  - Click the Window tab to select the application window from the list.

Select the Currently active window, Desktop, or Taskbar, or a window from the list of Available windows.

- Click the Variable tab to insert an existing window variable to specify the title of the application window you want to use.
3. Click Capture region.  
The selected window appears.
  4. Drag the mouse to select the area and right-click when done.  
The captured area appears in the Preview section with the image coordinates underneath.
  5. In the Wait before capturing the image (ms) field, specify the delay (in milliseconds) before searching for the image.
  6. Specify the target image (needle) that you want to find in the application window.
    - Control Room file: Uses an image file that is available on the Enterprise Control Room.
    - Desktop file: Uses an image file that is available on your device.

- Variable: Uses a file variable to specify the location of the image file you want to use.

Note: Images of .jpeg, .jpg, .jpe, .jif, .bmp, and .gif formats are supported.

7. In the Image occurrence field, enter a value to specify the occurrence of the target image on which you want to perform this action.
8. In the Match percentage field, specify the acceptable percentage of matching pixels between the two images. For example, if you specify  
20  
in the field as the match percentage, the system considers the images as matching even if there is up to 80% of pixel mismatch between the two images.
9. Select a click action:
  - Click match: During runtime, the bot clicks in the center of the matched image.
  - Offset from match: During runtime, the bot clicks in the specified offset coordinates.  
Note: The offset coordinates measure the number of pixels from the top left corner of the image.
10. Select an option from the Action list to specify the action you want to perform on the matched image in the application window.
11. Optional: Select the Repeat if image not found check box if you want the system to retry searching for the target image if it is not found.
  - a) In the Times field, specify the number of times the system must repeat the process to find the target image.
  - b) In the Wait between repeats field, specify the time period the system must wait before repeating the process of finding the target image.
12. Click Apply.

## Using the Find window in window action

Use the Find window in window action to search for a UI element in an application window using a screenshot of a window. This action enables you to capture an image of a UI element in an application and use the captured image to search for that UI element in another window.

### Procedure

To use an image available in an application window to find an image in another application window, follow these steps:

1. Double-click or drag the Find window in window action from the Image Recognition package in the Actions palette.
2. Specify the window (haystack) in which you want to find the image:  
Choose from the Window or Variable tab.
  - Click the Window tab to select the application window from the list.  
  
Select the Currently active window, Desktop, or Taskbar, or a window from the list of Available windows.
  - Click the Variable tab to insert an existing window variable to specify the title of the application window you want to use.
3. Click Capture region.  
The selected window appears.
4. Drag the mouse to select the area and right-click when done.  
The captured area appears in the Preview section with the image coordinates underneath.
5. Specify the target image (needle) that you want to find in the application window.

- Captured image: Enables you to capture the image relative to the screen or window.

If you select Captured image, perform steps 7 through 9. Otherwise, skip to step 10.

- Control Room file: Uses an image file that is available on the Enterprise Control Room.
- Desktop file: Uses an image file that is available on your device.
- Variable: Uses a file variable to specify the location of the image file you want to use.

Note: Images of .jpeg, .jpg, .jpe, .jif, .bmp, and .gif formats are supported.

- Specify whether to capture the image relative to the screen or window.  
Use the Window option when you work with varying screen resolutions (for example, a dual monitor).
- Click Capture image to capture the target image.
- Drag the mouse pointer over an area of the application window.  
The captured area appears in the Preview section.
- In the Image occurrence field, enter a value to specify the occurrence of the target image on which you want to perform this action.
- In the Match percentage field, specify the acceptable percentage of matching pixels between the two images.  
For example, if you specify  
20  
in the field as the match percentage, the system considers the images as matching even if there is up to 80% of pixel mismatch between the two images.
- Select a click action:
  - Click match: During runtime, the bot clicks in the center of the matched image.
  - Offset from match: During runtime, the bot clicks in the specified offset coordinates.  
Note: The offset coordinates measure the number of pixels from the top left corner of the image.
- Select an option from the Action list to specify the action you want to perform on the matched image in the application window.
- Optional: Select the Repeat if image not found check box if you want the system to retry searching for the target image if it is not found.
  - In the Times field, specify the number of times the system must repeat the process to find the target image.
  - In the Wait between repeats field, specify the time period the system must wait before repeating the process of finding the target image.
- Click Apply.

## Interactive forms package

The interactive forms package contains actions that handle exceptions encountered by a bot. All the actions performed by users on the interactive forms can be monitored to execute logic using subtasks.

Interactive forms are first-class citizens within the bot repository and have the same workflows for moving forms between public and private workspaces, and for export or import actions. Handling exceptions ensures that a bot completes a task when it encounters an error.

## Actions in the Interactive forms package

The interactive forms package includes form-level and element-level actions.

The following actions are available at the form level:



Action	Description
Display	<p>Loads and displays the selected form.</p> <p>Select or insert a variable in the Form name field to display the form when an event is triggered. Optionally, select the check box to always display the form window in front.</p>
Close	Closes the selected form.
Show	<p>Loads or displays a hidden form.</p> <p>Select or insert a variable in the Form name field for a hidden form, which is displayed on the desktop when an event is triggered.</p>
Hide	Hides the selected form from the user's view.
Reset	<p>Resets the selected form to its default value.</p> <p>Select or insert a variable in the Form name field for a form. All the values of this specified form are reset when an event is triggered.</p>

The following actions are available at the element level:

Action	Description
Enable	<p>Enables the specified element of the selected form.</p> <ul style="list-style-type: none"> <li>• Select or insert a variable in the Form name field.</li> <li>• Select a specific element of the form from the Form element field, which is enabled when an event is triggered.</li> </ul>
Disable	Disables the specified element of the selected form.
Get	<p>Retrieves the value from the specified element of the selected form.</p> <ul style="list-style-type: none"> <li>• Select or insert a variable in the Form name field and use the Form element to select a specific element of the form.</li> <li>• Use the Returns the value drop-down menu to assign this value to a variable when an event is triggered.</li> </ul> <p><a href="#">Create a variable</a></p>
Set	<p>Assigns the user-defined or global variable to the selected element of the specified form.</p> <ul style="list-style-type: none"> <li>• Select or insert a variable in the Form name field and use the Form element to select a specific element of the form.</li> <li>• Use the Input value to assign a user-defined or system variable into this element when an event is triggered.</li> </ul>

Action	Description
Set focus	<p>Sets the focus on the selected element of a form.</p> <ul style="list-style-type: none"> <li>• Select or insert a variable in the Form name field.</li> <li>• Use the Form element to select the specific element of the form and set the focus: <ul style="list-style-type: none"> <li>• For form elements such as a text box, text area, number or date, the cursor appears on the element.</li> <li>• For form elements such as a check box or radio button, the emphasis is on the first option.</li> </ul> </li> </ul>
Highlight	<p>Highlights the specified element of the selected form.</p> <ul style="list-style-type: none"> <li>• Select or insert a variable in the Form name field.</li> <li>• Use the Form element to select the specific element of the form to highlight it when an event is triggered.</li> </ul>
Unhighlight	Removes the highlight from the specified element of the selected form.

## IQ Bot package

The IQ Bot package enables you to upload and download documents from an IQ Bot server.

## Actions in the IQ Bot package

The IQ Bot package includes the following actions:

Action	Description
Download all documents	See <a href="#">Download all documents action</a> .
Upload document	See <a href="#">Upload document action</a> .

## Download all documents action

Use the Download all documents action to download the extracted results from an IQ Bot server that were created by running a Bot with the Upload Document action.

IQ Bot extracts fields from documents and exports them as CSV files. This action can also download any unclassified, untrained, and invalid documents to your local directory.

## Procedure

Follow these steps to download extracted results from the IQ Bot server:

1. In the Actions palette, double-click or drag the Download all documents action from the IQ Bot package.
2. In the Learning instance name field, select the name.
3. In the Local output folder field, provide a path to your local folder.
4. In the Document status, select the appropriate status for the documents.
  - Success: Documents have been processed and are in .CSV format.
  - Invalid: Documents were marked as invalid during the validation process.
  - Unclassified: Documents could not be classified.
  - Untrained: Documents were classified into new groups during processing and require training.
5. In the Delete files from the server after downloading check box, select the option to delete documents.
6. Optional: In the Save the response to a variable field, add a variable. For example, select prompt-assignment - string from the drop-down list.  
A variable value in this field provides information on whether the download was successful or failed, and the reason for the failure.
7. Click Update.
8. Click Save.
9. Click Run now.
10. Click Close.  
Note: If the download fails, verify the variable value using a Message Box or Log to File action. See the Save the response to a variable description.
11. Navigate to the local output folder to review the downloaded files.

## Upload document action

The Upload Document action enables you to upload a document with IQ Bot. IQ Bot extracts fields from the document and exports them to CSV files.

### Prerequisites

- Gain access to an Enterprise Control Room.
- Ensure your local host is a registered device in the Enterprise Control Room.

Use the Upload Document action to upload a single document to the Enterprise Control Room.

### Procedure

Follow these steps to upload a document:

1. In the Actions palette, double-click or drag the Upload Document action from the IQ Bot package.  
Note: A file size of 50 MB is supported for the upload action.
2. In the Learning instance name field, select the name.
3. In the File path field, specify the location or type of the file.
4. Optional: In the Save the response to variable field, add a variable. For example: select prompt-assignment - string from the drop-down list.  
A variable value in this field provides information about the file upload process: if the upload was successful or failed, and the reason for the failure.
5. Click Apply.
6. Click Save.
7. Click Run now.
8. Click Close.

## Next steps

To upload multiple files, see [Upload multiple files with IQ Bot using Loop action](#).

## JavaScript package

The JavaScript package contains actions to run a JavaScript from a bot.

## Before you start

1. Open a JavaScript file, or specify the script you want to run using the Open action. You must associate the details of the file or script you want to run with a session name. Use this same session name for other JavaScript actions.
2. Use the Run JavaScript action to run a function within a script or an entire script. You must use the same JavaScript session name established in the Open action.
3. Close the JavaScript session after running the script.

## Actions in the JavaScript package

The JavaScript package includes the following actions:

Note: The Open action must be the first action to use the JavaScript in a task. These actions can run a JavaScript on Windows, Linux, and UNIX based devices.

Action	Description
Close	<p>Closes the session.</p> <p>Specify the same session name from the Open action.</p>
Open	<p>Opens a JavaScript file.</p> <ul style="list-style-type: none"> <li>• In the JavaScript session field, specify a session name. Use this same session name for other JavaScript actions.</li> <li>• In the JavaScript, choose one of the following options: <ul style="list-style-type: none"> <li>• In the Import existing file option, select an existing JavaScript file. Note: If you are uploading a script from a file on your desktop, the file and any dependencies must be in a standalone folder. When you select a file for upload, all files and folders at the same folder level are uploaded.</li> <li>• In the Manual input option, enter the JavaScript.</li> </ul> </li> </ul>
Run JavaScript	<p>Runs a function within the JavaScript.</p> <ul style="list-style-type: none"> <li>• In the JavaScript session field, specify a session name. Use the same session name from the Open action.</li> <li>• Optional: Specify the function name to run and the arguments to pass to the function. Note: You can pass only a list variable as an argument for the function. You can use the list variable to pass multiple arguments of different data types such as Boolean, datetime, number, and string.</li> </ul>

Action	Description
	<ul style="list-style-type: none"> <li>Optional: In the Assign the output to variable field, specify the variable.</li> </ul>

## Resources

To learn more, see [Training - Write inline scripts in a bot](#). This course introduces you to writing inline scripts within a command.

Note: You must log in with a registered A-People Community [account](#) to access course.

### Simulate keystrokes package

Use the Simulate keystrokes package to simulate keystrokes in Chinese (simplified and traditional), English, French, German, Japanese, Korean, Italian, or Spanish characters.

### Action in the Simulate keystrokes package

The Simulate keystrokes package includes the following action:

Action	Description
Keystrokes	See <a href="#">Using Keystrokes action</a> .

Watch the following video on how to use Simulate keystrokes:

Using Simulate keystrokes

- [Using Keystrokes action](#)  
Use the Keystrokes action to simulate keystrokes.

Related tasks

[Build a basic bot that uses a desktop application](#)

## Using Keystrokes action

Use the Keystrokes action to simulate keystrokes.

### Procedure

Follow these steps to add a Keystrokes action:

1. In the Actions palette, double-click or drag the Keystrokes action from the Keystrokes package.
2. In the Select window field, choose the Window or Variable tab.
  - Click the Window tab to select the application window from the list.

Select the Currently active window, Desktop, or Taskbar, or a window from the list of Available windows.

- Click the Variable tab to insert an existing Window variable to specify the title of the application window you want to use.
- 3. In the Keystrokes, select a radio button:
  - Enter keystrokes here or use the on-screen keyboard: Type or use the keyboard to enter keystrokes.
  - Select a credential: Uses a password stored in the Credential Vault.
- 4. Optional: In the Delay between each keystroke in ms field, the delay time.  
Note: The default is set to 10 milliseconds.
- 5. Click Apply.
- 6. Click Save.

## List package

The List package contains actions that enable you to perform various operations on a variable of the list data type.

## Working with variables of list data type

A list is a collection of ordered values. The values can be of Boolean, number, or string data subtype. When initializing a list variable, you can select the Any data subtype in order to hold any of the three data subtypes. You can manually enter values by creating a new variable or selecting an existing one from the Variables menu, and then clicking Add.

Common uses of list variables include:

- Sending an email to multiple recipients.
- Searching multiple web addresses.

## Actions in the List package

If you are using a List action in a Loop action, you must use the For each item in the list iterator.

The following actions are available in the List package:

Action	Description
Add item	<p>Inserts an item into a list variable. You can choose to add the item at the end of the list or specify a position in the list.</p> <ul style="list-style-type: none"> <li>• Select the list variable in which to add an item from the List variable list.</li> <li>• Select the variable that contains the value to add from the Item to be added list. Note: The variable must be of the same sub-data type as the other list items.</li> <li>• Select the To end of list option to insert the item at the end of the list or select At list index to specify the index in the list where to insert the item.</li> </ul> <p>The list index starts from 0. The first item in the list is at position '0', the second item at '1', and so on. For example, to add an item at the fourth position in a list, enter <code>'3'</code> in the At list index field.</p>

Action	Description
Assign	<p>Assigns the value of the source list to the destination list variable. Note: The source and destination list variables must be of the same sub-data type.</p> <p>Select the source list variable from the drop-down list. Select a variable or create one to hold the output.</p>
Clear	Clears all items from the selected list variable.
Get item	<p>Retrieves a value at the specified position in a list and stores the output to a variable.</p> <ul style="list-style-type: none"> <li>• Select the list variable from which you want to retrieve a value from the List variable list.</li> <li>• Specify the position in the list from where you want to retrieve the value in the Index number field. The list index starts from 0. The first item in the list is at position '0', the second item at '1', and so on.</li> <li>• Select a variable that you want to use to store the output from the Assign the output to variable list.</li> </ul>
Join items	<p>Combines all the available values in a list variable and stores the output to a string variable. You can specify the delimiter you want to use as a separator in the output.</p> <ul style="list-style-type: none"> <li>• Select the list variable that you want to use from the List variable list.</li> <li>• Specify the delimiter you want to use to separate values in the output in the Delimiter field.</li> <li>• Select a variable that you want to use to store the output from the Assign the output to variable list.</li> </ul>
Remove item	<p>Removes an item from a list and assigns the output to a variable.</p> <ul style="list-style-type: none"> <li>• Select the list variable from which you want to remove an item from the List variable list.</li> <li>• Specify the position in the list from where you want to remove the value in the Index number field. The list index starts from 0. The first item in the list is at position '0', the second item at '1', and so on.</li> <li>• Select a variable that you want to use to store the output from the Assign the output to variable list.</li> </ul>
Set item	<p>Sets an item at the specific position in a list and stores the output in a variable.</p> <ul style="list-style-type: none"> <li>• Select the list variable in which you want to set a value from the List variable list.</li> <li>• Specify the position in the list where you want to set the item in the Index number field. The list index starts from 0. The first item in the list is at position '0', the second item at '1', and so on.</li> </ul>

Action	Description
	<ul style="list-style-type: none"> <li>Select a variable that you want to use to store the output from the Assign the output to variable list.</li> </ul>
Size	<p>Retrieves the number of items in a list and assigns the output to a number variable.</p> <ul style="list-style-type: none"> <li>Select the list variable for which you want to retrieve the size from the List variable list.</li> <li>Select a number variable that you want to use to store the output from the Assigns the number of items to variable list.</li> </ul>

Related reference

[Loop package](#)

[Variables overview](#)

## Log To File package

Use the Log To File package to create a log file with data.

The Log To File package enables the following:

- Verify that a bot ran properly.
- Create a new log file.
- Specify custom text to be included in the log file.
- Add a time stamp to the log file.
- Use a log file as a variable.

Note: The package supports ANSI, Unicode, and UTF8. It can save files as .csv and .txt.

## Actions in the Log To File package

The Log To File package includes the following action:

Action	Description
Log to file	See <a href="#">Using Log To File action</a> .

- [Using Log To File action](#)

Using the Log To File action, you can create a log file with data about the events that occur when a TaskBot runs.

## Using Log To File action

Using the Log To File action, you can create a log file with data about the events that occur when a TaskBot runs.



## Procedure

Follow these steps to log text into a file:

1. In the Actions palette, double-click or drag the Log To File action from the Log To File package.
2. In the File path field, specify the file location or variable.
3. In the Enter text to log field, enter the text to log in the file.
4. Select the Append timestamp check box to add a time stamp.
5. In the When logging field, choose Append to existing file to append the log file or Overwrite existing log file to overwrite the content in the log file.
6. In the Encoding field, specify the value.
7. Click Apply.
8. Click Save.

## Loop package

Use the Loop package to run a sequence of actions repeatedly for a specific number of times or until a specific condition is met.

The Loop package enables you to repeatedly run specific actions within a bot. For example, repeat the set of actions that read data from each row of a Microsoft Excel file, rename all files in a folder, and save each email in a mailbox. You can also use the If action within the Loop action to validate a condition, and based on the outcome of it, skip the current iteration in the loop or even break the loop.

For conditional loops, different actions are taken depending on whether the conditional parameters are met. For loops that have a specified number of iterations, the loop exits on the last iteration and goes to the next step in the bot.

## Actions in the Loop package

The following Loop actions are available:

Action	Description
Loop	Repeats the action for a specific number of times based on the option selected from the Iterator list, or until the condition is met based on the option selected from the Condition list. See <a href="#">Using Loop action</a> .
Continue	Use the Continue action along with the If action to skip the current iteration and continue with the next iteration in the loop based on the condition you have specified in the If action. See <a href="#">If package</a> .
Break	You can use the Break action along with the If action to terminate the loop based on the condition you have specified in the If action. When you terminate the loop, the actions immediately following the Loop action run. See <a href="#">If package</a> .

## Iterator related conditions within the Loop action

Select the Iterator option to specify the number of times the set of actions will be repeated as part of the Loop action:

Iterator	Description
For each row in CSV/TXT	Repeats the set of actions for each row in the specified CSV or text file and assigns the values in the current row to a Record variable. See <a href="#">Using Read action in loop</a> .
For each row in Data Table	Repeats the set of actions for each row in the specified table and assigns the values in the current row to a Record variable.
For each row in a SQL query dataset	Repeats the set of actions for each row in the specified SQL query dataset and assigns the values in the current row to a Record variable. Provide the session name that you have used to establish a connection with the database.
For each key in the Dictionary	Repeats the set of actions for each key in the specified Dictionary variable and assigns the name of the current key to a variable.
For each value in the Dictionary	Repeats the set of actions for each value in the specified Dictionary variable and assigns the current value to a variable. Because this variable type can hold various subtypes (String, Number, Boolean, and so on), if you quick-create a variable when configuring the action, the variable is of Any type..
For each mail in mailbox	Repeats the set of actions for each email in the specified mailbox. See <a href="#">Using Email action in loop</a> .
For each row in worksheet (Excel basic Basic and Advanced)	Repeats the set of actions for each row that contains data in a worksheet and assigns the values in the current row to a Record variable. Provide the session name that you have used to open the Excel basic worksheet. Specify whether to repeat the actions for all the rows, specified rows, or a specified cell range.
For each file in folder	Repeats the set of actions for each file in the specified folder and assigns the properties of the current file to a Dictionary variable containing two keys: the <code>name</code> key holds the file name and the <code>extension</code> key holds the file extension.
For each folder in folder	Repeats the set of actions for each folder in the specified folder and assigns the current folder name to a String variable.
For each item in the list	Repeats the set of actions for each item in the specified list and assigns the current item to a variable. Specify whether to repeat the action for all items in the list or only for the range of items in the list. Because this variable type can hold various subtypes (String, Number, Boolean, and so on), if you quick-create a variable when configuring the action, the variable is of Any type. Note: The index in a list starts from zero. For example, to repeat the action for items from the third position to the sixth position of the list, specify '2' and '5' in the appropriate fields.
For n times	Repeats the actions in the container the specified number of times. You can assign the iteration count to a Number variable.
For each value in record	Repeats the actions for each value in the specified record and assigns the current value to a variable. Because this variable type can hold various subtypes (String, Number, Boolean, and so on), if you quick-create a variable when configuring the action, the variable is of Any type..
For each meeting in calendar	Use this option to repeat the set of actions for each meeting in the specified calendar. See <a href="#">Using Office 365 Calendar actions in a loop</a> .
For each row in worksheet (Office 365 Excel)	Repeats the set of actions for each row that contains data in a worksheet. <ul style="list-style-type: none"> <li>Provide the session name that you have used to open the worksheet.</li> </ul>

Iterator	Description
	<ul style="list-style-type: none"> <li>Specify whether to repeat the actions for all the rows or specific rows.</li> <li>You can assign the values in the current row to a Record variable.</li> </ul>
For each node in a XML dataset	Repeats the set of actions for each node in an XML dataset and assigns the current node to String variable. Provide the session name that you have used to open the XML file.

## While related conditions in the Loop action

You can configure multiple while related conditions within a single Loop action:

- Click Add condition.
- Select either of the following options:
  - And: Both of the conditions must be met for the actions to run.
  - Or: Either of the conditions must be met for the actions to run.
- Select the conditions from the drop-down list.

Select the While option to use the following conditions:

While	Description
Application is or is not running	Repeat the set of actions based on whether an application is running or not. Enter the application path or specify the path using a variable along with the number of seconds to wait for this condition to be true.
Boolean variable	Use this condition to repeat the set of actions based on the value of a Boolean variable. Use this condition to compare the values of two Boolean variables by entering or selecting the second variable from the Select an existing Boolean field.
Datetime	Use the Datetime variable condition to execute actions based on whether the value of the source datetime variable is Equal to or Not Equal to, is Greater than or Equal to, or is Lesser than or Equal to the value of the target datetime variable.
Dictionary	Use this condition to repeat the set of actions based on whether the selected dictionary variable contains the specified key.
File	<p>Use the following conditions to execute the action:</p> <ul style="list-style-type: none"> <li>File date           <p>Use this condition to verify the date and time that the specified file was created or modified. Specify a date range using the Is within last, Is between, or Is before options. For the Is within last option, specify the number of days or time (in hours, minutes, and seconds). Enter the amount of time (in seconds) to wait for this condition to be true.</p> </li> <li>File exists and File does not exist           <p>Use these conditions to execute an action based on whether a file exists. For example, if a data file exists, format the file and upload it to a database.</p> </li> <li>File size</li> </ul>

While	Description
	<p>This condition verifies if the specified file is larger, smaller, not the same, or the same as the size you specify.</p> <p>Enter the amount of time (in seconds) to wait for this condition to be true.</p>
Folder does or does not exists	Use these conditions to repeat the set of actions based on whether a folder exists.
Image Recognition	<p>Use these conditions to verify whether:</p> <ul style="list-style-type: none"> <li>• Image file is found in the Image file or not.</li> <li>• Image file is found in the Window or not.</li> <li>• Window is found in the Image file or not.</li> <li>• Window is found in the Window or not.</li> </ul>
JavaScript	Use the Script is successful or Script is unsuccessful condition to execute actions based the status of the specified JavaScript. Select the file that contains the script and optionally specify the parameters by selecting a list variable.
Legacy automation	<p>Use the Legacy automation conditions to execute actions on bots migrated from Version 11.3. Use the following conditions to verify the following:</p> <ul style="list-style-type: none"> <li>• Whether Web control exists or not.</li> <li>• Whether Web control is active or not.</li> <li>• Whether Window control is active or not.</li> </ul>
List	Use this condition to repeat the set of actions based on whether the specified List variable contains a particular value. The value can be of Number, String, or Boolean.
Number	Use this condition to repeat the set of actions based on whether the specified Number variable is equal to/not equal to/is greater than or equal to/is lesser than or equal to a particular value.
Ping is successful/Ping is unsuccessful	Use these conditions to verify if a machine or server is running, and repeat the set of actions based on the result. Enter the number of seconds to wait for this condition to be true.
Recorder	Use this condition to detect an Object in a window. Select a window or variable to capture the object. Enter the amount of time (in seconds) to wait for this condition to be true.
String	Use this condition to repeat the set of actions based on whether the specified String variable Equals to/Not equal to/Includes/Does not include a particular value.
Task Bot	Use the Task successful or Task unsuccessful condition to execute actions based on the status of the specified Task Bot.
VBScript	Use the Script is successful or Script is unsuccessful condition to execute actions based on the status of the specified Visual Basic script. Select the file that contains the script and optionally specify the parameters by selecting a variable.

While	Description
Window exists/Window does not exist	Use these conditions to verify if a specific Windows application is open by entering the window title or using a Window variable. Enter the number of seconds to wait for this condition to be true.

- [Using Loop action](#)

Use the Loop action to repeatedly run a sequence of actions for a specific number of times or until a condition is met.

## Using Loop action

Use the Loop action to repeatedly run a sequence of actions for a specific number of times or until a condition is met.

### Procedure

To repeatedly run a sequence of actions, do the following:

1. Double-click or drag the Loop action from the Actions palette.
2. Set the desired conditions for the loop.

Option	Action
Select the Iterator option to repeat the sequence of actions for a specific number of times.	<p>Select an option from the Iterator list to specify how many times the sequence of actions is to be repeated.</p> <p>See <a href="#">Loop package</a> for the list of available options.</p>
Select the While option to repeat the sequence of actions until a condition is met.	<p>a) Select an option from the Condition list to specify the condition.</p> <p>b) Select the Check the condition at the end of the iteration to validate the condition at the end of each iteration. This option enables you to ensure that the set of actions run at least one time, even if the condition is not met.</p>

3. Double-click or drag the actions to be repeated within the Loop.
4. Optional: Double-click or drag the Continue action from the Loop package to skip the current iteration and continue with the next iteration of the loop.
5. Click Apply.

Watch the following video on how to use the Loop action:

Using Loop action

## Message box package

Use the Message box action from the Message box package to insert a message box that shows a message when the task runs. For example, you can insert a Message Box action to follow a web form so that the action displays the message: `Web Form Filled and Complete`.

For more information, see [Using the Message box action](#).

Use the following examples to learn how to use the Message box action when building automations.

- [Build a Go be Great bot](#)

Build a basic TaskBot using a Message Box action and a variable.

- [Build a basic bot that uses a desktop application](#)

An example of how to build a basic TaskBot that uses the calculator application.

## Using the Message box action

Use the Message box action to insert a message box that shows a message when the task runs.

### Procedure

1. Double-click or drag the Message box action from the Message box package in the Actions palette.
2. Enter a window title for the message box.
3. Enter the message to display.  
Note: The single dollar sign (\$) is reserved for the Automation Anywhere back-end system. If you want to display a dollar sign (\$) in your message, you must enter two dollar signs (\$\$). For example, if you want to display the message, `Pay $5.00`, that message must say `Pay $$5.00` in A2019 for it to display properly to users.
4. Specify the number of lines at which to show a scroll bar.
5. Optional: Select the Close message box after option and specify the number of seconds after which the message box closes automatically.  
Important: Select this check box if the bot will run on an unattended machine.
6. Click Apply.

## Microsoft LUIS NLP package

The Microsoft LUIS NLP package contains actions that enable you to connect to and consume the Microsoft Cognitive Services Text Analytics API to identify the language, sentiment, key phrases, and entities. This package supports the following languages: English, Chinese (Simplified), French, German, and Spanish.

### Before you start

You must have an Azure Cognitive Services resource to use these actions. See [Create a Cognitive Services resource using the Azure portal](#).

You also require the following in order to use the actions:

- Service Endpoint URL: the endpoint URL that identifies the Azure service.
- Subscription Key: the unique key that authenticates Automation Anywhere Enterprise.

## Actions in the Microsoft LUIS NLP package

Action	Description
Detect language	Identifies the language of the provided content and returns it in ISO 639-1 language code. The output is stored in a string variable.
Get key phrases	Identifies the main points and returns a list of key phrases. For example, if the input text is about a basketball game, this action returns the names of teams, the name of the venue, and the final score.
Get named entities	Identifies the entities in the provided content such as people, places, organizations, date/time, quantities, branded products, and book titles.  The output is stored in a dictionary variable, where each name is a key, and the corresponding entity is the value.
Get sentiment	Analyzes the provided content and returns a sentiment and score. <ul style="list-style-type: none"> <li>• If the score is greater than 0.6, the sentiment is Positive.</li> <li>• If the score is in the 0.2 through 0.6 range, the sentiment is Neutral.</li> <li>• If the score is less than 0.2, the sentiment is Negative.</li> </ul> The output is stored in a dictionary variable containing two keys and their corresponding values: <code>sentiment</code> and <code>score</code> .

## Mouse package

Use the Mouse package to simulate mouse actions.

## Actions in the Mouse package

The Mouse package includes the following actions:

Action	Description
Click	See <a href="#">Using the Click action</a> .
Move	See <a href="#">Using the Move action</a> .
Scroll	See <a href="#">Using the Scroll action</a> .

## Secure Recording

When secure recording mode is enabled, bots do not capture object images or values. This ensures that sensitive data is not stored in the bots.

When you record a task in secure recording mode, the Preview window temporarily shows the captured area. This image is deleted after you click Apply and navigate away from the action editor window.

A user with admin privileges must enable this setting. See [Settings](#).

## Using the Click action

Use the Click action to enable a bot to simulate mouse clicks. This action enables to capture the UI element such as the screen or window, and to specify the click event and click button.

### Procedure

Follow these steps to add a Click action:

1. In the Actions palette, double-click or drag the Click action from the Mouse package.
2. Select either the Screen, Window, or Variable tab to specify the window in which to capture the mouse click:
  - Click the Screen tab to capture coordinates on the device screen.
  - Click the Window tab to capture coordinates in the application window you select from the list.

Select the Currently active window, Desktop, or Taskbar, or a window from the list of Available windows.

Note: It is recommended to use the Window option because screen sizes can change, which distorts the captured coordinates.

- Click the Variable tab to insert an existing Window variable..
3. Click Capture coordinate.  
The selected window appears.
  4. Drag the mouse to select the click spot and left-click to capture it.  
The captured spot appears in the Preview section with the coordinates underneath.
  5. In the Button option, specify the button to click.  
Choose from Left Button, Right Button, or Middle Button.
  6. In the Event option, specify the event.  
Choose from Click, Double click, Button up, or Button down.
  7. Click Apply.
  8. Click Save.

## Using the Move action

Use the Move action to simulate moving the mouse pointer from one location to another.



## Procedure

Follow these steps to add a Move action:

1. In the Actions palette, double-click or drag the Move action from the Mouse package.
2. In the Mouse from and Mouse to fields, specify the X and Y coordinates.
3. Click Capture coordinate to capture the starting point location.
4. Click Capture coordinate to capture the end point location.
5. Optional: In the Delay in milliseconds field, specify the total duration of the movement in milliseconds.
6. Click Apply.
7. Click Save.

## Using the Scroll action

Use the Scroll action to simulate scrolling the mouse wheel up or down.

## Procedure

Follow these steps to add a Scroll action:

1. In the Actions palette, double-click or drag the Scroll action from the Mouse package.
2. In the Select scroll option, select the Up or Down option.
3. In the Number of iterations field, specify the number of times to scroll.
4. Optional: In the Delay in milliseconds field, specify the total duration of the scrolling in milliseconds.
5. Click Apply.
6. Click Save.

## Number package

The Number package contains actions that enable you to perform various operations on a number variable. A number variable holds numeric values, including integers and decimals. It holds values from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807, and up to 15 decimal digits.

## Actions in the Number package

The actions in the Number package accept a variable as an input and assign the output to a variable. These actions enable you to assign a value to a Number variable, decrement or increment a number, or convert a Number variable to a String variable.

The Number package includes the following actions:

Action	Description
Assign	Assigns a specified number or result of an expression to a user-defined Number variable. You can use expressions built using the +, -, *, and / operators and use parentheses to group expressions. For example, (3*4)+5 or (\$Variable2*\$Variable3)/\$Variable1.

Action	Description
	<ul style="list-style-type: none"> <li>Enter a number or select a predefined Number variable.</li> <li>Select a predefined number variable or create a new one to hold the output.</li> </ul>
Decrement	<p>Decrements (decreases by set intervals) a number by a user-specified value.</p> <ul style="list-style-type: none"> <li>Enter a number or select a predefined Number variable.</li> <li>Enter the decrement value or select a predefined Number variable.</li> <li>Select a predefined number variable or create a new one to hold the output.</li> </ul>
Increment	<p>Increments (increases by set intervals) a number by a user-specified value.</p> <ul style="list-style-type: none"> <li>Enter a number or select a predefined Number variable.</li> <li>Enter the increment value or select a predefined Number variable.</li> <li>Select a predefined number variable or create a new one to hold the output.</li> </ul>
Random	<p>Generates a random integer from a user-specified range and assigns it to a number variable.</p> <ul style="list-style-type: none"> <li>In the Beginning of range field, enter a number or select a predefined number variable. Note: This field accepts values in the range of -9,223,372,036,854,775,808 through 9,223,372,036,854,775,807, and up to 15 decimal digits.</li> <li>In the End of range field, enter a number or select a predefined number variable that is greater than the value in the Beginning of range field. Note: This field accepts values in the range of -9,223,372,036,854,775,808 through 9,223,372,036,854,775,807, and up to 15 decimal digits.</li> <li>Select a predefined number variable or create a new one to hold the output.</li> </ul>
To string	<p>Converts a user-specified number to a string.</p> <ul style="list-style-type: none"> <li>Enter a number or select a predefined Number variable.</li> <li>Enter the amount of digits after the decimal.</li> </ul> <p>To remove all the digits after the decimal, leave the default value of 0.</p> <ul style="list-style-type: none"> <li>Assign the output to a String variable. You can use prompt-assignment during building and testing of the bot. Important: Ensure that you reassign the values from prompt-assignment to a user-created variable before deploying the bot into production.</li> </ul>

## OCR package

The OCR package contains actions that enable you to extract text from images or applications.

The OCR package enables you to:

- Extract text from a window or a specific area of an application.
- Extract text from images or files stored on a local machine, a website, or the Enterprise Control Room folder.
- Filter extracted text and store it as a variable.

Note: You can extract text from images in .jpeg, .jpg, .bmp, .gif, and .png formats.

ABBYY FineReader 12 is installed along with A2019 and does not require any additional setup.

## Actions in the OCR package

The OCR package includes the following actions:

Action	Description
Capture image by path	See <a href="#">Using Capture image by path action</a> .
Capture image by url	See <a href="#">Using Capture image by URL action</a> .
Capture window	See <a href="#">Using Capture window action</a> .
Capture area	See <a href="#">Using Capture area action</a> .

## Secure Recording

When secure recording mode is enabled, bots do not capture object images or values. This ensures that sensitive data is not stored in the bots.

When you record a task in secure recording mode, the Preview window temporarily shows the captured area. This image is deleted after you click Apply and navigate away from the action editor window.

A user with admin privileges must enable this setting. See [Settings](#).

## Using Capture image by path action

Use the Capture image by path action to extract text from an image on a device or a folder in the Enterprise Control Room. The extracted text can assigned as a variable.

### Procedure

Follow these steps to use the Capture image by path action:

1. In the Actions palette, double-click or drag the Capture image by path action from the OCR package.
2. In the Image path field, select an option:
  - Control Room file: Uses a file that is available on the Enterprise Control Room.
  - Desktop file: Uses a file that is available on a device.
  - Variable: Uses a file variable to specify the file location.
3. Select the Filter the captured text check box to filter the captured text.  
Specify the text to filter out in the Before and After fields. For example, if the extracted text is Name : ABC Inc. Location:, to retrieve ABC Inc., you must specify Location: in the Before field and Name : in the After field.

4. Select the Trim the captured text check box to trim the extra spaces.
5. Select the Load Profile check box and select an option to load the locale based on the ABBYY profile:
  - Control Room file: Uses a file that is available on the Enterprise Control Room.
  - Desktop file: Uses a file that is available on a device.
  - Variable: Uses a file variable to specify the file location.
6. In the Select locale list, select the language of the device.  
 Note: Currently, only English and Japanese locales are supported. If you have selected an option from the Select locale list and the Load Profile option, the system considers the locale based on the Load Profile selection.
7. In the Assign value to variable list, select a string variable.
8. Click Apply.
9. Click Save.

## Using Capture image by URL action

Use the Capture image by url action to extract text from an online image. You can filter the extracted text and assign it to a string variable.

### Procedure

Follow these steps to use the Capture image by url action:

1. In the Actions palette, double-click or drag the Capture image by url action from the OCR package.
2. In the Image url field, specify the URL.  
 Note: A URL of an image on a shared drive or an FTP server is not supported.
3. Select the Filter the captured text check box to filter the captured text.  
 Specify the text to filter out in the Before and After fields. For example, if the extracted text is `Name : ABC Inc. Location:`, to retrieve `ABC Inc.`, you must specify `Location:` in the Before field and `Name :` in the After field.
4. Select the Trim the captured text check box to trim the extra spaces.
5. Select the Load Profile check box and select an option to load the locale based on the ABBYY profile:
  - Control Room file: Uses a file that is available on the Enterprise Control Room.
  - Desktop file: Uses a file that is available on a device.
  - Variable: Uses a file variable to specify the file location.
6. In the Select locale list, select the language of the device.  
 Note: Currently, only English and Japanese locales are supported. If you have selected an option from the Select locale list and the Load Profile option, the system considers the locale based on the Load Profile selection.
7. In the Assign value to variable list, select a string variable.
8. Click Apply.
9. Click Save.

## Using Capture window action

Use the Capture window action to extract text from an application window. You can filter the extracted text and assign it to a string variable.

## Procedure

Follow these steps to use the Capture window action:

1. In the Actions palette, double-click or drag the Capture window action from the OCR package.
2. In the Window title field, select an option:
  - Window: Captures a window on a desktop. In the Window title field, specify the application title.
  - Variable: Inserts an existing window variable. Click Variable tab to create a new string variable.
3. In the Wait before capturing the image (ms) field, specify the delay time.
4. Select the Filter the captured text check box to filter the captured text.  
Specify the text to filter out in the Before and After fields. For example, if the extracted text is `Name : ABC Inc. Location :`, to retrieve `ABC Inc.`, you must specify `Location :` in the Before field and `Name :` in the After field.
5. Select the Trim the captured text check box to trim the extra spaces.
6. Select the Load Profile check box and select an option to load the locale based on the ABBYY profile:
  - Control Room file: Uses a file that is available on the Enterprise Control Room.
  - Desktop file: Uses a file that is available on a device.
  - Variable: Uses a file variable to specify the file location.
7. In the Select locale list, select the language of the device.  
Note: Currently, only English and Japanese locales are supported. If you have selected an option from the Select locale list and the Load Profile option, the system considers the locale based on the Load Profile selection.
8. In the Assign value to variable list, select a string variable.
9. Click Apply.
10. Click Save.

## Using Capture area action

Use the Capture area action to extract text from a specific area in an application window. You can filter the extracted text and assign it as a variable.

## Procedure

Follow these steps to use the Capture area action:

1. In the Actions palette, double-click or drag the Capture area action from the OCR package.
2. In the Window title field, select an option:
  - Window: Captures a window on a desktop. In the Window title field, specify the application title.
  - Variable: Inserts an existing window variable. Click Variable tab to create a new string variable.
3. Specify the X, Y, Width, and Height coordinates.
4. Click Capture region.
5. In the Wait before capturing the image (ms) field, specify the delay time.
6. Select the Filter the captured text check box to filter the captured text.  
Specify the text to filter out in the Before and After fields. For example, if the extracted text is `Name : ABC Inc. Location :`, to retrieve `ABC Inc.`, you must specify `Location :` in the Before field and `Name :` in the After field.
7. Select the Trim the captured text check box to trim the extra spaces.
8. Select the Load Profile check box and select an option to load the locale based on the ABBYY profile:
  - Control Room file: Uses a file that is available on the Enterprise Control Room.
  - Desktop file: Uses a file that is available on a device.

- Variable: Uses a file variable to specify the file location.
9. In the Select locale list, select the language of the device.  
Note: Currently, only English and Japanese locales are supported. If you have selected an option from the Select locale list and the Load Profile option, the system considers the locale based on the Load Profile selection.
  10. In the Assign value to variable list, select a string variable.
  11. Click Apply.
  12. Click Save.

## Office 365 Calendar package

The Office 365 Calendar package contains actions that enable you to automate meeting-related tasks in Office 365 Calendar.

### Before you start

Perform the following actions within the Office 365 Calendar package as part of using the set of available actions:

1. Use the Connect action to establish a connection to the Office 365 server.  
See [Using the Connect action](#).  
Note: Office 365 packages do not currently support Multi-Factor Authentication or Single Sign-on.
2. Use a combination of actions available in this package to automate tasks.  
Note: To use actions from other Office 365 packages, establish a connection using the Connect action from that package.
3. Use the Disconnect action to terminate the connection.

## Actions in the Office 365 Calendar package

The Office 365 Calendar package includes the following actions:

Note: Some of the actions must be used within a Loop action to apply the action to each meeting in the calendar. See [Using Office 365 Calendar actions in a loop](#).

Action	Description
Add attachment	<p>Adds one or more attachments to a meeting. Use this action within a Loop action to repeat this operation for each meeting in the calendar. See <a href="#">Loop package</a>.</p> <ul style="list-style-type: none"> <li>• Provide the session name that you used in the Connect action.</li> <li>• Select the files to attach from the Enterprise Control Room or your desktop, or insert a file variable. Separate each file path with a comma.</li> </ul>
Add attendees	<p>Adds one or more attendees to a meeting and specifies whether attendance is optional or required. During run time, this action triggers an email notification to meeting attendees. Use this action within a Loop action to repeat this operation for each meeting in the calendar. See <a href="#">Loop package</a>.</p> <ul style="list-style-type: none"> <li>• Provide the session name that you used in the Connect action.</li> <li>• Enter the attendee emails into the Required or Optional fields. Separate each email with a comma.</li> </ul>

Action	Description
Cancel meeting	Cancels the meeting. During run time, this action triggers an email notification to meeting attendees. Use this action within a Loop action to repeat this operation for each meeting in the calendar. See <a href="#">Loop package</a> .
Connect	Establishes a connection with the Office 365 server using your organization's client ID and tenant ID, and user credentials.  See <a href="#">Using the Connect action</a> .
Create meeting	See <a href="#">Using the Create Meeting action</a> .
Delete attendees	Deletes attendees from a meeting. During run time, this action triggers an email notification to meeting attendees. Use this action within a Loop action to repeat this operation for each meeting in the calendar. See <a href="#">Loop package</a> . <ul style="list-style-type: none"> <li>• Provide the session name that you used in the Connect action.</li> <li>• Select an option to delete: meeting or attendees. To delete attendees, provide the email addresses of the attendees, separated by commas.</li> </ul>
Delete meeting information	Deletes information related to a meeting such as the title or location. Use this action within a Loop action to repeat this operation for each meeting in the calendar. See <a href="#">Loop package</a> . <ul style="list-style-type: none"> <li>• Provide the session name that you used in the Connect action.</li> <li>• Select the meeting information to delete. The options are: <ul style="list-style-type: none"> <li>• Title</li> <li>• Location</li> <li>• Agenda</li> <li>• Reminder</li> <li>• Recurrence</li> <li>• Is all day</li> </ul> </li> </ul>
Disconnect	Terminates the connection with the Office 365 server. Enter the session name that you used in the Connect action. Insert this action at the end of automation sequences that use actions from this Office 365 package.
Get available meeting slot(s)	See <a href="#">Using the Get available meeting slots action</a> .
Modify meeting	Modifies information for a meeting. Use this action within a Loop action to repeat this operation for each meeting in the calendar. See <a href="#">Loop package</a> . <ul style="list-style-type: none"> <li>• Provide the session name that you used in the Connect action.</li> <li>• Modify any of the following: <ul style="list-style-type: none"> <li>• Meeting start or end date.</li> <li>• Convert the meeting to all day.</li> <li>• Revise the meeting title.</li> <li>• Revise the agenda.</li> <li>• Update the location.</li> </ul> </li> </ul>

Action	Description
	<ul style="list-style-type: none"> <li>• Update the reminder.</li> <li>• Update the recurrence, frequency, start date, and end date.</li> </ul>
Respond to meeting	Responds to a meeting as accepted, rejected, or tentative. You can also add a message to the response. Use this action within a Loop action to repeat this operation for each meeting in the calendar. See <a href="#">Loop package</a> .

## Using the Create Meeting action

Use the Create Meeting action to specify a meeting agenda, attachments, attendees, duration, location, recurrence, and title. During run time, this action sends an invitation email to meeting attendees.

### Prerequisites

This action uses two Datetime variables to set the meeting start and end. Define the values before you start. The recurrence option also uses two Datetime variables to set the start and end. For more information, see [User-defined variables](#).

To create a meeting, do the following:

### Procedure

1. Enter the session name that you used in the Connect action.
2. Enter the calendar name.  
This field is case-sensitive.  
The prepopulated value is Calendar; this is the default Office 365 setting.
3. Specify the meeting start date and time.  
Select a DateTime variable from the drop-down list.
4. Specify the meeting end date and time.  
Select a DateTime variable with a value that is the same or after the start time.
5. Optional: Mark the All day meeting option.  
For an all day meeting, ensure that there is a one-day difference between the start and end dates.
6. Enter attendee email addresses, separated by commas.
7. Optional: Enter the following details:
  - Attach a file from the Enterprise Control Room, Desktop, or insert a File variable.
  - Enter a meeting title.
  - Enter a meeting location.
  - Enter an agenda.
  - Set a reminder.
  - Set a recurrence with the following options:
    - Select the recurrence type: Daily, Weekly, Monthly, or Yearly.
      - For Weekly recurrence, mark the days of the week on which the meeting takes place.
      - For Monthly recurrence, either enter the numerical date or select a day.  
For example, to schedule a meeting that takes place on the twenty-fifth day of the month enter  
25  
in the Specific date field.



Note: If you enter

31

in the Specific date field, for the months with less than 31 days, the meeting will schedule for the last day of the month.

To schedule it for the last Monday of the month, select Last from the Occurrence drop-down list and Monday from the Weekday drop-down list.

- For Yearly recurrence, either enter the numerical date and month or select a day and month.

For example, to schedule a meeting that takes place on March 15, enter

15

in the Specific Date field and

March

in the Month field. To schedule it for the second Friday in March, select Second from the Occurrence drop-down list, select Friday from the Weekday drop-down list, and March from the Month drop-down list.

- Enter the recurrence frequency.
- Specify the recurrence start by selecting a DateTime variable from the drop-down list.
- If the recurrence has an end date, select a DateTime variable from the drop-down list.

## Using the Get available meeting slots action

Use the Get available meeting slots action to retrieve available time slots for attendees in a specified date and time range. This action supports scheduling across time zones.

This action considers a time slot as available if it is within an attendee's working hours and if the attendee has not already accepted a meeting for that time. If an attendee tentatively responds or does not respond to an invitation, this action considers them available.

### Procedure

To retrieve available time slots, perform the following steps:

1. Enter the session name that you used in the Connect action.
2. Enter the email addresses of the attendees, separated by commas.
3. Enter the duration of the meeting in minutes.
4. Specify the range start date and time by selecting a DateTime variable from the drop-down list.
5. Specify the range end date and time by select a DateTime variable with a value that is the same or after the start time.
6. Optional: Select the Check for timeslots outside working hours option.
7. Select a Table variable from the drop-down list.
8. Click Apply.

## Using Office 365 Calendar actions in a loop

You must use certain Office 365 Calendar actions within a Loop action.

## Prerequisites

This action uses two Datetime variables to set the meeting start and end. Define the values before you start. For more information, see [User-defined variables](#).

## Procedure

To use an Office 365 Calendar action within a loop, follow these steps:

1. Double-click or drag the Loop action from the Loop package in the Actions palette.
2. Select the For each meeting in calendar iterator.
3. Enter the session name that you used in the Connect action.
4. Enter the calendar name.  
This field is case-sensitive.  
The prepopulated value is Calendar; this is the default Office 365 setting.
5. Select whether to loop through All meetings in the calendar or only meetings with the specific title.
6. Specify the meeting start date and time.  
Select a DateTime variable from the drop-down list.
7. Specify the meeting end date and time.  
Select a DateTime variable with a value that is the same or after the start time.
8. Optional: Enter the location.
9. Optional: Enter the duration in minutes.
10. Optional: Mark the All day meeting option.  
For an all day meeting, ensure that there is a one-day difference between the start and end dates.
11. Optional: Enter the meeting owner's email address.
12. Select or create a record variable to hold the output.

## Using the Connect action

Use the Connect action to establish a connection with the Office 365 server using your organization's (client ID and tenant ID) and user credentials. All of the fields in this action accept a credential from the Credential Vault or a user-input value.

## Prerequisites

Note:

- The Connect action does not currently support Multi-Factor Authentication or Single Sign-on.
  - The best practice is for a system administrator to perform the following steps.
1. Log in to the [Azure portal](#) using your Office 365 credentials.
  2. Register your application with the Microsoft identity platform to obtain the client ID and tenant ID. See [Register your app](#).
  3. Grant the required permissions for Microsoft Graph to the application. See [Permissions for application](#).
  4. To add a client secret, you must first subscribe to Azure. See [Subscribe to Azure](#).
  5. Add a client secret. See [Add a client secret](#).

## Procedure

To establish a connection to the Office 365 server, follow these steps:

1. In the Actions palette, double-click or drag the Connect action from the package you want to use.
2. In the User session field, enter a name for this session.  
Provide this session name in subsequent actions from this package.
3. In the Username and Password fields, enter your user credentials or insert them from the Credential Vault.
4. In the Tenant ID field, enter the unique ID for your Office 365 subscription or insert it from the Credential Vault.
5. In the Client ID field, enter the Office 365 client or insert it from the Credential Vault.
6. In the Client Secret Key field enter your access token or insert it from the Credential Vault.
7. Click Apply.

## Permissions for application

After registering the Office 365, you must grant certain permissions to the application. These permissions are required to enable Office 365 packages to perform various operations.

To grant permissions to an application, see [Add permissions](#).

The following [Delegated permissions](#) for Microsoft Graph are required for Office 365 packages:

Common permissions	
	<a href="#">openid</a>
	<a href="#">offline_access</a>
	<a href="#">User.ReadWrite.All</a>
	<a href="#">Group.Read.All</a>
	<a href="#">Group.ReadWrite.All</a>
Excel	
	<a href="#">Files.ReadWrite.All</a>
	<a href="#">Sites.ReadWrite.All</a>
One Drive	
	<a href="#">Files.ReadWrite.All</a>
	<a href="#">Sites.ReadWrite.All</a>
Calendar	
	<a href="#">Calendars.Read</a>
	<a href="#">Calendars.Read.Shared</a>
	<a href="#">Calendars.ReadWrite</a>
	<a href="#">Calendars.ReadWrite.Shared</a>

Apart from the above permissions, you can grant additional permissions based on your requirements. See, [Microsoft Graph permission reference](#).

## Office 365 Excel package

The Office 365 Excel package contains actions that enable you to automate tasks in the online version of Microsoft Excel.

## Choosing the Excel package in Enterprise A2019

Enterprise A2019 includes packages to support three types of Microsoft Excel usage. For optimal results, use the package that corresponds to the type of Excel that is available on the device you are running bots on.

- No Excel installed: If you do not have Microsoft Excel installed on the device on which you are running bots to automate Excel-related processes, use the Excel basic package.
- Desktop Excel installed: If you have a desktop version of Microsoft Excel installed on your computer, use the Excel advanced package in your bots.
- Online Office 365 Excel only: If you are using Microsoft Excel 365 on a web browser, use the Office 365 Excel package for automating tasks related to Excel.

## Before you start

1. Use the Connect action to establish a connection to the Office 365 server.  
See [Using the Connect action](#).  
Note: Office 365 packages do not currently support Multi-Factor Authentication or Single Sign-on.
2. Use the Open action to select a workbook, or the Create action to create a new workbook. See [Workbook operations](#).
3. Optional: If the workbook contains more than one worksheet, use the Activate sheet action to specify which worksheet to use.
4. Use a combination of actions available in this package to automate tasks.  
Note: To use actions from other Office 365 packages, establish a connection using the Connect action from that package.
5. Use the Close action to exit from the workbook.
6. Use the Disconnect action to terminate the connection.

## Actions in the Office 365 Excel package

The actions in the Office 365 Excel package enable you to perform the following operations:

Operations	Description
Cell	Perform operations related to cell and range operations, such as append, delete, format, get properties, and insert.  See <a href="#">Cell operations</a>
Row and column	Perform operations related to column and row operations such as autofit, delete, and read.

Operations	Description
	See <a href="#">Column/Row operations</a> .
Table	Perform operations related to the table operations such as create, delete, filter, get properties, rename, and sort. See <a href="#">Table operations</a> .
Workbook	Automate opening, closing, and creating a workbook. See <a href="#">Workbook operations</a> .
Worksheet	Perform operations related to worksheet operations, such as activate, delete, find, get worksheet names, hide, retrieve worksheet count, and show. See <a href="#">Worksheet operations</a> .

## Cell operations

The Office 365 Excel package contains actions that you can use to automate tasks related to cell and range operations, such as append, delete, format, get properties, and insert.

The Office 365 Excel package includes the following actions:

Action	Description
Delete cell	<p>Deletes the Active cell or a Specific cell from the current worksheet.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>If you select Specific cell, enter the cell location. For example A1</li> <li>Select one of the following options: <ul style="list-style-type: none"> <li>Shift cells left: Deletes the specified cell and shifts the cell one position to left.</li> <li>Shift cells up: Deletes the specified cell and shifts the cell one position up.</li> <li>Entire row: Deletes the entire row that contains the cell that you have specified to delete.</li> <li>Entire column: Deletes the entire column that contains the cell that you have specified to delete.</li> </ul> </li> </ul>
Delete range	<p>Deletes a specific range of cells.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Enter the cell range. For example A1 : B4</li> <li>This deletes an area consisting of the first four rows by the first two columns.</li> <li>Use the radio buttons to specify whether to shift the cells up or left. <ul style="list-style-type: none"> <li>Shift cells up: Deletes the specified cells and shifts the cells up by the number of rows</li> </ul> </li> </ul>

Action	Description
	<ul style="list-style-type: none"> <li>Shift cells left: Deletes the specified cells and shifts the cells left by the number of columns deleted.</li> </ul>
Format cell	See <a href="#">Using the Format cell action</a> .
Get cell	<p>Retrieves the value of a cell.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Select the Active cell or Specific cell option and enter the cell location.</li> <li>Assign the output to a String variable. To perform mathematical operations, convert the string to a number. See the <a href="#">String &gt; Convert action</a>.</li> </ul>
Get cell/text color	See <a href="#">Using the Get cell/text color action</a> .
Get multiple cells	<p>Retrieves the value(s) of multiple cells within a worksheet and assigns the output to a variable.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Select the Multiple cells or All cells option. If you select Multiple cells, enter the cell range, such as <code>A1:D1</code>. This retrieves the values of the first four cells in the top row.</li> <li>Assign the output to a String variable. To convert the string to a number, see the <a href="#">String &gt; Convert action</a>.</li> </ul>
Go to cell	<p>Moves the cursor to a specific cell in the worksheet.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Select from the following options: <ul style="list-style-type: none"> <li>Specific cell: Moves to the specified cell address.</li> <li>One cell to the left: Moves one cell left.</li> <li>One cell to the right: Moves one cell right.</li> <li>One cell above: Moves one cell up.</li> <li>One cell below: Moves one cell down.</li> <li>Beginning of row: Moves to the first cell in the same row.</li> <li>End of row: Moves to the last cell that contains data in the same row.</li> <li>Beginning of column: Moves to the first cell in the same column.</li> <li>End of column: Moves to the last cell that contains data in the same column.</li> </ul> </li> </ul>
Insert cell	<p>Inserts a value to the Active cell or a Specific cell in the current worksheet without overwriting the existing value.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>If you select Specific cell, enter the cell location. For example <code>A1</code>.</li> <li>After inserting the cell, you can: <ul style="list-style-type: none"> <li>Shift cells down: Shifts the existing values down by the specified number of rows.</li> <li>Shift cells right: Shifts the existing values right by the specified number of columns.</li> </ul> </li> </ul>
Insert range	Inserts the range into the current worksheet without overwriting the existing value.

Action	Description
	<ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Enter the cell range. For example A1:B4. This inserts a range of cells consisting of the first two columns.</li> <li>• After inserting the cell, you can: <ul style="list-style-type: none"> <li>• Shift cells down: Shifts the existing values down by the specified number of rows.</li> <li>• Shift cells right: Shifts the existing values right by the specified number of columns.</li> </ul> </li> </ul>
Paste cell	<p>Copies a value from a cell and pastes to a specified cell. If there is a value in the destination cell, the value will overwrite the value.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the Active cell or Specified cell option to specify the cell from which to copy the value. If you have selected the Specified cell option, specify the address of the cell in the field.</li> <li>• Enter the destination cell address to paste the value. For example, B3.</li> </ul>
Read cell format	<p>Gets the format of the Active cell or Specified cell and assigns the output to a string variable. This action returns a blank value if the specified cell does not contain a formula.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the Active cell or Specified cell option to specify the cell from which to read the format. If you have selected the Specified cell option, specify the address of the cell in the field.</li> <li>• Select a String variable to store the cell format from the Assign the output to variable list.</li> </ul>
Read cell formula	<p>Gets the formula available in the Active cell or Specified cell and assigns the output to a string variable. This action returns a blank value if the specified cell does not contain a formula.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the Active cell or Specified cell option to specify the cell from which to read the formula. If you have selected the Specified cell option, specify the address of the cell in the field.</li> <li>• Select a string variable to store the name of the formula from the Assign the output to variable list.</li> </ul>
Set cell	<p>Sets a value in the Active cell or Specific cell in a Microsoft Excel spreadsheet or a CSV file. You can use this action to set a formula.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the Active cell or Specified cell option to specify the cell in which to set the value. If you have selected the Specified cell option, specify the address of the cell in the field.</li> <li>• Enter the value to set in the Cell value field.</li> </ul>
Set cell color	<p>Sets a color to the background or text of the Active cell or Specific cell.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the Active cell, Specified cell, or Cell range option to specify the cell in which to set the color. If you have selected the Specified cell option, specify the address of the cell in the field.</li> <li>• Select which to apply the color to: the Cell or Text within cell.</li> <li>• Enter the value to set in the Cell value field. Use either the color name or the hex value. For a list of color names and corresponding hex values, see <a href="#">Frequently used cell colors</a>.</li> </ul>

Action	Description
Set cell formula	<p>Sets the formula of the specified cell.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the Active cell, Specified cell, or Cell range option to specify the cell in which to set the formula. If you have selected the Specified cell option, specify the address of the cell in the field.</li> <li>• Enter the value to set in the Cell formula field without an "=".</li> </ul>

## Using the Format cell action

Use the Format cell action to set the format of a cell or group of cells.

To format a cell or group of cells, do the following:

### Procedure

1. Double-click or drag the Format cell action.
2. Specify the name of the session in which you want to perform the action in the Session name field.
3. Select the Active cell, Specific cell, or Multiple cells option and provide cell location or cell range, as necessary.
4. Mark the Font option to modify any of the following:
  - Font name
  - Font size (between 8 and 72)
  - Bold or italicize
  - Underline
  - Font color (accepts color name or hex value). For a list of color names and corresponding hex values, see [Frequently used cell colors](#).
5. Mark the Alignment option, then use the drop-down lists to modify the vertical or horizontal alignment.
6. Mark the wrap text option to expand the cell(s) vertically to show long strings of text.
7. Mark the Merge Type option to select the any of the following operations:
  - Merge and center
  - Merge across
  - Merge cells
  - Unmerge
8. Click Apply.

## Frequently used cell colors

## Frequently used cell colors

The table below contains color names and their corresponding hex values. These are the most frequently used options for formatting worksheets.



Name	Hex
Aqua	#00FFFF
Black	#000000
Blue	#0000FF
Gray	#808080
Green	#008000
Fuchsia	#FF00FF
Lime	#00FF00
Maroon	#800000
Navy	#000080
Olive	#808000
Purple	#800080
Red	#FF0000
Silver	#C0C0C0
Teal	#008080
White	#FFFFFF
Yellow	#FFFF00

## Using the Get cell/text color action

Use the Get cell/text color action to get the color of the background or text in a cell. This action retrieves the color of a cell as RGB values. For example, if the background or text in a cell is of red color, the value retrieved is 255,0,0.

To get the color of the background or text in a cell, do the following:

### Procedure

1. Double-click or drag the Get cell/text color action.
2. Specify the name of the session in which you want to perform the action in the Session name field.
3. Select the Active cell option to get the color from the current active cell or the Specific cell option to get the color from the address of the cell you have specified.
4. Select the Cell color option to get the background color of the cell, the Text color option to get the color of the text, or both.
5. Specify whether to retrieve the cell color by name or as an RGB value.
6. Select a variable from the Assign the output to variable list to assign the cell/text color to a List variable.
7. Click Apply.
  - If both cell and text colors are retrieved, the List variable contains the cell color at index 0 and text color at index 1.

## Column/Row operations

The Office 365 Excel package contains actions that you can use to automate tasks related to column and row operations such as autofit, delete, insert, and read.

The Office 365 Excel package includes the following actions:

Actions	Description
Autofit columns	Adjusts the width of the columns in the worksheet in the specified session. Use the name of the session used to open the current workbook with the Open action.
Autofit rows	Adjusts the height of the rows in the worksheet in the specified session. Use the name of the session used to open the current workbook with the Open action.
Delete row/ column	See <a href="#">Using Delete action for rows or columns</a> .
Get row count	Retrieves the number of rows that contain data in the worksheet. <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Assign the output value to a Number variable. To convert the variable to a String variable, see the <a href="#">Number &gt; Convert</a> action.</li> </ul>
Insert row/ column	See <a href="#">Using Insert action for rows or columns</a> .
Read column	Extracts data from a column and stores it in a List variable. <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the From active cell or From specific cell option to specify the starting point.</li> <li>• You can also select the Read full column option to extract data for the entire column.</li> <li>• Assign the output value to a List variable.</li> </ul>
Read row	Extracts data from a row and stores it in a Record variable. <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Select the From active cell or From specific cell option to specify the starting point.</li> <li>• You can also select the Read full row option to extract data for the entire row.</li> <li>• Assign the output value to a Record variable.</li> </ul>

# Using Delete action for rows or columns

Use the Delete action to remove rows or columns from the current worksheet.

To delete rows or columns in a worksheet, do the following:

## Procedure

1. Double-click or drag Office 365 Excel > Delete.
2. Specify the name of the session in which you want to perform the action in the Session name field.
3. Select one of the following:
  - Row operations
    - a) Select the Delete Row(s) at option to delete all of the cells in a specific row. Specify the row number to delete in the field. For example, to delete the tenth row in the worksheet, enter 10 in the field.
    - b) Select the Delete Row(s) by option to delete all of the cells in either:
      - c) the row of the active cell.
      - d) a specific range of cells. Specify the range to delete. For example, to delete the first five rows, enter 1 : 5 in the field.
  - Column operations
    - a) Select the Delete Column(s) at option to delete all of the cells in a specific column. Specify the address of the column to delete in the field. For example, to delete column 'D' in the worksheet, enter D in the field.
    - b) Select the Delete Columns(s) by option to delete all of the cells in either:
      - c) the column of the active cell.
      - d) a specific range of cells. Specify the range to delete. For example, to delete the first five columns, enter A : E in the field.
4. Click Apply.

## Next steps

# Using Insert action for rows or columns

Use the Insert action to create rows or columns in the current worksheet.

To insert or delete rows or columns in a worksheet, do the following:

## Procedure

1. Double-click or drag Office 365 Excel > Insert.
2. Specify the name of the session in which you want to perform the action in the Session name field.
3. Select one of the following:
  - Row operations
    - a) Select the Insert Row(s) at option to insert a row and specify the location where to insert the row in the field. For example, to insert a row of cells in the tenth row in the worksheet, enter 10 in the field.
    - b) Select the Insert Row(s) by option to insert all of the cells in either:

- c) the row of the active cell.
- d) a specific range of cells. Specify the range where to insert the row. For example, to insert a row of cells in the first five rows in the worksheet, enter 1 : 5 in the field.
- Column operations
  - a) Select the Insert Column(s) at option to insert a column and specify the location where to insert the row in the field. For example, to insert a column of cells in column 'D' in the worksheet, enter D in the field.
  - b) Select the Insert Columns(s) by option to insert all of the cells in either:
    - c) the column of the active cell.
    - d) a specific range of cells. Specify the range to insert the column. For example, to insert a row of cells in the first five columns, enter A : E in the field.

4. Click Apply.

## Next steps

## Table operations

The Office 365 Excel package contains various actions that you can use to automate tasks related to the table operations such as create, delete, filter, get properties, rename, and sort.

The Office 365 Excel package includes the following actions:

Actions	Description
Create table	<p>Creates a new table in the specified table range.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Enter the table name.</li> <li>• Enter the cell range in which to create the table. For example, A1:D4.</li> <li>• Optional: Select the Table has headers option to establish the first row as the header row.</li> </ul>
Delete table	<p>Deletes the table in the active worksheet.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Enter the table name.</li> </ul>
Get table column	<p>Retrieves the column values of a specified table and column index, and assigns the values to a List variable.</p> <ul style="list-style-type: none"> <li>• Enter the name of the session used to open the current workbook with the Open action.</li> <li>• Enter the table name.</li> <li>• Enter the column index. For example, enter 5 to indicate the fifth column from the left.</li> </ul>

Actions	Description
	<ul style="list-style-type: none"> <li>Select a List variable to store the values.</li> </ul>
Get table names	<p>Retrieves the names of all the tables in a workbook and assigns them to a List variable.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Optional: Select whether to specify a sheet either by index or name.</li> <li>Select a List variable to store the values.</li> </ul>
Get table row	<p>Retrieves the row values of a specified table and row index, and assigns the values to a Record variable.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Enter the table name.</li> <li>Enter the row index. For example, enter 5 to indicate the fifth row from the top.</li> <li>Select a Record variable to store the values.</li> </ul>
Get table row count	<p>Retrieves the row count of a specified table and assigns the values to a number variable.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Enter the table name.</li> <li>Select a Number type variable to store the values. To convert the variable to a String type variable, see the <a href="#">Number &gt; Convert</a> action.</li> </ul>
Rename table	<p>Renames a table.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Enter the current table name.</li> <li>Enter a new table name.</li> </ul>

## Workbook operations

The Office 365 Excel package contains actions that you can use to automate opening, closing, or creating a new workbook.

The Office 365 Excel package includes the following actions:

Actions	Description
Close	Closes the workbook. Use the name of the session used to open the current workbook with the Open action.

Actions	Description
Create workbook	<p>Creates a new workbook.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Provide a File Path containing the new workbook name with one of the following extensions: .xls, .xlsm, .xlsx. Either enter the file path or select a String variable.</li> </ul> <p>For example, to perform this action on a file named SalesReport.xlsx, located in the Accounts folder, enter</p> <p><code>Accounts/SalesReport.xlsx</code></p> <p>.</p>
Open	See <a href="#">Using the Open action</a> .

## Using the Open action

Use this action to open a workbook. Insert this action at the start of your automation sequences.

To open a workbook, do the following:

### Procedure

1. Double-click or drag the Open action.
2. Provide the username either by selecting a Credential variable or entering an unencrypted value.
3. Enter a Session name.  
Use this same session name in subsequent actions to associate them with this workbook.
4. Enter the File Path or insert a file variable.  
For example, to perform this action on a file named SalesReport.xlsx, located in the Accounts folder, enter  
`Accounts/SalesReport.xlsx`  
.
5. If the sheet contains a header row, mark the Sheet contains header option.
6. Click Apply.

## Worksheet operations

The Office 365 Excel package contains actions that you can use to automate tasks related to worksheet operations, such as delete, find, get worksheet names, hide, retrieve worksheet count, and show.

The Office 365 Excel package includes the following actions:

Actions	Description
Copy worksheet	See <a href="#">Using the Copy worksheet action</a> .
Create worksheet	<p>Adds an empty sheet in the current workbook.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Enter a Worksheet name of up to 31 characters. The name cannot contain the following characters: \ / ? * [ ]</li> </ul>
Delete worksheet	<p>Deletes a spreadsheet from the current workbook.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Specify either an index number in the Sheet by Index field or a name in the Sheet by Name field for the worksheet. The index number is the number assigned to a worksheet. For example, if you want to perform an operation on the worksheet at the third position in the workbook, enter 3 in the field.</li> </ul> <p>Note: You can delete a worksheet only if the workbook contains more than one worksheet.</p>
Get current worksheet name	<p>Retrieves the name of the currently active worksheet.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Select a string variable that you want to use to store the name of the worksheet from the Assign the output to variable list.</li> </ul>
Get worksheet names	<p>Retrieves the names of all the worksheets in the workbook.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Select a list variable to store the worksheet names from the Assign the output to variable list.</li> </ul>
Hide worksheet	<p>Hides a worksheet from the current workbook.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Specify the name of the worksheet to hide in the Enter worksheet name to hide field.</li> </ul> <p>Note: You can hide a worksheet only if the workbook contains more than one worksheet.</p>
Rename worksheet	Renames a worksheet in the current workbook.

Actions	Description
	<ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Specify the index number or name of the worksheet to rename in the Sheet by Index or Sheet by Name field. The index number is the number assigned to a worksheet. For example, if you want to perform an operation on the worksheet at the third position in the workbook, enter 3 in the field. Note: You cannot rename the worksheet if a worksheet with the same name already exists in the workbook.</li> <li>Enter the new worksheet name that is under 31 characters.</li> </ul>
Retrieve sheet count	<p>Gets the number of sheets available in the current workbook and stores it in a number variable.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Select the appropriate option to specify whether to include the hidden worksheet or not and assign the count to a variable.</li> </ul>
Show worksheet	<p>Shows the hidden worksheet.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Enter the Worksheet name.</li> </ul>
Switch to sheet	<p>Activates a particular sheet in a Microsoft Excel file.</p> <ul style="list-style-type: none"> <li>Enter the name of the session used to open the current workbook with the Open action.</li> <li>Specify whether to activate the Sheet by Index (numerical value) or Sheet by Name.</li> </ul>

## Using the Copy worksheet action

Use the Copy worksheet action to copy values of a worksheet from one worksheet to another. This action overwrites any existing values. To move values from one worksheet to another without overwriting existing values, use the Append worksheet action.

### Prerequisites

Open the workbook(s) containing the source and target worksheets. See [Using the Open action](#).



Note: This action identifies the destination and source workbooks by the session names used to open them. If opening two workbooks, enter a session name for the source workbook that is different than the one used to open the destination workbook.

To copy the values of a worksheet into another, do the following:

## Procedure

1. Double-click or drag the Copy worksheet action.
2. Enter the name of the session that you used to open the destination workbook in the Open action.
3. If the workbook contains more than one worksheet, mark the Specific Sheet option, then enter the index number or name of the source worksheet.
  - The index number represents the position of the worksheet in the workbook. For example, to perform an operation on the worksheet at the third position in the workbook, enter  

3

 in the field.
  - The sheet name field is case insensitive.
4. Enter the name of the session that you used to open the source workbook in the Open action.
5. If the workbook contains more than one worksheet, mark the Specific Sheet option, then enter the index number or name of the destination worksheet.
6. Click Apply.

## Office 365 One Drive package

The One Drive package contains actions that enable you to automate many of the repetitive tasks in Microsoft cloud storage.

## How to use the actions in the One Drive package

Perform the following actions within the One Drive package as part of using the set of available actions:

1. Use the Connect action to establish a connection to the Office 365 server.  
See [Using the Connect action](#).  
Note: Office 365 packages do not currently support Multi-Factor Authentication or Single Sign-on.
2. Use a combination of actions available in this package to automate tasks.  
Note: To use actions from other Office 365 packages, establish a connection using the Connect action from that package.
3. Use the Disconnect action to terminate the connection.

## Actions in the One Drive package

Use the actions in the One Drive package to perform an operation on a single file or folder. To perform the operation on every file in a folder, use that action in a loop. See [Using One Drive actions in a loop](#).

The One Drive package includes the following actions:

Action	Description
Check permission	Checks for read, write, or delete permission for a file or folder in your OneDrive. <ul style="list-style-type: none"> <li>• Enter the user session that you provided in the Connect action.</li> </ul>

Action	Description
	<ul style="list-style-type: none"> <li>Enter the file or folder name and path, or select a predefined string variable.</li> <li>Specify which permission type to verify: Read or Read &amp; Write.</li> <li>Select a predefined Boolean type variable to hold the output, or click the icon on the right to create a new variable.</li> </ul> <p>The Boolean type variable will provide a true (user has permission) or false (user does not have permission) response.</p>
Connect	<p>Establishes a connection with the Office 365 server using your organization's client ID and tenant ID, and user credentials.</p> <p>See <a href="#">Using the Connect action</a>.</p>
Copy file or folder	<p>Copies a file or folder from one folder to another in your OneDrive.</p> <ul style="list-style-type: none"> <li>Enter the user session that you provided in the Connect action.</li> <li>Select whether to copy a file or folder.</li> <li>Provide the source and destination file/folder names and paths. Either enter the values or insert a string variable.</li> </ul> <p>For example, to perform this action on a file named SalesReport.xlsx, located in the Accounts folder, enter</p> <pre>/Accounts/SalesReport.xlsx</pre> <p>.</p> <p>Note: Use the forward slash to indicate that a file or folder is located in the root directory. For example,</p> <pre>/Accounts</pre> <p>.</p> <ul style="list-style-type: none"> <li>Optional: If there is file or folder with the same name in the destination folder, select the Override existing file/folder option which replaces the existing file or folder with the one being copied.</li> </ul>
Create folder	<p>Creates a folder in a specific directory in your OneDrive.</p> <ul style="list-style-type: none"> <li>Enter the user session that you provided in the Connect action.</li> <li>Enter the parent folder name and path, or folder link. Either enter the values or insert a string variable.</li> </ul> <p>For example, to perform this action on a folder named Q1, located in the Survey Results folder, enter</p> <pre>/SurveyResults/Q1</pre> <p>.</p> <ul style="list-style-type: none"> <li>Enter the new folder name. Either enter the values or insert a string variable.</li> </ul>
Delete file or folder	<p>Deletes a file or folder in a specific directory in your OneDrive.</p> <ul style="list-style-type: none"> <li>Enter the user session that you provided in the Connect action.</li> </ul>

Action	Description
	<ul style="list-style-type: none"> <li>• Select whether to delete a file or folder.</li> <li>• Provide the name and path, or file/folder link. Either enter the values or insert a string variable. For example, to perform this action on a file named SalesReport.xlsx, located in the Accounts folder, enter <code>/Accounts/SalesReport.xlsx</code></li> <li>• Note: Use the forward slash to indicate that a file or folder is located in the root directory. For example, <code>/Accounts</code></li> <li>•</li> </ul>
Download file	<p>Downloads a file from a specific directory in your OneDrive.</p> <ul style="list-style-type: none"> <li>• Enter the user session that you provided in the Connect action.</li> <li>• Provide the file name and path, or file link. Either enter the values or insert a predefined string variable. For example, to perform this action on a file named SalesReport.xlsx, located in the Accounts folder, enter <code>/Accounts/SalesReport.xlsx</code></li> <li>• Note: Use the forward slash to indicate that a file or folder is located in the root directory. For example, <code>/Accounts</code></li> <li>•</li> <li>• Enter the destination folder path or select a predefined string variable.</li> <li>• Optional: If there is file or folder with the same name in the destination folder, select the Override existing file/folder option which replaces the existing file or folder with the one being copied.</li> </ul>
Export as PDF	<p>Exports an existing file in your OneDrive as a PDF. This action supports the following file extensions: doc, docx, oentry, odp, pps, ppt, pptx, tf, xls, and.xlsx.</p> <ul style="list-style-type: none"> <li>• Enter the user session that you provided in the Connect action.</li> <li>• Provide the file name and path, or file link. Either enter the values or insert a predefined string variable. For example, to perform this action on a file named SalesReport.xlsx, located in the Accounts folder, enter <code>/Accounts/SalesReport.xlsx</code></li> <li>• Note: Use the forward slash to indicate that a file or folder is located in the root directory. For example, <code>/Accounts</code></li> <li>•</li> <li>• Enter the export folder name and path or select a predefined string variable.</li> <li>• Optional: If there is file or folder with the same name in the destination folder, select the Override existing file/folder option which replaces the existing file or folder with the one being copied.</li> </ul>

Action	Description
Find files and folders	<p>Finds files and folders in a specific directory in your OneDrive.</p> <ul style="list-style-type: none"> <li>• Enter the user session that you provided in the Connect action.</li> <li>• Provide directory details in the Folder to search field: enter the folder name and path, or folder link. Either enter the values or insert a string variable.</li> </ul> <p>For example, to perform this action on a folder named Q1, located in the Survey Results folder, enter</p> <div> <div>/Survey</div> <div>Results/Q1</div> </div> <ul style="list-style-type: none"> <li>• Specify whether to search for All items in folder or a Specific file (or folder). Note: The Drive Item Name field accepts wildcard characters in the file name and extension. For example, to search for all documents with extension docx, enter *.docx in the Specific file (or folder) field.</li> <li>• Assign the output to a table variable.</li> </ul> <p>The variable will hold the following file or folder details: Name, ID, CreatedBy, CreationDate, LastModifiedBy, LastModificationDate, and ParentFolderPath.</p>
Get file or folder information	<p>Retrieves information for a specific file or folder from OneDrive.</p> <ul style="list-style-type: none"> <li>• Enter the user session that you provided in the Connect action.</li> <li>• Provide the file or folder name and path, or file/folder link. Either enter the values or insert a string variable.</li> </ul> <p>For example, to perform this action on a file named SalesReport.xlsx, located in the Accounts folder, enter</p> <div> <div>/Accounts/SalesReport.xlsx</div> <div></div> </div> <p>Note: Use the forward slash to indicate that a file or folder is located in the root directory. For example,</p> <div> <div>/Accounts</div> <div></div> </div> <ul style="list-style-type: none"> <li>• Assign the output to a record variable.</li> </ul> <p>The variable will hold the following file or folder details: Name, ID, CreatedBy, CreationDate, LastModifiedBy, LastModificationDate, and ParentFolderPath.</p>
Move file or folder	<p>Moves a file or folder from one folder to another in OneDrive.</p> <ul style="list-style-type: none"> <li>• Enter the user session that you provided in the Connect action.</li> <li>• Select whether to move a file or folder.</li> <li>• Provide the source and destination file or folder names and paths, or file/folder links. Either enter the values or insert a string variable into the two fields.</li> </ul> <p>For example, to perform this action on a file named SalesReport.xlsx, located in the Accounts folder, enter</p> <div> <div>/Accounts/SalesReport.xlsx</div> <div></div> </div>

Action	Description
	<p>.</p> <p>Note: Use the forward slash to indicate that a file or folder is located in the root directory. For example, /Accounts</p> <p>.</p> <ul style="list-style-type: none"> <li>Optional: If there is file or folder with the same name in the destination folder, select the Override existing file/folder option which replaces the existing file or folder with the one being copied.</li> </ul>
Rename file or folder	<p>Renames a file or folder in a specific directory in OneDrive.</p> <ul style="list-style-type: none"> <li>Enter the user session that you provided in the Connect action.</li> <li>Select whether to rename a file or folder.</li> <li>Provide the current and new file or folder names, or file/folder link. Either enter the values or insert a string variable.</li> <li>For example, to perform this action on a file named SalesReport.xlsx, located in the Accounts folder, enter /Accounts/SalesReport.xlsx</li> </ul> <p>.</p> <p>Note: Use the forward slash to indicate that a file or folder is located in the root directory. For example, /Accounts</p> <p>.</p> <ul style="list-style-type: none"> <li>Optional: If there is file or folder with the same name in the destination folder, select the Override existing file/folder option which replaces the existing file or folder with the one being copied.</li> </ul>
Restore last version	<p>Restores a file to the last version. If you restore a file to its previous version, you will lose the current version.</p> <ul style="list-style-type: none"> <li>Enter the user session that you provided in the Connect action.</li> <li>Provide the file name and path, or file link. Either enter the values or insert a predefined string variable.</li> </ul> <p>For example, to perform this action on a file named SalesReport.xlsx, located in the Accounts folder, enter /Accounts/SalesReport.xlsx</p> <p>.</p> <p>Note: Use the forward slash to indicate that a file or folder is located in the root directory. For example, /Accounts</p> <p>.</p>
Upload file	<p>Uploads a file to a specific directory in OneDrive.</p> <ul style="list-style-type: none"> <li>Enter the user session that you provided in the Connect action.</li> <li>Provide the file name and path on the local directory. Either enter the values or insert a predefined string variable.</li> </ul> <p>For example, D:/Mydata/HR/EmployeeSurvey.xlsx</p>

Action	Description
	<ul style="list-style-type: none"> <li>• Provide the upload folder name and path, or folder link. Either enter the values or insert a string variable.</li> </ul> <p>For example, to perform this action on a folder named Q1, located in the Survey Results folder, enter</p> <div> <div>/Survey</div> <div>Results/Q1</div> </div> <ul style="list-style-type: none"> <li>• Optional: If there is file or folder with the same name in the destination folder, select the Override existing file/folder option which replaces the existing file or folder with the one being copied.</li> </ul>

## Using One Drive actions in a loop

Use the Find files and folder action to return a table of files and their properties inside of a specified folder, the Loop action to loop through each row in the table, and a One Drive action to perform an operation on every file in the table.

### Procedure

To perform an action on every file in a folder, do the following steps:

1. Double-click or drag One Drive > Find files and folders.
2. Provide your username with either a credential variable or an unencrypted value.
3. Enter the file path of the folder in which to perform the search.
4. Select the All items in folder option.  
You can alternatively select Specific file (or folder) to limit the search to a specific folder on your OneDrive.
5. Select or create a table variable to hold the output.  
The table variable holds information on files and folders in rows, under the following columns: name, id, createdBy, creationDate, lastModifiedBy, lastModificationDateTime, parentFolder, and Path.
6. Click Apply.
7. Double-click or drag the Loop action.
8. Select the For each row in table iterator.
9. Select the table variable holding the output from step 5.
10. Select or create a record variable to temporarily hold each row.
11. Click Apply.

Insert the action inside of the Loop container that you want to repeat on every file in the folder. This example uses the Download file action.

0. Double-click or drag Outlook > Download file.
1. Provide your username with either a credential variable or an unencrypted value.

2. Enter the following in the File name including path field, replacing the generic variable names with the names of the record variables from step 10:  
`$RecordVariable[7]$/ $RecordVariable[0]`
3. Enter the path to the folder where to download the files.
4. Click Apply.
5. Click Save.

## PDF package

Use the PDF package to automate various operations on a PDF file.

The PDF package enables you to perform the following tasks:

- Encrypt or decrypt a PDF file.
- Extract text from a PDF file.
- Convert a PDF file to an image.
- Split a single PDF file into multiple files.

It is not necessary to have a PDF reader installed on your device.

## Actions in the PDF package

The PDF package includes the following actions:

Action	Description
Decrypt document	See <a href="#">Using Decrypt document action</a> .
Encrypt document	See <a href="#">Using Encrypt document action</a> .
Extract image	See <a href="#">Using Extract image action</a> .
Extract text	See <a href="#">Using Extract text action</a> .
Split document	See <a href="#">Using Split document action</a> .

When an operation is performed on a PDF file, the file properties are stored in a dictionary variable. See [Using dictionary variable for PDF properties](#).

## Using Encrypt document action

Use the Encrypt document action to encrypt a PDF file.

### Procedure

To encrypt a PDF file, follow these steps:

1. In the Actions palette, double-click or drag the Encrypt document action from the PDF package.
2. In the PDF path, select one of the following options to specify the location of the PDF:

- Control Room file: Enables you to select a PDF file that is available in a folder.
  - Desktop profile: Enables you to select a PDF file that is available on your device.
  - Variable: Enables you to specify the file variable that contains the location of the PDF file.
3. Optional: In the User password or Owner password field, enter a password to restrict access to the encrypted PDF file.
    - User password: Allow users to perform specific operations on the encrypted PDF file.
    - Owner password: Allow users to use a password to open the file.
  4. In the User Permissions to Apply field, select the following operations:
    - Print: Allows users to print the document.
    - Modify: Allows users to edit the document.
    - Copy: Allows users to copy the document.
    - Form Fill: Allows users to fill a form in the document.
    - Document Assembly: Allows users to combine multiple PDF files, attach files, and so on.
    - Annotation: Allows users to apply annotations in the document.
    - Accessibility: Allows users to read text from the document using accessibility devices.
  5. In the Encryption level, select the RC4 40-bit, RC4 128-bit, or AES 128-bit option to specify the encryption level.
  6. In the Save encrypted PDF as field, specify a name and location for the encrypted file.  
You must include the .pdf extension in the name of the encrypted file. For example, if the file name is June\_Quarter\_report, the .pdf extension is June\_Quarter\_report.pdf.
  7. Select the Overwrite files with the same name check box to overwrite existing files with the same name.  
Note: If this option is not selected, the system appends a numeric value at the end of the file name. For example, if you save the June\_Quarter\_report file in a location that has a file with the same name, the system saves the file as June\_Quarter\_report\_(1).pdf. The numeric value is incremented each time you save the file as long as the option is selected.
  8. Optional: From the Assign PDF properties to a dictionary variable list, select a dictionary variable to hold the file properties.  
For more information, see [Using dictionary variable for PDF properties](#).
  9. Click Apply.
  10. Click Save.

#### Related tasks

[Using Decrypt document action](#)

## Using Extract text action

Use the Extract text action to extract text from a PDF file and save it as a text file.

### Procedure

To extract text from a PDF file, follow these steps:

1. In the Actions palette, double-click or drag the Extract text action from the PDF package.
2. In the PDF path, select one of the following options to specify the location of the PDF:
  - Control Room file: Enables you to select a PDF file that is available in a folder.
  - Desktop profile: Enables you to select a PDF file that is available on your device.
  - Variable: Enables you to specify the file variable that contains the location of the PDF file.
3. Optional: In the User password or Owner password field, enter a password to restrict access to the encrypted PDF file.
  - User password: Allow users to perform specific operations on the encrypted PDF file.
  - Owner password: Allow users to use a password to open the file.



4. In the Text type field, select one of the following options:
  - Plain text: Enables you to extract the text and copy it to a text file.

It is similar to copying and pasting text from a PDF file to a text file.

- Structured text: Enables you to preserve the original formatting of the extracted text from the PDF file.
5. In the Page range field, select one of the following options:
    - All pages: Enables you to save all the pages in the PDF file as an image.
    - Pages: Enables you to enter the page numbers of the pages that you want to save as an image.
  6. In the Export data to text file field, specify a name and location for the text file.  
 Note: You must include the .txt extension in the name of the text file. For example, if the file name is June\_Quarter\_report, the .txt extension is June\_Quarter\_report.txt.
  7. Select the Overwrite files with the same name check box to overwrite existing files with the same name.  
 Note: If this option is not selected, the system appends a numeric value at the end of the file name. For example, if you save the June\_Quarter\_report file in a location that has a file with the same name, the system saves the file as June\_Quarter\_report\_(1).pdf. The numeric value is incremented each time you save the file as long as the option is selected.
  8. Optional: From the Assign PDF properties to a dictionary variable list, select a dictionary variable to hold the file properties.  
 For more information, see [Using dictionary variable for PDF properties](#).
  9. Click Apply.
  10. Click Save.

## Using Extract image action

Use the Using Extract image action to save a PDF file as an image.

### Procedure

To save a PDF file as an image, follow these steps:

1. In the Actions palette, double-click or drag the Extract image action from the PDF package.
2. In the PDF path, select one of the following options to specify the location of the PDF:
  - Control Room file: Enables you to select a PDF file that is available in a folder.
  - Desktop profile: Enables you to select a PDF file that is available on your device.
  - Variable: Enables you to specify the file variable that contains the location of the PDF file.
3. Optional: In the User password or Owner password field, enter a password to restrict access to the encrypted PDF file.
  - User password: Allow users to perform specific operations on the encrypted PDF file.
  - Owner password: Allow users to use a password to open the file.
4. In the Page range field, select one of the following options:
  - All pages: Enables you to save all the pages in the PDF file as an image.
  - Pages: Enables you to enter the page numbers of the pages that you want to save as an image.
5. In the Type of image to be converted to list, select one of the following options to specify the format:
  - TIFF: If you selected this option, select an option from the TIFF compression type list.
    - NONE
    - LZE
    - RLE
    - CCITT Group 3

- CCITT Group 4
  - BMP
  - JPEG: If you selected this option, in the JPEG quality field, enter a value (between 0 and 100) to specify the quality of the compressed image.
  - GIF
  - PNG
  - WMF
  - EMF
  - EXIF
6. In the Folder path field, specify the location.
  7. In the File prefix field, enter a value.  
The image files are suffixed with index numbers. For example, if you have specified Report as the File prefix, the name of the image files will be Report\_1, Report\_2, and so on.
  8. Select the Overwrite files with the same name check box to overwrite existing files with the same name.  
Note: If this option is not selected, the system appends a numeric value at the end of the file name. For example, if you save the June\_Quarter\_report file in a location that has a file with the same name, the system saves the file as June\_Quarter\_report\_(1).pdf. The numeric value is incremented each time you save the file as long as the option is selected.
  9. In the X Resolution (dpi) and Y Resolution (dpi) fields, specify the resolution.
  10. In the Image output field, choose Color or Grayscale to specify the image output type:
    - a) In the Color property list, select an option.
      - b) True color (32 bits)
      - c) True color (24 bits)
      - d) High color (16 bits)
      - e) 56 color (8 bits)
      - f) 16 color (4 bits)
      - g) 2 color (1 bit, black/white)
 Note: This option is available only with the Color image output type. Select this option only if RLE, CCITT Group 3, or CCITT Group 4 compression type is selected for the TIFF image format.
  11. Optional: From the Assign PDF properties to a dictionary variable list, select a dictionary variable to hold the file properties.  
For more information, see [Using dictionary variable for PDF properties](#).
  12. Click Apply.
  13. Click Save.

## Using Decrypt document action

Use the Decrypt document action to decrypt a PDF file that is encrypted using the Encrypt document action.

### Procedure

To decrypt an encrypted PDF file, follow these steps:

1. In the Actions palette, double-click or drag the Decrypt document action from the PDF package.
2. In the PDF path, select one of the following options to specify the location of the PDF:
  - Control Room file: Enables you to select a PDF file that is available in a folder.
  - Desktop profile: Enables you to select a PDF file that is available on your device.
  - Variable: Enables you to specify the file variable that contains the location of the PDF file.
3. Optional: In the User/Owner password field, enter a password to restrict access to the decrypted PDF file.

4. In the Save the decrypted PDF file as field, specify a name and location for the decrypted file.  
You must include the .pdf extension in the name of the decrypted file. For example, if the file name is June\_Quarter\_report, the .pdf extension is June\_Quarter\_report.pdf.
5. Select the Overwrite files with the same name check box to overwrite existing files with the same name.  
Note: If this option is not selected, the system appends a numeric value at the end of the file name. For example, if you save the June\_Quarter\_report file in a location that has a file with the same name, the system saves the file as June\_Quarter\_report\_(1).pdf. The numeric value is incremented each time you save the file as long as the option is selected.
6. Optional: From the Assign PDF properties to a dictionary variable list, select a dictionary variable to hold the file properties.  
For more information, see [Using dictionary variable for PDF properties](#).
7. Click Apply.
8. Click Save.

#### Related tasks

[Using Encrypt document action](#)

## Using Split document action

Use the Split document action to split a PDF file into multiple files.

### Procedure

To split a PDF file into multiple files, follow these steps:

1. In the Actions palette, double-click or drag the Split document action from the PDF package.
2. In the PDF path, select one of the following options to specify the location of the PDF:
  - Control Room file: Enables you to select a PDF file that is available in a folder.
  - Desktop profile: Enables you to select a PDF file that is available on your device.
  - Variable: Enables you to specify the file variable that contains the location of the PDF file.
3. Optional: In the User password or Owner password field, enter a password to restrict access to the encrypted PDF file.
  - User password: Allow users to perform specific operations on the encrypted PDF file.
  - Owner password: Allow users to use a password to open the file.
4. In the Output file creation options, select one of the following options to specify how to split the PDF file:
  - Number of pages per extracted PDF: Enables you to specify the number of pages each new file will contain. For example, you can enter 8 if you want each output file to contain only 8 pages.
  - Single file with selected pages: Enables you to create a single output file that contains the pages you have specified from the original file.
  - Blank page as a separator: Creates a new output file whenever a blank page is encountered in the original PDF file.
  - Bookmark level per file: Enables you to split the PDF file according to the bookmark level available in the file. For example, if a file contains three bookmark levels, you can split the PDF file based on these three levels.

If you selected this option, specify the bookmark level in the Bookmark Level field.

5. In the Folder path field, specify the location.
6. In the File prefix field, enter a value.

The image files are suffixed with index numbers. For example, if you have specified Report as the File prefix, the name of the image files will be Report\_1, Report\_2, and so on.

7. Select the Overwrite files with the same name check box to overwrite existing files with the same name.  
Note: If this option is not selected, the system appends a numeric value at the end of the file name. For example, if you save the June\_Quarter\_report file in a location that has a file with the same name, the system saves the file as June\_Quarter\_report\_(1).pdf. The numeric value is incremented each time you save the file as long as the option is selected.
8. Optional: From the Assign PDF properties to a dictionary variable list, select a dictionary variable to hold the file properties.  
For more information, see [Using dictionary variable for PDF properties](#).
9. Click Apply.
10. Click Save.

## Using dictionary variable for PDF properties

When you automate a PDF-related task, Automation Anywhere Enterprise retrieves various properties of the file and stores the values of these properties in a dictionary variable.

Automation Anywhere Enterprise retrieves the PDF file name and extension, title, subject, and author. The file properties are stored in a dictionary variable within the following keys:

- pdfTitle
- pdfFilename
- pdfSubject
- pdfAuthor

The system automatically associates the properties of a PDF with the appropriate dictionary keys.

For example, if you create a dictionary variable Test and want to log the file name and author, you must enter

```
Test{pdfFilename}
```

and

```
Test{pdfAuthor}
```

in the appropriate fields.

## Play Sound package

The Play Sound package contains actions that enable you to play a beep sound and media file before or after an action is executed in a bot.

The Play Sound package includes the following actions:

Action	Description
Play beep	Plays a beep sound before or after an action is executed.
Play media file	Plays an audio file before or after an action is executed. Note: Only .wav and .mp3 files are supported and do not require any media player installed on the device.

Action	Description
	<p>Select any of the following tabs to specify the location of the media file:</p> <ul style="list-style-type: none"> <li>Control Room file: Use a file that is available on the Enterprise Control Room.</li> <li>Desktop file: Use a file that is available on a device.</li> <li>Variable: Use a file variable to specify the file location.</li> </ul>

## PGP package

Use the PGP (Pretty Good Privacy) package to automatically encrypt and decrypt files for security.

The PGP package supports two types of encryption, symmetric (passphrase) and asymmetric (public/private keys):

- PGP Passphrase: Requires a passphrase to encrypt and decrypt files.
  - PGP Public/Private Key: Requires PGP public and private key files to encrypt and decrypt files.
- Note: The system outputs an error when the proper file extension is not present in the path.
- Public key: File name specified for Public Key is not valid.
  - Private key: File name specified for Private Key is not valid.

## Actions in the PGP package

The PGP package includes the following actions:

Action	Description
Create keys	<p>Creates a pair of public and private encryption keys.</p> <ul style="list-style-type: none"> <li>In the Location to save public key file field, specify the path to the text file where to save the public key.</li> <li>In the Location to save private key file field, specify the path to the text file where to save the private key.</li> <li>Optional: In the Password to protect private key file field, select a credential variable or enter a value.</li> <li>Optional: Select the Overwrite Files check box to replace files with the same name.</li> </ul>
Decrypt files	Decrypts encrypted files. See <a href="#">Using the Decrypt files action</a>
Encrypt files	Encrypts files to enable users to send them over the internet securely. See <a href="#">Using the Encrypt files action</a>

## Using the Decrypt files action

Use the Decrypt files action to retrieve the contents of encrypted files and folders.

## Procedure

To decrypt a file or folder, do the following steps:

1. Double-click or drag the Decrypt action from the PGP package.
2. Select either the Passphrase or Public key file encryption type.
  - If you have selected Passphrase, enter the value or select a credential variable.
  - If you have selected Private key file, provide the private key file path.

Optional: In the Password to protect private key file field, select a credential variable or enter a value.

3. In the Source folder/file(s) field, specify the folders or files to decrypt.  
This field supports wildcard characters in the file name and extension. Use an asterisk (\*) to substitute for one or more unknown alphanumeric characters or symbols.
4. In the Destination folder/file(s) field, specify the location where to save the decrypted folders and files.  
This field supports wildcard characters in the file name and extension. Use an asterisk (\*) to substitute for one or more unknown alphanumeric characters or symbols.
5. Select the Overwrite Files option to replace files with the same name.
6. Select the Remove file extension option and specify the extension name.
7. Click Apply.

## Using the Encrypt files action

Use the Encrypt files action to prepare files and folders to be shared securely.

### Prerequisites

To encrypt the files or folder using a public key, create a public-private key pair using the Create keys action.

## Procedure

To encrypt a file or folder, do the following steps:

1. Double-click or drag the Encrypt action from the PGP package.
2. Select either the Passphrase or Public key file encryption type.
  - If you have selected Passphrase, enter the value or select a credential variable.
  - If you have selected Public key file, provide the public key file path.
3. Select the Encryption algorithm.

Choose from:

- AES128
  - AES192
  - AES256
  - Blowfish
  - CAST5
  - Idea
  - TripleDES
  - TwoFish256
4. In the Source folder/file(s) field, specify the folders or files to encrypt.

This field supports wildcard characters in the file name and extension. Use an asterisk (\*) to substitute for one or more unknown alphanumeric characters or symbols.

For example:

- Enter

```
C:\PGP\encrypt*.csv
```

to select the .csv files that start with the word encrypt.

- Enter

```
C:\PGP\*encrypt*.*
```

to select files of any extension that contain the word encrypt.

- In the Destination folder/file(s) field, specify the location where to save the encrypted folders and files. This field supports wildcard characters in the file name and extension. Use an asterisk (\*) to substitute for one or more unknown alphanumeric characters or symbols.
- Enter a value to append to the extension of the destination file.
- Select the Overwrite Files check box to replace files with the same name.
- Select the Compression type.  
Choose from:
  - None
  - zip
  - bzip2
  - zlib
- Select the Armor data (text output) check box to enable ASCII Armor output.
- Click Apply.

## Printer package

Use the actions in the Printer package to automate retrieving and setting the default printer.

## Actions in the Printer package

The Printer package includes the following actions:

Action	Description
Get Default	Retrieves the default printer and assigns the value to a string variable.
Set Default	Sets the default printer.

## Prompt package

Use the Prompt package to accept an input value, a yes/no response, or to open a file or folder.

## Actions in the Prompt package

The Prompt package includes the following actions:

Action	Description
For file	Prompts the user to input a value.

Action	Description
	<ul style="list-style-type: none"> <li>• In the Prompt window caption field, enter a window caption.</li> <li>• In the Prompt message field, enter a message.</li> <li>• In the Assign the value to a variable field, specify a list variable.</li> </ul>
For folder	<p>Prompts the user to choose a folder.</p> <ul style="list-style-type: none"> <li>• In the Prompt window caption field, enter a window caption.</li> <li>• In the Prompt message field, enter a message.</li> <li>• In the Assign the value to a variable field, specify a variable.</li> </ul>
For value	<p>Prompts the user to enter a value.</p> <ul style="list-style-type: none"> <li>• In the Prompt window caption field, enter a window caption.</li> <li>• In the Prompt message field, enter a message.</li> <li>• In the Assign the value to a variable field, specify a variable.</li> </ul>
For yes/no	<p>Prompts the user to choose a Yes/No response.</p> <ul style="list-style-type: none"> <li>• In the Prompt window caption field, enter a window caption.</li> <li>• In the Prompt message field, enter a message.</li> <li>• In the Assign the value to a variable field, specify a variable.</li> </ul>

Watch the following video on how to use the Prompt actions:

Using Prompt actions

Related reference

[Message box package](#)

## Python Script package

The Python Script package contains actions that enable Python Script functions in a task.

## Before you start

Ensure the following requirements are met before using the Python Script package:

- The appropriate version of Python is installed on the device on which you want to execute the script.  
Note: Python versions 2.x and 3.x are supported.
- The "PATH" environment variable is updated to the path of the Python folder.

To verify if the Python environment variable is set correctly, enter

`Python`

in the command prompt. If it returns the Python version that is installed, the path environment is set up correctly.

Perform the following actions within the Python Script package as part of using the set of available actions:



1. Open a Python Script file, or manually enter the script you want to run using the Open action. You must associate the details of the file or script you want to run with a session name.
2. Run the script using the Execute function or the Execute script action.
  - If you uploaded a file containing the script, use the Execute script action.
  - If you manually input the script, use the Execute function action.

You must use the same Python Script session name established in the Open action.
3. Close the Python Script session after running the script.

## Actions in the Python Script package

The Python Script package includes the following actions:

Note: The Open action must be the first action to use the Python Script in a task. These actions can run a Python Script on Windows, Linux, and UNIX-based devices.

Action	Description
Close	<p>Closes the session. Specify the same session name from the Open action. Important: It is mandatory to close the session after the script is executed.</p>
Execute function	<p>Executes a function within the Python Script.</p> <ul style="list-style-type: none"> <li>• In the Python session field, specify a session name. Use the same session name from the Open action.</li> <li>• Optional: Specify the function name to run and the arguments to pass to the function. Note: You can pass only one argument for the function. The argument can be a Boolean, dictionary data type, list, number, or string.</li> <li>• Optional: In the Assign the output to variable field, specify the variable.</li> </ul> <p>For an example, see <a href="#">Use Python script to join a list</a>.</p>
Execute script	<p>Executes a script within the Python Script.</p> <ul style="list-style-type: none"> <li>• In the Python session field, specify a session name. Use the same session name from the Open action.</li> <li>• Optional: In the Assign the output to variable field, specify the variable.</li> </ul> <p>For an example, see <a href="#">Create a PDF using Python script</a>.</p>
Open	<p>Opens a Python Script file.</p> <ul style="list-style-type: none"> <li>• In the Python session field, specify a session name. Use this same session name for other Python Script actions.</li> <li>• In the Python field, choose one of the following options:           <ul style="list-style-type: none"> <li>• In the Import existing file option, select an existing Python Script file. Note: If you are uploading a script from a file on your desktop, the file and any dependencies must be in a standalone folder. When you select a file for upload, all files and folders at the same folder level are uploaded.</li> <li>• In the Manual input option, enter the Python Script.</li> </ul> </li> <li>• In the Python runtime version field, select 2 or 3 to specify the runtime version.</li> </ul>

Action	Description
	Note: Select the runtime version based on the version of Python installed on the device.

## Resources

To learn more, see [Training - Write inline scripts in a bot](#). This course introduces you to new features in the Python Script command.

Note: You must log in with a registered A-People Community [account](#) to access the course.

- [Use Python script to join a list](#)  
Build a bot that uses a Python function to print the message `Go Be Great!`, the Automation Anywhere Enterprise version of `Hello World`. In this example, the bot combines a list of string values and prints them to a message box.
- [Create a PDF using Python script](#)  
Build a bot that runs Python script from a file and generates a PDF.

## Use Python script to join a list

Build a bot that uses a Python function to print the message `Go Be Great!`, the Automation Anywhere Enterprise version of `Hello World`. In this example, the bot combines a list of string values and prints them to a message box.

## Prerequisites

To run Python script from Enterprise A2019, you must already have the latest version of Python 3.x installed on your device.

## Procedure

1. Open a new bot:
  - a) From the Automation Anywhere Enterprise web interface, select Bots > My bots.
  - b) Click Create TaskBot.
  - c) Enter a bot name.
  - d) Accept the default folder location `\Bots\`.  
To change where your bot is stored, click Choose and follow the prompts.
  - e) Click Create and Edit.
2. Create a variable to hold the list values:
  - a) Click the Create variable icon.
  - b) Enter `lArgument` in the Name field.  
Recommendation: Prefix the variable name with a lowercase character to indicate the variable data type.

[Variable naming recommendations](#)

- c) Select the List type and String subtype.
- d) In the Default value field, enter the following values:

- a) Value at 0:

Go

- b) Value at 1:

Be

- c) Value at 2:

Great

- d) Value at 3:

!

- e) Click Create.

3. Provide the script with a Python Script > Open action:
  - a) Double-click or drag the Python Script > Open.
  - b) Select the Manual input option.
  - c) Copy and paste the following text into the Enter script here field.

```
def data ( str ) :
    x = " ".join( str )
    return x
```

- d) Click Apply.

4. Use a Python Script > Execute function action to tell the bot to run the script:
  - a) Double-click or drag Python Script > Execute function.
  - b) Enter
    - data
 in the Enter name of function to be executed field.
  - c) Select the lArgument variable from the Arguments to the function drop-down list.
  - d) Create the variable sOutput for the Assign the output to variable field.
  - e) Click Apply.
5. Insert a Message box action to hold the Python function output:
  - a) Double-click or drag the Message box > Message box action.
  - b) In the Enter the message to display field, select and insert the variable sOutput.
  - c) Select the Close message box after option. Retain the default value of 5 seconds in the field.
  - d) Click Apply.
6. Close the script execution session with a Python Script > Close action:
  - a) Double-click or drag Python Script > Close.
  - b) Click Save.
7. Click the Run icon.
 

The bot generates a message box with the text Go Be Great!. After 5 seconds, the message box disappears.

## Create a PDF using Python script

Build a bot that runs Python script from a file and generates a PDF.

## Prerequisites

- To run Python script from Enterprise A2019, you must already have the latest version of Python 3.x installed on your device.
- This example uses the FPDF library to generate the PDF. Therefore, install it before you start building this bot by copying and pasting the following line in the Windows command prompt:

```
pip install fpdf
```

See [FPDF project page](#).

## Procedure

1. Create a file in a standalone folder to hold the following Python script:  
Insert your device username in the angle brackets.

```
from fpdf import FPDF
pdf = FPDF()
pdf.add_page()
pdf.set_font("Arial", size=12)
pdf.cell(200, 10, txt="Go Be Great!", ln=1, align="C")
pdf.output("C:/Users/<yourusername>/Desktop/sample_demo.pdf")
```

Note: If you are uploading a script from a file on your desktop, the file and any dependencies must be in a standalone folder. When you select a file for upload, all files and folders at the same folder level are uploaded.

2. Open a new bot:
  - a) From the Automation Anywhere Enterprise web interface, select Bots > My bots.
  - b) Click Create TaskBot.
  - c) Enter a bot name.
  - d) Accept the default folder location \Bots\.

To change where your bot is stored, click Choose and follow the prompts.

  - e) Click Create and Edit.
3. Provide the script with a Python Script > Open action:
  - a) Double-click or drag the Python Script > Open.
  - b) Select the Import existing file option.
  - c) Click Browse to select the .py file you created in Step 1.
  - d) Click Apply.
4. Use a Python Script > Execute script action to tell the bot to run the script:
  - a) Double-click or drag Python Script > Execute script.
  - b) Click Apply.
5. Close the script execution session with a Python Script > Close action:
  - a) Double-click or drag Python Script > Close.
  - b) Click Save.
6. Click the Run icon.  
The bot creates a PDF on the desktop with the text `Go Be Great!`.  
If the bot does not create a file, change the last line in the Python script:

```
pdf.output(r"C:/Users/<yourusername>/Desktop/sample_demo.pdf")
```

## Recorder package

Use the Capture action from the Recorder package to capture an interaction with a user interface (UI) object such as a text box, button, table, menu, radio button, combo box, check box, list view, link, tree, and page tab. The Capture action replaces the Object Cloning command from Version 11.3.

The Capture action enables you to add a single interaction when building your bot (such as if you missed a step when recording a process). See [Using the Capture action](#). For a common use case, see [Enter data into a web form from a worksheet](#).

To record a process consisting of multiple steps, see [Recording a task](#). To learn more about Universal Recorder capabilities, see [Universal Recorder overview](#).

## Secure Recording

When secure recording mode is enabled, bots do not capture object images or values. This ensures that sensitive data is not stored in the bots.

When you record a task in secure recording mode, the Preview window temporarily shows the captured area. This image is deleted after you click Apply and navigate away from the action editor window.

A user with admin privileges must enable this setting. See [Settings](#).

## Object properties

When you select an object to capture, the Universal Recorder collects data on the object's properties in order to identify the object during runtime. You can do the following with the object properties:

- Verify that the captured object properties match your intended object.

For example, when capturing a table from a website, ensure the Control Type and HTML Tag values are TABLE.

- Retrieve a property value.

For example, to retrieve the text of a link, use the Get property action and enter

```
HTML InnerText
```

in the Property name field.

- Specify the properties to search for. Mark the only the properties that will always resolve to true.

## Object controls and actions

There are two parts to an interaction: the UI object and the action. Each UI object has a list of possible actions. For example, when you capture a click in a textbox, you can select either to click the link or to retrieve the link text. Refer to the table below for the objects and their actions.

Table 1. UI object controls and possible actions

Object Control	Action
Button	<ul style="list-style-type: none"> <li>• Get property: retrieves the value of the specified object property (such as the link text) and optionally saves it to a variable. For more information, see the Object properties section.</li> <li>• Click</li> <li>• Left click: use this action if the Click action does not work during Runtime.</li> <li>• Right click</li> <li>• Double click</li> </ul>
Checkbox	<ul style="list-style-type: none"> <li>• Get property: retrieves the value of the specified object property (such as the link text) and optionally saves it to a variable. For more information, see the Object properties section.</li> <li>• Get status: retrieves whether the check box is selected. Returns checked or unchecked.</li> <li>• Check</li> <li>• Uncheck</li> <li>• Toggle: switches the check box to the opposite status. For example, if the check box is checked, use the Toggle action to clear it.</li> <li>• Left click: use this action if the Toggle action does not work during Runtime.</li> <li>• Right click</li> <li>• Double click</li> </ul>
Client	<ul style="list-style-type: none"> <li>• Get property: retrieves the value of the specified object property (such as the link text) and optionally saves it to a variable. For more information, see the Object properties section.</li> <li>• Get all children names</li> <li>• Get all children values</li> <li>• Set text: enters text into the UI object. It supports <a href="#">Credentials</a>.</li> <li>• Click</li> <li>• Left click: use this action if the Click action does not work during Runtime.</li> <li>• Right click</li> <li>• Double click</li> </ul>
ComboBox (appears as a drop-down list)	<ul style="list-style-type: none"> <li>• Get property: retrieves the value of the specified object property (such as the link text) and optionally saves it to a variable. For more information, see the Object properties section.</li> <li>• Get total items: retrieves the number of items in the box.</li> <li>• Get selected index: retrieves the item's position in the menu. Note: Item index counts start at 1.</li> <li>• Get selected text: retrieves the data from the selected item.</li> <li>• Select item by index: selects the item that is located at the specified index. Note: Item index counts start at 1.</li> <li>• Select item by text selects the item that matches the specified text. Note: The Assign value field is case-sensitive.</li> <li>• Expand</li> </ul>

Object Control	Action
	<ul style="list-style-type: none"> <li>• Click</li> <li>• Left click: use this action if the Click action does not work during Runtime.</li> <li>• Right click</li> <li>• Double click</li> </ul>
Link	<ul style="list-style-type: none"> <li>• Get property: retrieves the value of the specified object property (such as the link text) and optionally saves it to a variable. For more information, see the Object properties section.</li> <li>• Click</li> <li>• Left click: use this action if the Click action does not work during Runtime.</li> <li>• Right click</li> <li>• Double click</li> </ul>
ListView	<ul style="list-style-type: none"> <li>• Get property: retrieves the value of the specified object property (such as the link text) and optionally saves it to a variable. For more information, see the Object properties section.</li> <li>• Get total items: retrieves the number of items in the list.</li> <li>• Get selected index: retrieves the item's position in the menu. Note: Item index counts start at 1.</li> <li>• Get selected text: retrieves the data from the selected item.</li> <li>• Select item by index: selects the item that is located at the specified index. Note: Item index counts start at 1.</li> <li>• Select item by text selects the item that matches the specified text. Note: The Assign value field is case-sensitive.</li> <li>• Click</li> <li>• Left click: use this action if the Click action does not work during Runtime.</li> <li>• Right click</li> <li>• Double click</li> </ul>
Menu	<ul style="list-style-type: none"> <li>• Get property: retrieves the value of the specified object property (such as the link text) and optionally saves it to a variable. For more information, see the Object properties section.</li> <li>• Get total items: retrieves the number of items in the menu.</li> <li>• Get selected index: retrieves the item's position in the menu. Note: Item index counts start at 1.</li> <li>• Get selected text: retrieves the data from the selected item.</li> <li>• Select item by index: selects the item that is located at the specified index. Note: Item index counts start at 1.</li> <li>• Select item by text selects the item that matches the specified text. Note: The Assign value field is case-sensitive.</li> <li>• Left click: use this action if the Click action does not work during Runtime.</li> <li>• Right click</li> </ul>

Object Control	Action
Page Tab	<ul style="list-style-type: none"> <li>• Get property: retrieves the value of the specified object property (such as the link text) and optionally saves it to a variable. For more information, see the Object properties section.</li> <li>• Get total items: retrieves the number of items in the tab.</li> <li>• Get selected index: retrieves the item's position in the menu. Note: Item index counts start at 1.</li> <li>• Get selected text: retrieves the data from the selected item.</li> <li>• Select item by index: selects the item that is located at the specified index. Note: Item index counts start at 1.</li> <li>• Select item by text selects the item that matches the specified text. Note: The Assign value field is case-sensitive.</li> <li>• Left click: use this action if the Click action does not work during Runtime.</li> <li>• Right click</li> <li>• Double click</li> </ul>
RadioButton	<ul style="list-style-type: none"> <li>• Get property: retrieves the value of the specified object property (such as the link text) and optionally saves it to a variable. For more information, see the Object properties section.</li> <li>• Get status: retrieves whether the radio button is selected. Returns selected or deselected.</li> <li>• Select</li> <li>• Left click: use this action if the Select action does not work during Runtime.</li> <li>• Right click</li> <li>• Double click</li> </ul>
Table	<ul style="list-style-type: none"> <li>• Get property: retrieves the value of the specified object property (such as the link text) and optionally saves it to a variable. For more information, see the Object properties section.</li> <li>• Get table: retrieves the table data and saves it to a Table variable. For more information, see <a href="#">Extract data from a web table and save it to a file</a>.</li> <li>• Get cell text by index: retrieves the data in the specified cell located at the row and column index. Note: Row and column index counts start at 0. For example, to retrieve the data in cell A2, enter  <div>0</div> in the Row field and  <div>1</div> in the Column field.</li> <li>• Get cell index by text: retrieves the index of the cell containing the specified text. Note: The Cell Text field is case-sensitive.</li> <li>• Get total rows: retrieves the number of rows that contain values.</li> <li>• Get total columns: retrieves the number of columns that contain values.</li> <li>• Set cell by index: enters text into the cell located at the row and column index. Note: Row and column index counts start at 0. For example, to enter text into cell A2, enter  <div>0</div> </li> </ul>



Object Control	Action
	<p>in the Row field and 1 in the Column field.</p> <ul style="list-style-type: none"> <li>Set cell by text: enters text into the cell containing the specified text. Note: The Find Text field is case-sensitive.</li> <li>Left click: use this action if the Click action does not work during Runtime.</li> <li>Right click</li> <li>Double click</li> </ul>
TextBox	<ul style="list-style-type: none"> <li>Get property: retrieves the value of the specified object property (such as the link text) and optionally saves it to a variable. For more information, see the Object properties section.</li> <li>Set text: enters text into the UI object. It supports <a href="#">Credentials</a>.</li> <li>Append text: adds text to the end of existing text in the field, instead of overwriting it.</li> <li>Click</li> <li>Left click: use this action if the Click action does not work during Runtime.</li> <li>Right click</li> <li>Double click</li> </ul>
Tree (Role is 10 but control is displayed as ListView)	<ul style="list-style-type: none"> <li>Get property: retrieves the value of the specified object property (such as the link text) and optionally saves it to a variable. For more information, see the Object properties section.</li> <li>Get total items: retrieves the number of items in the tree.</li> <li>Get selected index: retrieves the item's position in the menu. Note: Item index counts start at 1.</li> <li>Get selected text: retrieves the data from the selected item.</li> <li>Select item by index: selects the item that is located at the specified index. Note: Item index counts start at 1.</li> <li>Select item by text selects the item that matches the specified text. Note: The Assign value field is case-sensitive.</li> <li>Left click: use this action if the Click action does not work during Runtime.</li> <li>Right click</li> <li>Double click</li> </ul>

## Using the Capture action

Use the Capture action to capture a single interaction (click, keystroke, or mouse movement) with an object control including a text box, button, table, menu, radio button, combo box, check box, list view, link, tree, or page tab.

### Prerequisites

- Configure device display and font scale to 100%.

- If you are automating a task using a browser, configure the zoom level to 100%.

Considerations when recording a task:

- If you are automating a task using a browser, do not use autofill to enter values into fields.
- Use mouse clicks, keystrokes, and shortcuts when possible.
- Record the task at low speed.
- Avoid dragging windows during the recording process.
- Avoid clicking on applications that are not part of the process you are recording and automating.

## Procedure

To record a single interaction with an object control, follow these steps:

1. Double-click or drag Recorder > Capture.
2. In the Select window field, choose the Window or Variable tab.
  - Click the Window tab to select the application window from the list.

Select the Currently active window, Desktop, or Taskbar, or a window from the list of Available windows.

  - Click the Variable tab to insert an existing Window variable to specify the title of the application window you want to use.
3. Optional: If you have selected the Window tab, insert a wildcard character (\*) in the window title that is subject to change, such as for online invoices.

For example,

Sample\* - Google Chrome

Note: During runtime, verify that the TaskBot identifies the correct window.

4. Click Capture object.  
The selected window appears.
5. Click the object control, for example, a button, form field, or a table.  
A box highlights the control when you mouse over it.  
Note: If the box does not appear and you are capturing in the Google Chrome browser, verify that you have enabled the Automation Anywhere Google Chrome extension. See [Supported browsers for Enterprise A2019](#).
6. Verify that the Control Type matches your intended object.  
For example, when capturing a table from a website, ensure the Control Type and HTML Tag values are TABLE.  
If the Control Type does not match your intended object, recapture the object control.  
Note: When it is not possible to capture a UI element with the Recorder, use the [Image Recognition package](#). For example, when the Recorder cannot capture a Submit button as a BUTTON control type.
7. Review the Object properties list.  
The bot will use the selected properties to identify the object control during runtime.  
Ensure that only the properties that do not change are selected.
8. Select the Action from the drop-down list.

If the selected Action supports background processes, a Run in the background check box is available to enable this process. If the desired Action does not appear in the drop-down list, recapture the object. See [Background processing](#)

For the full list of possible actions by object, see [Object controls and actions](#).

9. Enter a wait time for the object control to appear.
10. Optional: Assign the output to a variable.
11. Click Apply.

Watch the following video on how to use the Capture action:

Using Capture action

## Extract data from a web table and save it to a file

Build a bot to open a browser window to the NASDAQ website, extract the data from a table, and write it to a CSV file on your desktop. This example uses actions from the Browser, Data Table, Recorder, and Window packages.

To extract data from a table, do the following steps:

### Procedure

1. Open a new bot:
  - a) From the Automation Anywhere Enterprise web interface, select Bots > My bots.
  - b) Click Create TaskBot.
  - c) Enter a bot name.
  - d) Accept the default folder location \Bots\.

To change where your bot is stored, click Choose and follow the prompts.

  - e) Click Create and Edit.
2. Open a browser window to the web page from which you will extract the table.
  - a) Double-click or drag the Browser > Launch website action.
  - b) In the URL field, enter  
`https://old.nasdaq.com/`
  - c) Specify the Internet Explorer browser.

Note: It is recommended to use Internet Explorer because it reliably launches the website in a new window, even if there is already an open window. Other browsers might launch the website in a new tab if there is an open window.

  - d) Click Apply.
  - e) Click Save.
  - f) Click Run.

The bot opens the window.
3. Specify the table.
  - a) Double-click or drag the Recorder > Capture action.
  - b) Click the Window tab and select the Daily Stock Market Overview window from the drop-down list.
  - c) Click Capture object.

The Daily Stock Market Overview window activates.

  - d) Hover over the table below the Stock Market Overview heading.

An orange box appears, surrounding the table.

  - e) Click the table.

The Object Processing message box appears.

  - f) Return to the Enterprise Control Room.
  - g) In the Object properties table, verify the Control Type is TABLE.

If it is not, click Recapture object.

  - h) From the Action drop-down list, select Get table.
  - i) In the Assign output to variable field, create the `tNasdaqTable`.
  - j) Click Apply.

The Daily Stock Market Overview window is saved as the variable `window-1`.

4. Specify the file where to save the data.
    - a) Double-click or drag the Data Table > Write to file action.
    - b) From the Data table name list, select `tNasdaqTable`.
    - c) Provide a file path to create a CSV file.  
For example, `C:\Users\<username>\Desktop\NasdaqTable.csv`.
    - d) Select the Create folders/files if it doesn't exist option.
    - e) Select to overwrite the existing file.
    - f) Click Apply.
  5. Close the Daily Stock Market Overview window.
    - a) Double-click or drag the Window > Close action.
    - b) Select the Variable tab and insert `window-1`.
    - c) Click Apply.
  6. Click Save.
  7. Click Run.
- The bot creates a CSV file on the desktop with data on seven indexes, their values, and net change.

## Enter data into a web form from a worksheet

In this example, you build a bot to enter multiple rows of data from an XLSX sheet into a web form. Use actions from the Excel advanced, Loop, and Recorder packages.

To retrieve values from an Excel file and input them into a web form, do the following:

### Procedure

1. Open a new bot:
  - a) From the Automation Anywhere Enterprise web interface, select Bots > My bots.
  - b) Click Create TaskBot.
  - c) Enter a bot name.
  - d) Accept the default folder location `\Bots\`.  
To change where your bot is stored, click Choose and follow the prompts.
  - e) Click Create and Edit.
2. Open the Excel file.
  - a) Double-click or drag the Excel advanced > Open action.
  - b) Enter a session name.
  - c) Select the Excel file.
  - d) Mark the Sheet contains a header option.
  - e) Click Apply.

By marking the Sheet contains a header option, you enable the bot to search for the column by the header name during run time.
3. Launch the website.
  - a) Double-click or drag the Browser > Launch website action.
  - b) Enter the website URL.
  - c) Click Apply.
4. Retrieve the worksheet values and store them in a Table variable.
  - a) Double-click or drag the Excel advanced > Get multiple cells action.
  - b) Enter the same session name you used in the Excel advanced > Open action.
  - c) Select All rows from the drop-down list.
  - d) Create a Table variable using the icon to the right of the Assign to variable drop-down list.

- e) Click Apply.
5. Instruct the bot to process the data row by row.
  - a) Double-click or drag the Loop action.
  - b) Select the For each row in table iterator.
  - c) Select the same Table variable that you used in Get multiple cells.
  - d) Create a Record variable using the icon to the right of the Assign to variable drop-down list.
  - e) Click Apply.

The Record variable holds all of the values for one row. With each iteration of the Loop, the bot retrieves the values of the next row and stores them in the Record variable, overwriting the values from the previous row.
6. Map the first column header to the web form textbox.
  - a) Double-click or drag the Recorder > Capture action.
  - b) Select the same window you opened with the Launch website action.
  - c) Click Capture object.
  - d) Hover over the textbox until a red outline appears.
  - e) Click the textbox.
  - f) Return to the Enterprise Control Room.
  - g) Verify that the Control Type value is TEXTBOX.
  - h) Select Set text from the Action drop-down list.
  - i) In the Keystrokes field, insert the same Record variable that you used in the Loop.
  - j) Select the By name option and copy-paste the first column header into the field.
  - k) Click Apply.
7. Repeat the sub-steps in step 5 to map the other columns, with the following differences:
  - a) Instead of searching for the browser window title, insert the Window variable generated by the Recorder.
  - b) When inserting the Record variable in the Keystrokes field, copy-paste the subsequent column header into the "By name" field.
8. Capture the Submit button.
  - a) Double-click or drag the Recorder > Capture action.
  - b) Insert the Window variable generated by the Recorder.
  - c) Click Capture object.
  - d) Verify that the Control Type value is BUTTON.
  - e) Select the Click action.
  - f) Click Apply.
9. Click Save.

## REST Web Service package

The REST Web Service methods (actions) work similarly to a representational state transfer (REST) client, using the REST request-response model.

## Before you start

You need to have the following information to add to the selected method to create and send a REST request and receive a response. Not all parameters are required for all methods.

- URI: A unique address for an API resource.
- Authentication Mode: There are 3 supported authentication modes.
  - Basic: text strings can be used for basic authentication.
  - Logged-in AD User: Active Directory (AD) users that are authorized to access the related API are authenticated through AD. No credentials are needed in the request.

- Windows NT LAN Manager (NTLM) Authentication (AD User): a challenge/response authentication method that allows clients to provide their user name and password as encrypted credentials or plain text. It is recommended that you use credentials that are stored in the Automation Anywhere Credential Vault. NTLM requires a user name in the `domain\username` or `\username` format.
- Header: Not all methods require a header. Headers represent the meta-data associated with the request.
- Content type: When a header contains a content type it defines the content negotiation between the client and the server.
- Output variable: Response output is captured in a dictionary variable. A dictionary variable is a key value pair. Use the response header name as key to capture the header value, or "Body" as the key to capture the response body.

## Actions in the REST Web Service package

Action	Description
DELETE Method	Removes a resource that is identified in the URL or request body.
GET Method	Retrieves information identified by parameters included in the URI. There is no Content type for the get method because all the parameters are passed as part of the URI. Limitations and characteristics of the GET method: <ul style="list-style-type: none"> <li>• URI length is limited to 2048 characters</li> <li>• All parameters are passed in the URI</li> <li>• The GET method exposes data that is in the URI, making it less secure than the POST method.</li> <li>• GET does not change any data, making it safe for all users regardless of authorization.</li> </ul>
POST Method	<ul style="list-style-type: none"> <li>• Parameters passed in request body.</li> <li>• There is no limit on length for a request body.</li> <li>• POST can be used to update or delete data.</li> </ul>
PUT Method	Updates a record based on parameters passed in the URI or body.

## SAP package

The SAP package contains actions to automate tasks and processes on a SAP application.

The SAP package enables you to perform the following tasks:

- Reduce the time required to combine data from disparate systems.
- Eliminate human error and increase efficiency.
- Increase the number of transactions processed.
- Deliver real-time information to various stakeholders.

- Enhance decision-making through comprehensive reports.  
Note: You can use the actions in the SAP package with any version of SAP ECC and Oracle Database.

## Prerequisites

- Open the Enterprise client and SAP GUI application in the same mode. For example, if you open the Enterprise client in Administrator mode, you must also open the SAP GUI application in Administrator mode.
- Log in to the SAP GUI before capturing objects because the SAP Logon screen is not supported for object capture.
- Enable GUI scripting and accessibility.

See [Enabling Scripting on the Client Side](#), [Enabling Scripting on the Server Side](#), and [Enabling Accessibility settings](#).

- Ensure that one of the following SAP GUI for Windows is installed on the devices that you use to automate SAP-related tasks and to run these tasks:
  - SAP GUI 750 with patch 9
  - SAP GUI 760 with patch 0
  - SAP GUI 760 with patch 5
- Install a Scripting Tracker or a similar tool to retrieve the field path of the SAP object.

## Before you start

Perform the following actions within the SAP package as part of using the set of available actions:

1. Establish a connection with the SAP application using the Connect action.

Log in to the SAP application using the SAP GUI application. Then use the Connect action from the SAP package to use this connection and assign a session name. Use this same session name for the other actions.

2. Use the actions to automate a task.
3. After you have automated all the SAP-related tasks, terminate the connection to the SAP application using the Disconnect action.

## Actions in the SAP package

The SAP package includes the following actions:

Action	Description
Check/uncheck checkbox	<p>Selects or clears a check box.</p> <ul style="list-style-type: none"> <li>• In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>• In the Field path field, specify the location or a string variable that contains the location of the object.</li> <li>• In the Action option, select Check, Uncheck, or Toggle.</li> </ul>
Click	<p>Performs a click operation.</p> <ul style="list-style-type: none"> <li>• In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> </ul>

Action	Description
	<ul style="list-style-type: none"> <li>In the Field path field, specify the location or a string variable that contains the location of the object.</li> </ul>
Click menu	<p>Clicks a menu item by text or index.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>In the Field path field, specify the location or a string variable that contains the location of the object.</li> <li>In the Select option, select Name or Index to specify the name or index of the menu item.</li> </ul>
Connect	See <a href="#">Using Connect action for SAP</a> .
Disconnect	Terminates the connection to the SAP application. In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.
Double click	<p>Performs a double-click operation.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>In the Field path field, specify the location or a string variable that contains the location of the object.</li> </ul>
Expand	<p>Expands the item by text or index.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>In the Field path field, specify the location or a string variable that contains the location of the object.</li> <li>In the Select option, select Text or Index to specify the text or index of the item.</li> </ul>
Export table	See <a href="#">Using Export table action</a> .
Get cell count	<p>Gets number of cells in a table or grid.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>In the Field path field, specify the location or a string variable that contains the location of the object.</li> <li>In the Assign the output to a variable field, specify the variable.</li> </ul>
Get children name	<p>Gets children control names.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>In the Field path field, specify the location or a string variable that contains the location of the object.</li> </ul>



Action	Description
	<ul style="list-style-type: none"> <li>In the Assign the output to list variable field, specify the list variable.</li> </ul>
Get children text	<p>Gets the text associated with children control.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>In the Field path field, specify the location or a string variable that contains the location of the object.</li> <li>In the Assign the output to list variable field, specify the list variable.</li> </ul>
Get column count	<p>Gets number of columns in a table or grid.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>In the Field path field, specify the location or a string variable that contains the location of the object.</li> <li>In the Assign the output to a variable field, specify the variable.</li> </ul>
Get row count	<p>Gets number of rows in a table or grid.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>In the Field path field, specify the location or a string variable that contains the location of the object.</li> <li>In the Assign the output to a variable field, specify the variable.</li> </ul>
Get selected item	<p>Gets selected item index from a combo box, page tab, or a tree view control.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>In the Field path field, specify the location or a string variable that contains the location of the object.</li> <li>In the Assign the output to a variable field, specify the variable.</li> </ul>
Get status	<p>Gets status of a radio button or check box.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>In the Field path field, specify the location or a string variable that contains the location of the object.</li> <li>In the Assign the output to a variable field, specify the variable.</li> </ul>
Get table cell index	<p>Gets table cell index for text.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>In the Field path field, specify the location or a string variable that contains the location of the object.</li> </ul>

Action	Description
	<ul style="list-style-type: none"> <li>• In the Search text field, specify the text.</li> <li>• Select the Case sensitive search option to specify a case-sensitive search.</li> <li>• Select the All occurrences check box.</li> <li>• In the Assign the output to a variable field, specify the variable.</li> </ul>
Get table cell text	<p>Gets table cell text by index.</p> <ul style="list-style-type: none"> <li>• In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>• In the Field path field, specify the location or a string variable that contains the location of the object.</li> <li>• In the Row field, specify the number of row from which you want to get the text.</li> <li>• In the Column field, specify the number of column from which you want to get the text.</li> <li>• In the Assign the output to a variable field, specify the variable.</li> </ul>
Get text	<p>Gets text from a text box, label, or status bar.</p> <ul style="list-style-type: none"> <li>• In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>• In the Field path field, specify the location or a string variable that contains the location of the object.</li> <li>• In the Assign the output to a variable field, specify the variable.</li> </ul>
Get total item	<p>Gets total item from a combo box, page tab, or tree view control.</p> <ul style="list-style-type: none"> <li>• In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>• In the Field path field, specify the location or a string variable that contains the location of the object.</li> <li>• In the Assign the output to a variable field, specify the variable.</li> </ul>
Left click	<p>Performs a left-click operation.</p> <ul style="list-style-type: none"> <li>• In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>• In the Field path field, specify the location or a string variable that contains the location of the object.</li> </ul>
Right click	<p>Performs a right-click operation.</p> <ul style="list-style-type: none"> <li>• In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>• In the Field path field, specify the location or a string variable that contains the location of the object.</li> </ul>
Select item	Selects an item by text or index.

Action	Description
	<ul style="list-style-type: none"> <li>In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>In the Field path field, specify the location or a string variable that contains the location of the object.</li> <li>In the Select option, select Text or Index to specify the text or index of the item.</li> </ul>
Select radio option	<p>Selects a radio button.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>In the Field path field, specify the location or a string variable that contains the location of the object.</li> </ul>
Send virtual key	<p>Sends a virtual key.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>In the Select list, select a key.</li> </ul> <p>See <a href="#">Virtual keys in SAP GUI</a>.</p>
Set table cell text	See <a href="#">Set table cell text</a> .
Set text	<p>Sets the text in an editable field.</p> <ul style="list-style-type: none"> <li>In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.</li> <li>In the Field path field, specify the location or a string variable that contains the location of the object.</li> <li>In the Field value field, specify the text.</li> <li>Select the Append text check box.</li> </ul>

- [Using Connect action for SAP](#)  
Use the Connect action to establish a connection with a SAP system that you want to use to automate SAP-related tasks. This must be the first action you use to automate an SAP-related task.
- [Using Export table action](#)  
Use the Export table action to export a table to a datatable or CSV file.
- [Using Set table cell text action](#)  
Use the Set table cell text action to set the text in a specific cell of a table or a grid.

## Using Connect action for SAP

Use the Connect action to establish a connection with a SAP system that you want to use to automate SAP-related tasks. This must be the first action you use to automate an SAP-related task.

## Procedure

1. In the Actions palette, double-click or drag the Connect action from the SAP package.
2. In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.
3. Select an option to specify the connection type:
  - a) If you have selected Automatic, no additional information is required to connect to the SAP application. You must be already logged on to SAP logon application as the system uses the available SAP session to connect to a SAP system.
4. Click Apply.

Related tasks

[Using Export table action](#)

[Using Set table cell text action](#)

Related reference

[SAP package](#)

## Using Export table action

Use the Export table action to export a table to a datatable or CSV file.

## Procedure

To export a table, follow these steps:

1. In the Actions palette, double-click or drag the Export table action from the SAP package.
2. In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.
3. In the Field path field, specify the location or a string variable that contains the location of the object.
4. In the Export As option, select CSV or Database to specify the export option of the table cell:

Choice	Steps
CSV	a) In the File Path field, specify the location or file variable. b) Optional: In the Encoding field, specify the value. c) Select the Export data with header check box to export with header. d) In the When saving field, select Append to existing log file or Overwrite existing log file to append the log file or overwrite the content within the log file.
Database	In the Database field, specify the value.

5. Click Apply.
6. Click Save.

# Using Set table cell text action

Use the Set table cell text action to set the text in a specific cell of a table or a grid.

## Procedure

To set a value in a table cell, follow these steps:

1. In the Actions palette, double-click or drag the Set table cell text action from the SAP package.
2. In the Session name field, enter the session name you used to connect to the SAP application in the Connect action.
3. In the Field path field, specify the location or a string variable that contains the location of the object.
4. In the Select option, select Text or Index to specify the text or index of the table cell:
  - Text: In the Find Text field, specify the value.

The system searches for the cell that contains the value.

- Index: In the Row and Column fields, specify the value.
5. In the Set Text field, enter the text to set the table cell.
  6. Click Apply.
  7. Click Save.

## Screen package

Use the Screen package to automate the process of capturing screenshots. Using the actions in this package, you can capture an area of an application window, the entire computer screen, or an active open window and save it in a specified location in an image format.

## Actions in the Screen package

The Screen package includes the following actions:

Action	Description
Capture area	<p>Captures specified area of an open application.</p> <p>See <a href="#">Using Capture area action</a></p>
Capture desktop	<p>Captures an image of the full desktop.</p> <ul style="list-style-type: none"> <li>• In the File path to save image field, specify the path where you want to save the captured image.</li> </ul> <p>The following file extensions are supported: png, bmp, jpeg, tiff, gif, and wmf.</p> <ul style="list-style-type: none"> <li>• Select Overwrite file check box to overwrite an existing file with the same name.</li> </ul>

Action	Description
Capture window	Captures screenshots of an open application window.  See <a href="#">Using Capture window action</a>

## Secure Recording

When secure recording mode is enabled, bots do not capture object images or values. This ensures that sensitive data is not stored in the bots.

When you record a task in secure recording mode, the Preview window temporarily shows the captured area. This image is deleted after you click Apply and navigate away from the action editor window.

A user with admin privileges must enable this setting. See [Settings](#).

- [Using Capture area action](#)  
Use the Capture area action in the Screen package to capture a screenshot of the application window area.
- [Using Capture window action](#)  
Use the Capture window action of the Screen package to capture an open application window.

## Using Capture area action

Use the Capture area action in the Screen package to capture a screenshot of the application window area.

### Procedure

To capture a specified area of an application window, perform these steps:

1. In the Actions palette, double-click or drag the Capture area action from the Screen package.
2. In the Select window field, choose the Window or Variable tab.
  - Click the Window tab to select the application window from the list.

Select the Currently active window, Desktop, or Taskbar, or a window from the list of Available windows.

  - Click the Variable tab to insert an existing Window variable to specify the title of the application window you want to use.
3. Optional: If you have selected the Window tab, insert a wildcard character (\*) in the window title that is subject to change, such as for online invoices.

For example,

`Sample* - Google Chrome`

Note: During runtime, verify that the TaskBot identifies the correct window.

4. Click the Capture region.  
The pixel coordinates of the captured area are displayed in the X, Y, Width, and Height fields. You can modify these values.
5. Browse to select the File path to save image.  
The following file extensions are supported: png, bmp, jpeg, tiff, gif, and wmf.

6. Click Overwrite file to replace an existing file with the same name.
7. Click Apply.

## Using Capture window action

Use the Capture window action of the Screen package to capture an open application window.

### Procedure

To capture an application window, perform these steps:

1. In the Actions palette, double-click or drag the Capture window action from the Screen package.
2. In the Select window field, choose the Window or Variable tab.
  - Click the Window tab to select the application window from the list.

Select the Currently active window, Desktop, or Taskbar, or a window from the list of Available windows.

  - Click the Variable tab to insert an existing Window variable to specify the title of the application window you want to use.
3. Optional: If you have selected the Window tab, insert a wildcard character (\*) in the window title that is subject to change, such as for online invoices.  
For example,  

Sample\* - Google Chrome

.

Note: During runtime, verify that the TaskBot identifies the correct window.
4. Browse to select the File path to save image.  
The following file extensions are supported: png, bmp, jpeg, tiff, gif, and wmf.
5. Click Overwrite file to replace an existing file with the same name.
6. Click Apply, and then click Save.

### SNMP package

The SNMP package allows you to automate network management tasks, such as retrieving and modifying data, and sending notification messages.

Simple Network Management Protocol (SNMP) is used to find the network management component on one or more computers and the managed component on multiple network devices.

The Automation Anywhere SNMP action offers powerful network management. Using this action, users can easily monitor network devices configured with SNMP agent software. Network devices such as servers, workstations, printers, routers, bridges, and hubs, as well as services such as Dynamic Host Configuration Protocol (DHCP) or Windows Internet Name Service (WINS) can be monitored.

### Actions in the SNMP package

The SNMP package includes the following actions:

Action	Description
Get	Retrieves data from an SNMP agent and assigns the values to a string variable
Get next	Browses entire hierarchy of management objects and assigns the return value to a variable. Users can save the result in a log file. The default file is SNMPLog.txt. This action supports ASCII and Hexagonal data types.
Send trap	<p>Send trap messages to the SNMP Manager from an SNMP Agent. These messages are sent by an SNMP Agent whenever certain events occur, such as a system restart. You must specify the Trap Type:</p> <ul style="list-style-type: none"> <li>• Cold start: Occurs when the SNMP agent initializes its configuration tables.</li> <li>• Warm start: Occurs when the SNMP agent re-initializes its configuration tables.</li> <li>• Link down: Occurs when the state of a network adapter on the SNMP agent changes from up to down.</li> <li>• Link up: Occurs when the state of a network adapter on SNMP agent changes from down to up.</li> <li>• Authentication fail: Occurs when the SNMP agent receives a message from an SNMP manager with an invalid community name.</li> <li>• EGP neighbour loss: Occurs when the SNMP agent cannot communicate with its Exterior Gateway Protocol (EGP) peer.</li> <li>• Enterprise specific: Occurs when specific error conditions and error codes are defined in the system. The user provides a Trap Specific Number for this option.</li> </ul>
Set	<p>Modifies data on an SNMP agent. You must specify the Value type.</p> <p>Select the Value Type from the drop-down list and enter the set value.</p>
Walk	<p>Retrieves a sub-tree of management values using the Get Next action.</p> <p>You must specify the Walk Type:</p> <ul style="list-style-type: none"> <li>• Within: All variables in the sub-tree below the given OID are queried.</li> <li>• All: Displays the variables that are equal to the specified OID for the given tree type structure.</li> </ul>

Related reference

[Active Directory package](#)

[Printer package](#)

## SOAP web service package

Use the SOAP web service action from the SOAP web service package to access and exchange information between two systems in XML format.

With this action, you can:



- Consume SOAP web services written to provide structured data for further business processing, such as currency conversion, weather reports, and language translation.
- Connect to different existing applications and different platforms, irrespective of any underlying infrastructure requirements.

The SOAP Web Service establishes complete interoperability between clients/applications and the internet, supporting XML-based open standards, such as Web Services Description Language (WSDL), Simple Object Access Protocol (SOAP), and Universal Description Discovery and Integration (UDDI).

- [Using the SOAP web service action](#)  
Use the SOAP web service action to make a call. The steps below include an example call to an online calculator application.

## Using the SOAP web service action

Use the SOAP web service action to make a call. The steps below include an example call to an online calculator application.

### Procedure

To make a SOAP web services call, do the following steps:

1. Double-click or drag the SOAP web services action.
2. Select a connection method:
  - To connect using a URI, enter the URI. For example,  
`http://www.dneonline.com/calculator.asmx?WSDL`
  - To connect using a file, select a file from the Control Room or the Desktop, or insert a variable.
3. Optional: Enter the SOAP endpoint in the Address location field.
4. In the Service field, enter the service name.  
This field identifies the collection of ports supported by the web service. It holds the `@service name` value for an a SOAP web service call.  
For example,  
`Calculator`
5. In the Port field, enter the endpoint to connect with.  
This field holds the `@binding name` value for a SOAP web service call.  
For example,  
`CalculatorSoap`
6. Select the SOAP version of the endpoint that you specified in the Port field.  
Choose from 1.1 or 1.2.
7. In the Operation field, enter the name of the service function for the endpoint that you specified in the Port field..  
This field holds the `@operation name` value for a SOAP web service call.  
For example,  
`Add`
8. Select either the Operation parameters or Raw data parameters option to provide the parameter details.

- If you select Operation parameters, enter the name and value of each parameter.

For example, enter

intA

in the first Name field and

10

in the first Value field, and

intB

in the second Name field and

20

in the second Value field.

- If you select Raw data parameters, enter the XML.
9. Select either the No Authentication or Basic option from the Authentication Mode list.
    - If you select No Authentication, proceed to the next step.
    - If you select Basic, provide credentials or enter values in the Username and Password fields.
  10. Optional: Provide a Client certificate by selecting a file from the Control Room or your desktop, or insert a variable.
  11. Optional: Provide custom headers.
  12. Optional: Provide a file to save the XML output.
  13. Select either the Complete response or the Selected response option to narrow the response scope.
    - If you select Complete response, proceed to the next step.
    - If you select Selected response, do the following:
      - a) Provide the XPath expression.  
For example,

```
/*[local-name()='Envelope' and namespace-uri()='http://schemas.xml
soap.org/soap/envelope/']
/*[local-name()='Body' and namespace-uri()='http://schemas.xmlsoap
.org/soap/envelope/']
/*[local-name()='AddResponse' and namespace-uri()='http://tempuri.
org/']
/*[local-name()='AddResult' and namespace-uri()='http://tempuri.or
g/']
```

- b) Select a section of the XML output: Values, Inner XML, or Outer XML.
  - c) If you select Values, choose a delimiter option: Pipe or Semicolon.
14. Select a variable to store the XML output.  
For example, prompt-assignment.
  15. Click Apply.

## Next steps

Verify the SOAP web service action output by inserting a Message box action with variable prompt-assignment in the body message. When you run the bot, the Message box should show 30.

## Step package

The Step package groups various actions together and runs them in a specific order. You can provide a relevant name for a step to identify the operation performed by the actions included in that step.

## Action in the Step package

The Step package includes the following action:

Action	Description
Step	<p>Runs a sequence of actions.</p> <ul style="list-style-type: none"> <li>• Creates a container for actions without impacting the bot run.</li> <li>• Configures actions within the Step action.</li> <li>• Runs the arranged actions in a sequential order.</li> <li>• Groups various actions for better management.</li> <li>• Optional: In the Title field, specify the title.</li> </ul>

Watch the following video on how to use the Step actions:

Using Step actions

## String package

Use the String package to perform various operations such as comparing two strings, retrieving the string length, or converting a string to uppercase or lowercase.

## Actions in the String package

The String package includes the following actions:

Action	Description
Assign	<p>Assigns or concatenates strings.</p> <ul style="list-style-type: none"> <li>• In the Select the source string variable(s)/value field, specify the variable.</li> <li>• In the Select the destination string variable field, specify the variable.</li> </ul>
Compare	<p>Compares two strings.</p> <ul style="list-style-type: none"> <li>• In the Source string field, specify the source string.</li> <li>• In the Compare to string field, specify the source string.</li> <li>• In the When comparing field, select an option:             <ul style="list-style-type: none"> <li>• Match case: Matches capitalization.</li> <li>• Do not match case: Does not match capitalization.</li> </ul> </li> <li>• In the Assign the output to variable list, specify the Boolean variable.</li> </ul>

Action	Description
Extract text	See <a href="#">Using Extract text action</a> .
Find	See <a href="#">Using Find action</a> .
Length	Retrieves the string length. <ul style="list-style-type: none"> <li>In the Source string field, specify the source string.</li> <li>In the Assign the output to variable list, specify the number variable.</li> </ul>
Lowercase	Converts the source string to lowercase. <ul style="list-style-type: none"> <li>In the Source string field, specify the source string.</li> <li>In the Assign the output to variable list, specify the variable.</li> </ul>
Replace	See <a href="#">Using Replace action</a> .
Reverse	Reverses the source string. <ul style="list-style-type: none"> <li>In the Source string field, specify the source string.</li> <li>In the Assign the output to variable list, specify the variable.</li> </ul>
Split	See <a href="#">Using Split action</a> .
Substring	Extracts a substring from a string. <ul style="list-style-type: none"> <li>In the Source string field, specify the source string.</li> <li>In the Start from field, specify the starting point.</li> <li>Optional: In the Length field, specify the length.</li> <li>In the Assign the output to variable list, specify the variable.</li> </ul>
To boolean	Converts a string to a Boolean. <ul style="list-style-type: none"> <li>In the Source string field, specify the source string.</li> <li>In the Assign the output to variable list, specify the Boolean variable.</li> </ul>
To number	Converts a string to a number. This action supports positive, negative, and decimal numbers. Note: If the string contains commas, remove them by using the Replace action to replace each comma with an empty string. <ul style="list-style-type: none"> <li>In the Source string field, specify the source string.</li> <li>In the Assign the output to variable list, specify the number variable.</li> </ul>
Trim	Trims blanks and whitespaces from a string. <ul style="list-style-type: none"> <li>In the Source string field, specify the source string.</li> <li>In Trim from the beginning and Trim from the end, select options.</li> <li>In the Assign the output to variable list, specify the variable.</li> </ul>

Action	Description
Uppercase	<p>Converts the source string to uppercase.</p> <ul style="list-style-type: none"> <li>• In the Source string field, specify the source string.</li> <li>• In the Assign the output to variable list, specify the variable.</li> </ul>

Watch the following video on how to use the String actions:

Using String actions

- [Extract text](#)  
The Extract text action enables you to extract text from a source string. You can extract text before, after, or between two strings.
- [Using Find action](#)  
Use the Find action to find a substring within the specified string.
- [Using Replace action](#)  
Use the Replace action to find a piece of text from the source string and replace it with another text.
- [Using Split action](#)  
Use the Split action to split the specified string into multiple strings and store the output in a list variable.

Related tasks

[Using Extract text action](#)

[Using Find action](#)

[Using Replace action](#)

[Using Split action](#)

## Extract text

The Extract text action enables you to extract text from a source string. You can extract text before, after, or between two strings.

### Using the Before Option

This option enables you to extract the entire text followed by the value you have provided in the Start after text field. You can also extract the text based on the occurrence of the value you have provided in the field.

For example, `a:123a:123b:123c:` is the source string, you can specify ':' in the Start after text field and '2' in the Occurrence field to extract the string followed by the second occurrence of ':' in the source string. In this example, the extracted text is '123b:123c:'.

### Using the After Option

This option enables you to extract the entire text preceding the value you have provided in the End before text field. You can also extract the text based on the occurrence of the value you have provided in the field.

For example, `a:123a:123b:123c:` is the source string, you can specify ':' in the End before text field and '3' in the Occurrence field to extract the string preceding the second occurrence of ':' in the source string. In this example, the extracted text is 'a:123a:123b:'.

## Using the Before and/or after option

This option enables you to extract the text between the values you have provided in the Start after text and the End before text fields. You can also apply the AND or OR logical operator. You can select the AND operator to extract text only when both conditions specified for the Before and After options are satisfied. Otherwise, you can select the OR operator to extract text only when either of the conditions specified for the Before or the After option is satisfied.

For example, `a:123a:123b:123c:` is the source string, you can specify `'.'` in the Start after text and End before text fields and `'2'` in the Occurrence fields to extract the string after the second and before the fourth occurrence of `'.'` in the source string. In this example, the extracted text is `'123b:123c'`.

## Using Extract text action

Use the Extract text action to extract a range of text using logical operators from the source string you have specified.

To extract a substring from the specified source string, do the following:

### Procedure

1. In the Actions palette, double-click or drag the Extract text action from the String package.
2. In the Source string field, specify the source string.
3. In the Before option, select to extract the entire text followed by the value you have provided in the Start after text field.
  - a) In the Start after text field, enter a string to use as the starting point.
  - b) In the Occurrence field, enter a value to specify the number of occurrences.  
 For example, if the source string is `This is a test string which is used to extract specific sub-string` and you want to extract the entire text after `This`. To extract the required text, you must enter `This` in the Start after text field and `1` in the Occurrence field. This indicates to the system to extract the text that is available after the first occurrence of `This` in the source string.
4. In the Before or after option, select to extract the text between the values provided in the Start after text and the End before text fields.
  - a) In the Occurrence field, enter a value to specify the number of occurrences.  
 For example, if the source string is `This is a test string which is used to extract specific sub-string` and you want to extract the entire text after `This`. To extract the required text, you must enter `This` in the Start after text field and `1` in the Occurrence field. This indicates to the system to extract the text that is available after the first occurrence of `This` in the source string.
  - b) In the OR or AND field, select one of the options:
    - c) OR: Select to extract text if either of the values specified in the Start after text or the End before text field are available in the source string.
    - d) AND: Select to extract text if both the values specified in the Start after text and the End before text fields are available in the source string.
  - e) In the End before text field, specify a string to use as the endpoint for extracting the text.
  - f) In the Occurrence field, enter a value to specify the number of occurrences of the string you have provided in the End before text field.  
 For example, if the source string is `This is a test string which is used to extract specific substring` and you want to extract the entire text before `specific`. To extract the required text, you must enter `specific` in the End before text field and `1` in the Occurrence field. This

indicates to the system to extract the text available before the first occurrence of `specific` in the source string.

5. In the After option, select to extract the entire text preceding the value you have provided in the End before text field.

a) In the End before text field, specify a string to use as the endpoint for extracting the text.

b) In the Occurrence field, enter a value to specify the number of occurrences of the string you have provided in the End before text field.

For example, if the source string is `This is a test string which is used to extract specific substring` and you want to extract the entire text before `specific`. To extract the required text, you must enter `specific` in the End before text field and 1 in the Occurrence field. This indicates to the system to extract the text available before the first occurrence of `specific` in the source string.

6. In the If no match found, return field, select one of the options:
  - Source String: Return the source string.
  - Empty (null) String: Return the null string if no match is found.
7. In the Number of characters to get field, select one of the options:
  - All: Extracts all characters from the source string.
  - Only: Specify the number of characters to extract from the source string.
8. In the Trim the extracted text (remove blank spaces) check box, select to remove blank spaces from the extracted text.
9. In the Remove Enter from the extracted text check box, select to remove `Enter` from the extracted text.
10. In the Assign the output to variable list, specify the variable.
11. Click Apply.
12. Click Save.

## Using Find action

Use the Find action to find a substring within the specified string.

This action also enables you to perform a search based on a regular expression. A regular expression is a sequence of characters that define a search pattern. For example, to find all email addresses in the source string, specify the following as a regular expression: `\b[A-Z0-9._%]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b`.

Note: To search for strings that contain the dollar sign (\$), you must enter the sign twice. Otherwise, the bot interprets the dollar sign as a regular expression character.

## Procedure

To find a substring from the specified string, follow these steps:

1. In the Actions palette, double-click or drag the Find action from the String package.
2. In the Source string field, specify the source string.
3. In the Find string field, specify the substring.
4. In the When finding field, select one of the options:
  - Match case: Matches the case of the text.
  - Do not match case: Does not match the case of the text.
5. In the The "find string" is field, select one of the options:
  - A regular expression: The substring is a regular expression
  - Not a regular expression: The substring is not a regular expression.
6. Optional: In the Start from field, specify the starting point.

For example, you want to replace Red in the source string with Blue in a paragraph. Red is in 10 instances in the paragraph and you want to replace only the third occurrence. Enter

3

in the Start from field to identify the third occurrence.

7. In the Assign the output to variable list, specify the number variable.
8. Click Apply.
9. Click Save.

## Using Replace action

Use the Replace action to find a piece of text from the source string and replace it with another text.

A regular expression is a sequence of characters that define a search pattern. For example, to find all email addresses in the source string, specify the following as a regular expression: `\b[A-Z0-9._%]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b`.

### Procedure

To replace text from the specified string, follow these steps:

1. In the Actions palette, double-click or drag the Replace action from the String package.
2. In the Source string field, specify the source string.
3. In the Find string field, specify the substring.
4. In the When finding field, select one of the options:
  - Match case: Matches the case of the text.
  - Do not match case: Does not match the case of the text.
5. In the The "find string" is field, select one of the options:
  - A regular expression: The substring is a regular expression
  - Not a regular expression: The substring is not a regular expression.
6. Optional: In the Start from field, specify the starting point.  
 For example, you want to replace Red in the source string with Blue in a paragraph. Red is in 10 instances in the paragraph and you want to replace only the third occurrence. Enter  
 3  
 in the Start from field to identify the third occurrence.
7. Optional: In the Count field, specify the number of times the find text must be replaced.  
 For example, you want to replace Red in the source string with Blue in a paragraph. Red is in 10 instances in the paragraph; however, you want to replace Red from only the third occurrence to the sixth occurrence. Enter  
 3  
 in the Start from field and  
 4  
 in the Count field to replace the third occurrence four times.
8. In the Replace with field, specify the text to replace the string.
9. In the Assign the output to variable list, specify the variable.
10. Click Apply.
11. Click Save.



# Using Split action

Use the Split action to split the specified string into multiple strings and store the output in a list variable.

## Procedure

To split a string into multiple strings, perform these steps:

1. In the Actions palette, double-click or drag the Split action from the String package.
2. In the Source string field, specify the source string.
3. In the Delimiter field, specify the character to split the string.  
For example, comma (,), semicolon (;), pipe (|), or slash (/ \).
4. In the Delimiter is field, select one of the options:
  - Case sensitive: The delimiter is case-sensitive.
  - Not case sensitive: The delimiter is not case-sensitive.
5. In the Split into substrings field, select one of the options:
  - All possible: Splits the source string into as many substrings as possible.

For example, if the original string is a,b,c,d, each character becomes a substring.

- Only: Limits the number of substrings.

For example, if the original string is a,b,c,d, and you enter 3, the output is three strings: a, b, and c,d.

6. In the Assign the output to variable list, specify the list variable.
7. Click Apply.
8. Click Save.

## System package

Use the actions in the System package to automate locking, logging off, restarting, and shutting down the computer. Use these actions at the end of a task.

The System package includes the following actions:

Action	Description
Lock computer	Locks the computer.
Logoff	Logs the user off the computer.
Restart	Restarts the computer.
Shutdown	Shuts down the computer.

Related tasks

[Using Log To File action](#)

Related reference

[Application package](#)

## Task Bot package

Use the Run, Pause, and Stop actions in the Task Bot package to manage running one or more child bots from a parent bot or with a third-party software using an API.

Build bots of shorter automation sequences and run them from a parent bot. For example, build a `Login` child bot to enter the username and password into a web form and click submit. The `Login` child bot accepts credentials from the parent bot and returns a success message, and then the parent bot can call the subsequent `CreateInvoice` child bot. The `Login` child bot should also contain error handling logic in case the credentials are incorrect.

This practice of building smaller, self-contained bots enables a user to reuse the bot logic in a greater number of tasks, and makes error handling and troubleshooting easier. This also decreases the amount of time spent building and maintaining bots. As a result, an organization is able to rapidly scale their automation initiatives. The `Login` child bot from this example can be reused in any task that involves providing credentials to a login page on a website.

In addition, running child bots from a parent bot offers greater control over data in the following ways:

- Users can configure the variables to control the direction in which information can be exchanged by selecting from the following options:
  - Use as input: The variable holds a value that can be passed from a parent bot or third-party software to a child bot.
  - Use as output: The variable holds a value that can be passed from a child bot to the parent bot or third-party software.
  - Both: The value can be passed in both directions.
  - Neither: The variable is confined to this bot; it cannot be shared across other bots.
- See [Create a variable](#)
- Other users can reuse child bots without viewing their contents.

The Task Bot package includes the following actions:

Action	Description
Pause	<p>Temporarily pauses the running bot. Use the Pause action to modify data or to verify the status of a relevant component.</p> <p>When the bot reaches the Pause action during run time, a Resume button appears. You must click Resume for the bot to continue to the next action.</p>
Run	<p>Runs the selected child bot with the specified input values and saves the output values to a dictionary variable. Each key in the dictionary is the variable name and corresponding value from the child bot.</p> <p>Note:</p> <ul style="list-style-type: none"> <li>• You must know the variable names from the child bot to extract them from the parent bot. The interface does not automatically import the variable names to the parent bot.</li> <li>• Use a Dictionary variable to hold the output of the Run action in order to make the parent bot more versatile. This way a parent bot can handle different child bots regardless of whether they return a single value or several values.</li> </ul> <p>For an example, see <a href="#">Using the Run action</a>.</p>

Action	Description
Stop	<p>Stops the running bot.</p> <p>For example, use the Stop action to terminate the bot if a condition is met, such as if the bot encounters a file larger than 100 MB.</p>

## Using the Run action

The Run action from the TaskBot package enables you to run and pass values to one or more child bots. In this example, you use the Run action to pass two values from a parent bot to the child bot; the child bot adds the values and passes the sum back to the parent bot.

This example demonstrates the following:

### Modularization

Modularization separates a TaskBot into several bots, where each bot is built with all the actions necessary to perform one specific function of the greater task.

In this example, you build a single child bot to add the values. You can build several child bots, with each performing a different mathematical operation, and edit the parent bot to call whichever one you want to use.

### Reusability

Reusability enables the user to build a bot once, and then use that bot to automate many processes.

The child bot contains only the actions necessary to add the values and print them to a Message Box. Also, the bot adds two variables instead of hard-coded values. As a result, this bot can be reused for any task that involves an addition operation.

### Data control

Because the child bot accepts and returns values only through the use of variables, it does not hold data. This minimizes the chance of data leakage.

## Procedure

To run a TaskBot from the current task, perform these steps:

Build the child bot.

1. Open a new bot.
  - a) From Automation Anywhere web interface, select Bots > My bots.
  - b) Click Create a bot.
  - c) Enter the bot name  

AddChildBot
  - d) Enter the folder location  

\Bots\TaskBotExample
  - e) To change where your bot is stored, click Choose and follow the prompts.
  - e) Click Create and Edit.
2. Create the following variables:
  - v1: number type; use as input

- `v2`: number type; use as input
- `nSum`: number type
- `sSum`: string type; use as output

#### Create a variable

- Use a Number > Assign action to perform the mathematical operation.
  - Double-click or drag the Number > Assign action.
  - In the Select the source string variable/ value field, enter the following expression:
 

```
$v1$ + $v2$
```
  - Select `nSum` from the Select destination number variable list.
  - Click Apply.
- Use the Number > To string action to convert the value so it can be printed to a Message Box.
  - Double-click or drag the Number > To string action.
  - In the Enter a number field, enter the following expression:
 

```
$nSum$
```
  - Select `sSum` from the Assign output to variable list.
  - Click Apply.
- Click Save.
- Click Close.

Build the parent bot.

- Open a new bot.
  - From the web interface, select Bots > My bots.
  - Click Create a bot.
  - Enter the bot name
 

```
MathBot
```
  - Enter the folder location
 

```
\Bots\TaskBotExample
```
  - To change where your bot is stored, click Choose and follow the prompts.
  - Click Create and Edit.
- Create a Dictionary variable of Any subtype named `dSums` to accept the values passed from the child bot. Use the Any subtype to enable the parent bot to accept String, Number, or Boolean type values.
- Insert a Task Bot > Run action to specify the values for the mathematical operation and the output variable to hold the sum.
  - Double-click or drag the Task Bot > Run action.
  - Click Browse to navigate to `Bots\TaskBotExample\AddChildBot`.
  - Enter the following values in the Input values fields:
    - d) `v1`: 12
    - e) `v2`: 54
  - Select `dSums` from the Assign output to variable list.
 

Note: Use a Dictionary variable to hold the output of the Run action in order to make the parent bot more versatile. This way a parent bot can handle different child bots regardless of whether they return a single value or several values.
  - Click Apply.
- Use a Message Box to retrieve and print the sum.
 

The variable `sSum` retrieved from the child bot is a key in the Dictionary variable `dSums`.

Note: You must know the variable names from the child bot to extract them from the parent bot. The interface does not automatically import the variable names to the parent bot.

  - Double-click or drag the Message box action.

b) Enter

```
$dSums { sSum } $
```

in the Message to display field.

c) Click Apply.

4. Click Save.

5. Click Run.

The bot runs and the Message Box appears containing the sum 66.

## Terminal Emulator package

The Terminal Emulator package contains actions that enable you to connect to and automate tasks on another machine. Use these actions to access and control operations on a remote machine. For example, you can run applications and access files on a different operating system.

The Terminal Emulator enables a machine to connect to and communicate with another machine using a command-line or graphical interface. The Terminal Emulator uses the Telnet or SSH protocol to communicate with other machines.

Note: The Terminal Emulator supports only TN3270E and TN5250E terminal types.

## Before you start

Perform the following actions within the Terminal Emulator package as part of using the set of available actions:

1. Establish a connection with a host machine using the Connect action.

You must first establish a connection with a host machine to automate any Terminal Emulator related task. When establishing the connection, specify the details of the host machine and associate it with a session name. Use the session name provided in this action in the other actions so that you do not have to provide details of the host machine in those actions again. See [Using Connect action](#).

Important: If you use any other action from this package before establishing the connection, you will encounter an error.

2. Use the Terminal Emulator actions to automate a task. For example, use the Get field action to retrieve the value from a particular field.
3. After you have automated all the Terminal Emulator related tasks, terminate the connection to the host machine using the Disconnect action.

## Actions in the Terminal Emulator package

The Terminal Emulator package includes the following actions:

Action	Description
Clear terminal	Clears the screen of the terminal. Specify the Terminal emulator session name that you used to establish a connection with the terminal using the Connect action.
Connect	See <a href="#">Using Connect action</a> .

Action	Description
Disconnect	Terminates the connection with the terminal. Specify the Terminal emulator session name that you used to establish a connection with the terminal using the Connect action.
Get all fields	<p>Retrieves the values of all fields and assigns them to a table variable.</p> <ul style="list-style-type: none"> <li>Specify the Terminal emulator session name that you used to establish a connection with the terminal using the Connect action.</li> <li>Select the table variable to store the retrieved data from the Assign the value to an existing table variable list.</li> </ul> <p>Note: You can use this action with the TN3270E, TN5250E, ANSI, and VT100 terminal types.</p>
Get field	<p>Retrieves the value of a field based on the index or name of the field and assigns it to a string variable.</p> <ul style="list-style-type: none"> <li>Specify the Terminal emulator session name that you used to establish a connection with the terminal using the Connect action.</li> <li>Select the By index option to retrieve the value of a field based on its index, or select the By name option to retrieve the value of a field based on its name.</li> <li>Select a string variable to store the retrieved data from the Assign the value to an existing variable list.</li> </ul> <p>Note: You can use this action with the TN3270E, TN5250E, ANSI, and VT100 terminal types.</p>
Get text	See <a href="#">Using Get text action</a> .
Hide terminal	Hides the terminal screen. This action enables you to hide the terminal screen when the Show terminal window option from the Connect action is selected. It is useful when you do not want to display the terminal screen when a bot is performing certain tasks. Specify the Terminal emulator session name that you used to establish a connection with the terminal using the Connect action.
Send key	See <a href="#">Using Send key action</a> .
Send text	See <a href="#">Using Send text action</a> .
Set cursor position	<p>Sets the position of the cursor on the screen of the terminal.</p> <ul style="list-style-type: none"> <li>Specify the Terminal emulator session name that you used to establish a connection with the terminal using the Connect action.</li> <li>Specify the row number in which to set the cursor in the Set row field.</li> <li>Specify the column number in which to set the cursor in the Set column field.</li> </ul> <p>Note: You can enter a value of 1 through 999 to specify the row and column number where the cursor is set.</p>
Set field	See <a href="#">Using Set field action</a> .
Show terminal	Shows the terminal screen. This action enables you to show the terminal screen when the Show terminal window option from the Connect action is not selected.

Action	Description
	It is useful when you want to display the terminal screen when a bot is performing certain tasks. You must specify Specify the Terminal emulator session name that you used to establish a connection with the terminal using the Connect action.
Wait	See <a href="#">Using Wait action</a> .

## Using Connect action

Use the Connect action to establish a connection with a host machine on which you want to automate a task. You can use this action to establish a connection with the TN3270E, TN5250E, ANSI, and VT100 terminal types.

This action enables you to specify the details of the host machine and associate it with a session name. Use the session name provided in this action in the other actions, so that you do not have to provide details of the host machine in those actions again.

Important: If you use any other action from this package before establishing the connection, you will encounter an error.

To establish a connection with a host machine, do the following:

### Procedure

1. Double-click or drag the Connect action from the Terminal Emulator node in the Actions palette.
2. Specify a name for the session in the Terminal emulator session name field.
3. Specify the Host name of the machine you want to connect to:
  - a) Click the Credential tab to select an entry from the Credential Vault.
  - b) Click the Variable tab to use a variable to specify the host name.
4. Enter a value in the Port field to specify the port you want to use to connect to the host.
5. Select an option from the Terminal type list to specify the terminal type you want to use:
  - ANSI
  - VT100
  - TN3270E
  - TN5250E

For ANSI terminal type:

- a) Select an option from the Connection type list to specify the type of connection you want to establish:
  - b) Telnet
  - c) SSH2

For SSH2 connection type:

- a) For User authentication:
  - b) User name: Click the Credential tab to select an entry from the Credential Vault. You can also click the String tab to enter a value manually.
  - c) For Password: Click the Credential tab to select an entry from the Credential Vault. You can also click the String tab to enter a value manually.
- d) For Key file authentication: Click one of the following options to specify the location of the file:
  - e) Control Room file: Enables you to select a PDF file that is available in a folder.
  - f) Desktop profile: Enables you to select a PDF file that is available on your device.
  - g) Variable: Enables you to specify the file variable that contains the location of the PDF file.
  - h) User name: Click the Credential tab to select an entry from the Credential Vault. You can also click the String tab to enter a value manually.

For VT100 terminal type:

- a) Select an option from the Connection type list to specify the type of connection you want to establish:
  - b) Telnet
  - c) SSH2

For SSH2 connection type:

- a) For User authentication:
  - b) User name: Click the Credential tab to select an entry from the Credential Vault. You can also click the String tab to enter a value manually.
  - c) For Password: Click the Credential tab to select an entry from the Credential Vault. You can also click the String tab to enter a value manually.
- d) For Key file authentication: Click one of the following options to specify the location of the file:
  - e) Control Room file: Enables you to select a PDF file that is available in a folder.
  - f) Desktop profile: Enables you to select a PDF file that is available on your device.
  - g) Variable: Enables you to specify the file variable that contains the location of the PDF file.
  - h) User name: Click the Credential tab to select an entry from the Credential Vault. You can also click the String tab to enter a value manually.

For TN3270E terminal type:

- a) Select an option from the Host name security list to specify the security you want to use:
  - b) NONE
  - c) SSL
  - d) TLS
- e) Select the Enable TN3270E support check box if you want to choose a connection method.
- f) Select an option from the Connection method list:
  - g) GENERIC: Enables you to specify the Device name.
  - h) SPECIFIC: Enables you to specify the Device name and Resource (LU) Name.

For TN5250E terminal type:

- a) Select an option from the Host name security list to specify the security you want to use:
  - b) NONE
  - c) SSL
  - d) TLS
- e) Select the Enable TN5250E support check box if you want to specify details about the device and resource.
- f) Enter a value in the Device name and Resource (LU) Name fields.

6. Select an option from the Terminal model list to specify the terminal workstation you want to connect to.
7. Select an option from the Code page list to specify the encoding you want to use for the terminal.
8. Select the Show terminal window check box to show the terminal window.
9. Select the Wait for the terminal prompt to appear while connected check box to wait for the terminal prompt to appear on the screen of the terminal after the connection is established.
  - a) Enter the text you want to display when you connect to the terminal in the Terminal prompt field.
  - b) Enter a value in the Wait time out field to specify the period in (milliseconds) the system must wait before the connection request is timed out.
10. In the Assign value to variable list, select a string variable.

## Using Get text action

Use the Get text action to retrieve text from the terminal and store it in a string variable. This action enables you to retrieve text from the last line, all lines, or a range of lines. You can use this action with the TN3270E, TN5250E, ANSI, and VT100 terminal types.

To retrieve text from the terminal, do the following:



## Procedure

1. Double-click or drag the Get text action from the Terminal Emulator node in the Actions palette.
2. Enter the name of the session that you have used to establish a connection with the terminal using the Connect action in the Terminal emulator session name field.
3. Select an option from the Get text list to specify the lines from which you want to retrieve text:
  - Last line: Retrieves text from the last line of the terminal.
  - All lines: Retrieves text from all lines of the terminal.
  - Lines from-to: Retrieves text from the specified range of lines of the terminal. You must enter values in the Start row and End row fields to specify the range.  
Note: You can specify a value of 1 through 999.
4. Select a string variable from the Assign the value to an existing variable list to assign the retrieved text to that variable.
5. In the Assign value to variable list, select a string variable.

## Using Send key action

Use the Send key action to send a key to the terminal. You can use these keys to perform various operations on the terminal. You can use this action with the TN3270E, TN5250E, ANSI, and VT100 terminal types.

To send a key to the terminal, do the following:

## Procedure

1. Double-click or drag the Send key action from the Terminal Emulator node in the Actions palette.
2. Enter the name of the session that you have used to establish a connection with the terminal using the Connect action in the Terminal emulator session name field.
3. Select an option from the Select key to be send list to specify the key you want to send.
4. Enter a value in the Delay after send key command field to specify the period the system must wait (in milliseconds) after the selected key is sent.
5. Select the Wait for text or prompt to appear check box if you want the system to wait for the prompt or certain text to appear on the terminal.
  - a) Select the Prompt option if you want the system to wait till the prompt appears on the terminal.
  - b) Select the Text option if you want the system to wait till the specific text appears on the terminal. You must provide the text you want to appear on the terminal in the Text value to be appear on screen field.
  - c) Select the Wait before send key for prompt or text appear check box if you want the system to wait for the prompt or the specified text to appear on the terminal before sending the selected key.
  - d) Select the Wait after send key for prompt or text appear check box if you want the system to wait for the prompt or the specified text to appear on the terminal after sending the selected key.
  - e) Enter a value in the Time out for prompt or text to appear field to specify the period (in seconds) the system must wait before the operation times out.
6. In the Assign value to variable list, select a string variable.

## Using Send text action

Use the Send text action to send text to the terminal. This action also enables you to send predefined keys after the text to perform various operations in the terminal. You can use this action with the TN3270E, TN5250E, ANSI, and VT100 terminal types.

To send a text to the terminal, do the following:

### Procedure

1. Double-click or drag the Send text action from the Terminal Emulator node in the Actions palette.
2. Enter the name of the session that you have used to establish a connection with the terminal using the Connect action in the Terminal emulator session name field.
3. Specify the Text you want to send to the terminal:
  - a) Click the Credential tab to select an entry from the Credential Vault.
  - b) Click the Variable tab to use a variable that contains the text you want to send.
4. Select the Send a key after sending the above text check box if you want to send a key after sending the text.
  - a) Select the key you want to send from the list.
  - b) Enter a value in the Delay after send key command field to specify the period (in milliseconds) the system must wait after sending the selected key.
5. Select the Wait for text or prompt to appear check box if you want the system to wait for the prompt or certain text to appear on the terminal.
  - a) Select the Prompt option if you want the system to wait till the prompt appears on the terminal.
  - b) Select the Text option if you want the system to wait till the specific text appears on the terminal. You must provide the text you want to appear on the terminal in the Text value to be appear on screen field.
  - c) Select the Wait before send key for prompt or text appear check box if you want the system to wait for the prompt or the specified text to appear on the terminal before sending the selected key.
  - d) Select the Wait after send key for prompt or text appear check box if you want the system to wait for the prompt or the specified text to appear on the terminal after sending the selected key.
  - e) Enter a value in the Time out for prompt or text to appear field to specify the period (in seconds) the system must wait before the operation times out.
6. In the Assign value to variable list, select a string variable.

## Using Set field action

Use the Set field action to set a value in a particular field in the terminal. This action also enables you to send predefined keys after the value to perform various operations in the terminal. You can use this action with the TN3270E, TN5250E, ANSI, and VT100 terminal types.

To set a value for a field in the terminal, do the following:

### Procedure

1. Double-click or drag the Set field action from the Terminal Emulator node in the Actions palette.
2. Enter the name of the session that you have used to establish a connection with the terminal using the Connect action in the Terminal emulator session name field.
3. Specify the field for which you want to set a value:

- a) Select the By index option to specify the field based on its index.  
The index value starts from zero. For example, if you want to set the value for the third field, you must specify 2 in this field.
  - b) Select the By name option to specify the location of the field.  
For example, if you want to set the value for a field in the fifth column of the third row, you must specify R3C5 in the field.
4. Specify the Plain text value you want to set in the specified field:
  - a) Click the Credential tab to select an entry from the Credential Vault.
  - b) Click the Variable tab to use a variable that contains the text you want to send.
5. Select the Send enter key after setting field check box if you want to send the enter key after setting the field.  
This option performs the operation similar to pressing the Enter key after setting the value in the field.
6. Select the Send a key after sending the above text check box if you want to send a key after sending the text.
  - a) Select the key you want to send from the list.
  - b) Enter a value in the Delay after send key command field to specify the period (in milliseconds) the system must wait after sending the selected key.
7. In the Assign value to variable list, select a string variable.

## Using Wait action

Use the Wait action to introduce a delay till specific conditions are met in the terminal. You can use this action to wait before executing the next action till the specific text appears on the screen, the cursor moves to the specified location, and so on. You can use this action with the TN3270E, TN5250E, ANSI, and VT100 terminal types.

To wait till specific conditions are met in the terminal, do the following:

### Procedure

1. Double-click or drag the Wait action from the Terminal Emulator node in the Actions palette.
2. Enter the name of the session that you have used to establish a connection with the terminal using the Connect action in the Terminal emulator session name field.
3. Select an option from the Terminal event list to specify a condition the system should wait till it is met.  
The following table provides information about the options available in the list and the input required for each option:

Option	Input required
Wait till text appears	Waits till the text you have specified in the Text field appears on the screen of the terminal.
Wait till text disappears	Waits till the text you have specified in the Text field disappears from the screen of the terminal.
Wait till cursor moves to position	Waits till the cursor moves to the position you have specified in the Move to row and Move to column fields. Note: You can enter a value of 1 through 999 to specify the row and column number where the cursor is set.
Wait till cursor moves out of position	Waits till the cursor moves out of the position you have specified in the Move out of row and Move out of column fields.

Option	Input required
	Note: You can enter a value of 1 through 999 to specify the row and column number where the cursor is set.
Wait till screen gets blank	No input required.
Wait till screen contains text	Waits till the text you have specified in the Text field is available on the screen of the terminal.
Wait till terminal prompt appears	No input required.
Wait till terminal ready state	No input required.

4. Enter a value in the How long you would like to wait? field to specify the period (in milliseconds) the system must wait before the operation times out.
5. In the Assign value to variable list, select a string variable.

## Trigger loop package

The Trigger loop package enables you to run a series of actions when a trigger event occurs. You can insert multiple trigger loops within a bot or nest one trigger loop within another trigger loop.

## Actions in the Trigger loop package

Action	Description
Trigger loop	Opens the loop. Note: Events within the Trigger loop follow a sequential order and any parallel events that occur during this time are queued.
Handle	Specifies the trigger event and runs the actions that are inside of the container when that event occurs. Drag this action to the right of the Trigger loop action (Flow view) or below the Trigger loop action (List view).
Break	Terminates the loop so that the bot continues with the actions below the loop. Drag this action inside of the Trigger Handle container. Note: You must close the loop with a Break action or else the bot will run indefinitely.

## Triggers in the Handle action

Trigger	Description	Options
File	Runs the actions when a file event occurs.  File triggers support the use of wildcard characters in the file name or extension.	<ul style="list-style-type: none"> <li>• File is created.</li> <li>• File is renamed.</li> <li>• File is modified.</li> <li>• File is deleted.</li> </ul>

Trigger	Description	Options
Folder	Runs the actions when a folder event occurs.	<ul style="list-style-type: none"> <li>Folder is created.</li> <li>Folder is renamed.</li> <li>Folder is modified.</li> <li>Folder is deleted.</li> </ul>
Form	Runs the actions when a form field is modified. Select an existing form and element.  <a href="#">Using interactive forms</a>	<ul style="list-style-type: none"> <li>On click: Button is clicked.</li> <li>Got focus: Input field is clicked.</li> <li>Lost focus: User leaves the input field.</li> <li>Value changed: Check box or radio button status is changed.</li> </ul>
Hot key	Runs the actions when specific keys are pressed.	<ul style="list-style-type: none"> <li>Modifiers: Ctrl, Shift, Alt, AltGr, and Win</li> <li>Keys: <ul style="list-style-type: none"> <li>Letters A-Z</li> <li>Numbers 0-9</li> </ul> </li> </ul>
Object	Runs the actions when an event occurs on the selected interface element. For example, when a user clicks a button.  Interface triggers for objects are available only for native Windows applications.	

## VBScript package

The VBScript package contains actions that enable VBScript functions in a task.

## Before you start

1. Open a VBScript file, or specify the script you want to run using the Open action. You must associate the details of the file or script you want to run with a session name. Use this same session name for other VBScript actions.
2. Use the Run function action to execute a function within the script or execute the entire script. You must use the VBScript session name you established in the previous step.
3. After you have executed the script, close the VBScript session.

Note: Windows operating system must be installed on the device on which you want to run the VBScript.

## Actions in the VBScript package

The VBScript package includes the following actions:

Note: The Open action must be the first action to use the VBScript in a task.

Action	Description
Close	<p>Closes the session.</p> <p>Specify the same session name from the Open action.</p>
Run function	<p>Runs a function within the VBScript.</p> <ul style="list-style-type: none"> <li>In the VBScript session field, specify a session name. Use the same session name from the Open action.</li> <li>Optional: Specify the function name to run and the arguments to pass to the function. Note: You can pass only a list variable as an argument for the function. You can use the list variable to pass multiple arguments of different data types such as Boolean, datetime, number, and string.</li> <li>Optional: In the Assign the output to variable field, specify the variable.</li> </ul>
Open	<p>Opens a VBScript file.</p> <ul style="list-style-type: none"> <li>In the VBScript session field, specify a session name. Use this same session name for other VBScript actions.</li> <li>In the VBScript, choose one of the following options: <ul style="list-style-type: none"> <li>In the Import existing file option, select an existing VBScript file. Note: If you are uploading a script from a file on your desktop, the file and any dependencies must be in a standalone folder. When you select a file for upload, all files and folders at the same folder level are uploaded.</li> <li>In the Manual input option, enter the VBScript.</li> </ul> </li> </ul>

## Resources

To learn more, see [Training - Write inline scripts in a bot](#). This course introduces you to write inline scripts within a command.

Note: You must log in with a registered A-People Community [account](#) to access course.

## Wait package

Use the actions in the Wait package to add a condition to wait for an application screen to change, or a separate window to open or close before proceeding to the next action.

## Actions in the Wait package

The Wait package includes the following actions:

Action	Description
Wait for condition	Makes the bot wait until a specific condition is true before executing the next action.

Action	Description
	<p>In the Wait till field, specify the condition to meet. For example, the wait condition can be based on whether an application is running, a folder or file exists, a variable matches the specified value, an application window exists, or a machine or server is running.</p> <p>This action offers the same conditions as the If package. See the <a href="#">If package</a>.</p>
Wait for screen change	See <a href="#">Using Wait for screen change action</a> .
Wait for window	See <a href="#">Using Wait for window action</a> .

## Using Wait for screen change action

Use the Wait for screen change action to wait for the content of a specific screen or the entire window to change before executing the next action.

For example, a Human Resources (HR) personnel automating leave-management tasks wants to log in to the HR portal. After entering the user credentials, this action enables the automation task to wait until the next screen loads.

### Procedure

1. In the Actions palette, double-click or drag the Wait for screen change action from the Wait package.
2. In the Screen change relative to field, choose Window or Screen.
3. Select an application window:

Option	Steps
Window	<p>From the drop-down list, select the window title from the open applications.</p> <ul style="list-style-type: none"> <li>• The Window title field displays the title of the application window you have selected. You can use the wildcard character in this field to enable the action to identify an application window with a similar title.</li> </ul> <p>Example: Enter Balance Sheet* to identify all open windows having title starting with Balance Sheet in the window title.</p> <ul style="list-style-type: none"> <li>• The Executable field displays the complete path of the executable file to identify the right window.</li> </ul>
Variable	Use a window variable to specify the application window you want to use.

4. Enter values (in pixels) in the Left, Top, Width, and Height fields to specify the dimensions of the screen.

The measurement starts from the top-left corner of the screen. Move vertically downward to indicate the height and horizontally from left to right to indicate the width.

5. In the How long to wait before comparing screens? field, specify the wait time (in seconds) to begin comparing the window or screen.

When a difference is detected, the comparison stops and the task proceeds to the next action.

6. In the How long to wait before screen comparing is stopped? field, specify the wait time (in seconds) for the screen to change.

If the screen does not change within the time you specified, the task proceeds with the next action.

7. Select the Throw an exception if the screen is not changed option to show an error message and terminate the bot if the screen does not change.

If the option is deselected and the screen does not change, the task proceeds with the next action.

8. Click Apply, and then click Save.

## Using Wait for window action

Use the Wait for window action to wait for the specified window to open or close before executing the next action.

For example, a Human Resources (HR) personnel automating leave-management tasks might require the attendance application to open before they enter the user credentials. This action enables the automation task to wait until the application window opens.

### Procedure

1. In the Actions palette, double-click or drag the Wait for window action from the Wait package.
2. In the Wait for window field, select the option you want the automation to wait for:
  - Wait for window to open
  - Wait for window to close
3. Select an application window:

Option	Steps
Window	<p>From the drop-down list, select the window title from the open applications.</p> <ul style="list-style-type: none"> <li>• The Window title field displays the title of the application window you have selected. You can use wildcard character in this field to enable the action to identify an application window with a similar title.</li> </ul> <p>Example: Enter Balance Sheet* to identify all open windows with a title that starts with Balance Sheet.</p> <ul style="list-style-type: none"> <li>• The Executable field displays the complete path of the executable file to identify the right window.</li> </ul>
Variable	Use a window variable to specify the application window you want to use.



4. In the How long you would like to wait for this condition to be true? field, specify the wait time (in seconds) for the window to open or close.  
If the window does not open or close within the time you specify, the task proceeds to execute the next set of actions.
5. Select the Throw an exception if wait for window is unsuccessful option to show an error message and terminate the bot if the window does not open or close within the specified amount of time.  
If the option is deselected and the window does not open or close within the specified amount of time, the task proceeds with the next action.
6. Click Apply, and then click Save.

## Window package

Use the Window package to automate tasks relating to the window.

Window is a data type available for storing application window titles. It is the only data type that can be used in all the actions of the Window package. Create a variable of data type Window and select or assign a window title. Use this variable in any number of actions within the automation task. If the application window title changes, then change the value assigned to the variable. The new application title is reflected in all the actions where the variable is used.

## Actions in the Window package

The Window package includes the following actions:

Action	Description
Activate	<p>Activates a window.</p> <ul style="list-style-type: none"> <li>In the Window field, select an option: <ul style="list-style-type: none"> <li>Window: Inserts a source window variable value.</li> </ul> </li> </ul> <p>Select the Currently active window, Desktop, or Taskbar, or a window from the list of Available windows.</p> <ul style="list-style-type: none"> <li>Variable: Inserts a user-defined variable value.</li> <li>In the Window title field, specify the title.</li> </ul>
Assign	<p>Assigns a source window variable's value to a designated window.</p> <ul style="list-style-type: none"> <li>In the Select the source Window variable/value field, select an option: <ul style="list-style-type: none"> <li>Window: Inserts a source window variable value.</li> <li>Variable: Inserts a user-defined variable value.</li> </ul> </li> <li>In the Window title field, specify the title.</li> <li>In the Select the destination window variable/value field, specify the variable.</li> </ul>
Close	<p>Closes the application window.</p> <ul style="list-style-type: none"> <li>In the Window field, select an option: <ul style="list-style-type: none"> <li>Window: Inserts a source window variable value.</li> </ul> </li> </ul>

Action	Description
	<p>Select the Currently active window, Desktop, or Taskbar, or a window from the list of Available windows.</p> <ul style="list-style-type: none"> <li>Variable: Inserts a user-defined variable value.</li> </ul>
Get active window title	Retrieves the title of the active window. In the Assign the window title to variable field, specify the variable.
Maximize	<p>Maximizes a window.</p> <ul style="list-style-type: none"> <li>In the Window field, select an option: <ul style="list-style-type: none"> <li>Window: Inserts a source window variable value.</li> </ul> </li> </ul> <p>Select the Currently active window, Desktop, or Taskbar, or a window from the list of Available windows.</p> <ul style="list-style-type: none"> <li>Variable: Inserts a user-defined variable value.</li> </ul>
Minimize	<p>Minimizes a window.</p> <ul style="list-style-type: none"> <li>In the Window field, select an option: <ul style="list-style-type: none"> <li>Window: Inserts a source window variable value.</li> </ul> </li> </ul> <p>Select the Currently active window, Desktop, or Taskbar, or a window from the list of Available windows.</p> <ul style="list-style-type: none"> <li>Variable: Inserts a user-defined variable value.</li> </ul>
Resize	<p>Resizes a window.</p> <ul style="list-style-type: none"> <li>In the Window field, select an option: <ul style="list-style-type: none"> <li>Window: Inserts a source window variable value.</li> </ul> </li> </ul> <p>Select the Currently active window, Desktop, or Taskbar, or a window from the list of Available windows.</p> <ul style="list-style-type: none"> <li>Variable: Inserts a user-defined variable value.</li> <li>In the Left, Top, Width, and Height fields, specify values for resizing the window.</li> </ul>

## Workload package

The Workload package enables you to insert work items in a queue for workload automation. It also enables data chaining between multiple queues. You can orchestrate multiple bots, and enable optimal device utilization through the queueing mechanism of workload management.

For example, a business process of Employee Salary Processing:

- A payroll queue can process the employee salaries, and a payroll bot can add work items to a finance queue.
- The finance queue in turn can release the funds to employees and a finance bot in turn adds work items in a HR queue to email employees on the salary processing completion.

You must have the Queue Owner or Participant privileges to view the list of queues in the Workload package.

## Actions in the Workload package

The Workload package includes the following action:

Action	Description
Insert work item	Allows you to insert a work item from an existing queue to another queue as part of a bot execution.

- [Using Insert work item action](#)

The Insert work item action provides you the flexibility to insert a work item to another queue as part of a bot execution.

## Using Insert work item action

The Insert work item action provides you the flexibility to insert a work item to another queue as part of a bot execution.

### Prerequisites

You must have the Queue Owner or Participant privileges to view the list of queues in the Workload package.

Use the Insert work item to manage complex workflows by configuring a bot to add work items from multiple queues. For example, use a bot to read a list of invoices from a .csv file and add the invoices due for payment to another bot that manages information of payments due on specific dates. The automation can be used in the scenarios where work items are part of a different system (Database, Excel) and the bot reads them and adds them to a queue or where one queue adds work items to another queue as part of data chaining.

### Procedure

1. In the My bots page, Create a bot or Edit an existing bot.
2. In the Actions palette, double-click or drag the Insert Work Item action from the Workload package.
3. In the Workload: Insert work item window, add parameters such as the Queue and attribute values:
  - a) Click Browse to select a queue name from the list of queues to insert as a work item.
  - b) Click Confirm to add the queue.
  - c) Start entering work item values in the attribute fields.

Note: These attributes are populated based on the work item template associated with the selected queue.

You can also assign a variable value to these attributes by pressing the F2 key and selecting a variable name from the list.

Tip: Use UTC format (YYYY-mm-dd 00:00:00) for Date data type.
4. Click Apply.

The action is added to the bot.

5. Add more data using the Insert Work Item action.

After the required data is added, Save the bot and Check in the bot to the Public folder for further processing and deployment.

## XML package

Extensible Markup Language (XML) is a markup language designed to store and transport data. Use the actions in the XML package to automate the processing of XML data generated from web services and cloud computing applications.

An XML document is structured as an ordered and labeled tree. Each node of the tree is an XML element and is written with an opening and closing tag. In the following example, custname and custid are nodes:

```
<customer>
    <custname>XYZ Corp</custname>
    <custid>A001</custid>
</customer>
```

XPath is a query language that uses path expressions to select nodes or node-sets in an XML document. XPath includes built-in functions for manipulation of string, numeric, Boolean, date and time, and so on.

## Before you start

Perform the following actions within the XMLpackage.

1. Start the XML session using the Start XML Session action. Use this session name for all corresponding actions.
2. Use the different actions available in the XML package to automate XML-related tasks.
3. Save the session using the Save XML Session action to assign the data to a file or String-type variable.
4. End the session using the End XML Session action to complete a task.

### Action in the XMLpackage

The XML includes the following :

Action	Description
Delete node	<p>Deletes a specific node from XML file.</p> <ul style="list-style-type: none"> <li>• Enter the session name or select an existing window variable used in the Start XML session action.</li> <li>• Enter the XPath expression for the node to be deleted or select an existing window variable.</li> <li>• Enter the attribute (optional)</li> </ul>
End session	<p>Closes an XML session.</p> <ul style="list-style-type: none"> <li>• Enter the session name or select an existing window variable used in the Start XML session.</li> </ul>

Action	Description
	<ul style="list-style-type: none"> <li>• Save and close the session.</li> </ul>
Execute XPath function	<p>Executes an XPath function and stores the results in a variable.</p> <ul style="list-style-type: none"> <li>• Enter the session name or select an existing window variable used in the Start XML session action.</li> <li>• Enter the XPath expression or select an existing window variable.</li> <li>• Assign a String-type variable to the output.</li> </ul>
Get multiple nodes	<p>Retrieves the value of multiple nodes.</p> <ul style="list-style-type: none"> <li>• Enter the session name or select an existing window variable used in the Start XML session action.</li> <li>• Enter the XPath expression to retrieve multiple nodes or select an existing window variable.</li> <li>• In the Get each node field, select if you require the Text value, XPath expression, or Specific attribute name of each node.</li> </ul>
Get single node	<p>Retrieves the value of a single node.</p> <ul style="list-style-type: none"> <li>• Enter the session name or select an existing window variable used in the Start XML session action.</li> <li>• Enter the XPath expression, and attribute (optional) to retrieve a node.</li> <li>• Assign the output to a String-type variable.</li> </ul>
Insert node	See <a href="#">Using Insert node action</a> .
Save session data	<p>Saves the XML session data to a file or variable of type String.</p> <ul style="list-style-type: none"> <li>• Enter the session name or select an existing window variable used in the Start XML session action.</li> <li>• To save the session data to a file, select the Write XML data option and specify a file path.</li> <li>• Select Overwrite to replace the existing file if a file with the same name exists in the location you have specified.</li> <li>• Assign the output to a String-type variable.</li> </ul>
Start session	<p>Creates a new XML session based on an XML file or specified text.</p> <ul style="list-style-type: none"> <li>• Start the XML session.</li> <li>• Enter the session name or select an existing window variable. Use this session name for all corresponding actions.</li> <li>• In the Data Source field, select either File or Text: <ul style="list-style-type: none"> <li>• File: Select from the Control Room file, Desktop file, or an existing Variable of File type.</li> <li>• Text: Specify the text name or select an existing window variable.</li> </ul> </li> <li>• Save the XML session.</li> </ul>

Action	Description
Update node	<p>Updates the value of a node.</p> <ul style="list-style-type: none"> <li>• Type the session name or select an existing window variable used in the Start XML session action.</li> <li>• Enter the XPath expression for the node to be updated.</li> <li>• Enter a New value for the node.</li> <li>• Select the Updates attribute(s) option to create a new Dictionary or to add an existing Variable of Dictionary-type.</li> </ul>
Validate XML document	<p>The tags and document structure are defined when the XML document is created. Use this action to validate an XML document.</p> <ul style="list-style-type: none"> <li>• Enter the session name or select an existing window variable used in the Start XML session.</li> <li>• Select the validation type from the following options: <ul style="list-style-type: none"> <li>• XML schema files (.xsd)</li> <li>• Specify the schema using a List or Variable. Create a new variable of type List or use one that already exists.</li> <li>• Internal Document Type Definitions (DTDs)</li> <li>• Well formed</li> </ul> </li> <li>• Assign the output to a variable using Assign the output (Valid or Invalid) to variable</li> </ul>

- [Using Insert node action](#)

Use this action to insert a node in an existing XML file and assign it to a value. Optionally, assign a name space and attributes to the node.

## Using Insert node action

Use this action to insert a node in an existing XML file and assign it to a value. Optionally, assign a name space and attributes to the node.

### Procedure

To insert a node, do the following:

1. Enter the session name.  
Use the name of the session that you have used in the Start XML session action.
2. Specify an XPath expression to indicate where to insert the new node.
3. Enter a node name.
4. Enter a value for the node.
5. If the node name exists, select from the following options:
  - Insert it anyways
  - Skip it
  - Overwrite it
6. Specify the location to insert the node from the following options:

- Beginning of the child nodes
- End of the child nodes
- Before specific child node
- After specific child node

If Before Specific child node or After Specific child node is selected, specify the child node name before or after which the node must be inserted.

7. Optional: Enter the default name space to be mapped to the node.
8. Enter the attributes using a Dictionary-type variable.  
Create a new variable of type Dictionary or use one that already exists.
9. Optional: Enter the attribute name space using a Dictionary-type variable  
Create a new Dictionary-type variable or use one that already exists.
10. Click Apply.

## Universal Recorder overview

Use the Universal Recorder to record interactions, such as click, read (data extraction), and write (data entry) with user interface (UI) objects on the desktop, taskbar, or in an application or browser window.

The earlier Automation Anywhere RPA products such as Version 11.3 have three separate recorders to capture objects from various environments. Enterprise A2019 combines the capabilities of these three recorders into the Universal Recorder to streamline the recording process.

## Using the Universal Recorder

You can use the Universal Recorder in the following ways:

1. To record a process consisting of multiple steps, use the Universal Recorder video camera icon, which is located on the top-left of the workbench.

See [Recording a task](#).

2. To record a single interaction that you want to add into an existing TaskBot (for example, if you missed a step when recording a task), use Capture action from the Recorder package.

See [Using the Capture action](#).

For a common use case, see [Enter data into a web form from a worksheet](#).

The Universal Recorder is also used within actions to capture coordinates (Mouse > Click action) or a local file path (Application > Open program/file action).

Watch the following video on how to use the Universal Recorder:

Using the Universal Recorder

## Supported applications and browsers

The Universal Recorder supports the following:

## Applications

- HTML applications
- Java applet, web start, and desktop applications that run using Java Runtime Environment (JRE) 6, 7, 8, 9, 10, and 11 (32-bit and 64-bit versions)

See [Recording tasks in applications that run on JRE 6](#).

- Microsoft Active Accessibility and UI automation
- Oracle EBS and Forms
- SAP
- Citrix Virtual Apps

## Browsers

- Google Chrome
- Internet Explorer

## Desktop

The desktop refers to the device screen when all application and browser windows are minimized.

## Taskbar

The taskbar is the horizontal or vertical bar containing icons of open applications and browsers, as well as the notification area. You can capture application, browser, and system icons, such as Clock and Calendar, volume, and Wi-Fi.

## UI objects

See [Object controls and actions](#).

Watch the following video for a demo on using the Universal Recorder in various applications:

Get started with Universal Recorder

## Background processing

Background processing enables an automation to run in the background. Use it in recorded tasks that use Citrix Virtual Apps and Windows native applications. Some packages, such as the Excel basic and Excel advanced inherently support background processing. The following actions support background processing:

- Click
- Set Text
- Get Text

See [Object controls and actions](#).

Related reference

[Supported browsers for Enterprise A2019](#)

## Secure recording mode

When secure recording mode is enabled, the bots do not capture values of certain properties or store application images. This ensures that sensitive data is not stored in the bots. This setting only applies to bots that are created or edited after the mode is enabled.

A user with admin privileges must enable secure recording mode. See [Settings](#).



To capture objects in secure recording mode, ensure that the Recorder package is set to 2.0.0-20200318-020414 or higher. See [Manage Enterprise Control Room packages](#).

When you record a task in secure recording mode, the Preview window temporarily shows the captured area. This image is deleted after you close the bot.

You can update the following object properties after capturing an object:

Technology	Object property
Microsoft Active Accessibility/UI automation/Java	<ul style="list-style-type: none"> <li>• Item name</li> <li>• Item value</li> <li>• Name</li> <li>• Parent</li> <li>• Value</li> </ul>
HTML	<ul style="list-style-type: none"> <li>• HTML InnerText</li> <li>• HTML name</li> <li>• HTML value</li> </ul>

## Recording tasks in applications that run on JRE 6

Use the Universal Recorder to capture objects from Java applet, web start, and desktop applications that run using Java Runtime Environment (JRE) 6.

Enterprise A2019 supports JRE 6 on both 32-bit and 64-bit systems. If JRE 6 is installed on the registered device, Java Access Bridge is automatically installed on that device.

If you have more than one version of JRE 6 installed on your device, Enterprise A2019 configures Java Access Bridge with the default version. To configure it to a different version, you must manually copy the required files to the corresponding directories. See [Java Access Bridge](#).

The first time you record a task in an application running on JRE 6, an error message might appear if the application is launched before Java Access Bridge is installed. Restart the application and proceed with recording the task.

## Recording a task

Use the Universal Recorder to capture a series of interactions (clicks, keystrokes, and mouse movements) with object controls including text boxes, buttons, tables, menus, radio buttons, combo boxes, check boxes, list views, links, trees, and page tabs.

## Prerequisites

- Configure device display and font scale to 100%.
- If you are automating a task using a browser, configure the zoom level to 100%.

Considerations when recording a task:

- If you are automating a task using a browser, do not use autofill to enter values into fields.
- Use mouse clicks, keystrokes, and shortcuts when possible.
- Record the task at low speed.
- Avoid dragging windows during the recording process.
- Avoid clicking on applications that are not part of the process you are recording and automating.

## Procedure

1. Open a new bot.
  - a) From Automation Anywhere web interface, select Bots > My bots.
  - b) Click Create TaskBot.
  - c) Enter a bot name and click Create and Edit.
2. Select your device.  
The Device button is a laptop icon.  
Automation Anywhere remembers this device until you log out. Each time you log in, reselect your device.
3. Click Record.  
The Record button is a video camera icon.  
The Record application window appears.
4. Select a window from the drop-down list.

Select the Currently active window, Desktop, or Taskbar, or a window from the list of Available windows.

If you open an application when the recorder is running, click the Refresh icon.

5. Click Start recording.  
The Record Application window appears with Pause and Stop buttons. Pause and resume the recording as needed.  
The Recording actions window also appears. It shows the number of actions recorded.
6. Perform the actions to record.  
For example, click buttons, fill in forms, or search a website.  
While you are recording, the following will occur:
  - A highlighted box will surround a control when you mouse over it.  
Note: If the box does not appear, verify that you have enabled the Automation Anywhere Google Chrome extension. For more information, see [Supported browsers for Enterprise A2019](#).
  - A Capture action will be generated for each operation.
  - Where applicable, the Recorder will generate [User-defined variables](#).
7. Click Stop to end the recording.  
The recorded steps appear as Recorder actions.
8. Click Save Changes.

## Next steps

Next [edit the task](#).

## Editing a recorded task

After recording a task, edit the Capture actions to change the window, capture a different object, select a new action, enable background processing, enter a different wait time, or save the output to a variable.

## Prerequisites

Select the List view to see the full details of each action.

Use these steps to edit one action of the recorded task. To add a Capture action to the task, see [Using the Capture action](#).

## Procedure

To edit a recorded task, follow these steps:

1. Click the Capture action that corresponds with the step.  
The Capture edit window appears.
2. Make the following changes, as necessary:
  - Select a different window either from the drop-down list or by inserting a window type variable.
  - Change the window selection to the Currently active window option.
  - Insert a wildcard character (\*) in the window title that is subject to change, such as for online invoices. For example,  
Sample\* -

Google Chrome

Note: During run time, verify that the TaskBot identifies the correct window.

- Click Re-capture object to select a new object control.
- Select a different Action from the drop-down list.

If the selected Action supports background processes, a Run in the background check box is available to enable this process. If the desired Action does not appear in the drop-down list, recapture the object. See [Background processing](#)

For the full list of possible actions by object, see [Object controls and actions](#).

- Enter a new wait time for the object control to appear.
  - Assign the output to a variable.
3. Click Apply.
  4. Click Save.

## Next steps

[Run the bot](#).

## Recording tasks from remote applications using AISense

AISense is the artificial intelligence (AI) powered capability of Enterprise A2019 that helps you identify objects from an image or an application with a complex user interface (UI) and make automation in all environments faster and more accurate.

AISense is suitable when object-based automation fails or is not efficient for automating tasks on applications that are accessed remotely, in a Citrix environment, and legacy applications.

Some of the issues that affect the accuracy of an automation task in a remote environment are as follows:

- In a Citrix environment, the automation platform receives an image of the application and does not have access to the actual UI elements.
- Screen resolution and scaling might not be the same on the host and client machines.

- The position of a UI element might change for web applications due to a different resolution on the Bot Runner machine.

AI Sense uses computer vision to intelligently create dynamic linking between objects by determining their composition to deliver change-resilient automation. AI Sense enables you to accurately automate applications even when labels and text change their position.

- [Record a task with AI Sense Recorder](#)  
Use the AI Sense Recorder to record an end-to-end task on a remote application or an application with a complex user interface (UI). The recorder captures only the following objects from an application: text boxes, buttons, radio buttons, combo boxes, and check boxes.
- [Edit a task recorded using AI Sense](#)  
After you have recorded a task using the AI Sense Recorder, you can edit the recorded actions to perform various actions on the captured objects.

Related tasks

[Record a task with AI Sense Recorder](#)

[Edit a task recorded using AI Sense](#)

## Record a task with AI Sense Recorder

Use the AI Sense Recorder to record an end-to-end task on a remote application or an application with a complex user interface (UI). The recorder captures only the following objects from an application: text boxes, buttons, radio buttons, combo boxes, and check boxes.

## Prerequisites

- If you are automating a task using a browser, ensure that autofill is disabled for that browser.
- The screen resolution must be set to 1980x1080 or lower.
- Configure the device display settings to 100% DPI.

Important: This version of AI Sense Recorder supports a limited set of controls. Support for more controls is planned in upcoming Enterprise A2019 releases.

Considerations when recording a task:

- Record the task at low speed.
- Avoid dragging windows during the recording process.
- Avoid clicking applications that are not part of the process you are recording and automating.

## Procedure

1. Create a new bot or open an existing bot for which you want to record a task.
2. Click the AI Sense Recorder icon next to the Universal Recorder.  
If you are using the AI Sense Recorder for the first time, the system downloads the required dependent files on your device, which might take some time.
3. In the Automation Anywhere Recorder dialog box, select an option from the window or URL list to specify the application window you want to use for recording.  
If you opened an application recently and it is not available in the list, click the Refresh icon.
4. Click Start AI Sense recording.  
The recorder toolbar appears with Pause, End recording, Cancel, and Analyze window options.

The system analyzes the application window you selected to identify the UI objects on that window. The options on the toolbar are disabled when the system is analyzing the application window. After the analysis, the options are enabled and you can continue to record the task.

5. Perform the actions you want to record. For example, click buttons, fill in forms, or select an option. To perform an action on an object, move the mouse pointer on that object. The object is highlighted with a box. If the object is not highlighted, click the Analyze window option on the recording toolbar. After the system finishes analyzing the application window, you can capture the required object.

The system highlights an object along with the text associated with that object. For example, when you move the mouse pointer over a button or check box, the associated text with that button or check box is also highlighted and captured.

Note: If you want to capture an option from a list, click the list and wait till the highlight appears around the box containing the options. After the box is highlighted, move the mouse pointer over the option you want to capture and wait till the option is highlighted.

- If the associated text is not highlighted:
  - a) Drag the mouse pointer towards the associated text.

The Control Selection dialog box appears.

- b) Select an option from the Type list to specify the type of the object you have selected.
  - c) Select an option from the Anchors list to select the associated text with the object you have selected.
  - d) Click OK.
- If the recorder is not able to identify the type of object you have selected, the Control Selection dialog box is displayed. Manually specify the type of the object selected and its associated text.
6. Click End recording after you finish capturing all the required objects. The recorded steps appear as a separate action in the bot.
  7. You can edit the recorded action or click Apply to save the changes.
  8. Click Save to save the bot.

Related tasks

[Edit a task recorded using AISense](#)

Related reference

[Recording tasks from remote applications using AISense](#)

## Edit a task recorded using AISense

After you have recorded a task using the AISense Recorder, you can edit the recorded actions to perform various actions on the captured objects.

Editing a recorded task enables you to perform these actions:

- Change the application window in which you want to perform the operation
- Change the properties of the captured objects
- Specify the action you want to perform on the captured objects
- Introduce a delay or specify a wait time before the next action is performed
- Specify the variable you want to use to store the output

Important: Only the following types of objects are supported with AISense Recorder:

- Text boxes

- Check boxes
- Combo boxes
- Radio buttons
- Buttons

## Procedure

1. Open the bot that contains the recorded actions you want to edit.
2. Click the action you want to edit.
3. Click the Window tab or the Variable tab to specify a different application window on which you want to perform the operation.  
If you have used the Window tab to specify the application window, you can use a wildcard character in the Window title field. The wildcard character is useful if the title of an application changes. For example,

Sample\* - Google

Chrome

Note: During runtime, verify that the TaskBot identifies the correct window.

4. Update the object properties.  
You can only update the value available in the Anchor field.
5. Select an option from the Action list to specify the action you want to perform on the object.  
For a full list of possible actions by an object, see [Object controls and actions](#).
6. Enter a value in the Wait for control field to specify the period of time the system must wait for the object control to appear on the application window.
7. Select a variable from the Assign the output to variable to assign the output to that variable.
8. Perform the Steps 2 through 7 for other actions that you want to edit in the bot.
9. Click Apply.

## Working with bots

Create, record, run, and edit automated processes or bots.

### Overview

Perform the following operations:

#### New

Create bot via different recording methods.

#### Record

Record keystrokes, mouse clicks, and mouse movements.

#### Run

Run the selected bot.

#### Edit

Edit the selected bot from the Bot editor.

#### Delete

Delete a bot.

#### Action

Select an action for the selected bot:

#### Run

Run the selected bot (the automated process).

**Edit**

Edit the selected bot using the Bot editor.

**Upload Bot**

Upload the selected bot to the Server Task Repository.

**Copy Bot**

Create a copy of the selected task.

**Rename**

Rename the selected task.

**Send To**

Send a copy of a task to: Desktop(Create Shortcut), Mail Recipient, Startup Folder or My Documents.

**Delete**

Delete the task.

- [Run bot now](#)  
Run bots from the In progress, My devices, and My bots pages.
- [Run a bot](#)  
Test an automated process by running a bot.
- [Preload packages](#)  
You can preload packages on your local device to shorten the bot runtime.
- [Copy a bot](#)  
Copy a bot to create a duplicate sequence of actions within your private repository. The copy retains the meta-data of the original bot including captured images, recorded objects, called files, and child bots.
- [Editing bots](#)  
Edit bot logic using the Bot editor to add, modify, or delete actions and automation steps.
- [Attach work item template to TaskBot](#)  
Attach a work item template to a TaskBot to use the TaskBot in workload automation.
- [Viewing package versions available in the Enterprise Control Room](#)  
Actions are grouped into version specific packages. Bot Creators can select which package version to use for a specific bot.
- [Export bots](#)  
You can export a bot from one to another.
- [Import bots](#)  
You can import a bot from one Enterprise Control Room to another.

## Run bot now

Run bots from the In progress, My devices, and My bots pages.

Run a bot as follows:

## Procedure

1. Click Run bot now.  
The Run bot Now page appears.
2. Type the activity name and optionally, type the description.
3. Find and select the bot.  
Search the TaskBot folders, BOTS and/or DEVICES, to find a specific bot. Search for a specific item by name or type using the drop-down and search field.
4. Click → to run the bot.

Note: The system supports running only one interactive bot for each device at a given time. If a bot is already running on the device, you cannot deploy another bot on it. To deploy another bot, restart the Automation Anywhere Bot Agent service.

## Run a bot

Test an automated process by running a bot.

## Overview

When a bot starts, the Run-Time window is displayed with the following information:

- The selected and current bots being run.
- Actions being performed.
- The bot line number.

To manually run a bot, follow these steps:

1. From the Enterprise client, select the bot to run.
2. Run the bot by using one of these methods:
  - a) Click Run.
  - b) Click Run Task Bot.

If the default device is mapped to the Bot agent, a user with an attended Bot Runner license in the Enterprise Control Room can use the Run Task Bot option from the bot context menu (vertical ellipsis) to run the bot.

- c) Navigate the File menu and select Run.
- d) Click Actions and select Run from the list.
- e) Right-click on the bot and select Run.

If you are running bots on your local machine, enter your username in Device Login Credentials (computer icon in the upper right corner); no password is required. The username is needed because bots run inside the user session of that particular user. If you are using a Chrome plug-in and running bots on your local machine, your username is also required.

You can preload packages on your local device to shorten the bot runtime.

If you are running bots on an unattended device, then the device needs to be unlocked at the time of scheduling the bot run and both the device username and password are required. The user credentials are used to unlock the machine at run time.

## Pausing and stopping bots

To pause or stop a bot from running at any time.

### Pause

To pause the bot, press the Pause in the Run Time Window.

### Stop

The process does not end abruptly. The Run-Time window displays the message "Aborting" and completing any steps in progress. To stop a bot from running, use one of the following methods:



1. Use the Stop button.
2. Press Esc on the keyboard.

Note: Modify the hotkeys to use a different key to stop bot that are running.

- [Repeating a bot](#)  
Repeat a bot after run failure or when running routine bots.
- [Stopping a bot manually](#)  
While running a bot, you can pause or stop the process manually from the progress window.

## Repeating a bot

Repeat a bot after run failure or when running routine bots.

### Options for repeating a bot

Do Not Repeat

Default. The bot runs once.

Repeat

Repeats the bot a number of times.

Repeat until I stop it

Repeats the bot until being stopped manually by clicking the stop button or by pressing the ESC key.

Repeat for

Repeats the bot for a time period (hh:mm:ss).

Note: Specify up to 99 hours, 59 minutes, 59 seconds.

Time between repeats

Sets a duration of time to wait before repeating a bot.

Upon error, continue with next repeat

The bot is repeated regardless of run failure.

## Stopping a bot manually

While running a bot, you can pause or stop the process manually from the progress window.

To stop a bot manually, click Pause or Stop on the progress window.

Users with manage activity permission are enabled to stop an automation bot running on a Bot Runner machine remotely from the Enterprise Control Room.

- From the Enterprise Control Room, if you stop a bot that is prompting for input or when an error message or any other message is displayed, then the message box, prompt or error window closes automatically, and the bot is aborted. If there is a bot in queue, the next bot starts.
- When the Auto-Login option is used, and you stop the bot running on a Bot Runner machine from the Enterprise Control Room, then the message box, prompt or error window closes, the task is aborted AND the machine is restored to its original state.
- If an error occurs in a bot running on a Bot Runner machine, and you stop the bot from the Enterprise Control Room, the error message is displayed in the Audit log details section of the Enterprise Control Room.

## Preload packages

You can preload packages on your local device to shorten the bot runtime.

When you run a bot, the system downloads the bot and all the packages used in the bot. Bot execution begins when the download finishes. If the packages used in the bot are preloaded, the system skips the package download process and starts executing the bot, thereby reducing the bot execution timeframe. Preloaded packages are stored in `\ProgramData\AutomationAnywhere\GlobalCache` on the local machine.

If there are multiple versions of a package, only the default package is preloaded. You can only preload packages on your local devices.

## Procedure

1. Log in with a Bot Creator account.
2. Click the My Devices > Preload package icon in the menu item associated with your local device.  
Note: The Preload package option is also available from the device icon drop-down menu for Enterprise A2019 (Cloud deployed) using the Google Chrome browser with the Chrome plug-in enabled.
3. In the Preload packages window, preload either all packages listed on the Common packages page, or select individual packages from the Customize packages page.  
Common packages represent the most lightweight and most commonly used packages.
4. Click the Start preloading option.

## Copy a bot

Copy a bot to create a duplicate sequence of actions within your private repository. The copy retains the meta-data of the original bot including captured images, recorded objects, called files, and child bots.

## Procedure

To copy a bot, follow these steps:

1. From Automation Anywhere web interface, select Bots > My bots.
2. Mouse over the actions menu.  
The actions menu is the vertical stack of three dots to the left of each bot.
3. Click the Copy Task Bot icon.
4. In the Name field, enter a name for the duplicate bot.
5. Optional: Click Browse to select the folder where to save the bot.
6. Click Copy.  
The duplicate bot appears in the specified folder.

## Editing bots

Edit bot logic using the Bot editor to add, modify, or delete actions and automation steps.

## Open a bot

Open a bot in the Bot editor in one of the following ways:

- Select the bot and click Edit. The edit button is a pencil icon.
- Right-click on the bot and select Edit.
- Click Actions and select Edit.

## Attach work item template to TaskBot

Attach a work item template to a TaskBot to use the TaskBot in workload automation.

When you attach a work item template to a TaskBot, you can use the variable `$WorkItem$` that contains the attributes for the required workload automation when you run the bot using the Run bot with queue option.

[Use Work Item variables](#)

## Procedure

1. Create or edit a TaskBot from the My bots page.  
The TaskBot opens in edit mode.
2. Select the option Work item template from the menu at the top-right of the Workbench.  
The list of Work item templates appears.
3. Select the template that best suits your workload automation requirement.
4. Click the right arrow to add the template.  
The Work item template attributes are listed for the selected template.
5. Click Save.
6. Choose one of the following actions:
  - Click Close to return to the My bots page.
  - Click Return to editor to continue editing the TaskBot.

## Viewing package versions available in the Enterprise Control Room

Actions are grouped into version specific packages. Bot Creators can select which package version to use for a specific bot.

Basic users can only view the packages available in the Enterprise Control Room and the specific package details.

From Bots > Packages the following action buttons are enabled for all users:

Add package

Browse to the package file to upload to the Enterprise Control Room.

Bot Creators with AAE\_Basic permission can select specific packages to use within a bot.

- [Select the package version used in your bot](#)  
As a Bot Creator you have the ability to select which package version to use for a set of actions within your bot.

# Select the package version used in your bot

As a Bot Creator you have the ability to select which package version to use for a set of actions within your bot.

## Prerequisites

You need to have AAE\_Basic permission.

Tip: For existing bots, follow these steps to update to the default package.

## Procedure

1. Open or create a new bot from Bots > My bots.
2. Click the verticle elipse in the upper right corner, and select Packages.  
Note: The packages used in the selected bot are highlighted.
3. Click the drop-down list to view which version of the package is currently used for this bot.
4. To change the package version to be used for this bot, select the package version to use from the drop-down list of available packages.
5. Click Change version.
6. Click Save.

## Export bots

You can export a bot from one to another.

## Prerequisites

- Only bots within the public workspace can be exported. To export bots, first check these bots into the public workspace before exporting them into the private workspace.
- Ensure that you have the correct role and permissions to import and export bots, including View package permissions.
- Exporting bots requires SMTP server to be enabled for email.

See [Configuring email server](#).

The Enterprise Control Room maintains a maximum of 10 GB of historical export files in the download directory. Files are deleted on a first-in/first-out basis in order to maintain this threshold.

## Procedure

1. Navigate to BOTS > My bots.
2. From the public directory, select the bot you want to export. Click the Export bots icon.  
The selected bot as well as the required dependencies are displayed in the Bots and Files window.  
Note: You can export bots without having the AAE\_ADMIN role. However, the dashboard-related information will not be exported. To export dashboards, you must have the AAE\_ADMIN role assigned to you.
3. Click Next.  
The bot and its dependencies are ready for export and displayed in the Review Dependencies window.
4. Click Next.  
All related packages are displayed.

5. Select the necessary packages associated with the bot to export.  
Select Exclude bot packages to remove all packages from the export.
6. Click Export bots and files.  
An email is sent that contains a link to the zip file of the exported package.
7. Download the zip file and save it to a location for import to another control room.

## Next steps

Import the bots into an Enterprise Control Room.

Related tasks

[Import bots](#)

Related reference

[Roles](#)

## Import bots

You can import a bot from one Enterprise Control Room to another.

## Prerequisites

- You can import bots only into your private workspace. After the bots are imported, you must then check the imported files into the public workspace, to complete the import process.
- Ensure that the following rights are enabled to import bots:
  - Import bots permission
  - Bot editor license
  - Check-in to Public Workspace permission
  - Package management rights

The required bot has already been exported and an email is received with the link to the zip file containing the package to import.

Any previously exported bot is located in the public repository and must be imported and checked in so that it can be deployed from the new Enterprise Control Room.

## Procedure

1. Navigate to BOTS > My bots.
2. Click Import bots. Browse and select the bot to import.
3. Set the option for importing the bot based on whether or not a previous version of the bot resides in the destination location.  
During the import, if a file already exists, you can choose to either skip the bot or file during the import process or overwrite the existing bot and file.
4. Click the Import bots icon.
5. Navigate to the private repository to find and select the imported bot.
6. Click Check in Task Bot.  
All related dependencies will also be checked in.
7. Update the comment for the check in Task Bot.
8. Click Check in.  
On completion, the imported Task Bot is placed in the public repository and is ready for deployment from the new Enterprise Control Room.

Related tasks  
[Export bots](#)  
Related reference  
[Roles](#)

## Keyboard shortcuts

List of keyboard shortcuts supported.

### All modes - Flow or List view

Keys	Action
Esc	Closes node details or clear cursor
Tab	Toggles Flow or List view
Enter	Shows node details and focus first input
Space	Toggles node details
Ctrl A	Selects all nodes

### All modes - Flow view

Keys	Action
Up	Moves cursor up
Down	Moves cursor down
Left	Moves cursor left
Right	Moves cursor right

### All modes - List view

Keys	Action
Up	Moves cursor up
Down	Moves cursor down

### Edit mode

Keys	Action
Ctrl S	Saves file
F5	Run
F10	Enters debug mode

Keys	Action
Shift F9	Toggles all breakpoint

## Edit modes - Flow or List view in focus

Keys	Action
Ctrl Shift R	Starts recording
Ctrl /	Enables or disables mode
F9	Toggles node breakpoint
Ctrl C	Copy node
Ctrl X	Cut node
Ctrl V	Paste node
Delete	Delete node
Ctrl Z	Undo
Ctrl Y	Redo

## Debug mode

Keys	Action
F5	Play/pause/restart
F6	Step next
F7	Stop
F10	Exits debug mode
Shift F9	Toggles all breakpoints

## Debug mode - Flow or List view in focus

Keys	Action
F9	Toggles node breakpoints

## Node details

Keys	Action
Ctrl Enter	Accepts current input and saves the action or trigger

## Variables overview

Enterprise A2019 offers a variety of variables, each designed to hold specific types of data and is intended for specific use. Use the topics below to learn more about each variable and how to use them.

### Credentials

Use Credentials to hold and input sensitive information such as passwords and account numbers into certain action fields. Using Credentials separates the sensitive information from the bots and Bot Runners, which reduces the risk of data spillage or unauthorized user access.

### System variables

System variables are predefined variables that return specific values about the machine on which the bot is executed. Users cannot edit the values of a system variable.

### User-defined variables

Users and some actions create user-defined variables to temporarily hold values. Use this kind of variable to input values into an action (window title, login credential, or file path) or to accept the output of an action (values read from a file or a Boolean return).

#### Create a variable

### Global values

Global values enable users to reuse identical values between bots instead of creating new variables for each bot. A user with the `AAE_admin` role configures a global value with a default value and can enable non-admin users to overwrite the value to use in their bots.

### Work Item variables

You can use the Work Item variables to pass the Work Item attributes or values to the TaskBot from the Enterprise Control Room when you run the bot with the option Run bot with queue.

Watch the following video on how to use variables:

Using variables

- [System variables](#)  
System variables are predefined variables that return specific values about the machine on which the bot is executed. Users cannot edit the values of a system variable.
- [User-defined variables](#)  
Users and some actions create user-defined variables to temporarily hold values. Use this kind of variable to input values into an action (window title, login credential, or file path) or to accept the output of an action (values read from a file or a Boolean return).
- [Use Work Item variables](#)  
You can use the Work Item variables to pass the Work Item attributes or values to the TaskBot from the Enterprise Control Room when you run the bot with the option Run bot with queue.
- [Credentials](#)  
Use Credentials to hold and input sensitive information such as passwords and account numbers into certain action fields. Using Credentials separates the sensitive information from the bots and Bot Runners, which reduces the risk of data spillage or unauthorized user access.
- [Global values](#)  
Global values enable users to reuse identical values between bots instead of creating new variables for each bot. A user with the `AAE_admin` role configures a global value with a default value and can enable non-admin users to overwrite the value to use in their bots.



## System variables

System variables are predefined variables that return specific values about the machine on which the bot is executed. Users cannot edit the values of a system variable.

System variable types include:

### Clipboard

Use the actions in the Clipboard package to perform operations on the Clipboard variable. [Clipboard package](#)

Variable	Description
Clipboard	Returns the contents of the clipboard.

### Date time

Use the actions in the Datetime package to perform operations on the date time variables. [Datetime package](#)

Variable	Description
Date	Returns the date including hours, minutes, and seconds. Note: Hours can be in 24-hour or AM/PM format depending on the machine configuration.
Day	Returns the day in DD format.
Hour	Returns the hours in HH format.
Machine	Returns the device name as a string.
Millisecond	Returns the milliseconds with a value between 0 and 999.
Minute	Returns the minutes in MM format.
Month	Returns the month in MM format.
Second	Returns the seconds in SS format.
Year	Returns the year in YYYY format.

### String

Use the following variables to change how a string is displayed.

Variable	Description
Enter	Starts a new line.
Separator	Demarcates a separation between values with a <sep> value.
Tab	Creates large space.

## System settings/parameters

Use the following variables to return data from the computer that is connected to the running Bot agent:

Variable	Description
AAControlRoom	Returns the URL of the Enterprise Control Room.
AAInstallationPath	Returns the Bot agent installation path. For example, C:\Program Files\Automation Anywhere\Bot Agent.
AATaskName	Returns the path and name of the currently running TaskBot. For example, /Bots/Finance/combineSheets.atmx.
CPUUsage	Returns the percentage utilization of the CPU.  Use this variable in a Wait for condition action to make the bot wait until machine CPU usage decreases to a specific amount. See <a href="#">Wait package</a>
Machine	Returns the name of the computer.
RAMUsage	Returns the RAM usage in megabytes.  Use this variable in a Wait for condition action to make the bot wait until machine RAM usage decreases to a specific amount. See <a href="#">Wait package</a>
OSName	Returns the operating system. For example, Windows 10 64-bit.
TotalRAM	Returns the total amount of RAM available.

Note:

- Version 11.3 contained Email, File, and PDF system variables, which could hold a limited number of properties values. In Version A2019, an infinite amount of Email, File, and PDF properties values are stored in User-defined Dictionary variables. For more information, see [Using dictionary variable for email properties](#) and [Using dictionary variable for PDF properties](#).
- Version 11.3 contained Excel system variables to return the cell, column, or row location. In Version A2019, the Excel Advanced package contains the following actions to return location values: [Get cell address](#), [Get column](#), and [Get row](#).
- Version 11.3 contained the Counter system variable to return the loop iteration count. In Version A2019, the user must create and configure a string variable.

## User-defined variables

Users and some actions create user-defined variables to temporarily hold values. Use this kind of variable to input values into an action (window title, login credential, or file path) or to accept the output of an action (values read from a file or a Boolean return).

## Variable types

The variable type dictates what kind of data the variable holds. Most variable types have a package with a similar name, which contains actions used to perform operations on the values stored in the variable. For example, use the actions in the String package to work on String variables. Similarly, to work on Number variables, use the actions in the Number package.

## Variable naming recommendations

The following tables include recommendations for naming variables, in which each variable is prefixed with a lowercase character to indicate the variable data type.

Note: Variables cannot be named Java keywords such as String, Boolean, Integer, Public, and Finally.

### Simple variable types

Variable type with naming recommendation	Description	Use examples
Any aMyVariableName	Stores Boolean, Data table, Datetime, File, Number, Record, String, or Window data types. Use this variable type when you are uncertain of which data type an action will output.	<a href="#">Using the Run action</a>
Boolean bMyVariableName	Stores either a True or False value.	
DateTime dtMyVariableName	Stores a value containing a single date and time value.	You can format the values by selecting a predefined format or specifying a custom format.  <a href="#">Date time formats</a>
File fMyVariableName	Stores a file path.	
Number nMyVariableName	Stores numeric values, including integers and decimals. It holds values from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807, and up to 15 decimal digits.  You can assign a randomly generated value to this variable.  <a href="#">Random number action</a>	You can remove the digits following the decimal when converting the value to a string.  <a href="#">Number to string action</a>
String sMyVariableName	Stores alphanumeric, character, and empty values.	You can compare the value in a String variable to an empty String variable.  <a href="#">Perform an action based on cell value</a>
Window wMyVariableName	Stores a window title and URL.	Some actions, such as the Capture a screenshot action, create a Window variable to store the specified window title and URL.  <a href="#">Extract data from a web table and save to file</a>

## Complex variable types

The variable types described below store multiple values of a simple variable type.

Note: Complex variables can be configured to store data in the any variable type. The any variable type enables a bot to pass different types of data to the same dictionary or list variable including Boolean, numeric, and string.

Variable type with naming recommendation	Description	Use examples
Dictionary dMyVariableName	Stores data in the form of key-value pairs. The value can be boolean, number, or string.	<ul style="list-style-type: none"> <li>• <a href="#">Using dictionary variable for properties</a></li> <li>• <a href="#">Using dictionary variable for properties</a></li> </ul>
List lMyVariableName	Stores a sequence of boolean, number, or string values.	
Record rMyVariableName	Stores a single row of values extracted from a table. The values can be boolean, datetime, number, or string.	<a href="#">Enter data into a web form from a worksheet</a>
Table tMyVariableName	Stores multiple values in a table of rows and columns. The values can be boolean, datetime, number, or string.	Assign values to a table variable by extracting values from a <a href="#">CSV/TXT Excel file</a> , or a <a href="#">Web form</a> .

- [Build a bot that passes a variable between actions](#)

Build a bot that takes the output of an action and inputs it into another. This example uses the actions from the Excel advanced and Message box packages to retrieve the value of a cell, save it to a String variable, then show the value to the user via a pop-up.

- [Create a variable](#)

Create a variable to store values.

- [Type casting](#)

You can temporarily convert the values inside a Boolean, Number, or String variable from one type to another type to use within one action. For example, use `${YourStringVariable}.toNumber$` to convert the value of a string variable to a number variable to perform mathematical operations.

- [Recursive expressions](#)

Use a recursive expression to insert a variable in the place of an index or key of a dictionary, list, record, or table variable.

### Related tasks

[Create a variable](#)

[Build a bot that passes a variable between actions](#)

### Related reference

[Boolean package](#)

[Data Table package](#)

[Datetime package](#)

[Dictionary package](#)

[List package](#)

Number package  
String package

## Build a bot that passes a variable between actions

Build a bot that takes the output of an action and inputs it into another. This example uses the actions from the Excel advanced and Message box packages to retrieve the value of a cell, save it to a String variable, then show the value to the user via a pop-up.

### Prerequisites

Create a worksheet and enter some sample values.

### Procedure

1. Create a variable.
  - a) Click Variables.

The Variables accordion menu is located below the Actions menu.

  - b) Click the Create variable icon.
  - c) Enter  

CellValue

 in the name field.
  - d) Check both the Use as input and the Use as output options.
  - e) Click Create.
2. Open the worksheet.
  - a) Double-click or drag an Excel advanced > Open action.
  - b) Enter a session name.
  - c) Click Desktop file.
  - d) Click Browse and select the workbook that contains the worksheet containing your sample values.
  - e) If the workbook contains multiple worksheets, select the Specific sheet name option and enter the name.
  - f) Select the option to open the file in Read-write mode.
  - g) Click Apply.
3. Retrieve the cell values and save them to the String variable.
  - a) Double-click or drag an Excel advanced > Get single cell action.
  - b) Enter the session name that you used in the Excel advanced > Open action.
  - c) Select the Specific cell option and enter the coordinates of a cell that contains a value.

For example,

A3

.

- d) Select the CellValue variable to store the cell contents.
  - e) Click Apply.
4. Close the worksheet.
  - a) Double-click or drag an Excel advanced > Close action.
  - b) Enter the session name that you used in the Excel advanced > Open action.
  - c) Click Apply.

5. Assign the CellValue variable to the Message box.
  - a) Double-click or drag an Message box action.
  - b) Click inside the Enter the message to display field.
  - c) Either press the F2 key or click the Insert variable icon.

The Insert variable icon is located on the right side of the text field.

- d) Select CellValue from the drop-down list.
  - e) Click Yes, insert.
  - f) Click Apply.
6. Click Save in the top right corner.  
Your bot is now ready to run.
7. Click the Run icon to run the bot you just created.

## Create a variable

Create a variable to store values.

### Procedure

To create and configure a variable, follow these steps:

1. From the Bot editor, click the Create variable icon at the top of the Variables menu.  
The Variables accordion menu is located below the Actions menu.  
The Create variable window appears.
2. Enter a descriptive name for the variable that is prefixed by a lowercase letter to indicate the variable type.  
For example, sCellValue.  
Note:
  - Once the variable is initialized, you cannot change the name.
  - Variables cannot be named Java keywords such as String, Boolean, Integer, Public, and Finally.
3. Optional: Enter a description.
4. Optional: Select the Constant (read-only) option to ensure values cannot be edited or overwritten.
5. Select from the following options:  
This field pertains to using a bot to run other bots. For more information, see the [Task Bot package](#).
  - Use as input: The variable holds a value that can be passed from the parent bot to a child bot. (Applicable only to a child bot).
  - Use as output: The variable holds a value that can be passed from a child bot to the parent bot. (Applicable only to a child bot).
  - Both: The value can be passed in both directions.
  - Neither: The variable is confined to this bot; it cannot be shared across other bots.
6. Select a data type from the drop-down list.  
Note: Once the variable is initialized, you cannot change the type.  
For more information on data types, see [Variable types](#).
7. Optional: Enter a default value to assign to the variable.  
The values are NULL/empty by default.
8. Click Create.

## Next steps

Follow the steps in [Build a Go be Great bot](#) for an example of how to create a variable and assign it to an action.

## Type casting

You can temporarily convert the values inside a Boolean, Number, or String variable from one type to another type to use within one action. For example, use `$<YourStringVariable>.toNumber$` to convert the value of a string variable to a number variable to perform mathematical operations.

In the text field, enter

`$<variable name>.`

. After you enter the period, the auto-fill suggests possible actions such as `toNumber`.

Variable type	Type casting options
Boolean	<ul style="list-style-type: none"> <li>• <code>invert</code>: Converts the Boolean value to the opposite value (True to False and False to True).</li> <li>• <code>toNumber</code>: Converts the Boolean value to a numeric value (True to 1 and False to 0).</li> <li>• <code>toString</code>: Converts the Boolean value to a string value.</li> </ul>
Number	<ul style="list-style-type: none"> <li>• <code>decrement</code>: Decreases the number value by one.</li> <li>• <code>increment</code>: Increases the number value by one.</li> <li>• <code>toString</code>: Converts the number value to a string value.</li> </ul>
String	<ul style="list-style-type: none"> <li>• <code>length</code>: Returns the number of characters in the string as a numeric value.</li> <li>• <code>lowercase</code>: Converts the characters in the string to lowercase.</li> <li>• <code>reverse</code>: Reverses the characters in the string.</li> <li>• <code>toBoolean</code>: Converts the string value to a Boolean value.</li> <li>• <code>toNumber</code>: Converts the string value to a number value. The limits of the String to Number conversion apply here.</li> </ul> <p>See <a href="#">String to number action</a>.</p> <ul style="list-style-type: none"> <li>• <code>trim</code>: Trims blanks and whitespaces from the string.</li> <li>• <code>uppercase</code>: Converts the characters in the string to uppercase.</li> </ul>

## Recursive expressions

Use a recursive expression to insert a variable in the place of an index or key of a dictionary, list, record, or table variable.

A recursive expression contains a variable nested inside of another variable (the outer variable). The value of the outer variable is conditional based on the value of the nested variable.

For example, the list variable `$listOfPlanets$` has the following values: Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, and Neptune. The Number variable `$indexPosition$` has value of 2. The expression `$listOfPlanets[$indexPosition$]` returns Earth.

The following types of expressions are supported. You can create a recursive expression containing up to 10 expressions.

- `$dictionaryVariable{$key}$`
- `$listVariable[$index]$`
- `$recordVariable[$index]$`
- `$tableVariable[$rowIndex][$columnIndexOrName]$`

Note:

- You cannot use a Table variable within a List variable. For example, the following expression is not supported:  
`$vListStr[$vTable[0][0]$]`
- You cannot combine expressions with properties. For example, the following expression is not supported:  
`$dictionaryVar{$listStrVar[2]$}.String:reverse$`

## Use Work Item variables

You can use the Work Item variables to pass the Work Item attributes or values to the TaskBot from the Enterprise Control Room when you run the bot with the option Run bot with queue.

The Work Item variables are available in a TaskBot only after you attach a work item template to the TaskBot when you define the work item template in the work item structure during queue creation.

[Attach work item template to TaskBot](#)

## Procedure

1. Go to the My bots page.
2. Click Create a bot.
3. In the Create a Taskbot window, enter the required parameters such as Name, Description, and Folder location.
4. Click Create & edit to open the TaskBot in edit mode.
5. Attach the bot to a queue category by selecting a Work item template.  
[Attach work item template to TaskBot](#)
6. Press the function key F2 to open the Insert a variable window and add the following Workload variables to an action:

a) `workItem` to view the default values or attributes of the Work Item when you run the TaskBot using the Run bot with queue option.

The `workItem` is an input variable for debugging a Workload bot to be used for TaskBot deployment options Run now and Schedule a bot. You can add the values for the `workItem` variable when you create an automation.

On the other hand, when you use the option Run bot with queue, the `workItem` variable uses the values passed on by the Enterprise Control Room.

b) `workItemResult` to set the final outcome of the Work Item when you run the TaskBot using the Run bot with queue option.



Use the String > Assign action to set the value of workItemResult variable. This is an output variable type and you can use the string values as well as other variables to set the value of workItemResult variables. The variables are read-only and therefore cannot be edited or deleted from the TaskBot editor page. Double click a variable in the Variables panel to view the variable parameters:

- Variable name.
- Description of the variable.
- Read only if the Constant check box is selected when the variable was created.
- To be used as input or output parameter in a TaskBot during run time.
- Variable type.
- The default values or attributes that are configured with the Work Item template for a workItem variable or the default output values for a workItemResult variable.

#### Related tasks

[Define Work Item structure](#)

## Credentials

Use Credentials to hold and input sensitive information such as passwords and account numbers into certain action fields. Using Credentials separates the sensitive information from the bots and Bot Runners, which reduces the risk of data spillage or unauthorized user access.

## Overview

A Credential is comprised of up to 50 attributes. Each attribute is configured with a value that is either standard for all users, or one that is set by the user. Similar Credentials are grouped into lockers.

For example, you have the Credential `Mainframe connection parameters` which contains three attributes: `host name`, `user name`, and `password`. This Credential is assigned to the `Connection parameters`, which contains other similar Credentials.

The Enterprise Control Room administrator gives a user access to a locker, which allows the user to insert the Credential in certain action fields. To learn more, see [Credentials and lockers](#).

## Working with Credentials

- Only fields with a Credential tab below the field name accept a Credential.
- Action fields display the locker, Credential, and attribute name; users can not see attribute values in the Bot editor.
- A Credential cannot be appended to other variables or to a string.
- If the Masked option is selected when configuring an attribute, then that attribute can only be used in Password type fields.
  - If a secure attribute is used in a non-secure text box, detect this deviation during testing to eliminate an error when the bot runs in a production environment.
  - A bot user is required to update Flex, Chrome, and Edge plugins to use password-only attributes in actions that access User Interface objects in those browsers.

## Global values

Global values enable users to reuse identical values between bots instead of creating new variables for each bot. A user with the `AAE_admin` role configures a global value with a default value and can enable non-admin users to overwrite the value to use in their bots.

## Data types

Data type	Description
DateTime	Stores a value containing a single date and time value.
File	Stores a file path.
Number	Stores numeric values, including integers and decimals. It can hold up to 15 decimal digits.
String	Stores alphanumeric and character values.

## Working with global values

Users can perform the following tasks:

- Create a global value as an admin

[Create a global value](#)

- Update the default value as a non-admin

[Overwrite the default value](#)

- Insert global values into bots

All action fields that accept a user-created variable accept a global value. Insert a global value into an action field by pressing F2 or by clicking the Insert a value icon on the left side of the field.

- [Create a global value](#)

As a user with the `AAE_admin` role, you can create a global value, configure the data type, set the default value, and enable the value to be updated by non-admin users.

- [Overwrite the default value](#)

Global values can be configured to allow non-admin users to overwrite the default value. When building bots, use a global value for values that remain constant across all bots instead of creating a new variable for every bot.

## Create a global value

As a user with the `AAE_admin` role, you can create a global value, configure the data type, set the default value, and enable the value to be updated by non-admin users.

## Procedure

1. Navigate to BOTS > Global values.  
The All global values page appears with a table of global values.
2. Click Create global value.
3. Enter a descriptive name of up to 50 characters.  
Note:
  - When the global value is initialized, you cannot change the name.
  - Global values cannot be named Java keywords such as String, Boolean, Integer, Public, and Finally.
4. Optional: Enter a description of up to 255 characters.
5. Select the data type from the drop-down list.  
[Data types](#)
6. Enter the default value.  
Note: When the value is initialized, you cannot change the type.
7. Specify whether users can overwrite the value:
  - CANNOT be changed: The default value remains constant across all users and bots.
  - CAN be changed: Users can overwrite the default value to use in their bots.
8. Click Create global value.

## Overwrite the default value

Global values can be configured to allow non-admin users to overwrite the default value. When building bots, use a global value for values that remain constant across all bots instead of creating a new variable for every bot.

As a non-admin user, you can only overwrite the default value in global values with a changeable scope. You can identify which global values have a changeable scope by checking the value in the Scope changeable column in the All global values table.

## Procedure

To verify if a global value has a changeable scope and to overwrite the default value, do the following steps:

1. Navigate to BOTS > Global values.  
The All global values table appears. Each row shows the data type, value name, whether the scope is changeable, and the last modified date and time.
2. Find the global value that has a changeable scope.
3. Move your mouse over the vertical ellipsis and click Edit global value.  
The Edit global value page appears with the User value field enabled.
4. Enter the value that will overwrite the default value.  
Note: This value will overwrite the default value of this global value in all of your bots.
5. Click Save changes.

## Bot dependencies

Bots dependencies are files and other bots that are required to run that bot successfully.

- [Upload bot dependencies](#)  
You must upload all the files that are required to run a bot successfully to the Enterprise Control Room.
- [Add bot dependencies](#)  
You must add the files that are required to run a bot as its dependencies.
- [View bots dependencies](#)  
You can view the files and bots that are added to a bot as its dependencies.

#### Related tasks

[Upload bot dependencies](#)

[View bots dependencies](#)

#### Related reference

[Bot dependencies](#)

## Upload bot dependencies

You must upload all the files that are required to run a bot successfully to the Enterprise Control Room.

## Procedure

1. Click BOTS > My bots.
2. In the PRIVATE tab, expand the Bots folder and select the folder in which you want to upload the dependent files.
3. Click the Upload files icon.
4. On the Upload files page, click Add files.
5. From the Open dialog box, select the files you want to upload and click Open.
6. Click Upload.

#### Related tasks

[Add bot dependencies](#)

[View bots dependencies](#)

## Add bot dependencies

You must add the files that are required to run a bot as its dependencies.

## Prerequisites

Ensure that the files you want to add as dependencies are uploaded in the Enterprise Control Room, except for bots.

The system identifies the automatic dependencies for a bot and adds the required files and other bots as its dependencies. However, for manual dependencies, you must identify the files and bots that are required and add them as dependencies for a bot.

## Procedure

1. Click BOTS > My bots.
2. In the PRIVATE tab, expand the Bots folder that contains the bot for which you want to add dependencies.
3. Select the bot for which you want to add dependencies.
4. Select the Edit Task Bot icon from the actions menu (vertical ellipsis) on the right.
5. Select Dependencies from the actions menu (vertical ellipsis) on the top-right.

6. Expand the Bots folder and select the folder that contains the files you want to add as dependencies.
7. Select the files you want to add as dependencies from the Available files section and click the right arrow.
8. Click Save.
9. Click Return to editor if you want to update the bot or click Close to close the bot.

#### Related tasks

[View bots dependencies](#)

[Upload bot dependencies](#)

## View bots dependencies

You can view the files and bots that are added to a bot as its dependencies.

## Procedure

1. In the PRIVATE or PUBLIC tab, expand the folder that contains the bot for which you want to view dependencies.
  2. Select the bot for which you want to view dependencies.
  3. Select the View Task Bot icon from the actions menu (vertical ellipsis) on the right.
  4. Select Dependencies from the actions menu (vertical ellipsis) on the top-right.
- The system shows a list of files and bots that are added as dependencies for the bot.

#### Related tasks

[Add bot dependencies](#)

[Upload bot dependencies](#)

# Attended automation using Enterprise A2019

Robotic Process Automation (RPA) enables users to automate repetitive business processes with bots. The collaboration between humans and software bots is called attended automation.

Enterprise Control Room is the central interface in Enterprise A2019 that enables you to manage the processes of an RPA infrastructure. As Enterprise A2019 is a web-based interface, it does not require an installation or deployment on the user machines.

As part of the bot development functionality, Enterprise A2019 provides a form and bot creator interfaces. A form creator interface has various elements that enables users to design new forms. A bot creator interface enables users to create TaskBots. Interactive Forms and Trigger Loop packages that are integrated into Enterprise A2019 provides a list of actions to manage task logic within these TaskBots. Users with a Bot Creator license in Enterprise Control Room can create bots to manage data across various applications within their attended automation process.

Triggers are preset conditions that can be created to start bots. As part of attended automation, certain preset end user actions (modifying a file/folder or using combination of keys) or an incoming email can be used as triggers. Users with an attended Bot Runner license can run a bot with triggers. The bot is deployed on the user's machine and continuously checks for the trigger condition. When the trigger condition is met, the bot runs on the machine to complete the specified task.

- [Using interactive forms](#)

Interactive forms provide a form builder experience for users to build interfaces for submitting and regrouping data that is used to send and receive information from various applications within their attended automation process. No additional licenses or permission are required to use interactive forms.

- [Adding a trigger to run a bot](#)


Triggers enable a task to run automatically when certain predefined events occur. Enterprise A2019 enables you to add triggers that can automatically run the selected bot whenever a specific event occurs. For example, clicking a specific button or using a combination of keystrokes.

## Using interactive forms


Interactive forms provide a form builder experience for users to build interfaces for submitting and regrouping data that is used to send and receive information from various applications within their attended automation process. No additional licenses or permission are required to use interactive forms.

Interactive forms provide a list of actions that you can drag into a bot as a task logic. Specific function of each action is displayed when you hover over them. The actions within the interactive forms can be broadly classified into the following types:

### Form level actions

Represented by  and used within a task logic to manage the linked form.

### Element level actions

Represented by  and used to manage elements of a form.

- [Create a form](#)

The form builder in Enterprise A2019 enables you to create or edit forms through various form elements available through a browser interface. A form provides structural content with little or no changes in the layout. You can link a form to bots that perform predefined automated tasks.

- [Add a form to bot](#)

Adding an existing form to a bot enables users to collaborate with bots. Use the necessary action items to create a bot and build a task logic.

## Create a form

The form builder in Enterprise A2019 enables you to create or edit forms through various form elements available through a browser interface. A form provides structural content with little or no changes in the layout. You can link a form to bots that perform predefined automated tasks.

The form builder requires no additional installation over the Bot agent. Enterprise Control Room users with the Bot Creator license can access the form builder to design a new form or edit an existing form.

This task provides an example scenario where you create a new user registration form that captures all the necessary information. When you link this form to a bot, it can trigger a third-party application to store user information.

## Procedure

1. Click BOTS > My Bots > Create new > Form.
2. In the Create form page, enter a name for the new form.  
Forms are saved in the \Bots\ folder by default. Click Browse to change the default folder.
3. Click Create & edit.  
The form builder page appears with a single row Column layout.

4. Use the Properties panel on the right to update the form.

For example, enter

User Registration Form

in the Form title field and use the Display behaviors option to set the way the form is displayed on the user desktop.

You can also use the Hidden elements to set a variable. For example, if the registration form is linked to a Social Security Number (SSN) that cannot be displayed to users, select the hidden element Type as Number. A user with a Bot Creator license can then access this variable when creating a bot.

5. Optional: Drag Column to arrange elements horizontally.

For example, if you want the first name and last name fields to appear next to each other on the form, drag Column into the form. Then drag two text boxes into the columns. Use the row properties to customize the number of columns and the column width.

Note: Click Delete or select any element and press the Delete button from the keyboard to remove it from the form.

You can also use these advanced form elements:

- Document  
Use this element to render a document, such as an image (.jpg) or PDF that you want to preview along with the form. Ensure you enter the correct URL or the document location address. You can also zoom in or zoom out of the attached document.  
Note: Only one Document element can be used in a form.
- Password

Use this element if the form must have a confidential or masked field. Unlike the Textbox element that has an option to mask the content, the Password element uses the masking feature by default to capture the data in the field and CV encryption to transfer data.

6. In this scenario, drag the following elements into the form:

a) Textbox: Enter First Name in the Element label field of the first text box, and enter Last Name in the Element label field of the second text box.

b) Number: Enter Contact Number in the Element label field.

For a new user registration form, some of the fields such as contact number might be mandatory. Use the Advance behaviors to make it mandatory and the Formatting option to set the input format.

c) Radio Button: Update the Element label as Gender. Change the values for the Radiobutton1 as Male and Radiobutton2 as Female.

d) Date: Enter Date of Birth in the Element label field.

e) Textbox: Enter Email ID in the Element label field.

f) Dropdown: Update the Element label as Newsletter. Enter Daily, Weekly, Monthly and None in the Add dropdown content field.

In this scenario, the Dropdown element is used to create a newsletter option for the new user. This can be used to trigger an email based on the selected option.

g) Button: Enter Submit in the Button-text (required) field.

h) Drag another Button element and enter Cancel in the Button-text (required) field.

Use the Button type drop-down menu and set this as Secondary.

7. Optional: Click Preview.

Use this feature to preview the appearance of the form based on the resolution of the destination device.

8. Click Save.

The new registration form is created and is ready to be linked to a bot.


## Add a form to bot

Adding an existing form to a bot enables users to collaborate with bots. Use the necessary action items to create a bot and build a task logic.

This task describes a scenario where you want to design a bot that triggers an existing New User Registration form. A Hot key preset trigger starts the bot. An existing form that is linked to the bot is displayed where the user enters the information.

- If the user clicks Submit, a User Agreement form is displayed.
- If the user clicks Cancel, User registration is canceled message is displayed and the bot is terminated.

## Procedure

1. Click BOTS > My Bots > Create new > Bot.
2. Enter a name for the bot.  
All the bots are stored in the \Bots folder. Click Browse to change the folder.
3. Click Create & edit.  
The bot builder pane is displayed that has Flow (default view), List and Dual view. You can drag the required actions from the left pane to create a task.
4. Set Ctrl and L as the hot key combination.
5. Use the Search actions field to find Interactive forms.
6. Drag Display into the task logic.  
You can use the display action to show the form when the bot is triggered. The display action properties appear on the right.
7. Click the create variable icon (  ) to create a variable.  
You have to create a form variable by linking an existing form. This enables you to use the form variable and the associated form across multiple instances.  
Note: The form variable can be linked only to forms created using the Enterprise Control Room form builder.
8. On the Create variable screen, enter a name for the form variable you are creating.
9. Click Browse and select the New User Registration form.  
This links the New User Registration form to the form variable that you are creating.
10. Click Create & select.  
A new form variable is created. For any existing form variable, use the Form name drop-down menu to select it.
11. Use the Search actions field to find Trigger loop and drag it to add event monitoring.  
Event monitoring enables you to simultaneously set multiple events within the task logic of a bot.
12. Drag Handle into the Trigger loop.  
Handle enables you to run an action by configuring a trigger in the Trigger loop: Handle for the bot. You can set a trigger when an event occurs on the selected interface (selected application) or when a user clicks the selected form element.
13. For this scenario, use the Trigger on drop-down menu to select INTERACTIVE FORMS > Form trigger.
14. Click the Form name drop-down menu and select the New User Registration form.  
This enables you to set a trigger for the Submit option of the New User Registration form.
15. Click the Form element drop-down menu and select Submit.  
For the Button element of a form, the Select action trigger is set to Clicked by default. When the user clicks Submit on the New User Registration form, it triggers the User Agreement form.
16. Search for Interactive forms and drag Display into Trigger loop: Handle.
17. Click the Form name drop-down menu and select User Agreement form.  
The User Agreement form appears.



18. Search for Message and drag it into Trigger loop: Handle, after Interactive forms.  
A successful registration message appears when the user clicks Submit.
19. Terminate the trigger by dragging Break into the Trigger loop.  
Use Break to exit the current event loop and for actions where finishing a process is required.
20. Drag another Handle next to the Trigger loop: Handle.  
This enables you to set a trigger for the Cancel option of the New User Registration form.
21. Click the Form element drop-down menu and select Cancel.
22. Search for Message and drag it into Trigger loop: Handle. Update the Enter the message to display field as User registration is canceled.  
When the user clicks Cancel on the New User Registration form, it triggers a `User registration is canceled` message and terminates the bot.
23. Drag Trigger Loop > Break to terminate the loop.
24. Click Apply and then save the bot.
25. Click Run with triggers.  
The Run with triggers sets the Hot key trigger to start the bot, which is Ctrl plus L combination from the keyboard in this scenario.

Related reference

[Interactive forms package](#)

## Adding a trigger to run a bot

Triggers enable a task to run automatically when certain predefined events occur. Enterprise A2019 enables you to add triggers that can automatically run the selected bot whenever a specific event occurs. For example, clicking a specific button or using a combination of keystrokes.

Triggers integrate predefined events into your workflow and reduce the number of repetitive tasks that users must perform. Attended automation in Enterprise A2019 enables you to create unique triggers for various applications. You can then use these triggers to start a bot.

Use one of the following types of triggers to start a bot:

### Email

Starts a bot when a new email message is received in the specified email service such as Microsoft Outlook, Email server, or EWS server.

### Files and folders

Starts a bot when a predefined file or folder event occurs. You can set one of the following options as the trigger:

- When a new file or folder is created.
- When an existing file or folder is deleted.
- When a file or folder is renamed.
- When a file or folder is modified.

### Hot key

Starts a bot when a predefined combination of keystroke is performed on the keyboard. You can set a combination of the following keys as the trigger:

- Control keys such as Ctrl, Alt, Windows logo key, and Esc.
- Keys with regular alphabetical (English) characters.

## Interface

Starts a bot when a predefined event occurs on the selected user interface element. Some of the events that you can set as trigger in Microsoft Windows are:

- When a process starts running.
  - When a process stops running.
  - When an application window opens.
  - When an application window closes.
- [Add an email trigger](#)  
Use email triggers to start a bot when a new email message is received in a specified email service.
  - [Add a file and folder trigger](#)  
Use the files and folders trigger to start a bot when a predefined action such as create, delete, rename, or modify is performed on a file or a folder.
  - [Add a hot key trigger](#)  
Use the hot key trigger to start a bot when a predefined combination of keys is used on the keyboard.
  - [Add an interface trigger](#)  
Use interface triggers to start a bot when a predefined action occurs on a specific interface element, such as clicking a button or closing an application.

## Related reference

[Trigger loop package](#)

## Add an email trigger

Use email triggers to start a bot when a new email message is received in a specified email service.

Connect the Enterprise Control Room to one of the email services on your system and trigger the bot when you receive a new email.

## Procedure

1. Click BOTS > My Bots > Create new > Bot.
2. Enter a name for the bot.  
All the bots are stored under the \Bots folder. Click Browse to change the folder.
3. Click Create & edit.
4. Choose one of the following server types as the trigger for your incoming emails:

Server type	Steps
Microsoft Outlook	a) In the bot builder page, click Triggers > Email trigger b) Drag Email into Drag a trigger here. c) In the Email trigger pane on the right, click Outlook  Microsoft Outlook is set as the email service and the bot is triggered when you receive a new email.  d) Optional: Use the Check every option to configure the frequency for triggering the bot.
Email server	a) In the Email trigger pane on the right, click Email server. b) Enter the email service Host information.  For example, if you are using Office365 as your email service, enter

Server type	Steps
	<p>Outlook.office365.com</p> <p>.</p> <p>c) Select IMAP or POP3. IMAP and POP3 are protocols used to connect to the mail server that enables you to read your emails through an email client. Note: The POP3 protocol has a known limitation where any new calendar invite can also trigger the bot.</p> <p>d) Enter the configured port number for your email server.</p> <p>e) Select one of the following user authentication methods to sign in to the email server:</p> <ul style="list-style-type: none"> <li>f) Credential: Click Pick to select the stored Username and Password.</li> </ul> <p><a href="#">Credentials</a></p> <ul style="list-style-type: none"> <li>g) Insecure string: Enter the username and password.</li> </ul>
EWS server	<p>a) In the Email trigger pane on the right, click EWS server.</p> <p>b) From the Microsoft Exchange Version drop-down menu, select your current version.</p> <p>For example, if you are using Microsoft Exchange Service Pack 2, select Exchange2010_SP2 as your email service.</p> <p>c) Select one of the following user authentication methods to sign in to the email server:</p> <ul style="list-style-type: none"> <li>d) Credential: Click Pick to select the stored Username and Password.</li> </ul> <p><a href="#">Credentials</a></p> <ul style="list-style-type: none"> <li>e) Insecure string: Enter the username and password.</li> </ul>

- Optional: Use Select Conditions to specify an email event for the selected server type.  
Consider a scenario where you have selected Microsoft Outlook as the server type and want trigger the bot when you receive an email only from a particular sender. Under Select Conditions, select the Email from check box and specify the email address.  
Note: The Email in folder does not support sub-folders when you set the protocol as POP3 for Email Server,
- Click Apply.
- Click Run > Run with triggers.

Based on your selection Microsoft Outlook, Email server or EWS server is set as the trigger to start the bot for new emails.

## Add a file and folder trigger

Use the files and folders trigger to start a bot when a predefined action such as create, delete, rename, or modify is performed on a file or a folder.

This procedure is for a scenario where you want to trigger a bot when a user modifies a spreadsheet called Sales update.xls.

## Procedure

1. Click BOTS > My Bots > Create new > Bot.
2. Enter a name for the bot.  
All the bots are stored under the \Bots folder. Click Browse to change the folder.
3. Click Create & edit.
4. In the bot builder page, click Triggers > Files & folders.
5. Drag File trigger into Drag a trigger here.  
Note: Drag Folder trigger if you want to add a folder as a trigger.
6. Click Browse and select the Sales update.xls file.
7. From the Select the bot when the file is drop-down menu, select modified.  
This sets the trigger for the bot.  
Note: File triggers support the use of wildcard characters in the file name or type.
8. Click Apply.
9. Click Run > Run with triggers.  
Whenever a user modifies the Sales update.xls, the bot is triggered.

Related reference

[Using event triggers](#)

## Add a hot key trigger

Use the hot key trigger to start a bot when a predefined combination of keys is used on the keyboard.

This procedure is for a scenario where you want to start a bot when a user uses the Ctrl plus L keys from the keyboard.

## Procedure

1. Click BOTS > My Bots > Create new > Bot.
2. Enter a name for the bot.  
All the bots are stored under the \Bots folder. Click Browse to change the folder.
3. Click Create & edit.
4. In the bot builder page, click Triggers > Hot key.
5. Drag Hot key trigger into Drag a trigger here.
6. Click Ctrl to set it as the Hot key from the available key modifiers.
7. Use the drop-down menu to select L from the keyboard.  
The Hot key field displays the key combination that will trigger the bot.
8. Click Apply.
9. Click Run > Run with triggers.  
When the Ctrl and L keys are used on the keyboard, the bot is triggered.

## Add an interface trigger

Use interface triggers to start a bot when a predefined action occurs on a specific interface element, such as clicking a button or closing an application.

This procedure is for a scenario where you want to configure a trigger to start a bot when the user clicks the File > New in Microsoft Word. Ensure you have the Microsoft Word application open on your system.

Note: Interface triggers for objects are available only for native Windows applications.

## Procedure

1. Click BOTS > My Bots > Create new > Bot.
2. Enter a name for the bot.  
All the bots are stored under the \Bots folder. Click Browse to change the folder.
3. Click Create & edit.
4. In the bot builder page, click Triggers > Interface trigger.
5. Drag Object into Drag a trigger here.  
The Interface trigger: Object editing panel appears.
6. From the Window detail drop-down menu, select AVAILABLE WINDOWS > Word.  
This sets the trigger for the bot. You must configure the action to start the bot. The Window title and Executable options are updated based on the window you select.

If Microsoft Word is not displayed in the list, you can use the Refresh option to reload the list of applications.

7. Click Capture object.  
This initiates a built-in web recorder that captures the action. In this scenario, Microsoft Word is displayed. Click New, which is then set as the trigger to start the bot.
8. Click the Select main event drop-down menu to choose the desired event.  
The events are populated based on the object that you select. You can also select one of the listed keys as the trigger event.
9. Click Apply.
10. Click Run > Run with triggers.  
The preset trigger is created, and the bot is triggered when the user clicks New in Microsoft Word.

## Get started building bots

Create and edit bots in the cloud-based Bot editor. Use the examples below to become familiar with features and learn to build bots in Enterprise A2019.

- [Build a Go be Great bot](#)  
Build a basic bot using a Message Box action and a variable. Follow these steps to create your first bot that prints the message, `Go be great!`, the Automation Anywhere version of `Hello World!`
- [Build a basic bot that uses a desktop application](#)  
Build a bot that uses a conditional statement to verify that the calculator is open, then uses the calculator to multiply two numbers. This example uses actions from the Application, If, Simulate keystrokes, Message Box, and Window packages.
- [Extract data from a web table and save it to a file](#)  
Build a bot to open a browser window to the NASDAQ website, extract the data from a table, and write it to a CSV file on your desktop. This example uses actions from the Browser, Data Table, Recorder, and Window packages.
- [Enter data into a web form from a worksheet](#)  
In this example, you build a bot to enter multiple rows of data from an XLSX sheet into a web form. Use actions from the Excel advanced, Loop, and Recorder packages.

- [Perform an action based on cell value](#)  
In this example, you build a bot that prints a message based on whether a cell has a value or is empty. Use the actions from the Excel basic or Excel advanced, If, and Message Box packages.
- [Use Python script to join a list](#)  
Build a bot that uses a Python function to print the message `Go Be Great!`, the Automation Anywhere Enterprise version of `Hello World`. In this example, the bot combines a list of string values and prints them to a message box.

## Build a Go be Great bot

Build a basic bot using a Message Box action and a variable. Follow these steps to create your first bot that prints the message, `Go be great!`, the Automation Anywhere version of `Hello World`!

## Prerequisites

To build a bot you must already have done the following:

- [Register device and install Bot agent](#)
- [Set device credentials](#)

## Procedure

If you have already completed the steps in [Create your first bot](#), skip to step 6.

1. Open a new bot:
  - a) From the Automation Anywhere Enterprise web interface, select Bots > My bots.
  - b) Click Create TaskBot.
  - c) Enter a bot name.
  - d) Accept the default folder location `\Bots\`.  
To change where your bot is stored, click Choose and follow the prompts.
  - e) Click Create and Edit.
2. Insert a [Message box package](#) action.
  - a) Click Actions.
  - b) Search for the Message Box package.  
Click in the Actions search box and type the word, `message`. Click the arrow to expand the Message Box options.
  - c) Double-click or drag the Message Box action to the Bot editor (open space to the right).  
A dialog box to configure the action opens.
3. Specify the conditions for the Message Box action.
  - a) In the Enter the message box window title field, type `My first bot!`.
  - b) In the Enter the message to display field, type `Go be great!`.
  - c) Accept the defaults in the Scrollbar after lines field and Close message box after check box.
  - d) Click the Apply button to save your message edits.  
The Message Box action is added to the flowchart in the Bot editor.
4. Click Save.  
Your bot is now ready to run.
5. Test your bot.  
Click Run at the top right.  
The bot generates a pop-up Message box with the text `Go be great!`.

In the following steps, you configure a variable and insert it into the Message box.

0. Create a variable.

a) Click Variables.

The Variables accordion menu is located below the Actions menu.

b) Click the Create variable icon.

c) Enter

`vHelloWorld`

in the name field.

d) Check the Use as output option.

e) Type

`Say Go be Great! with a variable`

in the Default value field.

f) Click Create.

1. Assign the

`vHelloWorld`

variable to the Enter the message to display field.

a) Click the Message Box action icon.

A dialog box to configure the action opens.

b) Delete the text from the Enter the message to display field.

c) Either press the F2 key or click the Insert variable icon.

The Insert variable icon is located on the right side of the text field.

d) Select

`vHelloWorld`

from the drop-down list.

e) Click Yes, insert.

2. Click Apply.

3. Click Save.

4. Click the Run icon.

The bot generates a pop-up Message box with the text `Say Go be Great! with a variable`.

## Next steps

After you successfully run your bot, progress to [Build a basic bot that uses a desktop application](#).

## Build a basic bot that uses a desktop application

Build a bot that uses a conditional statement to verify that the calculator is open, then uses the calculator to multiply two numbers. This example uses actions from the Application, If, Simulate keystrokes, Message Box, and Window packages.

## Prerequisites

To build a bot you must already have done the following:

- [Register device and install Bot agent](#)
- [Set device credentials](#)

## Procedure

1. Open a new bot.
  - a) From Automation Anywhere web interface, select Bots > My bots.
  - b) Click Create TaskBot.
  - c) Enter a bot name and click Create and Edit.
2. Insert an conditional sequence that verifies that the calculator is open and activates the window.
  - a) Double-click or drag the If action.
  - b) Select Window exists from the Condition drop-down list.
  - c) Select Calculator from the Window drop-down list.
  - d) Click Apply.

The Calculator window is saved to the variable `window-1`.

  - e) Drag the Message box action into the If container.
  - f) In the Enter the message to display field, enter  
`The calculator is running`
  - g) Click Apply.
  - h) Drag the Window > Activate action into the If container below the Message box action.
  - i) In the Window field, insert the `window-1` variable.
  - j) Click Apply.
3. Insert an alternative sequence that opens the calculator.
  - a) Double-click the If > Else action.
  - b) Click Apply.
  - c) Drag the Message box action into the Else container.
  - d) In the Enter the message to display field, enter  
`The calculator is not running`
  - e) Click Apply.
  - f) Drag the Application > Open program/file action into the If container below the Message box action.
  - g) In the Location of the program/file field, enter  
`C:\Windows\System32\calc.exe`
  - h) Click Apply.
4. Insert the Simulate keystrokes action to perform the calculation.
  - a) Drag the Simulate keystrokes action below the If and Else containers.
  - b) In the Window field, insert the `window-1` variable.
  - c) In the Keystrokes field, enter  
`5*5=`
  - d) In the Delay field, enter  
`500`
  - e) Click Apply.
5. Click Save to save the bot.
6. Run the bot.  
 When the bot runs, the `The calculator is not running` message appears, then the Calculator window opens, and the calculation is performed.
7. Run the bot again.  
 When the bot runs, the `The calculator is running` message appears, then the Calculator window activates, and the calculation is performed.

If the bot does not input all of the numbers into the calculator, try increasing the keystrokes delay.



## Extract data from a web table and save it to a file

Build a bot to open a browser window to the NASDAQ website, extract the data from a table, and write it to a CSV file on your desktop. This example uses actions from the Browser, Data Table, Recorder, and Window packages.

To extract data from a table, do the following steps:

### Procedure

1. Open a new bot:
  - a) From the Automation Anywhere Enterprise web interface, select Bots > My bots.
  - b) Click Create TaskBot.
  - c) Enter a bot name.
  - d) Accept the default folder location \Bots\.

To change where your bot is stored, click Choose and follow the prompts.

  - e) Click Create and Edit.
2. Open a browser window to the web page from which you will extract the table.
  - a) Double-click or drag the Browser > Launch website action.
  - b) In the URL field, enter  
`https://old.nasdaq.com/`
  - c) Specify the Internet Explorer browser.

Note: It is recommended to use Internet Explorer because it reliably launches the website in a new window, even if there is already an open window. Other browsers might launch the website in a new tab if there is an open window.

  - d) Click Apply.
  - e) Click Save.
  - f) Click Run.

The bot opens the window.
3. Specify the table.
  - a) Double-click or drag the Recorder > Capture action.
  - b) Click the Window tab and select the Daily Stock Market Overview window from the drop-down list.
  - c) Click Capture object.

The Daily Stock Market Overview window activates.

  - d) Hover over the table below the Stock Market Overview heading.

An orange box appears, surrounding the table.

  - e) Click the table.

The Object Processing message box appears.

  - f) Return to the Enterprise Control Room.
  - g) In the Object properties table, verify the Control Type is TABLE.

If it is not, click Recapture object.

  - h) From the Action drop-down list, select Get table.
  - i) In the Assign output to variable field, create the `tNasdaqTable`.
  - j) Click Apply.

The Daily Stock Market Overview window is saved as the variable `window-1`.
4. Specify the file where to save the data.
  - a) Double-click or drag the Data Table > Write to file action.
  - b) From the Data table name list, select `tNasdaqTable`.
  - c) Provide a file path to create a CSV file.

For example, `C:\Users\<username>\Desktop\NasdaqTable.csv`.

- d) Select the Create folders/files if it doesn't exist option.
  - e) Select to overwrite the existing file.
  - f) Click Apply.
  - 5. Close the Daily Stock Market Overview window.
    - a) Double-click or drag the Window > Close action.
    - b) Select the Variable tab and insert `window-1`.
    - c) Click Apply.
  - 6. Click Save.
  - 7. Click Run.
- The bot creates a CSV file on the desktop with data on seven indexes, their values, and net change.

## Enter data into a web form from a worksheet

In this example, you build a bot to enter multiple rows of data from an XLSX sheet into a web form. Use actions from the Excel advanced, Loop, and Recorder packages.

To retrieve values from an Excel file and input them into a web form, do the following:

### Procedure

1. Open a new bot:
  - a) From the Automation Anywhere Enterprise web interface, select Bots > My bots.
  - b) Click Create TaskBot.
  - c) Enter a bot name.
  - d) Accept the default folder location \Bots\.

To change where your bot is stored, click Choose and follow the prompts.

  - e) Click Create and Edit.
2. Open the Excel file.
  - a) Double-click or drag the Excel advanced > Open action.
  - b) Enter a session name.
  - c) Select the Excel file.
  - d) Mark the Sheet contains a header option.
  - e) Click Apply.

By marking the Sheet contains a header option, you enable the bot to search for the column by the header name during run time.
3. Launch the website.
  - a) Double-click or drag the Browser > Launch website action.
  - b) Enter the website URL.
  - c) Click Apply.
4. Retrieve the worksheet values and store them in a Table variable.
  - a) Double-click or drag the Excel advanced > Get multiple cells action.
  - b) Enter the same session name you used in the Excel advanced > Open action.
  - c) Select All rows from the drop-down list.
  - d) Create a Table variable using the icon to the right of the Assign to variable drop-down list.
  - e) Click Apply.
5. Instruct the bot to process the data row by row.
  - a) Double-click or drag the Loop action.
  - b) Select the For each row in table iterator.
  - c) Select the same Table variable that you used in Get multiple cells.
  - d) Create a Record variable using the icon to the right of the Assign to variable drop-down list.

- e) Click Apply.
- The Record variable holds all of the values for one row. With each iteration of the Loop, the bot retrieves the values of the next row and stores them in the Record variable, overwriting the values from the previous row.
6. Map the first column header to the web form textbox.
  - a) Double-click or drag the Recorder > Capture action.
  - b) Select the same window you opened with the Launch website action.
  - c) Click Capture object.
  - d) Hover over the textbox until a red outline appears.
  - e) Click the textbox.
  - f) Return to the Enterprise Control Room.
  - g) Verify that the Control Type value is TEXTBOX.
  - h) Select Set text from the Action drop-down list.
  - i) In the Keystrokes field, insert the same Record variable that you used in the Loop.
  - j) Select the By name option and copy-paste the first column header into the field.
  - k) Click Apply.
7. Repeat the sub-steps in step 5 to map the other columns, with the following differences:
  - a) Instead of searching for the browser window title, insert the Window variable generated by the Recorder.
  - b) When inserting the Record variable in the Keystrokes field, copy-paste the subsequent column header into the "By name" field.
8. Capture the Submit button.
  - a) Double-click or drag the Recorder > Capture action.
  - b) Insert the Window variable generated by the Recorder.
  - c) Click Capture object.
  - d) Verify that the Control Type value is BUTTON.
  - e) Select the Click action.
  - f) Click Apply.
9. Click Save.

## Perform an action based on cell value

In this example, you build a bot that prints a message based on whether a cell has a value or is empty. Use the actions from the Excel basic or Excel advanced, If, and Message Box packages.

## Prerequisites

Before building this bot, save an empty Excel worksheet to the desktop.

Because this bot does not create a new Excel worksheet, you can use either the Excel basic or Excel advanced packages.

Note: All of the actions must be from the same package.

## Procedure

1. Open a new bot.
  - a) From Automation Anywhere web interface, select Bots > My bots.
  - b) Click Create TaskBot.
  - c) Enter a bot name and click Create and Edit.
2. Use the Open action from the Excel basic or the Excel advanced package to open the Excel sheet.

- a) Double-click or drag the Open action.
  - b) Enter a session name.
  - c) Click Browse to provide the file path to the empty Excel worksheet on the desktop.
  - d) Click Apply.
3. Use the Get single cell action to assign the value of a cell to a string variable.
  - a) Double-click or drag the Get single cell action from the same package that you used for the Open action.
  - b) Provide the session name that you used in the Open action.
  - c) Select the Active cell option.
  - d) In the Store cell contents to field, create the variable `Output`.
  - e) Click Apply.
4. Use the If action to configure the conditional statement.
  - a) Double-click or drag the If action.
  - b) Select String from the Condition drop-down list.
  - c) In the Source value field, insert the variable `Output`.
  - d) Select Equals to as the Operator.
  - e) Leave the Target value field empty.
  - f) Click Apply.
5. Insert a Message box into the If container.
  - a) Drag the Message box action.
  - b) In the Enter the message to display field, enter  
`Cell is empty`
  - c) Click Apply.
6. Use the Else and Message box actions to configure the alternative sequence of actions.
  - a) Drag the Else action next to the If action.
  - b) Drag the Message box action into the Else container.
  - c) In the Enter the message to display field, enter  
`Cell is not empty`
  - d) Click Apply.
7. Click Save.
8. Run the bot.  
As the bot runs, the message box appears with the message `Cell is empty`.
9. Enter a value into the cell located at A1 in the Excel sheet and save the sheet.
10. Run the bot.  
As the bot runs, the message box appears with the message `Cell is not empty`.

## Use Python script to join a list

Build a bot that uses a Python function to print the message `Go Be Great!`, the Automation Anywhere Enterprise version of `Hello World`. In this example, the bot combines a list of string values and prints them to a message box.

### Prerequisites

To run Python script from Enterprise A2019, you must already have the latest version of Python 3.x installed on your device.

## Procedure

1. Open a new bot:
  - a) From the Automation Anywhere Enterprise web interface, select Bots > My bots.
  - b) Click Create TaskBot.
  - c) Enter a bot name.
  - d) Accept the default folder location \Bots\.

To change where your bot is stored, click Choose and follow the prompts.

  - e) Click Create and Edit.
2. Create a variable to hold the list values:
  - a) Click the Create variable icon.
  - b) Enter  
lArgument  
in the Name field.

Recommendation: Prefix the variable name with a lowercase character to indicate the variable data type.

### Variable naming recommendations

- c) Select the List type and String subtype.
  - d) In the Default value field, enter the following values:
    - a) Value at 0:  
Go
    - b) Value at 1:  
Be
    - c) Value at 2:  
Great
    - d) Value at 3:  
!
  - e) Click Create.
3. Provide the script with a Python Script > Open action:
  - a) Double-click or drag the Python Script > Open.
  - b) Select the Manual input option.
  - c) Copy and paste the following text into the Enter script here field.

```
def data ( str ):
    x = " ".join( str )
    return x
```
- d) Click Apply.
4. Use a Python Script > Execute function action to tell the bot to run the script:
  - a) Double-click or drag Python Script > Execute function.
  - b) Enter  
data  
in the Enter name of function to be executed field.
  - c) Select the lArgument variable from the Arguments to the function drop-down list.
  - d) Create the variable sOutput for the Assign the output to variable field.
  - e) Click Apply.
5. Insert a Message box action to hold the Python function output:
  - a) Double-click or drag the Message box > Message box action.
  - b) In the Enter the message to display field, select and insert the variable sOutput.

- c) Select the Close message box after option. Retain the default value of 5 seconds in the field.
  - d) Click Apply.
  - 6. Close the script execution session with a Python Script > Close action:
    - a) Double-click or drag Python Script > Close.
    - b) Click Save.
  - 7. Click the Run icon.
- The bot generates a message box with the text `Go Be Great!`. After 5 seconds, the message box disappears.

## Build advanced bots and packages

Learn how to build action packages and advanced bots that include custom features such as scripting, and API calls. Find recommendations on bot and action package design and reusability.

### Bot developer resources

- [Package Software Development Kit \(SDK\)](#)  
This document explains the requirements and process for creating and uploading an action package to your Enterprise Control Room.
- [Build a bot using REST Web Services and JavaScript actions](#)  
Use the Azure Cognitive Text Analytics API to get a subscription key and use the REST API and Enterprise A2019 JavaScript actions to build a bot.
- [Post to Salesforce through custom app with OAuth 2.0](#)  
Create a Salesforce custom app to get authentication credentials and use the Enterprise A2019 REST Web Service and String Operation actions to build a bot. After creating the app, you will get the authentication credentials. Use the credential values to create variables and build a bot using the Enterprise A2019 actions.
- [Use Python to build a bot to parse JSON response](#)  
Use the Enterprise A2019 Python script to execute Python functions to build a bot. Use the Python functions to parse the JSON response from a REST Web Services GET request.
- [Use JavaScript to build a bot to take user input](#)  
Use the JavaScript actions Enterprise A2019 to execute JavaScript functions to build a bot. Use the actions to create a bot that takes user input and provides the appropriate output.

## Package Software Development Kit (SDK)

This document explains the requirements and process for creating and uploading an action package to your Enterprise Control Room.

Packages are Java Archive (JAR) files containing the executable third-party applications used to create bots. Actions are available in the Enterprise Control Room under the Action Panel.

The Automation Anywhere Package Development Kit provides detailed instructions for users to independently develop custom actions and upload and manage packages in their Enterprise Control Room.

The SDK includes sample code and supporting files for Java developers to create and validate custom actions.

[April 2020, Release \(2019.12.1\)](#)

[April 2019-package-sdk-2.0.1.zip](#)

[April 2020, Release \(2019.12\)](#)[A2019-package-sdk-2.0.0.zip](#)[March 2020, Release \(2019.11\)](#)[A2019-package-sdk-1.0.11.zip](#)

Note: All the components for this update to the Package SDK are included in this single zip file.

[February 2020, Release \(A2019.10\)](#)[A2019.10-package-sdk-1.0.0.zip](#)

Note: All the components for the Package SDK are included in a single zip file starting with the A2019.10 release.

[January 2020, Release \(A2019.09\)](#)

- SDK Demo Package: [A2019.09-packageSDK-1.0.0.zip](#)
- Documentation: [A2019.09-package-annotations-javadoc.zip](#)

[November 2019 Release \(A2019.08\)](#)

- SDK Demo Package: [A2019.08-packageSDK-1.0.0.zip](#)
- Documentation: [A2019.08-package-annotqtions-javadoc.zip](#)

[October 2019 Release \(A2019.07\)](#)

- SDK Demo Package: [A2019.07-Package-Sdk-1.0.0.zip](#)
- Documentation: [A2019.07-package-annotations-javadoc.zip](#)

[August 2019 Release \(A2019.06\)](#)

- SDK Demo Package: [A2019DemoPackage.zip](#)
- Documentation: [A2019-package-annotations-javadoc.zip](#)

For detailed release notes for the SDK Packages, see [Enterprise A2019 Package Development Kit Release Notes](#).

Click a title to read details about each task in the process.

[Setting up the Java project](#)

Set up an Integrated Develop Environment (IDE) for Java, including Automation Anywhere custom annotations to enable the development of action packages that can be uploaded to your Enterprise Control Room.

[Standard coding practices and guidelines for developing packages](#)

This topic covers standard coding practices and guidelines that help to ensure the development of high quality packages.

[Develop a sample package](#)

Develop your own package and upload it to an Enterprise Control Room to provide custom actions for bots.

[How to examples](#)

This section contains code examples and explanations about how to code some basic bot capabilities.

[Annotations](#)

This section provides reference information about the annotations used to create Automation Anywhere packages.

[Build and test a demo package and bot](#)

This practical how to section demonstrates that creating, changing, and managing packages allow you to customize actions and efficiently manage packages for all Enterprise Control Room users.

[Build and test a custom package](#)

Use IntelliJ to build a custom package and use Enterprise A2019 actions to test the package.

Related reference

[Enterprise A2019 Package Development Kit Release Notes](#)

## Setting up the Java project

Set up an Integrated Develop Environment (IDE) for Java, including Automation Anywhere custom annotations to enable the development of action packages that can be uploaded to your Enterprise Control Room.

## Prerequisites

A working knowledge of Java and Gradle is needed in order to successfully build an action package. You need the following software and file:

- Java Developer Kit (JDK) 11
- Java IDE
  - [Eclipse](#)
  - [Community edition of IntelliJ](#)
- Gradle plug-in v.5.\* in the IDE.
- Automation Anywhere Bot Agent installed and connected to valid Enterprise Control Room account
- Download and extract the zip archive files for the release you need:

[April 2020, Release \(2019.12.1\)](#)

[April 2019-package-sdk-2.0.1.zip](#)

[April 2020, Release \(2019.12\)](#)

[A2019-package-sdk-2.0.0.zip](#)

[March 2020, Release \(2019.11\)](#)

[A2019-package-sdk-1.0.11.zip](#)

Note: All the components for this update to the Package SDK are included in this single zip file.

[February 2020, Release \(A2019.10\)](#)

[A2019.10-package-sdk-1.0.0.zip](#)

Note: All the components for the Package SDK are included in a single zip file starting with the A2019.10 release.

[January 2020, Release \(A2019.09\)](#)

- SDK Demo Package: [A2019.09-packageSDK-1.0.0.zip](#)
- Documentation: [A2019.09-package-annotations-javadoc.zip](#)

[November 2019 Release \(A2019.08\)](#)

- SDK Demo Package: [A2019.08-packageSDK-1.0.0.zip](#)
- Documentation: [A2019.08-package-annotqtions-javadoc.zip](#)

[October 2019 Release \(A2019.07\)](#)

- SDK Demo Package: [A2019.07-Package-Sdk-1.0.0.zip](#)
- Documentation: [A2019.07-package-annotations-javadoc.zip](#)

[August 2019 Release \(A2019.06\)](#)

- SDK Demo Package: [A2019DemoPackage.zip](#)
- Documentation: [A2019-package-annotations-javadoc.zip](#)

Initial set up of your IDE is important so that you have the correct environment for creating Automation Anywhere action packages.



Important: The listed prerequisites are recommendations for use with all the sample code and instructions included in this package development kit.

## Procedure

1. Extract the content of the associated sample zip files to a folder you can access from your IDE.
2. Import the content in any java IDE of your choice as a Gradle project.
3. Edit the settings.gradle and change the root project name to something meaningful to you.  
Tip: The settings.gradle file is included in the A2019DemoPackage.zip file.
4. Go to src > main > resources > package.template and change the package name and related information to something meaningful.

Original package.template	Updated package.template
<pre>{   "name": "A2019DemoPackage",   "label": "A2019DemoPackage",   "description": "Provides actions for                                 A2019DemoPackage operations.",   "group": "",   "artifactName": "",   "packageVersion": "",   "codeVersion": "",   "commands": [] }</pre>	<pre>{   "name": "YourPackageName",   "label": "Appropriate label",   "description": "Meaningful description for the actions                                 contained in the package.",   "group": "",   "artifactName": "",   "packageVersion": "",   "codeVersion": "",   "commands": [] }</pre>

Tip: The package template file controls the following names and labels of your package.

- "name" is the JAR file name  
Package file naming conventions:
  - No spaces
  - No special characters
- "label" is what appears in the Action panel of your Enterprise Control Room

Related concepts

[Standard coding practices and guidelines for developing packages](#)

[How to examples](#)

[Annotations](#)

[Build and test a demo package and bot](#)

Related tasks

[Develop a sample package](#)

## Standard coding practices and guidelines for developing packages

This topic covers standard coding practices and guidelines that help to ensure the development of high quality packages.

### Testing

Ensure high quality code. Write sufficient unit tests and integration tests for your package.

### Icons

Set proper icon for your package.

### Dependencies

Embed all the dependencies in your package JAR. Load the dependencies at run time by extracting them to a temporary location. Be sure to clean the temporary location after the dependencies are loaded.

### Dependent JAR files

Add dependent JAR files under dependencies in in the build.gradle file as implementation so that the the dependant JAR files are packaged.

```

. . .
dependencies {
    compileOnly name: 'command-annotations'
    compileOnly name: 'bot-runtime'
    compileOnly name: 'bot-api'
    implementation name: 'i18n-api'
    implementation name: 'mydependentjavafile.jar'
    apt name: 'command-processor'
    compileOnly group: 'org.apache.logging.log4j', name: 'log4
j-core', version: "$loggerVersion"
    testImplementation "org.testng:testng:$testNgVersion"
    testImplementation name: 'bot-runtime'
    testImplementation name: 'bot-api'
}
. . .

```

### Add new actions to exiting package

When adding new actions to existing package, make sure to do clean before packaging. Its always a good practice to do clean build - gradlew.bat clean build shadowJar

### Error messages

Provide meaningful error messages.

- Do throw meaningful error messages. For example, in local language using i18n APIs with `BotCommandException`, throw a new exception  
`BotCommandException(MESSAGES.getString("Run.Exception.InvalidWorkingDirPath"))`.
- Do not throw generic error messages, such as `ex.message`.

#### Basic validation

Use the validation annotation rules such as `@NotEmpty` included with this development kit. Do not add basic validations for your code. Read [Validation annotations](#) for more details.

#### Loops

Avoid long running loops in your code. Long running loops can cause high CPU usage, leading to errors such as, "Bot is unresponsive."

#### Add logging

Use the default log4J logger provided in the bot run time framework. Do not add your own logger. See the sample code for more details.

#### Related concepts

[How to examples](#)

[Annotations](#)

#### Related tasks

[Setting up the Java project](#)

[Develop a sample package](#)

## Develop a sample package

Develop your own package and upload it to an Enterprise Control Room to provide custom actions for bots.

## Prerequisites

Download and extract the contents from [A2019DemoPackage.zip](#). This package contains the necessary source code for the sample package.

You need to have a project created in a Java IDE. For details about setting up a project, read [Setting up the Java project](#).

The following high-level tasks provide the basic workflow for creating a package.

## Procedure

1. Create a java class.  
This class is the action you plan to publish in your package.  
Important: It is required that the class support the default constructor.
2. Add required business logic to the class.  
The following are the supported return types:
  - Void: Use this return type if your action does not return any value.
  - Value: Use this return type if your action returns any type of value.
3. Annotate the class with `BotCommand` and `CommandPkg` annotations to make the class eligible to be converted to an action.
4. Annotate the variable that accept values with `Idx` and `Pkg`.
5. Annotate the entry method with the `Execute` annotation.
6. From the action prompt, run `gradlew.bat clean build shadowJar`.  
The JAR file created from the build is located in `build/lib`.
7. From the Enterprise Control Room on the Bots > Package, click the Add package icon to upload JAR file.

## Next steps

From the Enterprise Control Room on the Bots > Package page, click the Add package icon to upload the JAR file.

Tip: To upload a package to a Enterprise Control Room, you need Upload package permission. Read details about how to add a package to an Enterprise Control Room: [Add packages to the Enterprise Control Room](#).

Related concepts

[Standard coding practices and guidelines for developing packages](#)

[How to examples](#)

[Annotations](#)

Related tasks

[Setting up the Java project](#)

## How to examples

This section contains code examples and explanations about how to code some basic bot capabilities.

### [Return a value from an action](#)

Set the following properties on CommandPkg to store the action's output in a variable.

### [Expose an action as a property](#)

An action can be exposed as property if it does not accept any parameter. This can be done by setting the following properties on CommandPkg.

### [Add a condition in a custom package for If condition](#)

Add conditions in a custom package.

### [Add an iterator in a custom package for Loop action](#)

Add an iterator in your package for Loop action.

### [Add debug logs of custom packages to bot\\_launcher.log file](#)

Logging can be added using log4j. The dependency is already added in the sample build.gradle.

### [Handle session in custom package](#)

Actions need to extract the required session from the SessionsMap by the session name.

There are three types of actions:

- Command\Action (default choice)
- Iterator
- Condition

Any action class supports only one method as an entry point. Annotate all parameters of the entry point method with `Idx`.

CAUTION: If you do not provide a public setter to member variables with Inject, compilation errors occur.

Related concepts

[Standard coding practices and guidelines for developing packages](#)

[Annotations](#)

Related tasks

[Setting up the Java project](#)

[Develop a sample package](#)

## Return a value from an action

Set the following properties on CommandPkg to store the action's output in a variable.

## Action return values

### return\_type

Defines the return type of the action. It usually matches the entry method's return type.

### return\_required

Makes the assignment operation compulsory when true.

### return\_label

The UI label when asking for the variable to store the value in.

```
@BotCommand
@CommandPkg(label = "Uppercase", name = "uppercase", description="Converts the
source string to upper case.",
icon = "uppercase.svg", node_label="Convert {{sourceString}} to upper case| an
d assign the result to {{returnTo}}|",
return_type=DataType.STRING, return_required = true, return_label="Assign the o
utput to variable",
property_name="uppercase", property_description="Converts the string to upper c
ase", property_type=DataType.STRING,
property_return_type=DataType.STRING) public class UpperCase {

    @Execute
    public Value<String> convert(
        @Idx(index = "1", type=TEXT)
        @Pkg(label="Source string")
        @NotEmpty
        String sourceString){
        return new StringValue(sourceString.toUpperCase());
    }
}
```

### Related concepts

[How to examples](#)

## Expose an action as a property

An action can be exposed as property if it does not accept any parameter. This can be done by setting the following properties on CommandPkg.

## Action property values

### property\_name

The name of the property, unique at action level, in auto-complete box this name would appear.

### property\_description

A description of the property.

### property\_type

The data type on which property operates, only if the type matches, the property will be appear in the auto-complete box.

### property\_return\_type

The data type for what property returns. If this type does not match with the field type where it is used, there will be validation error.

```
@BotCommand
@CommandPkg(label = "Uppercase", name = "uppercase", description="Converts the
source string to upper case.",
icon = "uppercase.svg", node_label="Convert {{sourceString}} to upper case| an
d assign the result to {{returnTo}}|",
return_type=DataType.STRING, return_required = true, return_label="Assign the o
utput to variable",
property_name="uppercase", property_description="Converts the string to upper c
ase", property_type=DataType.STRING,
property_return_type=DataType.STRING) public class UpperCase {

    @Execute
    public Value<String> convert(
        @Idx(index = "1", type=TEXT)
        @Pkg(label="Source string")
        @NotEmpty
        String sourceString){
        return new StringValue(sourceString.toUpperCase());
    }
}
```

### Related concepts

[How to examples](#)

## Add a condition in a custom package for If condition

Add conditions in a custom package.

## Create condition values in an Action

- To create a condition set `commandType` property of `BotCommand` annotation with value as `Condition`.
- To define the entry method of the condition use the annotation `ConditionTest`.

```
@BotCommand(commandType = Condition)
@CommandPkg(label = "File exists", name = "fileExists",
    description = "Checks the file exists condition.",
    node_label = "file exists at {{sourceFilePath}}", icon = "")
public class Exist extends AbstractCondition {
    @ConditionTest
    public boolean test(@Idx(index = "1", type = FILE) @LocalFile @Pkg(label =
"File path") @NotEmpty String sourceFilePath,
        @Idx(index = "2", type = NUMBER) @Pkg(label = "How lon
g you would like to wait for this condition
to be true?(Seconds)",
            default_value = "0", default_value_type = DataT
ype.NUMBER)
            @GreaterThanEqualTo("0") @LessThanEqualTo("99999") @Not
Empty @NumberInteger Double waitTimeout) {

        // Logic to check for the condition goes here

    }
}
```

Related concepts

[How to examples](#)

## Add an iterator in a custom package for Loop action

Add an iterator in your package for Loop action.

### Add an iterator to a Loop action

- To create an iterator, set `commandType` property of `BotCommand` annotation with value as `Iterator`.
- There are two methods required by iterator, and they are defined by `HasNext`, and `Next` annotations.

```

@BotCommand(commandType = BotCommand.CommandType.Iterator)
@CommandPkg(name = "loop.iterators.files",
    label = "For each file in folder",
    node_label = "for each file and assign file name and extension to {{returnTo}}",
    description = "Iterator for each file in folder.",
    return_type = DataType.DICTIONARY,
    return_sub_type = DataType.STRING,
    return_required = true,
    return_description = "Note: Access the 'name' key to access file name and 'extension'
    key to access the file extension.",
    return_label = "Assign file name and extension to this variable")
public class FileLoop extends AbstractCommandFileIterator {

    @Idx(index = "1", type = AttributeType.TEXT)
    @Pkg(label = "Folder path")
    @Inject
    @NotEmpty
    private String folderPath;

    @HasNext
    public boolean hasNext() {
        return getFileIterator(folderPath).hasNext();
    }

    @Next
    public Value<?> next() {
        Map<String, Value> returnValueMap = new HashMap<>();

        FileIterator fileIterator = getFileIterator(folderPath);
        String fileName = fileIterator.getNext();

        returnValueMap.put(FILE_NAME, new StringValue(fileIterator.getFileName(fileName)));
        returnValueMap.put(EXTENSION, new StringValue(fileIterator.getExtensio

```



```

n(fileName));

        return new DictionaryValue(returnValueMap);

    }

    public void setFolderPath(String folderPath) {
        this.folderPath = folderPath;
    }
}

```

Related concepts

[How to examples](#)

## Add debug logs of custom packages to bot\_launcher.log file

Logging can be added using log4j. The dependency is already added in the sample build.gradle.

```

@BotCommand
@CommandPkg(label = "Copy to",icon="assigntoclipboard.svg" ,name = "assignToClipboard", description
= "Accepts user input or a variable and assigns it to Clipboard", node_label="{{value}}")
public class AssignToClipboard {

    private static Logger logger = LogManager.getLogger(AssignToClipboard.class
);

    @Execute
    public static void assign(@Idx(index = "1", type = TEXT) @Pkg(label = "Value") @NotEmpty String
value) {

        logger.trace("Assigning '{}' value to clipboard.", value);
    }
}

```

```

    }
}

```

## Handle session in custom package

Actions need to extract the required session from the SessionsMap by the session name.

SessionsMap instance can be received using the Sessions attribute. The annotation can only be applied to class field and a corresponding public setter is expected. The variable must be of type Map<String,Object>.

```

@BotCommand
@CommandPkg(label = "Start session", name = "startSession", description = "Start new session",
icon = "pkg.svg", node_label = "start session {{sessionName}}|") public class Start {

    @Sessions
    private Map<String, Object> sessions;

    @Execute
    public void start(@Idx(index = "1", type = TEXT) @Pkg(label = "Session name",
        default_value_type = STRING, default_value = "Default") @NotEmpty String sessionName) {

        // Check for existing session
        if (sessions.containsKey(sessionName))
            throw new BotCommandException(MESSAGES.getString("xml.SessionNameInUse", sessionName));

        // Do some operation

        // Create new session
        sessions.put(sessionName, new Session(operation));

    }
}

```

```

    public void setSessions(Map<String, Object> sessions) {
        this.sessions = sessions;
    }
}

```

```

@BotCommand
@CommandPkg(label = "End session", name = "endSession", description = "End session", icon =
"pkg.svg", node_label = "End session {{sessionName}}|")
public class EndSession {

    @Sessions
    private Map<String, Object> sessions;

    @Execute
    public void end(
        @Idx(index = "1", type = TEXT) @Pkg(label = "Session name", default_value_type = STRING,
            default_value = "Default") @NotEmpty String sessionName) {

        sessions.remove(sessionName);

    }

    public void setSessions(Map<String, Object> sessions) {
        this.sessions = sessions;
    }
}

```

Related concepts

[How to examples](#)

## Annotations

This section provides reference information about the annotations used to create Automation Anywhere packages.

[Creation and function annotations](#)

List of the available creation and function annotations.

[Validation annotations](#)

Validates annotated strings and values used in your Java code.

## Related concepts

[Standard coding practices and guidelines for developing packages](#)

[How to examples](#)

Related tasks

[Setting up the Java project](#)

[Develop a sample package](#)

## Creation and function annotations

List of the available creation and function annotations.

Annotation	Description
BotCommand	<p>Makes the type eligible to be treated as an <code>action</code>. You can define 3 types of actions <code>commandType</code> property.</p> <ul style="list-style-type: none"> <li>• <code>Command\Action</code></li> <li>• <code>Condition</code></li> <li>• <code>Iterator</code></li> </ul> <p>Examples:</p> <ul style="list-style-type: none"> <li>• <code>@BotCommand(commandType = BotCommand.CommandType.Iterator)</code></li> <li>• <code>@BotCommand(commandType = Condition)</code></li> </ul>
CommandPkg	<p>Makes the type eligible for creation of <code>action package.json</code>. This annotation must be used with <code>BotCommand</code> to take effect.</p> <p>Pkg would participate in the activity only when this annotation is present.</p> <p>Example:</p> <pre>@CommandPkg(label = "Create", name = "createFile", description = "Creates a file", node_label = "{{filePath}}", icon = "file.svg")</pre>
ConditionTest	<p>Method annotated with this annotation will participate in the execution of <code>Condition</code>.</p>

Annotation	Description
	<p>This annotation can only be used when the BotCommand has commandType set as Condition.</p> <p>Exactly one method needs to be annotated when BotCommand annotation is present on the type. Failure to do so will result in compilation error.</p>
Execute	<p>Method annotated with this annotation will participate in the execution of BotCommand. Exactly one method needs to be annotated when BotCommand annotation is present on the type. Failure to do so will result in compilation error.</p> <p>Example:</p> <pre data-bbox="451 743 1299 1094"> @Execute public void create( @Idx(index = "1", type = FILE) @LocalFile @Pkg(label = "File", description = "e.g. C:\\MyDoc\\MyFile.doc") @ NotEmpty String filePath, @Idx(index = "2", type = CHECKBOX) @Pkg(label = "Over write an existing file") @NotEmpty Boolean isOverwrite) { createFile(filePath , isOverwrite); } </pre>
GlobalSessionContext	<p>Can only be applied to member variables and fetches the GlobalSessionContext through a setter.</p> <p>Example:</p> <pre data-bbox="451 1365 1299 1890"> @com.automationanywhere.commandsdk.annotations.Global SessionContext private GlobalSessionContext globalSessionContext ;  public void setSessionMap(Map &lt; String, Object &gt; sessionMap) { this.sessionMap = sessionMap; }  public void setGlobalSessionContext(com.automatio nanywhere.bot.service.GlobalSessionContext globalSess </pre>

Annotation	Description
	<pre>ionContext) {     this.globalSessionContext = globalSessionContext ; }</pre>
HasNext	<p>Method annotated with this annotation will participate in the execution of Iterator.</p> <p>This annotation can only be used when the BotCommand has commandType set as Iterator.</p> <p>Requires annotation Next to be present.</p> <p>Exactly one method needs to be annotated when BotCommand annotation is present on the type. Failure to do so will result in compilation error.</p>
Idx	Makes the annotated element part of hierarchy utilized for code and resource generation. In other words without this annotation no BotCommand related element annotations is processed.
Idx.Option	<p>An option represents the elements that would play in the hierarchy, but lend the values to the parents.</p> <p>Examples:</p> <ul style="list-style-type: none"> <li>RADIO</li> </ul> <pre>@Idx(index = "1", type = RADIO, options = {     @Idx.Option(index = "1.1", pkg = @Pkg(node_label = "[Delay.delayType.1.1.node_label]", label = "[Delay.delayType.1.1.label]", value = REGULAR)),     @Idx.Option(index = "1.2", pkg = @Pkg(node_label = "[Delay.delayType.1.2.node_label]", label = "[Delay.delayType.1.2.label]", value = RANDOM)) }) @Pkg(label = "[Delay.delayType.label]", default_value = "REGULAR", default_value_type = DataType.STRING)</pre>

Annotation	Description
	<pre data-bbox="516 279 915 352">@Inject private String delayType;</pre> <p data-bbox="477 405 584 430">• SELECT</p> <pre data-bbox="516 483 1300 1192">@Idx(index = "2", type = SELECT, options = {     @Idx.Option(index = "2.1", pkg = @Pkg(label = "[[ LaunchWebsite.browser.2.1.label]]", value = "DEFAULT")),     @Idx.Option(index = "2.2", pkg = @Pkg(label = "[[ LaunchWebsite.browser.2.2.label]]", value = "INTERNET_EXPLORER")),     @Idx.Option(index = "2.3", pkg = @Pkg(label = "[[ LaunchWebsite.browser.2.3.label]]", value = "FIREFOX")),     @Idx.Option(index = "2.4", pkg = @Pkg(label = "[[ LaunchWebsite.browser.2.4.label]]", value = "CHROME")) }) @Pkg(label = "[[LaunchWebsite.browser.label]]" , default_value = "DEFAULT", default_value_type = DataType.STRING) @NotEmptyStringbrowser)</pre>
Inject	<p data-bbox="431 1276 1312 1402">Makes an element eligible for injection into the annotated type's object. The injection is setter-based so a corresponding setter in the type is mandatory. The injected values would form the BotCommand parameter map using the name provided in Idx.</p>
Next	<p data-bbox="431 1455 1247 1512">Method annotated with this annotation will participate in the execution of Iterator.</p> <p data-bbox="431 1549 1320 1606">This annotation can only be used when the BotCommand has commandType set as Iterator.</p> <p data-bbox="431 1644 914 1669">Requires annotation HasNext to be present.</p> <p data-bbox="431 1707 1284 1764">Exactly one method needs to be annotated when BotCommand annotation is present on the type. Failure to do so will result in compilation error.</p>

Annotation	Description
Pkg	Makes an element participate in creation of package.json. This annotation is ignored when Idx is not present.

Related concepts

[Annotations](#)

## Validation annotations

Validates annotated strings and values used in your Java code.

Annotation	Description
CodeType	The MIME-type of the code to format.
CredentialOnly	Can only accept a credential value, no plain string allowed.
Equals	Validates the given String is equal to annotated String variable.
FileExtension	Validates the annotated String value ends with the supported extension type.
GreaterThan	Validates that the annotated number variable value is always greater than given numeric value.
GreaterThanEqualTo	Validates that the annotated number variable value is always greater than or equal to given numeric value.
LessThan	Validates that the annotated number variable value is always less than given numeric value.
LessThanEqualTo	Validates that the annotated number variable value is always less than or equal to given numeric value.
LocalFile	Can only accept local paths and no file expression.
MatchesRegex	Validates the annotated String value matches the for given regular expression.
NotEmpty	Validates and throws exception when the annotated variable value is null.  <pre>@Execute public Value&lt;Double&gt;length(@Idx(index="1",     type=TEXT)@Pkg(label="Source string")@NotEmpty String     sourceString) {}</pre>
NotEquals	Validates that the given String is not be equal to annotated String variable.



Annotation	Description
NotMatchesRegex	Validates the annotated String value does not match the for given regular expression.
NumberInteger	Ensures the UI accepts only integers not double for the annotated variable value.
RepositoryFile	Can only accept repository paths and no file expression.
VariableNotPackage	Cannot choose a variable from this package.
VariablePackage	Can only choose a variable from this package.
VariableSubType	The variable subtype that must match
VariableType	The variable type that must match
VariableUserDefined	Can only choose user-defined a variable.

Related concepts

[Annotations](#)

## Build and test a demo package and bot

This practical how to section demonstrates that creating, changing, and managing packages allow you to customize actions and efficiently manage packages for all Enterprise Control Room users.

Here is a list of all the necessary tasks to create a package, add the package to your Enterprise Control Room, and verify your work in a bot. Complete the listed tasks in order.

Tip: Click the title of each step to go the detailed task.

### Step 1 [Update related workflow and build files](#)

Follow the detailed steps for updating workflow and build files for this project using your integrated development environment (IDE).

### Step 2 [Choose your favorite IDE](#)

You can compile a package from the IDE of your choice. Here are two possible ways you can compile a package:

#### [Compile a demo JAR file from the Eclipse UI](#)

Use Eclipse to compile a demo JAR file that you can add as a package to your Enterprise Control Room.

#### [Compile a demo JAR file from the command line](#)

Compile the demo Java code provided with this software development kit.

### Step 3 [Add your demo package to an Enterprise Control Room](#)

Users with Upload package permission can add packages to the Enterprise Control Room for use by all Bot Creators.

### Step 4 [Create a demo bot with the demo package](#)

Create a bot using the demo package to verify the actions that were created.

### Step 5 [Change the Java file used to create the package JAR file](#)

Modify and compile the Java code used to create a package to fix issues and create new functionality.

### Step 6 [Upload new demo package](#)

Package management allows you to upload package updates. The new package has the same name, but a different version number.

### Step 7 [Update the demo bot with the updated package](#)

Update bots to use specific package versions.

# Update related workflow and build files

Follow the detailed steps for updating workflow and build files for this project using your integrated development environment (IDE).

## Prerequisites

Complete all the steps for project setup detailed in [Setting up the Java project](#).

This task shows you how to update the appropriate build and workflow files.

## Procedure

1. Open the project "A2019DemoPakage" that you created in [Setting up the Java project](#).  
You can find the project file in the directory where you extract the zip files to, for example c : \A2019DemoPackage.
2. From inside the project, Open the settings.gradle file.
3. Replace the project name with A2019DemoPackageFirstnameLastname.  
If your name is John Developer it would look like this, A2019DemoPackageJohnDeveloper.
4. Open src > main > resources > package.template.
5. Update the name, label, and description values.

Original package.template	Updated package.template
<pre>{   "name": "A2019DemoPackage",   "label": "A2019DemoPackage",   "description": "Provides actions for A2019DemoPackage operations.",   "group": "",   "artifactName": "",   "packageVersion": "",   "codeVersion": "",   "commands": [] }</pre>	<pre>{   "name": "A2019DemoPackageFirstnameLastname",   "label": "A2019DemoPackageFirstnameLastname",   "description": "A2019DemoPackageFirstnameLastname",   "group": "",   "artifactName": "",   "packageVersion": "",   "codeVersion": "",   "commands": [] }</pre>

Tip: The package template file controls the following names and labels of your package.

- "name" is the JAR file name  
Package file naming conventions:

- No spaces
  - No special characters
  - "label" is what appears in the Action panel of your Enterprise Control Room
6. Save the changes.

## Next steps

After you have setup the build files, you need to compile the demo Java code, [Compile a demo JAR file from the command line](#).

# Compile a demo JAR file from the Eclipse UI

Use Eclipse to compile a demo JAR file that you can add as a package to your Enterprise Control Room.

## Prerequisites

Before starting this task complete the steps in [Update related workflow and build files](#).

Build a package file using a Gradle project in the Eclipse IDE.

## Procedure

1. Import the A2019DemoPackage as a Gradle project, File > Import > Gradle > Existing Gradle Project and click Finish.
2. From the Gradle Tasks tab, go to <your project> > build and run the following tasks in order.
  - a) <your project> > build > clean
  - b) <your project> > build > build
3. From the Gradle Tasks tab, go to <your project> > shadow and run the shadowJar task.

Your compiled package file is located in file: \A2019DemoPackage\build\libs\ . The package file has named after your project name (<your project>-1.0.0.jar).

## Next steps

To add your custom package to your Enterprise Control Room follow the instructions in [Add packages to the Enterprise Control Room](#).

Related concepts

[Build and test a demo package and bot](#)

# Compile a demo JAR file from the command line

Compile the demo Java code provided with this software development kit.

## Prerequisites

Before starting this task complete the steps in [Update related workflow and build files](#).

## Procedure

1. Open a terminal window and go to where the gradlew.bat file is located.  
...\A2019DemoPackage > gradlew.bat
2. In the terminal window, type gradlew.bat clean build shadowJar, and press Enter. Here is an example of what you see:

```
> . . .\A2019DemoPackage>gradlew.bat clean build shadowJar

> Task :compileJava
Note: Starting hierarchy discovery for 'com.automationanywhere.botcommand.
demo.Concatenate'
Note: Starting non-hierarchical element discovery for 'com.automationanywh
ere.botcommand.demo.Concatenate'
Note: Starting hierarchy discovery for 'com.automationanywhere.botcommand.
demo.Uppercase'
Note: Starting non-hierarchical element discovery for 'com.automationanywh
ere.botcommand.demo.Uppercase'
Note: Starting Command Java generator...
Note: Starting Json generator...
Note: Generating command json for Concatenate
Note: Generating command json for Uppercase

> Task :commandCodeGen
mergeJsonFiles: updatePackage: group com.automationanywhere , artifactNam
e A2019DemoPackageFirstnameLastname ,
packageVersion 1.0.0-20190816-101906
```

The compiled file is located in file: \A2019DemoPackage\build\libs\.

## Next steps

To add your custom package to your Enterprise Control Room follow the instructions in [Add packages to the Enterprise Control Room](#).

Related concepts

[Build and test a demo package and bot](#)

## Add your demo package to an Enterprise Control Room

Users with Upload package permission can add packages to the Enterprise Control Room for use by all Bot Creators.

## Prerequisites

Before you can upload a package, you need valid user login credentials with Upload package permission for the Enterprise Control Room you are adding the package to.

## Procedure

1. From the Bots > Packages page, click the Add package icon.
2. Browse to the location of the package to add.  
Packages are Java Archive (JAR) files that contain actions used to create bots.
3. Select the package to add, and click Upload package.
4. On the Bots > Packages > Confirm package page, choose any of the following options:
  - Reject  
Stops the upload process.
  - Accept, enable and set as default  
Uploads and enables the selected package, and setting it to the default package for the Enterprise Control Room.
  - Accept and enable  
Uploads and enables the package, but the package is not set as the default package. Bot Creators have to specifically select non-default packages to use them for creating bots.

## Next steps

After successfully uploading your demo package, create a bot to test the actions you just created. For detailed step about how to create a bot, read [Create a demo bot with the demo package](#)

# Create a demo bot with the demo package

Create a bot using the demo package to verify the actions that were created.

## Prerequisites

Here are the minimum prerequisites for building this demo bot:

- Access to a Control Room
- User credentials with AAE\_Basic permission
- Your local host (workstation) is a registered device in the Control Room
- Ensure that the demo package A2019DemoPackageFirstnameLastname is available in the Enterprise Control Room

This task uses the following actions and components:

- Uppercase (demo package)
- [Message box](#)
- [Variables overview](#)

## Procedure

1. Go to Bots > My bots and click the My Task Bot icon.
2. Type  
`MyDemoBot1`  
in the Name field.
3. Click Create & Edit.
4. Expand A2019DemoPackageFirstnameLastname and double Uppercase.
5. Type  
`hello world, go be great!`  
in all lower case letters.
6. Create the variable  
`vMyDemoVar1`  
.
7. Click Apply.
8. Add a Message box and insert the variable vMyDemoVar1 in the Enter the message to display field.
9. Click Apply and Save.
10. Click the Run icon.  
A message box with "HELLO WORLD, GO BE GREAT!" in all upper case letters is displayed. The custom action Uppercase converted all the letters from lower case letters to upper case letters.

## Next steps

The task, [Change the Java file used to create the package JAR file](#), gives instruction on how to modify the Uppercase action to convert all upper case letters to lower case letters.

# Change the Java file used to create the package JAR file

Modify and compile the Java code used to create a package to fix issues and create new functionality.

## Prerequisites

## Procedure

1. Open the project "A2019DemoPakcage."
2. From inside the project, open src/main/java/com.automationanywhere.botcommand.samples.commands/basic/Uppercase .
3. Change the function from upper case to lower case.

Original function	Updated function
<pre>String result = "ALL".equals(caseType) ?     sourceString.toUpperCase() : (sourceString     .substring(0, 1).toUpperCase</pre>	<pre>String result = "ALL".equals(caseType) ?     sourceString.toLowerCase() : (sourceString     .substring(0, 1).toUpperCase</pre>

Original function	Updated function
<code>e() + sourceString .substring(1));</code>	<code>e() + sourceString .substring(1));</code>

4. Save the changes and re-compile the package.

## Next steps

You can now upload the changed package to the Enterprise Control Room. [Upload new demo package](#)

Related tasks

[Upload new demo package](#)

## Upload new demo package

Package management allows you to upload package updates. The new package has the same name, but a different version number.

## Prerequisites

You need AAE\_Basic permissions to create and edit bots.

## Procedure

1. From the Bots > Packages page, click the Add package icon.
2. Browse to the location of the package to add.  
Packages are Java Archive (JAR) files that contain actions used to create bots.
3. Select the package to add, and click Upload package.
4. On the Bots > Packages > Confirm package page, click Accept, enable and set as default.

## Next steps

You can select specific packages to be used from within a bot. Read detailed steps about managing packages for specific bots in [Update the demo bot with the updated package](#) task.

## Update the demo bot with the updated package

Update bots to use specific package versions.

## Prerequisites

- Access to the bot created in the task [Create a demo bot with the demo package](#).
- AAE\_Basic permission.

## Procedure

1. Go to Bots > My bots and double-click MyDemoBot1, the demo bot you created in an earlier task.
2. Click the vertical eclipses in the upper right corner and click Packages.
3. Expand the row for the package A2019DemoPackageFirstnameLastname.
4. From the drop-down list of package versions, select the Default version.  
Because you added the updated package as the default version, you are selecting the new version of the package you created.
5. Click Change Version and Save.
6. Go to Bots > My bots and double-click MyDemoBot1.
7. Click A2019DemoPackageFirstnameLastname and type  
HELLO WORLD, GO BE GREAT!  
in the Source string field.
8. Click Apply and Save.
9. Click the Run icon.  
The message box displayed by the bot displays "hello world, go be great!" This verifies that the action from the updated package is being used.

## Build and test a custom package

Use IntelliJ to build a custom package and use Enterprise A2019 actions to test the package.

Complete the following tasks to create a custom package, upload the package to your Enterprise Control Room, and build a bot to test it. After you update the IntelliJ files, create directories, and create and update java classes, you can build the custom package.

### 1. [Create and build a custom package using IntelliJ](#)

Use IntelliJ to import the Automation Anywhere SDK, create a new package, and compile a JAR file that you can upload to your Enterprise Control Room.

### 2. [Add custom package to your Enterprise Control Room](#)

Use the compiled JAR file and upload it to the Enterprise Control Room.

### 3. [Create a bot to test the custom package](#)

Use Enterprise A2019 actions to create a bot to test the custom package.

Related tasks

[Create and build a custom package using IntelliJ](#)

[Add custom package to your Enterprise Control Room](#)

[Create a bot to test the custom package](#)

## Create and build a custom package using IntelliJ

Use IntelliJ to compile a JAR file that you can upload as a package to your Enterprise Control Room in Enterprise A2019.



## Prerequisites

A basic understanding of JDK and Java IntelliJ is required in order to build an action package. You require the following software and files:

- [Java SE Development Kit 11 Downloads](#)
- Java IDE [Community edition of IntelliJ](#)
- Automation Anywhere A2019 SDK. Download and extract the zip files for the release you require: [Enterprise A2019 Package Development Kit Release Notes](#)

## Procedure

1. Unzip the contents of the SDK package to your IdeaProjects directory and rename the folder from A2019.10-package-sdk-1.0.0 to MetricToImperial.  
By default, the package is located at: C:\Users\<User>\IdeaProjects.
2. In IntelliJ IDEA, go to File > Open and open the project at C:\Users\<User>\IdeaProjects\MetricToImperial.
3. Open the settings.gradle file in the project root. Set the `rootProject.name = 'MetricToImperial'`
4. Update the package.template file located at `src > main > resources > package.template`.
5. Change the package name from A2019DemoPackage to MetricToImperial.
6. Update the package name in locales json: go to `src > main > resources > locales > en_US.json`.
  - a) Change the label and description fields as follows:

Original en_US.json	Updated en_US.json
<pre>{   "label" : "A2019DemoPackage" ,   "description" : "Provides actions for A2019DemoPackage operations." }</pre>	<pre>{   "label" : "Metric To Imperial",   "description" : "Converts lengths and distances from Metric format to Imperial format" }</pre>

- b) Delete all other remaining labels in the en\_US.json file.
7. Create a new Java Class, right-click the `metrictoimperial.commands` package, and select New > Java Class.  
Enter the name for the new class `CMtoINCH`:
    - a) Open the `CMtoINCH` class. Copy and paste the following code above the class definition statement:

```
import static com.automationanywhere.commandsdk.model.DataType.NUMBER;
//BotCommand makes a class eligible for being considered as an action.
@BotCommand
//CommandPks adds required information to be displayable on GUI.
@CommandPkg(
    //Unique name inside a package and label to display.
```

```

        name = "CM to Inch", label = "[[CMtoINCH.label]]",
        node_label = "[[CMtoINCH.node_label]]", description = "[[CMtoINCH.description]]", icon = "ruler_icon.svg",
        //Return type information. return_type ensures only the right
        kind of variable is provided on the UI.
        return_label = "[[CMtoINCH.return_label]]", return_type = NUMBER, return_required = true)

```

b) Inside the CMtoINCH class, copy and paste the following code:

```

//Identify the entry point for the action. Returns a Value<String> because the return type is String.
@Execute
public NumberValue action(
    //Idx 1 would be displayed first, with a text box for entering the value.
    @Idx(index = "1", type = AttributeType.NUMBER)
    //UI labels.
    @Pkg(label = "[[CMtoINCH.CMInput.label]]")
    //Ensure that a validation error is thrown when the value is null.
    @NotEmpty
    Double CMInput) {
    //Internal validation, to disallow empty inputs. No null check needed as we have NotEmpty on CMInput.
    if ("".equals(CMInput.toString().trim()))
        throw new BotCommandException("Input of CM is required");
    Number result;
    try {
        //Conversion logic
        result = CMInput * 0.393701;
    } catch (Exception e) {
        //Throw custom error message
        throw new BotCommandException("Unable to convert " + CMInput.toString() + "cm to inches");
    }
    //Return NumberValue.
    return new NumberValue(result);
}

```

The code automatically imports namespaces based on the annotations and datatypes.

To manually import namespaces, select a highlighted line, and press these keys simultaneously alt and the enter key.

8. Go to src > main > resources > locales > en\_US.json and add the following fields after the label and description values:

```
"CMtoINCH.label" : "cm to inches",
"CMtoINCH.node_label": "cm to inches",
"CMtoINCH.description" : "Convert centimeters to inches",
"CMtoINCH.return_label" : "Assign the Output in Inches to a Number Variable",
"CMtoINCH.CMInput.label" : "Centimeters to Convert to Inches"
```

9. Go to src > main > java > com.automationanywhere.botcommand, and delete the samples.commands package. Additionally, delete the samples package.
10. Update the CommandPkg annotation.
  - a) Download ruler\_icon.svg from [github](#) and right-click the image and save the image as ruler\_icon.svg.
  - b) Download iconwhite.svg from [github](#), right-click the image and save the image iconwhite.svg.
  - c) Copy both the files into the src > main > resources > icons folder.
11. Open the build.gradle in the project root. After the dependencies section, but before the last closing tag, copy and paste the following code:

```
test {
    testLogging {
        exceptionFormat = 'full'
    }
    useTestNG() {}

    afterSuite { desc, result ->
        if (!desc.parent)
            println("${result.resultType} " +
                "${result.testCount} tests, " +
                "${result.successfulTestCount} successes, " +
                "${result.failedTestCount} failures, " +
                "${result.skippedTestCount} skipped)")
    }
    maxHeapSize "3g"
}
```

12. In the project explorer, right-click src and select New > Directory.

a) In the Name field, enter

`test\java`

, or select the `test\java` suggestion from the Gradle Source Sets.

b) Right-click the `java` directory and select **New > Package**.

c) Enter the name for the newly created package:

`com.automationanywhere.botcommand.metrictoimperial.commands`

.

d) Right-click the new package and select **New > Java Class**. Enter the name for the new class

`CMtoINCHTest`

.

13. Inside the `CMtoINCHTest` class, copy and paste the following code:

```
@Test
public void testCMtoINCH() {
    Double validInput = 10.00;
    Double expectedOutput = 0.393701 * validInput;
    CMtoINCH testCM = new CMtoINCH();
    Value<Double> result = testCM.action(validInput);
    Assert.assertEquals(result.get(), expectedOutput);
}
```

14. Save the project **File > Save All**.

15. Build the package.

You can use the IntelliJ UI or the command line. If you are using the command line:

a) Open a terminal window and navigate to the `MetricToImperial` directory and enter:

`cd`

`"%USERPROFILE%\IdeaProjects\MetricToImperial"`

b) To build the project, enter the following command:

`gradlew.bat clean build shadowJar`

A **BUILD SUCCESSFUL** message appears.

Sometimes a build might fail because existing files could not be automatically deleted and a system message appears indicating the execution failed for the task: `clean`. If this occurs, close the explorer windows and run the build again.

## Next steps

[Add custom package to your Enterprise Control Room](#)

# Add custom package to your Enterprise Control Room

Use the compiled JAR file and upload it as a package to your Enterprise Control Room in Enterprise A2019.

## Prerequisites

- Complete the steps in [Create and build a custom package using IntelliJ](#).
- Ensure you have the following:
  - Access to the Enterprise Control Room.
  - Credentials with AAE\_Bot Store Publisher permission.

## Procedure

1. From Windows Explorer, go to C:\Users\<Username>\IdeaProjects\MetricToImperial\build\libs and locate MetricToImperial-1.0.0.jar.
2. Log in to your Enterprise Control Room as a user with permissions to add a new package.
3. Click BOTS > Packages.
4. In the All packages page, move your mouse over the plus sign and click Add package.
5. In the Add package page, click Browse and locate the MetricToImperial-1.0.0.jar file.  
By default, the file is located at: C:\Users\<Username>\IdeaProjects\MetricToImperial\build\libs.
6. Select the MetricToImperial-1.0.0.jar file and click Open.
7. Click Upload Package.

## Next steps

[Create a bot to test the custom package](#)

Related concepts

[Build and test a custom package](#)

Related tasks

[Add custom package to your Enterprise Control Room](#)

[Create a bot to test the custom package](#)

# Create a bot to test the custom package

Use the actions in Enterprise A2019 to create a bot and test the custom package.

## Prerequisites

Ensure you have the following to build the bot:

- Access to the Enterprise Control Room.
- Credentials with AAE\_Bot Store Publisher permission.
- Your workstation is a registered device in the Enterprise Control Room.
- Your package MetricToImperial-1.0.0.jar is available in the Enterprise Control Room.

## Procedure

1. Log in to the Enterprise Control Room.
2. Go to Bots > My bots, and click the Create a bot icon.
3. In the Name field, enter  
MetricToImperialTest
4. Click Create & Edit.

5. In the Actions pane, find Metric to Imperial and drag Convert centimeters to inches into the bot flow.
  - a) In the CM to Convert field, enter  
1
  - b) Click Create variable (next to the Output field) to create a new number variable.
  - c) In Create Variable, enter  
nInchesOutput  
and click Create & Select.
  - d) Click Apply.
6. In the Actions pane, find the Number package and drag the To String command below Convert centimeters to inches.
  - a) In the Enter a number field, enter F2 and select nInchesOutput.
  - b) In the Enter number of digits after decimal field, enter  
5
  - c) In the Assign the output to variable field, select prompt-assignment - String from the drop-down list.
  - d) Click Apply.
7. From the Actions pane, drag Message box, click F2 and insert the prompt-assignment variable.
8. In the Enter the message to display field, click F2 and select the prompt-assignment variable.
9. Click Apply and Save.
10. Click Run.  
The bot displays the 0 . 39370 message, which is a successful build.

#### Related concepts

[Build and test a custom package](#)

#### Related tasks

[Create and build a custom package using IntelliJ](#)

[Add custom package to your Enterprise Control Room](#)

## Enterprise A2019 Package Development Kit Release Notes

These release notes describe new features, changed features, fixed features, security fixes, deprecated features, and known limitations in the Enterprise A2019 Package Development Kit.

### Enhancements A2019.12.1

Feature	Description
Trigger samples	Added trigger samples in the Package SDK download file. Download the latest Package SDK to access the sample files.

[April 2019-package-sdk-2.0.1.zip](#)

### Enhancements A2109.12

Feature	Description
Run-time libraries	Added the latest run-time libraries.
Custom trigger	Creating custom triggers is supported in this release.

Feature	Description
Sample	We added more code samples.

[A2019-package-sdk-2.0.0.zip](#)

## Enhancements A2019.11

Feature	Description
Run-time libraries	This package contains the latest run-time libraries.

[A2019-package-sdk-1.0.11.zip](#)

## Enhancements A2019.10

Feature	Description
Updated annotation support	Added new attributes annotation support.
Improved bundling of SDK	Bundled java doc inside package SDK zip. You only need to download one zip file now.
Run-time libraries	Bundled latest bot run-time libraries.
More examples	Added new examples inside zip.
Datatypes	Added new datatypes.

[A2019.10-package-sdk-1.0.0.zip](#)

Note: All the components for the Package SDK are included in a single zip file starting with the A2019.10 release.

## Enhancements A2019.09

Feature	Description
Bot run-time libraries	Bundled latest bot run-time libraries for A2019.09 Package SDK.

- SDK Demo Package: [A2019.09-packageSDK-1.0.0.zip](#)
- Documentation: [A2019.09-package-annotations-javadoc.zip](#)

## Enhancements A2019.08

Feature	Description
Bot run-time libraries	Bundled latest bot run-time libraries for A2019.08 Package SDK.
Properties support	Extended support for properties.

Feature	Description
Comments expanded and improved	Added more comments to sample commands to help use SDK.
Comment field formatting	Added text color and background color to comment fields.

- SDK Demo Package: [A2019.08-packageSDK-1.0.0.zip](#)
- Documentation: [A2019.08-package-annotqations-javadoc.zip](#)

## Enhancements A2019.07

Feature	Description
Bot run-time libraries	The A2019.07 bot run-time libraries are bundled in the SDK package.
CREDENTIAL attribute	We provide support for credential attributes that require input from action screens.
Localized error messages and actions UI text	Enables developing packages with i18n error messages and localized (l10n) actions UI with localized text.
Java Development Kit 11	The A2019.07 Package Development Kit supports JDK 11.

- SDK Demo Package: [A2019.07-Package-Sdk-1.0.0.zip](#)
- Documentation: [A2019.07-package-annotations-javadoc.zip](#)

## Build a bot using REST Web Services and JavaScript actions

Use the Azure Cognitive Text Analytics API to get a subscription key and use the REST API and Enterprise A2019 JavaScript actions to build a bot.

Complete the following tasks to create a resource to get a subscription key, and use the Enterprise A2019 REST Web Services and JavaScript actions to build the bot. After you create a resource and deploy it, you will get a subscription key. Use the subscription key to create a variable and use Enterprise A2019 actions to build the bot.

### 1. [Set up the Azure portal](#)

Use the Azure Cognitive Text Analytics API to create a resource, get a subscription key, and then use a Reference API to send a POST request.

### 2. [Build a bot to parse JSON response using JavaScript](#)

Use the subscription key created in the Microsoft Azure to build a bot to parse the JSON response using Enterprise A2019 REST Web Services and JavaScript actions.



## Set up Azure portal

Use the Azure Cognitive Text Analytics API to create a resource, get a subscription key, and then use a Reference API to send a POST request.

Create an account with [Azure Cognitive Services](#).

## Procedure

1. Log in to your Azure account and click Portal.
2. Search and select Text Analytics from the drop-down list.
3. Click Create a Resource:
  - a) Enter a Name:  

`TextAnalyticsDemo2020`
  - b) Select the Subscription--use the default Azure subscription 1, select the Location, for example West US, and select the Pricing tier, for example F0 (5K Transactions per 30 days).
  - c) Create a Resource Group and provide a name, for example,  

`TextAnalyticsDemo2020RG`
  - d) Click Create.  
 When the Overview window opens, you should be able to see the following status: Your Deployment is complete.
4. Go to your resource  

`TextAnalyticsDemo2020`

 and select it.  
 You should see: `TextAnalyticsDemo2020 | Quick Start`.
5. From section 1, copy the K1 contents, for example, `0abfa73d93f1469d9d4b5db459394315`.  
 The Text Analytics API (v.2.1) page opens.
6. From section 2, click the API Console (V2) link:
  - a) Select API Reference > POST Sentiment.
  - b) In Sentiment, Select the testing console in the region where you created your resource:, select West US.  
 The Text Analytics API (v.2.1) Sentiment page opens.
  - c) In the Headers section, enter the value Ocp-Apim-Subscription-Key that you copied from the K1 field.
  - d) In the Request body, delete the existing content and replace it with the following code:

```
{
  "documents": [
    {
      "language": "en",
      "id": "1",
      "text": "I love this world"
    }
  ]
}
```

e) Click Send.

The response status is displayed as 200 OK. The Response content field shows: { "documents": [{ "id": "1", "score": 0.96714282035827637 }], "errors": [] }.

## Next steps

[Build a bot to parse JSON response using JavaScript](#)

### Build a bot to parse JSON response using JavaScript

Use the subscription key created in the Microsoft Azure to build a bot to parse the JSON response using Enterprise A2019 REST Web Services and JavaScript actions.

## Prerequisites

Ensure you have the following to build the bot:

- Created an account in the Azure portal and have the subscription key available.

[Set up Azure portal](#)

- Access to the Enterprise Control Room or Community Control Room
- Your workstation is a registered device in the Enterprise Control Room.

## Procedure

1. Log in to the Enterprise Control Room.
2. Go to Bots > My bots, and click the Create a bot icon.
3. In the Name field, enter  
API\_Sentiment\_Score\_Bot
4. Click Create & Edit.
5. In the Actions page, click Variables to create a new variable:
  - a) In the Name field, enter  
sURI
  - b) In the Type field, select String.
  - c) In the Default value field, access your previously created resource and paste the value here: <https://westus.api.cognitive.microsoft.com/text/analytics/v2.1/sentiment>.
  - d) Click Create.
6. Create a new Variable:
  - a) In the Name field, enter  
sSubscriptionKey
  - b) In the Type field, select String.
  - c) In the Default value field, paste the value from the K1 field:  
0abfa73d93f1469d9d4b5db459394315.
  - d) Click Create.
7. From the Actions pane, select REST Web Services > Post method and place it at the Start of the bot.
  - a) In the URI field, select \$sURI\$ and click Yes, insert.
  - b) In the Authentication Mode drop-down list, select No Authentication.

c) Click Header > Custom Headers > (Add):

a) In the Name field, enter

Ocp-Apim-Subscription-Key

.

b) In the Value field, select \$\$SubscriptionKey\$ and click Yes, Insert.

d) In the Content type drop-down list, select JSON (application/json).

e) In Custom Parameters, enter the following code:

```
{
  "documents": [
    {
      "language": "en",
      "id": "1",
      "text": "I love this world"
    }
  ]
}
```

f) Create a new dResponse variable of Type > Dictionary and Subtype > String.

g) In the Assign the output to a variable drop-down list, select dResponse -Dictionary of Strings.

h) Click Create & Select.

8. From the Actions pane, drag Message box to the bot. In the Enter the message to display field, select dResponse -Dictionary of Strings, and click Yes, insert.

9. Click Apply.

10. Create a new sOutput variable of Type > String.

11. From the Action pane, click Dictionary > Get.

a) In the Dictionary variable drop-down list, select dResponse -Dictionary.

b) In the Key field, enter

Body

.

c) In the Assign the output to a variable drop-down list, select sOutput -String.

d) Click Apply.

12. Create a new lParseOutput variable of Type > List and Subtype > String.

13. Create a new sSentimentValue variable of Type > String.

14. From the Action pane, click List > Add item.

a) In the List variable drop-down list, select lParseOutput -List.

b) In the Item to be added drop-down list, select sOutput -String.

c) In the Add Item field, select To end of list.

d) Click Apply.

15. From the Action pane, click JavaScript > Open.

a) In the JavaScript session field, select Default.

b) In JavaScript, select Manual input and enter the following script:

```
function extractValue(list) {var output = JSON.parse(list);return output.documents[0].score;}
```

c) Click Apply.

16. From the Action pane, click JavaScript > Run JavaScript.
  - a) In the JavaScript session, select Default.
  - b) In the Enter name of function to be executed, enter  
`extractValue`
  - c) In the Parameters drop-down list, select IParseOutput - List.
  - d) In the Assign the output to variable drop-down list, select sSentimentValue - String.
  - e) Click Apply.
17. From the Actions pane, drag Message box to your bot.
  - a) In the Enter the message to display field, enter:  
`Sentiment score: $sSentimentValue$`
  - b) Click Apply.
18. Click Save and click Run.  
 The following message appears: Your bot has run successfully!Sentiment score:  
 0.9671428203582764

## Post to Salesforce through custom app with OAuth 2.0

Create a Salesforce custom app to get authentication credentials and use the Enterprise A2019 REST Web Service and String Operation actions to build a bot. After creating the app, you will get the authentication credentials. Use the credential values to create variables and build a bot using the Enterprise A2019 actions.

### 1. [Create a custom app with OAuth 2.0 for authentication](#)

Create a custom app in Salesforce and get the access token to interact with REST APIs, and then use REST Web Service and String Operation actions to build a bot.

### 2. [Build a bot using REST web services and String actions](#)

Use the access token created in the custom app and build a bot using the REST Web Service and String Operation actions.

## Create a custom app with OAuth 2.0 for authentication

Create a custom app in Salesforce and get the access token to interact with REST APIs.

## Prerequisites

Ensure you have the following to build the bot:

- A basic knowledge of REST API and an understanding of OAuth concepts.
- An account created in Salesforce and a user name and password available.

[Salesforce Developers](#)

## Procedure

1. Log in to Salesforce and click Switch to Lightning Experience.
2. Click your Profile image and click Settings.
3. On the left side, click Reset My Security Token.  
After resetting the token action, a new token is sent to your registered email.
4. Check your email and save the security token.
5. In the Salesforce application, click Home, and search and select App manager.
6. Click New Connected App:
  - a) In the Connected App Name field, enter  
`OAuth_Validation_App`
  - .
  - b) In the Contact Email field, enter your email.
  - c) In the API section, select Enable OAuth Settings.
  - d) In the Callback URL field, enter  
`no:callbackURL`
  - .
  - e) In the Selected OAuth Scopes, select Access and manage your data (api).
  - f) Click Add and verify that the Access and manage your data (api) is shown on the right side.
  - g) Click Save.  
A system message is displayed: Allow from 2-10 minutes for your changes to take effect on the server before using the connected app.
  - h) Wait until the changes are applied and then click Continue.
7. When the OAuth\_Validation\_App page is displayed, go to the API section and copy the Consumer Key and Consumer Secret values. If it is hidden, select Click to reveal and copy the Consumer Secret key.
8. Click Manage and validate the OAuth Policies: verify that the Permitted Users field includes All users may self-authorize. If not, click Edit Policies and select All users may self-authorize.
9. Click Save.  
You can build a bot using the values from the  
`OAuth_Validation_App`  
, such as, ClientId = Consumer Key, ClientSecret = Consumer Secret, SecurityToken, and your Salesforce user name and password.

## Next steps

[Build a bot using REST web services and String actions](#)

### Build a bot using REST web services and String actions

Use the custom app authentication to generate the access token to interact with Salesforce REST APIs and build a bot using the Enterprise A2019 REST Web Service and String Operation actions.

## Prerequisites

Ensure you have the following to build the bot:

- Created a connected app and credentials available.  
[Create a custom app with OAuth 2.0 for authentication](#)
- Access to the Enterprise Control Room or Community Control Room

- Your workstation as a registered device in the Enterprise Control Room.

## Procedure

1. Log in to the Enterprise Control Room.
2. Go to Bots > My bots, and click the Create a bot icon.
3. In the Name field, enter  
`ConnectToAPIwithOAuth`
4. Click Create & Edit.
5. In the Actions page, click Variables to create a new variable:
  - a) In the Name field, enter  
`sClientId`
  - b) In the Type field, select String.
  - c) In the Default value field, paste the value from the Consumer Key field of the  
`OAuth`
  
`validation app`
  - d) Click Create.
6. Create a new sClientSecret variable, Type > String, Default value > Consumer Secret, paste the value.
7. Create a new sUserName variable, Type > String, Default value > enter the Salesforce login user name.
8. Create a new sPassword variable, Type > String, Default value > enter the Salesforce login password.
9. Create a new sSecurityToken variable, Type > String, Default value > Security Token, paste the value.
10. Create a new dResponse variable, Type > Dictionary and Subtype > String.
11. Create a new sExtractedValue variable, Type > String.
12. From the Actions pane, select REST Web Services > Post method and place it under the Start of the bot flow.
  - a) In the URI field, enter `https://ap17.salesforce.com/services/oauth2/token?grant_type=password&client_id=$sClientId&client_secret=$sClientSecret&username=$sUsername&password=$sPassword$sSecurityToken$`. Replace  
`ap17`  
 with your Salesforce instance number.
  - b) In the Authentication Mode drop-down list, select No Authentication.
  - c) In the Content type drop-down list, select JSON (application/json).
  - d) In the Assign the output to a variable drop-down list, select dResponse -Dictionary of Strings.
  - e) Click Apply.
13. From the Actions pane, drag Message box to the bot flow. In the Enter the message to display field, enter  
`$dResponse{Body}$`  
 . Click Apply.
14. From the Action pane, click String > Extract Text.
  - a) In the Source string field, enter  
`$dResponse{Body}$`
  - b) In the Before or after Start after text, enter  
`"access_token": "`  
 , Occurrence, enter  
`1`  
 , select AND, End before text, enter  
`", "`  
 , Occurrence, enter  
`1`

- . If no match found, return, select Source String. Number of characters to get, select All, click Trim the extracted text (remove blank spaces), and click Remove Enter from the extracted text.
  - c) In the Assign the output to a variable drop-down list, select sExtractedValue - String.
  - d) Click Apply.
15. From the Actions pane, drag Message box to the bot flow. In the Enter the message to display field, enter `$sExtractedValue$`. Click Apply.
16. From the Actions pane, select REST Web Services > Post method.
- a) In the URI field, enter `https://ap17.salesforce.com/services/data/v36.0/subjects/Account`. Replace `ap17` with your Salesforce instance number.
  - b) In the Authentication Mode drop-down list, select No Authentication.
  - c) Click Header > Custom Headers > (Add):
    - a) In the Name field, enter `Authorization`.
    - b) In the Value field, enter `Bearer $sExtractedValue$`.
  - d) In the Content type drop-down list, select JSON (application/json).
  - e) In Custom Parameters, enter the following code:
 

```
{
  "Name" : "Text passed through A2019 REST API"}
```
  - f) In the Assign the output to a variable drop-down list, select dOutput -Dictionary of Strings.
  - g) Click Apply.
17. From the Actions pane, drag Message box. In the Enter the message to display field, enter `$dOutput{Body}$`. Click Apply.
18. From the Action pane, select String > Extract text.
- a) In the Source string field, enter `$dOutput{Body}$`.
  - b) In the Before or after Start after text, enter `"id": "`, Occurrence, enter `1`, select AND, End before text, enter `", "`, Occurrence, enter `1`.
  - . If no match found, return, select Source String, Number of characters to get, select All, click Trim the extracted text (remove blank spaces), and click Remove Enter from the extracted text.
  - c) In the Assign the output to a variable drop-down list, create a new sURLValue by clicking the plus sign, Type > String.
  - d) Click Create & Select.
  - e) Click Apply.
19. From the Actions pane, drag Message box. In the Enter the message to display field, enter `$sURLValue$`.

. Click Apply.

A message from the bot displays a URL value 0014R00002gIcsZQAS. Use the value to validate the data in your Salesforce account: `https://<yourinstance>.salesforce.com/<sURLValue>`. For example: `https://ap17.salesforce.com/0014R00002gIcsZQAS`.

## Use Python to build a bot to parse JSON response

Use the Enterprise A2019 Python script to execute Python functions to build a bot. Use the Python functions to parse the JSON response from a REST Web Services GET request.

### Prerequisites

Ensure you have the following to build the bot:

- Basic understanding of Python programming language.
- Basic experience with creating Automation Anywhere bots.
- Download and install Python 3.

[download Python](#)

Add a path to the Environment variable, and select Add Python 3.8 to PATH during the Python installation.

### Procedure

1. Log in to the Enterprise Control Room.
2. Go to Bots > My bots, and click the Create a bot icon.
3. In the Name field, enter  

`PythonTutorial`
4. Click Create & Edit.
5. In the Actions pane, click Variables to create the following new variables:
  - a) Name dResponse, Type > Dictionary and Subtype > String.
  - b) Name dRetrieveValue, Type > Dictionary and Subtype > Any.
  - c) Name sResponseBody, Type > String.
  - d) Name sFullName, Type > String.
  - e) Name sLocation, Type > String.
  - f) Name sTotalUserCount, Type > String.
  - g) Name nTotalUserCount, Type > Number.
  - h) Name nCurrentUser, Type > Number.
  - i) Name sCurrentUser, Type > String.
6. From the Actions pane, select REST Web Services > Get method and place it under the Start of the bot flow.
  - a) In the URI field, enter <https://randomuser.me/api/?results=5&inc=name,email,location&nat=us>. This is a sample API that returns random user details to the calling application.
  - b) In the Authentication Mode drop-down list, select No Authentication.
  - c) In the Assign the output to a variable drop-down list, select dResponse -Dictionary of Strings.
  - d) Click Apply.
7. From the Actions pane, click String > Assign.
  - a) In the Select the source string variable value field, enter  

`$dResponse { Body } $`



- b) In the Assign the output to a variable drop-down list, select dResponseBody - String.
  - c) Click Apply.
8. From the Actions pane, click Python script > Open.
- a) In the Python field, select Manual input.
  - b) In the Enter script here field, copy and paste the following code.

```
import json

def get_node_count(response):
    #parse response as json
    response_dict=json.loads(response)
    # Create list from JSON body
    response_body = response_dict['results']
    #return the count of entries in JSON body as string
    lengthasString = str(len(response_body))
    return lengthasString

def get_full_name(dictRequest):
    itemCount = int(dictRequest['count'])
    #parse response as json
    response_dict=json.loads(dictRequest['response'])
    # Create list from JSON body
    response_body = response_dict['results']
    #Extract values to return
    return response_body[itemCount]['name']['first'] + " " + response_
body[itemCount]['name']['last']

def get_location(dictRequest):
    itemCount = int(dictRequest['count'])
    #parse response as json
    response_dict=json.loads(dictRequest['response'])
    # Create list from JSON body
    response_body = response_dict['results']
    #Extract values to return
    return response_body[itemCount]['location']['city'] + ", " + respo
nse_body[itemCount]['location']['state']
```

- c) In the Python runtime version field, retain the default value as 3.
- d) Click Apply.

9. From the Actions pane, click Python script > Execute function.
  - a) In the Python session field, retain Default.
  - b) In the Enter the name of function to be executed field, enter  
`get_node_count`
  - c) In the Arguments to the function drop-down list, select dResponseBody - String.
  - d) In the Assign the output to a variable drop-down list, select sTotalUserCount - String.
  - e) Click Apply.
10. From the Actions pane, click Dictionary > Put.
  - a) In the Dictionary variable field, select dRetrieveValue -Dictionary.
  - b) In the Associate to this key field, enter  
`response`
  - c) In the New value drop-down list, select dResponseBody - String.
  - d) In the Assign previous value to a variable drop-down list, select prompt-assignment - String.
  - e) Click Apply.
11. From the Actions pane, click String > To number.
  - a) In the Enter the string field, enter  
`sTotalUserCount`
  - b) In the Assign the output to a variable drop-down list, select nTotalUserCount - Number.
  - c) Click Apply.
12. From the Actions pane, click Loop > Loop.
  - a) In the Loop Type > Iterator, select For n times from the drop-down list.
  - b) In the times field, enter  
`$nTotalUserCount$`
  - c) In the Assign the current value to a variable drop-down list, select nCurrentUser - Number.
  - d) Click Apply.
13. From the Actions pane, click Number, select Decrement and place it inside the Loop action.
  - a) In the Enter number field, enter  
`$nCurrentUser$`
  - b) In the Enter decrement value field, enter  
`1`
  - c) In the Assign output to a variable drop-down list, select nCurrentUser - Number.
  - d) Click Apply.
14. From the Actions pane, click Number, select To string and place it inside of the Loop action, below Number > Decrement.
  - a) In the Enter a number field, enter  
`$nCurrentUser$`
  - b) In the Enter number of digits after decimal field, enter  
`0`
  - c) In the Assign output to a variable drop-down list, select sCurrentUser - String.
  - d) Click Apply.
15. From the Actiond pane, click Dictionary > Put and place it inside of the Loop action.
  - a) In the Dictionary variable field, select dRetrieveValue -Dictionary.
  - b) In the Associate to this key field, enter  
`count`

- c) In the New value drop-down list, select sCurrentUser - String.  
 d) In the Assign previous value to a variable drop-down list, select prompt-assignment - String.  
 e) Click Apply.
16. From the Actions pane, click Python script > Execute function and place it inside of the Loop action.  
 a) In the Python session field, retain Default.  
 b) In the Enter the name of function to be executed field, enter  
`get_full_name`  
 .  
 c) In the Arguments to the function drop-down list, select dRetrieveValue - Dictionary.  
 d) In the Assign the output to a variable drop-down list, select sFullName - String.  
 e) Click Apply.
17. From the Actions pane, click Python script > Execute function and place it inside of the Loop action.  
 a) In the Python session field, retain Default.  
 b) In the Enter the name of function to be executed field, enter  
`get_location`  
 .  
 c) In the Arguments to the function drop-down list, select dRetrieveValue - Dictionary.  
 d) In the Assign the output to a variable drop-down list, select sLocation - String.  
 e) Click Apply.
18. From the Actions pane, click Message box and place it inside of the Loop action.  
 a) In the Enter the message to display field, enter the following:  
`Full Name: $sFullName$ Location: $sLocation$`  
 b) Select Close message box after > Seconds, enter  
`5`  
 .  
 c) Click Apply.
19. From the Actions pane, click Python script > Close and place it outside of the Loop action.  
 a) In the Python session field, retain Default.  
 b) Click Apply.
20. Click Save to save your bot, and then click Run.  
 The bot runs, displaying five full user names with locations for approximately 5 seconds each before completing its successful execution.

## Use JavaScript to build a bot to take user input

Use the JavaScript actions Enterprise A2019 to execute JavaScript functions to build a bot. Use the actions to create a bot that takes user input and provides the appropriate output.

### Prerequisites

Ensure you have the following to build the bot:

- Basic understanding of JavaScript programming language.
- Basic experience with creating Automation Anywhere bots.

## Procedure

1. Log in to the Enterprise Control Room.
2. Go to Bots > My bots, and click the Create a bot icon.
3. In the Name field, enter

Hello

User

4. Click Create & Edit.
5. In the Actions pane, click Variables to create the following new variables:
  - a) Name lInputList, Type > List and Subtype > String.
  - b) Name sInputName, Type > String.
  - c) Name sOutput, Type > String.
6. From the Actions pane, click Prompt > For value:
  - a) In the Prompt window capture field, enter
- Enter your name here
- b) In the Prompt message field, enter
- Please enter your name here
- c) In the Assign the output to a variable drop-down list, select sInputName - String.
- d) Click Apply.
7. From the Actions pane, click List > Add item:
  - a) In the List variable drop-down list, select lInputList - List.
  - b) In the Item to be added drop-down list, select sInputName - String.
  - c) In the Add item field, select To end of list.
  - d) Click Apply.
8. Create a new JavaScript file.
  - a) Open a notepad, and copy and paste the following code:

```
function response(str) {
  return "Hello " + str
}
```

- b) Save the file in your desktop as Salutation.js, and select Save as type: as All Files.
9. From the Actions pane, click JavaScript > Open:
  - a) In the JavaScript session field, leave as Default.
  - b) In JavaScript > Import existing file, for the JavaScript file field, select Desktop file.
  - c) Click Browse and select the Salutation.js file from your desktop or where you saved the file.
  - d) In the Assign the output to a variable drop-down list, select sOutput - String.
  - e) Click Apply.
10. From the Actions pane, click JavaScript > Run JavaScript:
  - a) In the Enter name of the function to be executed field, enter
  - response
  - b) In the Parameters drop-down list, select lInputList - List.
  - c) Click Apply.
11. From the Actions pane, click Message box:
  - a) In the Enter the message to display field, enter
  - \$sOutput\$

- b) Click Apply.
- 12. Click Save to save your bot, and then click Run.
- 13. When the bot prompts to enter a name, enter the name and click OK.  
A message from the bot appears: Hello <User>.

## Bot developer recommendations

Automation Anywhere provides a flexible platform for bot and package development. The information in this topic provide guidelines and recommendations on how to structure and develop robust and reusable bots and packages.

- [Bot Store submissions checklist](#)  
Use the checklist to ensure that your Enterprise A2019 Bot Store submission is correctly created, processed, and accepted. If you do not follow these requirements, your submission will be rejected and will not be published on the Bot Store page.
- [Building reusable bots](#)  
Review the guidelines to gain a better understanding of how to develop bots or subtasks for reusability, from designing and creating through reusing.
- [Building reusable packages](#)  
Review the guidelines to understand how to build packages for reusability.

## Bot Store submissions checklist

Use the checklist to ensure that your Enterprise A2019 Bot Store submission is correctly created, processed, and accepted. If you do not follow these requirements, your submission will be rejected and will not be published on the Bot Store page.

Note: Ensure you have Enterprise A2019.11 or later for the Bot Store submission.

Item number	Item	Submission requirement
1	Register as a vendor (required for the first submission only)	<p>Every developer or an organization that posts bots on the Bot Store must register as a vendor.</p> <p>Use the Bot Store partner profile page to register using the <a href="#">Bot Store Vendor Profile page</a>. In this page, enter the details about the organization as well as the logos that can be used with the bots.</p> <p>This is a one-time process that has to be set up as a vendor. Bots submitted after this initial vendor registration process can all be linked to the existing registered vendor.</p>
2	Use the template for Readme file content and format	<p>Use the provided Readme template for submitting a bot or a package to the Bot Store: <a href="#">Bot Store template</a>. The Readme file template contains the necessary details on how to use the template.</p> <p>The Readme file must include all information that is required for your bot to be submitted to the Bot Store:</p>

Item number	Item	Submission requirement
		<ul style="list-style-type: none"> <li>Describe all included files and the purpose of each file.</li> <li>List the required inputs, outputs, and provide examples.</li> <li>If required, highlight any bot or package dependencies, such as, API keys, xls or csv files.</li> <li>For package: List actions and their purpose. Include the expected inputs and outputs.</li> </ul> <p>Submit the final Readme file in the PDF format. Do not include it as your bot dependency when uploading bots from the Enterprise Control Room.</p>
3	Follow file and folder structure	<p>Once imported, the Bot Store template is automatically installed to a directory named BotName - VendorName in the root of the Private Bot Store workspace of your Enterprise Control Room. Rename the folder to reflect the Vendor Name that you registered with in Step 1 of this checklist as well as the Bot Name for your submission. Also, rename the bot installed from the template as BotShell. The template contains basic error handling, logging, and snapshot files with customizable root logging location for maintaining older log files.</p> <ul style="list-style-type: none"> <li>All bot files and their dependencies must be contained within a single parent folder.</li> <li>Include all the dependencies, such as subfolders, subtasks, DLLs, scripts, and Python files.</li> <li>For package development, include a sample bot that demonstrates the package use. Use the same the naming structure as used for the bot submission.</li> <li>You can modify files and folders within the parent folder. When the bot and all its dependencies are ready for the submission to the Bot Store, it must be checked into the Public Bot Store workspace of your Enterprise Control Room.</li> </ul> <p>Naming example:</p> <p>If in step 1 of this checklist, you registered as a vendor named Bot Factory and named your bot as Currency Converter then your bot folder name must be Currency Converter - Bot Factory.</p>
4	Include images and videos	<p>As part of the bot or package submission process, one main image or a video and three to five additional images are required.</p> <ul style="list-style-type: none"> <li>Videos can demonstrate the full capabilities of the bot so that potential customers will know how it can be applied.</li> <li>Images help customers to decide whether or not to download the bot or package.</li> </ul>
5	Include a demo bot	<p>As part of the bot submission process, when submitting a bot or a package include a demo bot to show:</p> <ul style="list-style-type: none"> <li>how your bot or subtask can be used.</li> <li>how the actions within your package can be invoked.</li> </ul> <p>If your bot is designed to call a subtask from another bot, include a sample master bot that demonstrates setting values, invoking a subtask, and handling the subtask response.</p>

Item number	Item	Submission requirement
6	Follow the submission process	<p>The submission process for the Enterprise A2019 bot or package starts with the Enterprise Control Room:</p> <ol style="list-style-type: none"> <li>1. Check in your bots and all dependencies into the Bot Store Public workspace of on your Enterprise Control Room. In the My bots page, move your mouse over the Action toolbar, and click Submit to Bot Store.</li> <li>2. In the Submit to Bot Store page, review the dependencies that will be bundled with your bot (including a parent bot) and make changes as required. In addition, ensure that the bot and all dependent bots and files are in the same folder to submit to the Bot Store.</li> <li>3. After bots or packages have been submitted, navigate to the Bot Store &gt; My Submissions to complete the submission form.</li> </ol>

## Building reusable bots

Review the guidelines to gain a better understanding of how to develop bots or subtasks for reusability, from designing and creating through reusing.

Define prerequisites, input, output, and variables

When you build bots for reusability, define the following:

- Document all necessary prerequisites on how to use your bot either on its own or as a subtask.
- When creating your bots, define values as input, output, or local. Input and output variables are used when your bot is designed to be used as a subtask, allowing it to receive and pass back values to or from another calling bot.
- Provide meaningful variable descriptions when defining input and output variables so that other developers know how to interact with your subtask.
- Adhere to an established standard for variable naming guidelines. Review the Automation Anywhere user-defined variables for variable naming guidelines. [User-defined variables](#)

Follow single - responsibility principle

Bots developed for reusability should follow the single - responsibility principle which states that each subtask or component should have responsibility over a single part of the functionality of the overall bot and that responsibility should be entirely encapsulated by that subtask or component.

Other examples of single - responsibility:

- A subtask that processes a single transaction, but can be called multiple times for each transaction on a list.
- A subtask that collects screen display data on a single page of a website, but can be called multiple times as a bot goes through pagination.

Opening and closing applications

Any applications, files, or windows that a bot or subtask opens must be closed by the same bot or subtask.

- For example, when a bot opens Microsoft Excel to perform a spreadsheet operation, verify that the spreadsheet and Excel are closed when the bot finishes processing.
- Close applications when the bot execution is successful or unsuccessful.
- Use the Finally block of the Try/Catch/Finally operation to ensure applications are closed regardless of success of the task processing.
- In the case that applications do not respond during testing, consider using the command prompt to forcefully close (kill) the applications. For example, to forcefully close power point, the command-line operation would be:

```
Taskkill /IM powerpnt.exe /F
```

### Error handling

After completing the task, verify that the bot successfully handles any failure or exceptions.

- Each task or subtask must handle its own errors.
- An unhandled exception in a subtask can cause issues in a parent task.
- Use Try/Catch/Finally blocks at the root level of every bot.
- Use Try/Catch blocks inside of a loop if you want to try an operation multiple times before reporting a failure.

### Running bots on other computers

When designing a bot, enable it so it runs on computers other than the computer on which a bot was created.

- Use variables for local file paths, network shares, or window titles so that your bot can successfully run from other machines.
- Consider using global values for environment markers or network shares that multiple bots need access to.
- Use wildcard characters for window titles where appropriate to enable bots to run regardless of specific environment or version of the target application. For example, instead of using

```
Salesforce - Professional Edition - Internet Explorer
```

use the following:

```
Salesforce - * - Internet Explorer
```

### Using prompts, message boxes, and infinite loops

Prompts and message box actions stop the bots from running when waiting for a user input. Unless a user input is required, design the bots without using prompt statements.

- When using loops, ensure all loops have a definite end by clearly defining their number of iterations or specifying where break loop actions need to exist.
- If your bot is intended to run as an unattended bot, remove or disable any prompts or message box actions.
- If you are building bots for an attended automation scenario, message boxes and prompts are often reasonable or required for bots to run as expected. Use message boxes to display different variables, such as, responses, outputs, or values.



### Storing sensitive data in the Credential Vault

The Enterprise Control Room includes the Credential Vault that can be used to store sensitive information, such as user names, passwords, API keys, and tokens.

- When building a bot, create a locker in the Enterprise Control Room using the Credential Vault to store credentials and retrieve them as required by referencing the credential and the attribute. This allows users to create bots that consume APIs or perform logins without the need for bot builders to directly hard-code the required credentials within a bot.
- Do not hard-code sensitive credentials into a bot, or a subtask, because hard-coded storage in a bot introduces a security risk.
- When Credential Vault values are required to be used in a bot, verify that all locker names and credentials are clearly defined in the bot documentation. If required, include details on how to obtain credentials, for example, an API key or a token.

### Testing independent tasks

When creating bots for reusability, design them in a way that they can be tested independently of other subtasks.

- Practice the test-driven development (TDD) approach: When adding a new bot, or a new feature in an application, write a test case for it.
- In a test case, define the specific function that validates that feature or functionality.
- For single-responsibility principle and reusability, create many smaller tasks that can be tested independently.

### Using comments and steps

Comments enable developers to provide descriptions within their bots so that other bot developers can better understand what each section, block of code, or subtask is designed for. Include clear comments to allow developers to understand the purpose of the function of a given code block.

- When bots are submitted to the Bot Store, commenting demonstrates how to customize the bot.
- Using comments makes code maintenance easier because section descriptions help identify where changes might be required to enable developers to work towards quicker issue resolution.
- Comments on bots that are a work in progress can be helpful when creating placeholders for future work. Consider using a TODO command as a reminder to add logic to the bot, but update the comments when the work is completed.
- Enterprise A2019 includes the Step action, which provides the capability to organize the code into logical groupings to improve readability and the flow.
- Create an outline of the major objectives of your bot by using empty, labeled step actions. When that is completed, go back to each step and complete the logic for the step.

### Creating logging files

Identifying problems without logs can be difficult when bots are running unattended on any number of Bot Runners. Software developers, support teams, and bot owners rely on logs to understand where their automations have issues and how to diagnose problems. Bots must log errors to get error details.

- Use error handling and screen captures to better understand when a bot or subtask encounters an error.
- Use the A2019 Bot Store template that contains basic error handling, logging, and snapshot capabilities with the customizable root logging location for maintaining older log files.

[A2019 Bot Store bot template](#)

- If required, create additional logging files and include a full audit history of everything a bot or subtask has done. The additional log files can include audit, debug, and performance information about the bot, as well as the following:
  - Main bot start and end time.
  - Subtask start and end time.
  - The completion time of specific milestones defined within the bot.
  - Number of transactions received in an input file.
  - Number of successfully processed or failed transactions.

Related tasks

[Using Loop action](#)

Related reference

[User-defined variables](#)

[Application package](#)

[Error handler package](#)

[Step package](#)

[Prompt package](#)

[Message box package](#)

## Building reusable packages

Review the guidelines to understand how to build packages for reusability.

Know your incoming data

When setting fields that your action package requires from the user, provide specifics in setting the attribute type to limit the kinds of data that your package receives.

- Limit the input to reduce the burden of checks that have to be done when the package is received.
- Javadoc includes 34 defined attribute types, so review those when you build your package to select the appropriate field types.
- Set your package so that it takes a stored value. For example, on behalf of the bot, your package is making API calls, verify that the AttributeType of the action input field for the API key or a token is set to credential. This way users are encouraged to use a value stored in the Credential Vault for sensitive input data that the package requires.

Use labels appropriately

In the CommandPkg annotation, use different labels, node\_labels, and descriptions appropriately.

- Use these labels as short descriptions of your action and use only a few words to describe an action.
- Replicate the same naming style as it is presented in the default Action packages.
- Each action is a child element of a package, and the action label is displayed along with the package icon in the Actions pane. Use short names to describe each action.
- Document an expected input format for certain fields. Use the parameter description for the @Pkg annotation. This allows package developers to review the format, requirement, or data that must be used for a specific input field. For example:

```
@Pkg(label = "Start Date", description="Date Format as MM/DD/YYYY")
```

### Unit test your components

During the package development, create unit tests to validate that each component and the action of the package is working as expected.

- Validate the behavior of the individual test unit, a single class, or a single action, to ensure that it is functioning as expected.
- Review and document any feature or functionality defects at early stages of the development process.

### Handling errors

Include the error handling in the bot logic to ensure that all errors are handled gracefully. If an error is not handled, it could prevent a bot runner from executing other tasks.

- Create meaningful error messages that can help bot consumers with error resolutions.
- As a package developer, keep in mind these recommendations:
  - Use Try/Catch block to accommodate for an error.
  - Use a multi-catch block to find specific errors, and use the BotCommandException to return customized error messages. For example:

```
//create array of 3 items
int[] myIntArray = new int[]{1, 0, 7};
try {
    //print 4th item in array
    System.out.println(myIntArray[3]);
    //Perform operation on first and second items in array
    int result = myIntArray[0] / myIntArray[1];
} catch (ArrayIndexOutOfBoundsException e) {
    //Throw custom message for IndexOutOfBounds
    throw new BotCommandException("The array does have the number
of expected items.");
} catch (ArithmeticException e) {
    //Throw custom message on Arithmetic Exception
    throw new BotCommandException("Math Operation Error with " + Integer.toString(myIntArray[0]) + " and " + Integer.toString(myIntArray[1]));
}
```

### Follow single-responsibility principle

A package is a collection of actions. Each action within a package must have a single responsibility and that responsibility must be encapsulated by that action.

- Following the single-responsibility principle helps your package consumers to implement it easily, simplifies testing, and avoids unnecessary modifications.
- The actions that you offer allow package consumers to customize the way they use your package within their bots, and can help their bots be as efficient as possible.

### Provide examples

When submitting packages to the Bot Store include a demo bot that demonstrates the use of the package.

- Use the Enterprise A2019 actions and allow package consumers to use these actions to expend their bot capabilities.
- Always provide sample bots with descriptions to help your package consumers with the knowledge they require to understand its proper use.

### Related reference

[Building reusable bots](#)

## Troubleshooting and debugging

Troubleshooting and debugging information.

- [Troubleshoot bot run issue](#)  
Issue: I'm unable to deploy a bot because I get the following – Unexpected error setting up a new user session.
- [Debugger features](#)  
The Automation Anywhere Debugger provides tools to help identify and fix issues during bot development.
- [Bot agent log files](#)  
Various types of information about the Bot agent are captured in different log files. You can analyze these log files when the Bot agent or a bot encounters an error and identify the root cause for that error.

### Troubleshoot bot run issue

Issue: I'm unable to deploy a bot because I get the following – Unexpected error setting up a new user session.

#### Cause:

Most often, this error is caused by invalid device credentials.

#### Solution:

Confirm the device is registered and connected.

1. Select Devices > My devices from the Automation Anywhere web interface.
2. Confirm that the Status column shows Connected for the relevant device.

Confirm that the device username and password are correct.

1. Select the profile icon on the Automation Anywhere web interface and select Edit profile.
2. Verify the device username.

Depending on how the device is configured within the network, you might need to prepend it with the domain name. For example, domainname\firstname.lastname.

3. Verify the device password.

## Debugger features

The Automation Anywhere Debugger provides tools to help identify and fix issues during bot development.

To run a bot in Debugger mode, select the bot to run the debug function against and select Edit TaskBot. From the Edit TaskBot page, do the following:

1. Click the Debugger icon.
2. Click the Start icon.

Important: Use the List view to debug bots. The list view provides access to all of the Debugger features and visual indications of which action is running.

Debugger features:

- To debug your task one action at a time, insert a breakpoint next to each action. This makes the task pause at the breakpoint.
  - To insert a breakpoint, click the vertical ellipses and select Enable breakpoint.
  - To remove a breakpoint, click the vertical ellipses and select Disable breakpoint.
- Tip: You can enable and disable breakpoints in the Debugger mode, or in the regular edit mode.
- To move one action at a time, click the Step over icon.
- To clear all breakpoints, click the Clear all breakpoints icon.
- To stop the current debugging session, click the Stop icon.
- To exit debugging, click the Exit debugger menu.

Note: You cannot edit actions in Debugger mode.

## Bot agent log files

Various types of information about the Bot agent are captured in different log files. You can analyze these log files when the Bot agent or a bot encounters an error and identify the root cause for that error.

### Overview

The Bot agent log files enable you to perform these actions:

- Determine whether a bot ran successfully
- Identify the issues that resulted in a bot failure
- Determine if the device is properly connected with the Enterprise Control Room

### Log file locations

The Bot agent log files are available at C:\ProgramData\AutomationAnywhere\BotRunner\Logs on your device. The following files are available at this location:

#### Bot\_Launcher

Captures information about the execution of a bot such as the operations performed, events triggered, and errors encountered by the bot.

#### Node\_Manager

Captures information about when a bot is run from the Enterprise Control Room and communication details between a Bot Runner and the Enterprise Control Room.

## Log file configuration

The configuration files `botlauncher-logging` and `nodemanager-logging` are available at `C:\Program Files\Automation Anywhere\Bot Agent\config`. The information captured for the Bot agent in the log files depends on the configuration set in these files. You can update the configuration files to change the level of information captured in these files by setting the value in the `ROOT level` tag in the files, for example, `ROOT level="INFO"`.

You can configure the following modes in the configuration files:

### INFO

By default, this mode is set in the log files. This mode captures information used for monitoring the normal operations of a task and auditing. You can use the information collected in this mode to determine whether a business process was completed properly or not.

### FATAL

This mode captures information about the exceptions encountered while executing a task.

### DEBUG

This mode captures information that you can use to debug a bot. Information in this file is typically used by a Bot Creator.

### ALL

This is the most verbose logging level and logs all types of logging information including INFO, FATAL, and DEBUG.