

Robotic Process Automation with Automation Anywhere





Table of Contents

1. About Automation Anywhere: 4
 2. Installing Automation Anywhere: 19
 3. Overview of Automation Anywhere Control Room: 44
 4. Overview of the Automation Anywhere Development Interface: 69
 5. Building Your First Bot: 120
 6. Introducing Variables: 150
 7. Interacting with Applications: 187
 8. String Manipulation and List Variables: 237
 9. Working with Conditional Logic, Loops, and the Filesystem: 287
- 



Table of Contents

- 10: Working with XML Files: 323
 - 11: Automating Excel
 - 12: Automation Using Word
 - 13: Working with Emails
 - 14: Working with PDF Files
 - 15: Working with Databases
 - 16: Building Modular Bots and Sub-Tasks
 - 17: Running External Scripts
 - 18: Managing Errors
- 

About Automation Anywhere



About Automation Anywhere

We will cover the following topics in this lesson:

- What is robotic process automation?
- Overview of Automation Anywhere
- Automation Anywhere Versions
- Community Edition

Technical requirements

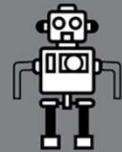
In order to use AA Community Edition, the following requirements are necessary:

- Windows OS version 7 or higher
- A processor with a minimum speed of 3 GHz
- A minimum of 4 GB RAM
- Internet Explorer v10 or higher, or Chrome v49 or higher
- An internet connection with a minimum speed of 10 Mb/second

What is robotic process automation?

- You probably already know what RPA is, but we will go through a quick overview here.
- The words automation or robot usually conjure up images of a physical machine performing repetitive tasks. We began to see this type of automation years ago, particularly in manufacturing. Physical robotic machines were built to help automate tasks usually done by humans.
- This form of industrial manufacturing automation was later adopted by many other industries including logistics, distribution, and packaging.

What is robotic process automation?

RPA	TRADITIONAL AUTOMATION
	
Requires no modifications in the current IT infrastructure	Requires certain customisations in current IT infrastructure
Lower costs of implementation when RPA is adopted	An option for companies with a flexible budget
A more efficient option since it can make improvements instantly	Requires more time, effort and considerable manpower
Eliminate the need to change the current processes in the system	Have to change the processes in the current system



Opening email
and attachments



Logging into web/
enterprise applications



Reading and writing
to databases



Copying and pasting



Filling in forms



Moving files and folders



Following "if/then"
decisions/rules



Collecting social
media statistics



Extracting structured
data from documents



Making calculations



Connecting to
system APIs



Scraping data from
the web

Overview of Automation Anywhere

	BluePrism	UiPath	Automation Anywhere
Free Community Edition	No	Yes	Yes
Front Office Development	No	Yes	Yes
Back Office Development	Yes	Yes	Yes
User Friendly Interface	Yes	Yes	Yes
Drag & Drop Development Feature	Yes	Yes	Yes
Interface Recorder Development Feature	No	Yes	Yes
Certification Available	Yes	Yes	Yes
Training Academy	No	Yes	Yes
Cloud Based Development	No	Yes	Yes
Bot Store Marketplace	No	Yes	Yes

The Digital Workforce

- A bot is referred to by AA as a Digital Worker as it clones the actions of a human to perform a given task. A Digital Worker is a member of the team designed to carry out a process just the same as any human worker.
- In a working environment, a team can consist of both humans and bots, hence the bot being referred to as a Digital Worker. As more bots are built within an organization, you can see a Digital Workforce being created.
- These bots can work side by side with a human or can be deployed to run on their own.

IQ Bot

- As well as utilizing condition-based decisions, more and more processes require a certain level of cognitive intelligence to make decisions.
- An example of this would be when dealing with unstructured data, A common scenario would be based on invoices, they all tend to have the same type of data such as supplier, items, costings, and dates, but the layout and format will vary between different suppliers.
- The consistency is not present when handling multiple suppliers. AA has developed a product called IQ Bot.
- This bot uses cognitive automation with RPA to learn how to handle unstructured data.

Bot Insight

- Designing and building bots is not the complete story, AA has also developed a platform that produces real-time analytics about your Digital Workforce, processes, and business-level processes, This is all a part of the Bot Insights tool, the RPA analytics tool for AA.
- Bot Insights is broken down into two categories: operational analytics and business intelligence.
- As bots are deployed, as well as executing tasks, they also process data, This data is related to each specific process and can provide valuable insight.
- Bot Insight analyzes this data and transforms it into meaningful insights.

Bot Store

- AA is the first RPA vendor to have a fully operational Bot Store.
- Bot Store is an online store with a collection of Digital Workers, The bots available here are built by independent developers from all around the world as well as AA themselves.
- AA Bot Store won the Silver award in the 2019 Edison Awards for developing the world's first and largest enterprise automation marketplace.
- These are complete bots out of the box that will perform a specific task or role.
- They are available as bots for specific applications, categories, or business processes.

Mobile Bot

- AA has also released a mobile app to work with your bots.
- It allows you to manage your Digital Workers from your mobile device.
- Bot Insight is available on the mobile app.
- This app will give you live alerts on bot performance as well as business insights on bot data.
- You can control your bots from the app including starting and stopping them.
- It also provides a platform for you to connect with the wider AA RPA community.

Automation Anywhere versions

ENTERPRISE VERSION 11	ENTERPRISE A2019	COMMUNITY EDITION A2019
Designed as an on-premises enterprise level RPA platform	Designed as a cloud-based enterprise level RPA platform	Designed for the developer and student
Free trial for 30 days	Free trial for 30 days	Free for small businesses, developers, and students
Included components: Enterprise – RPA on-premises IQ Bot, Bot Insight, Bot Store, Mobile App	Included components: Enterprise – fully web-based RPA in the cloud IQ Bot, Bot Insight, Bot Store, Mobile App	Included components: Enterprise – fully web-based RPA in the cloud IQ Bot, Bot Insight, Bot Store

Community Edition

- AA Community Edition is the latest free version available.
- The version prior, AA v11.x, used a client-server architecture where the management was done through the web-based Control Room app while the bot development was done through a client application installed on the desktop.
- Community Version is a fully cloud-based solution.
- The bot management and building are all done through the web application.
- No development client is installed on your desktop.

Summary

- You will now have a good understanding of AA and its competitors as well as what AA's capabilities are.
- Having registered with AA to use the free Community Edition , you must be keen to get AA up and running on your machine.
- In the next lesson, you will be guided through the installation process of AA.

Installing Automation Anywhere



Installing Automation Anywhere

We will cover the following topics in this lesson:

- Connecting to Control Room
- Preparing your device
- Configuring profile and device credentials

Technical requirements

In order to install the Automation Anywhere Community Edition Bot agent, the following requirements are necessary:

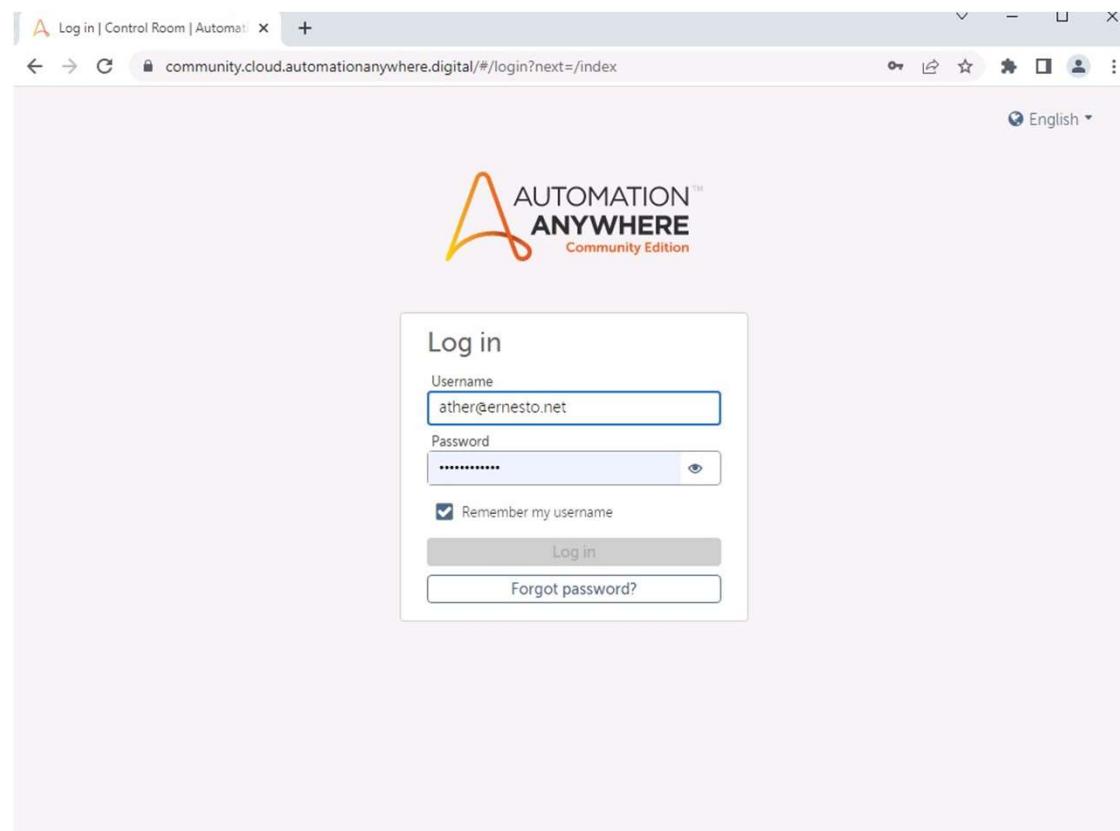
- Windows OS version 7 or higher
- A processor with a minimum speed of 3 GHz
- A minimum of 4 GB RAM
- At least 100 MB hard disk space
- Internet Explorer v10 or higher, or Chrome v49 or higher
- A minimum screen resolution of 1024*768
- An internet connection with a minimum speed of 10 Mb/second
- Have completed the registration process with Automation Anywhere for Community Edition AA 2019

Connecting to Control Room

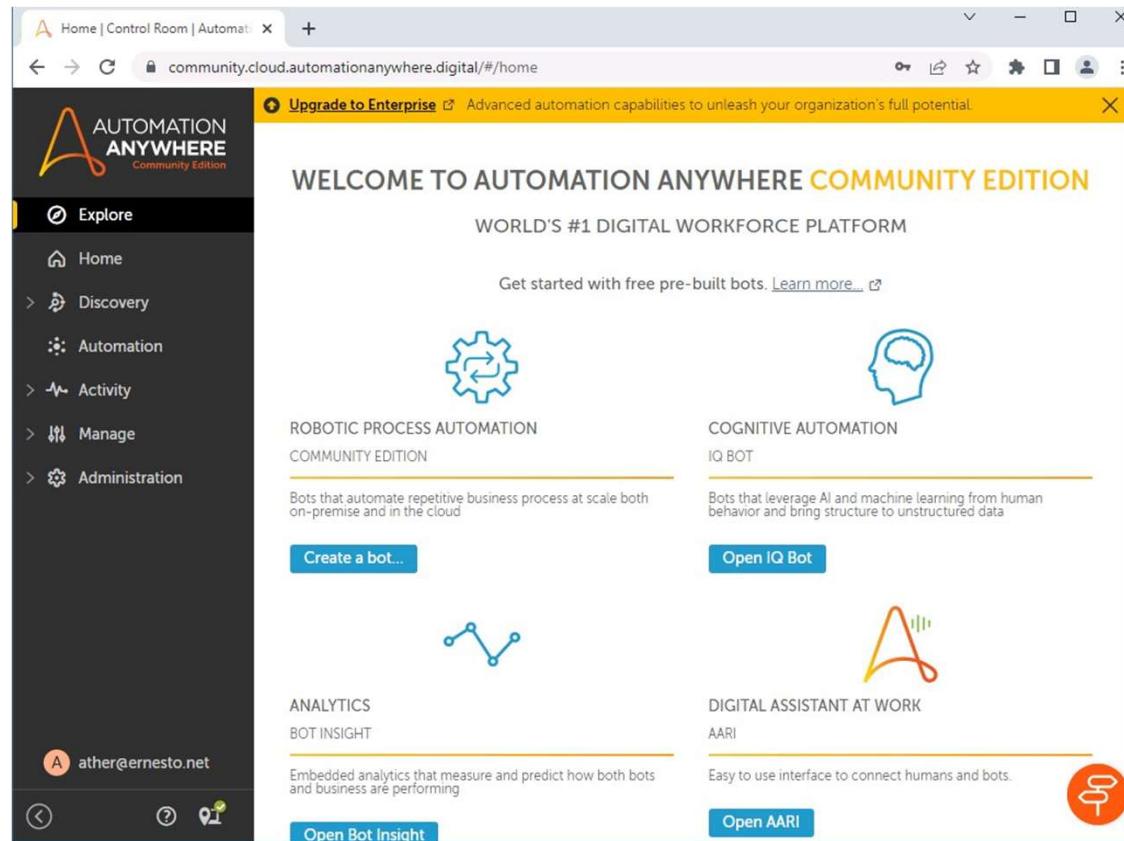
Continuing from lesson 1, About Automation Anywhere, after your registration with Automation Anywhere is complete, you should have received an email with the following details:

- Your Control Room URL
- Your username
- Your password

Launching and logging into Control Room

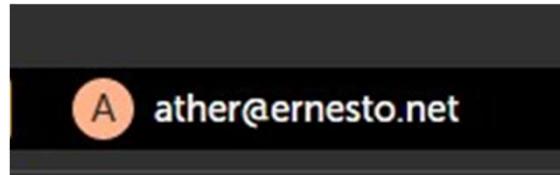


Launching and logging into Control Room

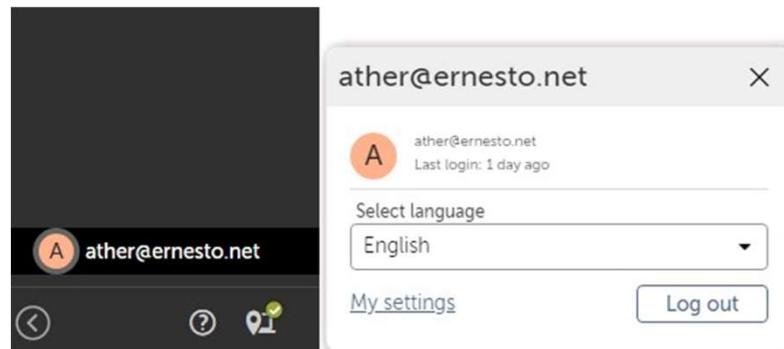


Updating your profile and password

- In the top right-hand corner of the Home page, click on your profile icon:



- From the dialog box, click on Go to My settings:



Updating your profile and password

- Will take you to your profile details interface; to make any changes to your profile, click on Edit:

The screenshot shows the 'My Settings' interface. At the top, there's a user icon (an orange circle with a white letter 'A') and the email address 'ather@ernesto.net'. To the right, there's a language dropdown set to 'English'. Below this, the 'My profile' section is expanded, showing 'General details' and 'Contact info'. Under 'General details', there are fields for 'First name' (N/A) and 'Last name' (N/A), with an 'Edit...' button. Under 'Contact info', there's an 'Email' field containing 'ather@ernesto.net'. In the 'Change password' section, there's a 'Password' field with five asterisks. At the bottom of the profile section, it shows 'Modified by' (ather@ernesto.net), 'Last login' (1 day ago), and 'Last modified' (1 day ago). Below the profile section, there's a partially visible 'Devices' section.

Updating your profile and password

- Will allow you to update your details.
- Here, you can update your password; once you have made your changes, click on Save changes to apply them:

My Settings

ather@ernesto.net

English ▾

My profile

General details

First name (optional) Last name (optional)
Maximum = 50 characters Maximum = 50 characters

Contact info

Contact email Confirm contact email
ather@ernesto.net ather@ernesto.net
Maximum = 255 characters

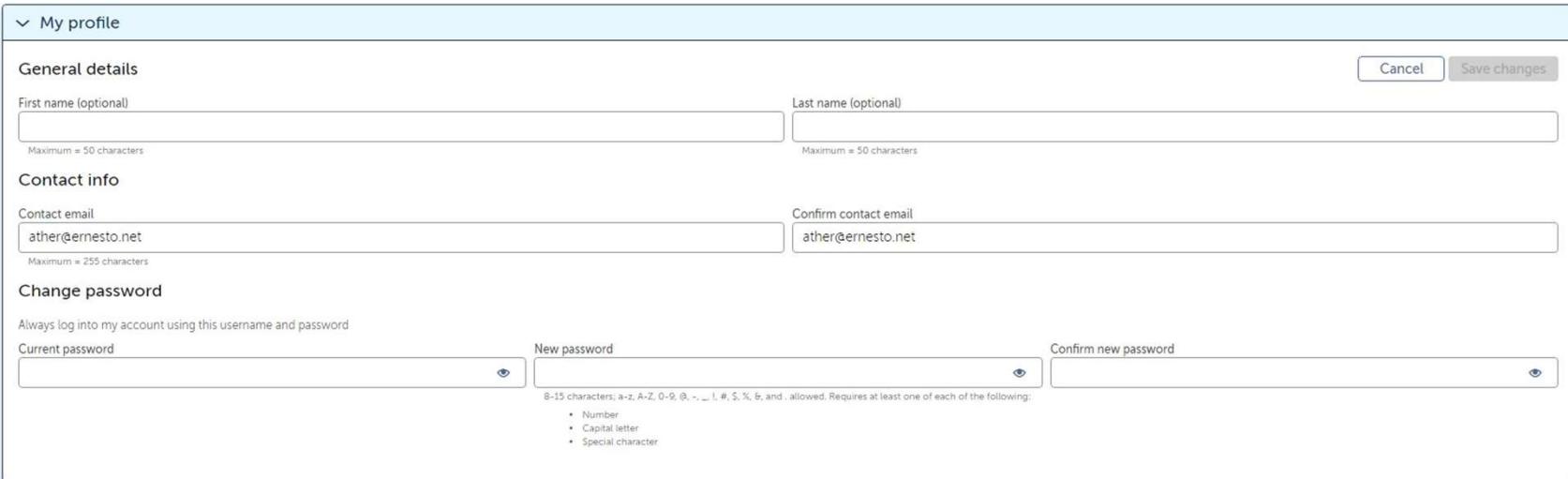
Change password

Always log into my account using this username and password

Current password New password Confirm new password
8-15 characters, a-z, A-Z, 0-9, @, ., _, #, \$, %, &, and . allowed. Requires at least one of each of the following:

- Number
- Capital letter
- Special character

Cancel Save changes



Preparing your device

To add a device, two stages are involved:

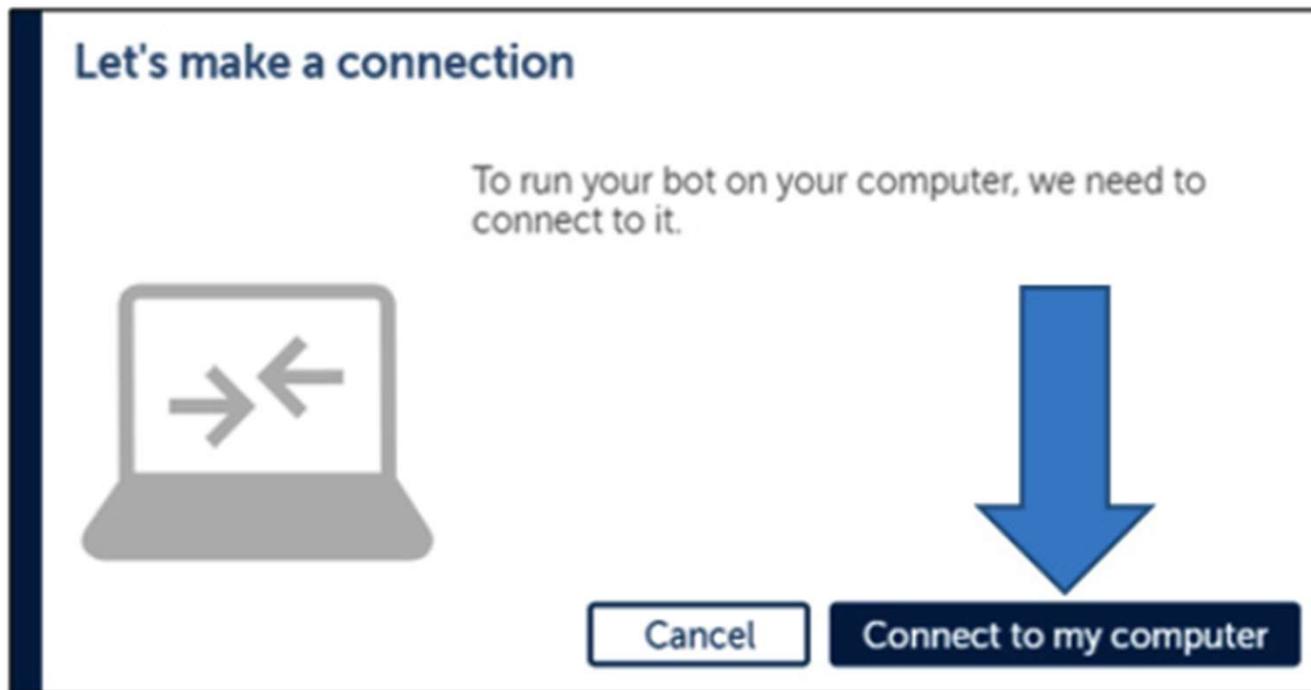
- Installing a Bot agent
- Enabling the extension

Installing a Bot agent

The screenshot shows the Automation Anywhere Community Edition web interface. The left sidebar menu includes Home, Discovery, Processes, Opportunities, Automation, Activity, Manage (with Learning Instances), Devices (which is selected and highlighted in yellow), Global values, Credentials, Packages, and Administration. The main content area is titled 'Devices' and shows a table with one device entry. The table columns are Status, Device name, Default users, Device nickname, and Lifespan. A yellow banner at the top right encourages upgrading to Enterprise with the text 'Upgrade to Enterprise' and 'Advanced automation capabilities to unleash your organization's full potential.' Below the banner are buttons for 'Run bot now...' and 'Connect local device...'. A search bar at the top allows filtering by 'Device name'.

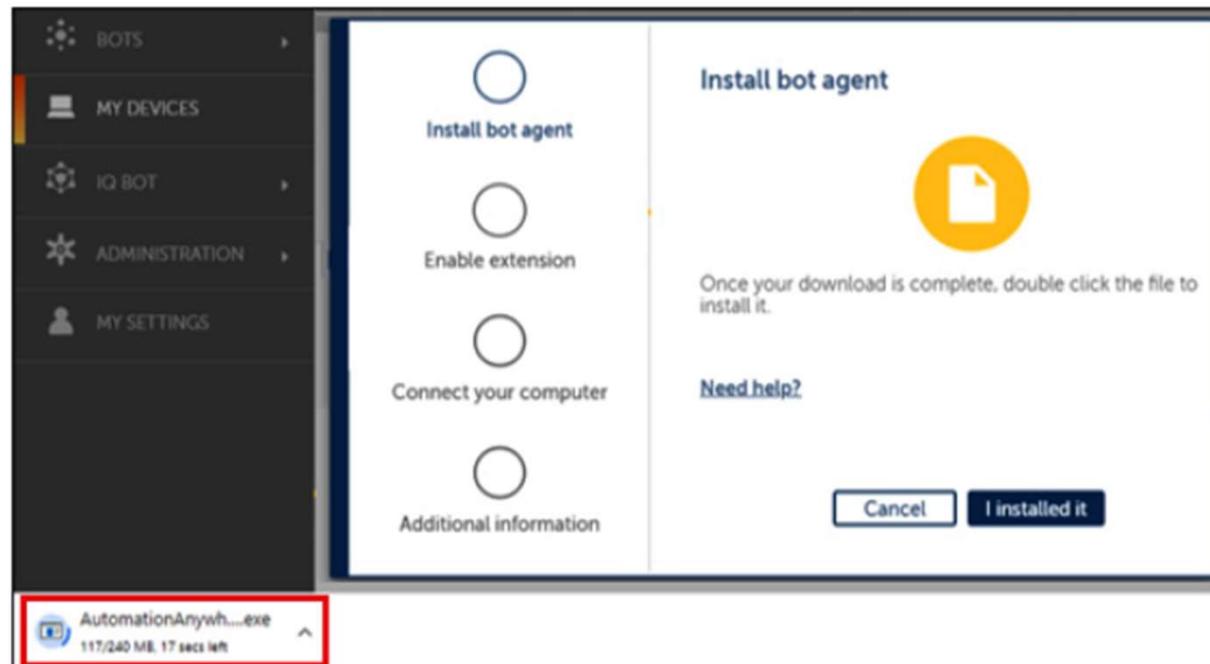
Installing a Bot agent

The connection wizard will pop up. Click on **Connect to my computer**:



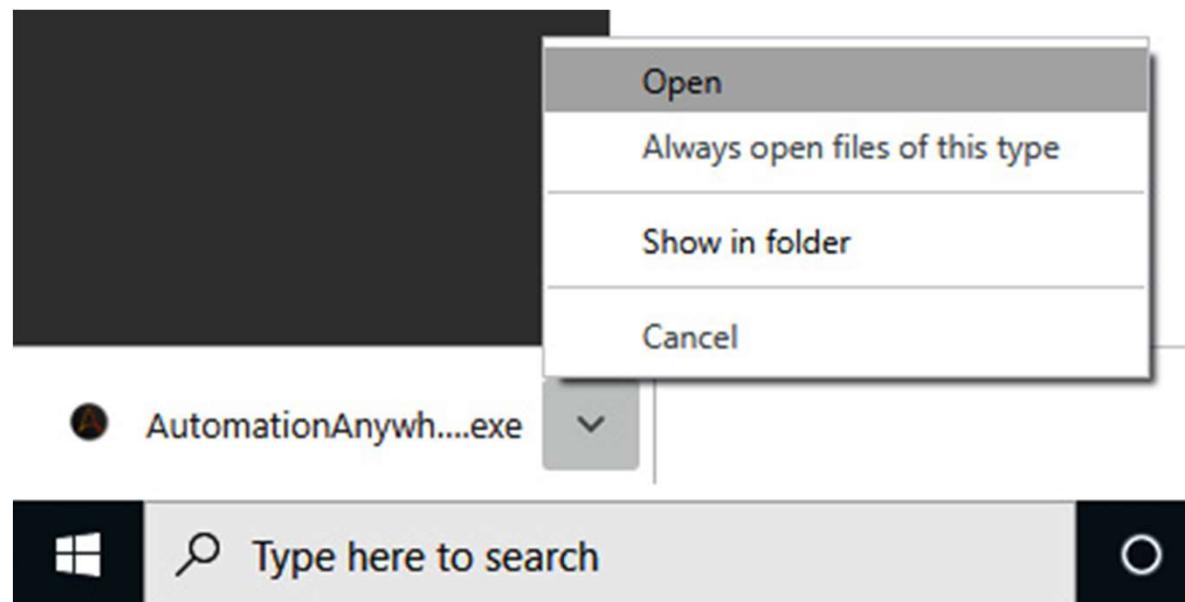
Installing a Bot agent

The wizard will start to download the Bot agent (into your default downloads folder) as shown in the following screenshot:



Installing a Bot agent

- Once the Bot agent has been downloaded, right-click on the downloaded file's icon and select Open:



Installing a Bot agent



Install bot agent



Enable extension



Connect your computer



Additional information

Enable Chrome extension

This will let bots run on this computer without device credentials. Without the extension a username and password will be required.

[Need help?](#)

Cancel

I enabled it

Enabling the extension



Install bot agent



Enable extension



Connect your computer



Additional information

Chrome extension not enabled

It might help if you get the extension from the Chrome store:

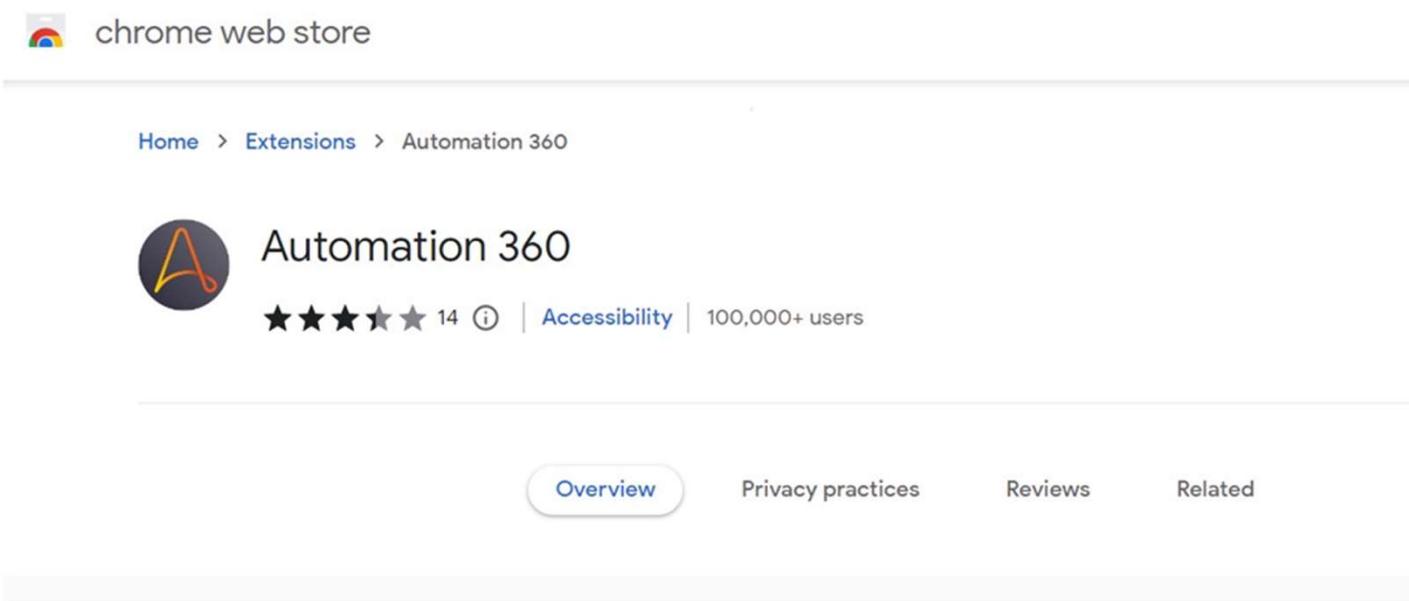
1. [Get the Automation Anywhere extension.](#)
2. Click "Add extension" to add it and enable it.
3. Once the extension is installed and enabled, we will automatically move to the next step.

[Skip this step.](#) I will add my username and password manually.

Cancel

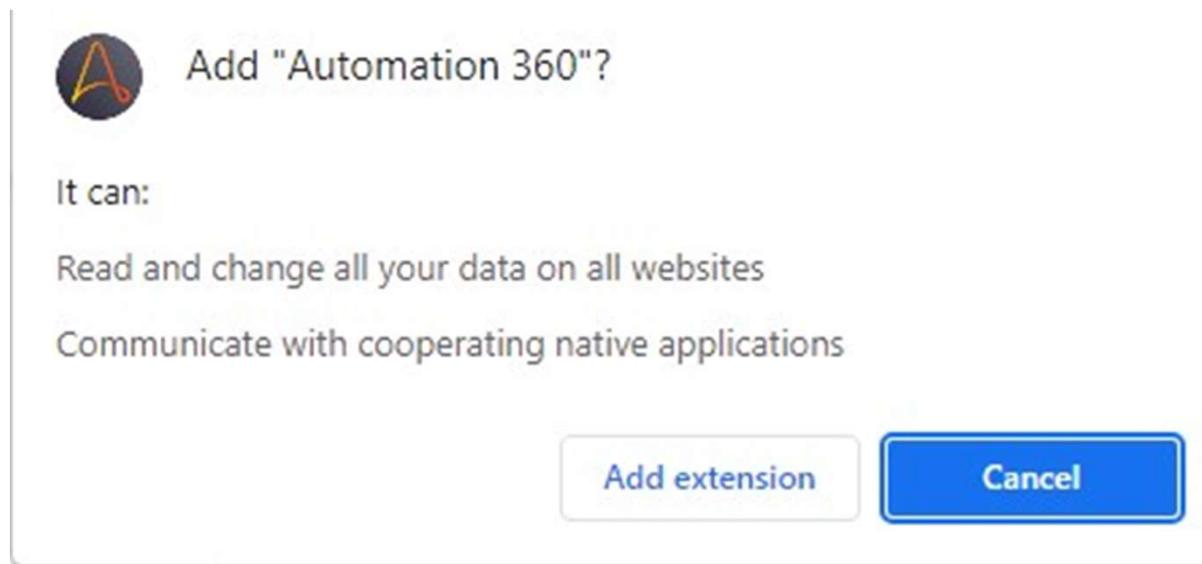
Enabling the extension

- Click on the Get the Automation Anywhere extension link to navigate to the Chrome Web Store:

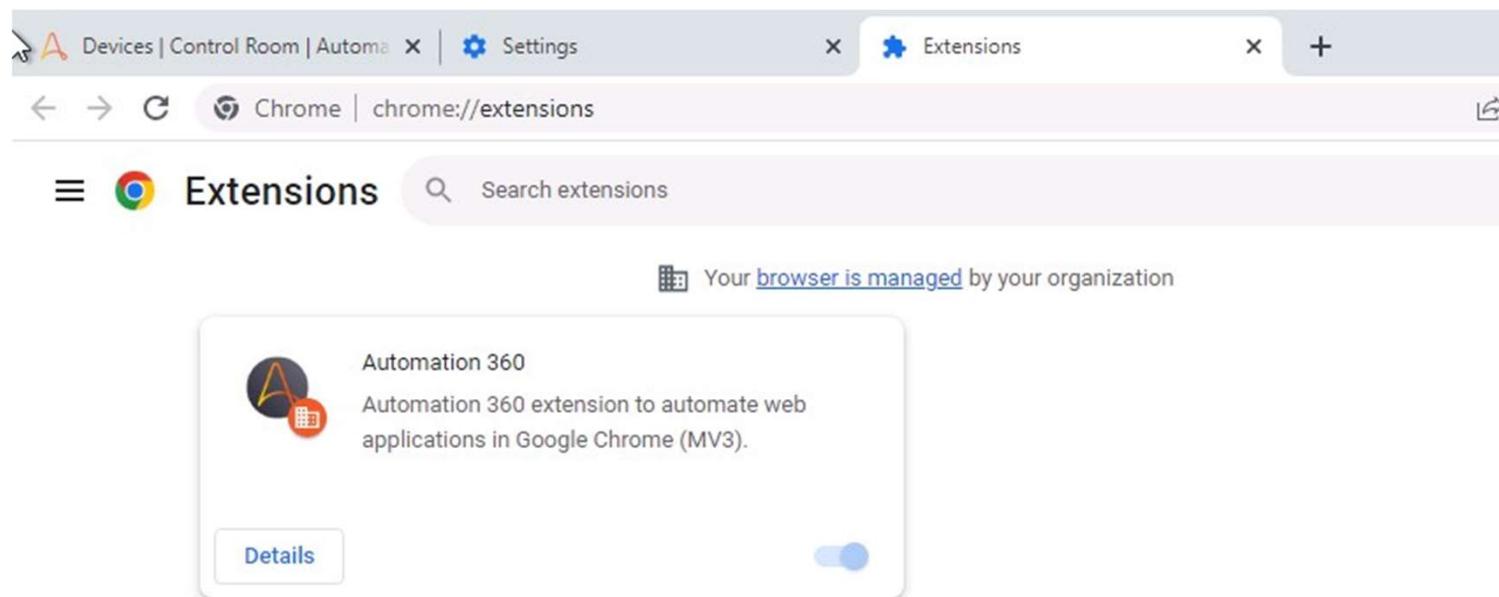


Enabling the extension

- Click on Add to Chrome button and you will get the following prompt; click on Add extension:



Enabling the extension



Enabling the extension

- Once the extension has been enabled, the progress indicator will be updated with further green ticks, shown as follows:



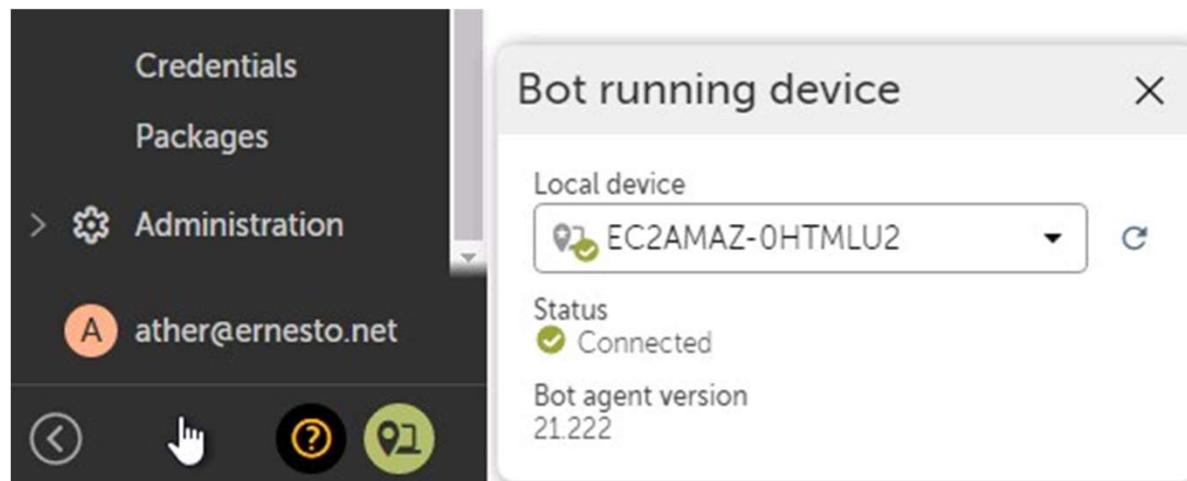
Enabling the extension

The screenshot shows the Automation Anywhere Community Edition web interface. The left sidebar has a dark theme with white icons and text. It includes links for Home, Discovery, Processes, Opportunities, Automation, Activity, Manage (with Learning Instances), and Devices, where 'Devices' is currently selected. The main content area has a yellow header bar with an 'Upgrade to Enterprise' button and a message about advanced automation capabilities. Below this is a section titled 'Devices' with a search bar and a 'Run bot now...' button. A table lists one device entry:

Status	Device name	Default users	Device nickname	Lifespan
Connected	EC2AMAZ-0HTMLU2	ather@ernesto.net	--	Persis :

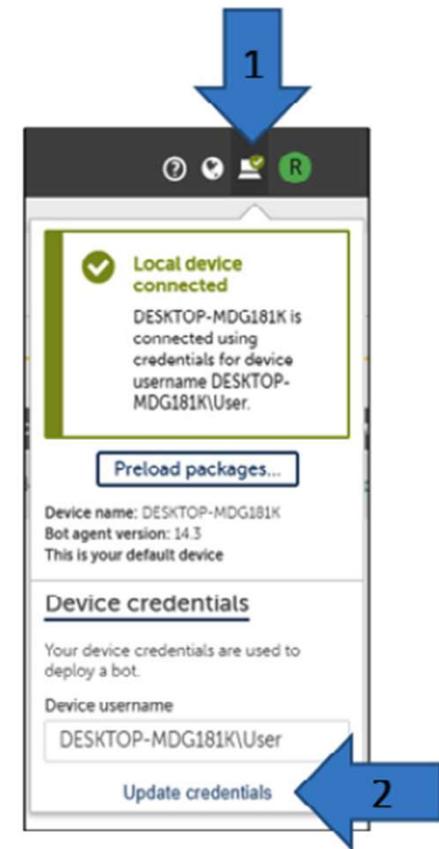
Enabling the extension

- The Bot agent should now be successfully installed on your device.
- You will also notice a little green tick against your local device icon on the top pane:



Configuring profile and device credentials

- Once in Control Room, from the Home screen, click on the local device icon, followed by Update credentials



Configuring profile and device credentials

- The dialog will allow you to update the Device password.
- Enter the user password for the device and click on Update

Device credentials

Your device credentials are used to deploy a bot.

Device username
DESKTOP-MDG181K\User 

Max characters = 255

Device password 

Device password is used for remote login.

Summary

- We are all ready to get going now.
- You will be comfortable in the future with setting up and configuring a device with Automation Anywhere Control Room, and can configure your device credentials as well as update your user profile.
- You could have a number of devices installed in Control Room.
- Once a bot is built, it can be deployed to any one of them or even a number of devices.
- Installing and configuring devices is an essential part of deploying and testing bots, and Control Room provides a centralized location to manage all your devices easily.

Overview of Automation Anywhere Control Room



Automation Anywhere Control Room

We will cover the following sections of Control Room in this lesson:

- Exploring the home screen
- Understanding the dashboard
- Viewing RPA activity
- Managing bots
- Managing My Devices
- Managing user administration

Technical requirements

- Windows operating system version 7 or higher
- A processor with a minimum speed of 3 GHz
- Minimum of 4 GB RAM
- At least 100 MB hard disk space
- Internet Explorer v10 or higher OR Chrome v49 or higher
- A minimum screen resolution of 1024*768
- An internet connection with a minimum speed of 10 Mb/sec
- Completed registration with Automation Anywhere Community Edition
- Logged on successfully to Automation Anywhere Community Edition
- A successfully registered local device

Exploring the home screen

- You will notice shortcuts to some of the exciting features available for you to explore:

The screenshot shows the home screen of the Automation Anywhere Community Edition web interface. The left sidebar has a dark theme with white text and icons. It includes sections for Explore (Home, Discovery, Processes, Opportunities, Automation, Activity), Manage (Learning Instances, Devices, Global values, Credentials, Packages), and Administration. A user profile for 'ather@ernesto.net' is at the bottom. The main content area has a light blue header with a yellow bar containing an 'Upgrade to Enterprise' button. Below this is a yellow banner with the text 'WELCOME TO AUTOMATION ANYWHERE COMMUNITY EDITION' and 'WORLD'S #1 DIGITAL WORKFORCE PLATFORM'. It features five main service cards: 'ROBOTIC PROCESS AUTOMATION COMMUNITY EDITION' with a gear icon and 'Create a bot...', 'COGNITIVE AUTOMATION IQ BOT' with a brain icon and 'Open IQ Bot', 'ANALYTICS BOT INSIGHT' with a line graph icon and 'Open Bot Insight', 'DIGITAL ASSISTANT AT WORK AARI' with a stylized 'A' icon and 'Open AARI', and 'PROCESS DISCOVERY DISCOVERY BOT' with a gear icon and 'Open Discovery Bot'. Each card has a brief description below it.

Exploring the home screen

- There are a number of options and information available in the top-right panel, These are as follows:



Access further documentation and help



Set your language preference for the application



View/Set local bot agent and device credentials



rpa_training@skysoftuk.net

View/Set user profile

Understanding the dashboard

The screenshot shows the Automation Anywhere Community Edition dashboard. The left sidebar includes links for Explore, Home (which is selected), Discovery, Automation, Activity, Manage, and Administration. The main content area features three main sections: 'Getting started' (with a 'Create a bot' button), 'Recently visited pages' (listing Home, My Settings, Devices, Packages, and Test | Edit Task Bot), and 'Insights' (describing Bot Insight). Below these are four metrics: '# of Task Bots created' (1), 'Most used actions' (represented by a yellow circle), '# of Task Bots run' (1), and 'Average time spent to create a Task Bot (across all users)' (15 min).

Understanding the dashboard

The available shortcuts on the dashboard are as follows:

- Creating a new bot
- Launching bot insights
- Your recently visited pages

Viewing RPA activity

The screenshot shows the Automation Anywhere web interface with the title bar "A In progress activity | Control Room". The URL in the address bar is "community.cloud.automationanywhere.digital/#/activity/inprogress". A yellow banner at the top right says "Upgrade to Enterprise" with a link to "Advanced automation capabilities to unleash your organization's full potential".

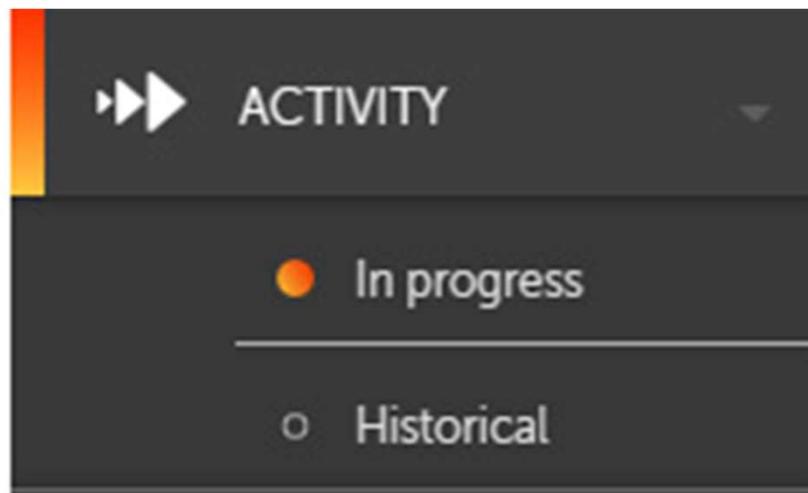
The left sidebar has a dark theme with the following navigation items:

- Explore
- Home
- Discovery
- Automation
- Activity
 - In progress
 - Historical
 - Insights
- Manage
- Administration

The main content area is titled "In progress activity". It features a search bar with "Status" dropdown and "Choose" button. Below is a table header for "Activity (0)" with columns: Status, Item name, Automation priority, Progress, Current action, Bot, Activity type, Started on, Device, and Username. A large circular button with two right-pointing arrows is centered. Below it is the text: "When there is activity, it will automatically appear here."

Viewing RPA activity

- There are two sub-sections to the Activity page, showing In progress and Historical:



Viewing RPA activity

- A number of options are available through the icons on the right, just above the activity list:



Export List to CSV file



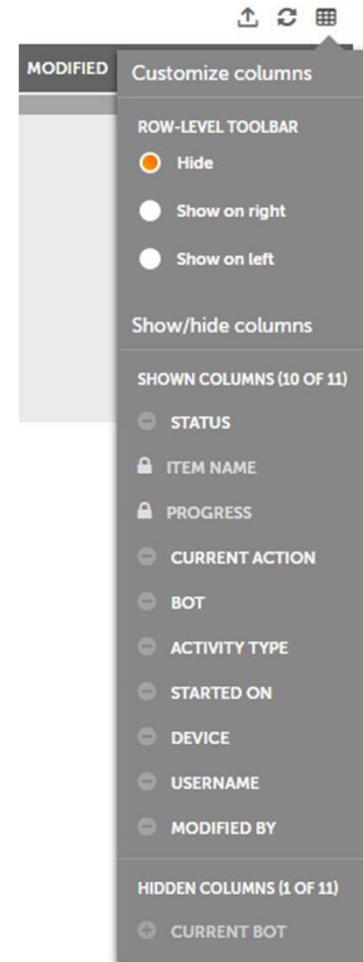
Refresh List



View/Hide Columns in List

Viewing RPA activity

- You can see all the columns that are available.
- Each one can be set to be visible or hidden:



Viewing RPA activity

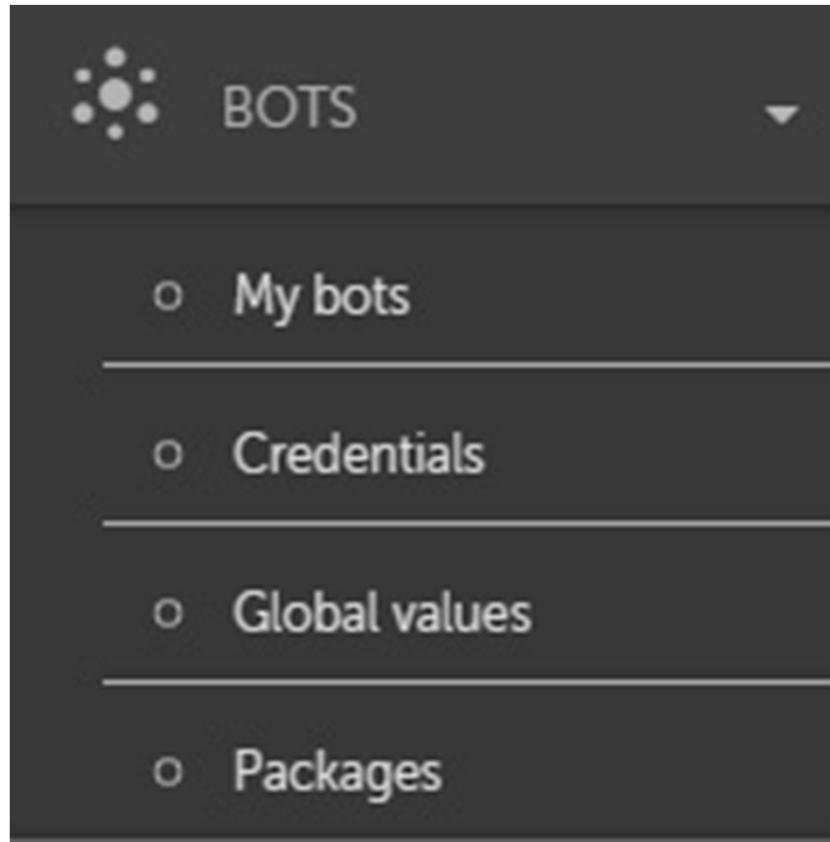
In progress activity

The screenshot shows a user interface for viewing RPA activity. On the left, there is a dropdown menu labeled "Status" with a downward arrow. The dropdown menu is open, displaying the following options: "Status", "Current action", "Current bot", "Bot", "Activity type", and "Device". The "Status" option is highlighted with a blue background. To the right of the dropdown is a table header with three columns: "NAME", "CURRENT ACTION", and "PROGRESS".

NAME	CURRENT ACTION	PROGRESS
------	----------------	----------

Managing bots

- Bot management and deployment are performed in this section.
- This is broken down into four sub-sections, which are accessible from the main menu pane on the left:



My bots

- The folder structure is displayed on the left pane and the contents on the right:

The screenshot shows the 'Control Room' interface with the title 'Control Room' at the top. On the right side of the top bar are icons for help, system status, and user information (rpa_training@skysoftuk.net). Below the title, the navigation path 'Bots > My bots' is shown. The main area is titled 'My bots' with a 'Create a bot...' button. On the left, there's a 'Folders' section with a tree view showing a single folder named 'Bots'. The main content area displays a table titled 'Files and folders (1)'. The table has columns: Name, Status, Size, Last Modified, and Modified By. The single entry is 'Sample_bots', which is a folder. The table also includes a search bar at the top and various icons for file operations like create, move, copy, delete, and refresh.

Name	Status	Size	Last Modified	Modified By
Sample_bots	N/A	N/A	12:23:57 GMT 2020-02-14	

My bots

- There are a few options available for each file and folder within this interface, as shown in the following screenshot:



Create new bot



Delete checked item



Create sub-folder



Refresh list



Upload files



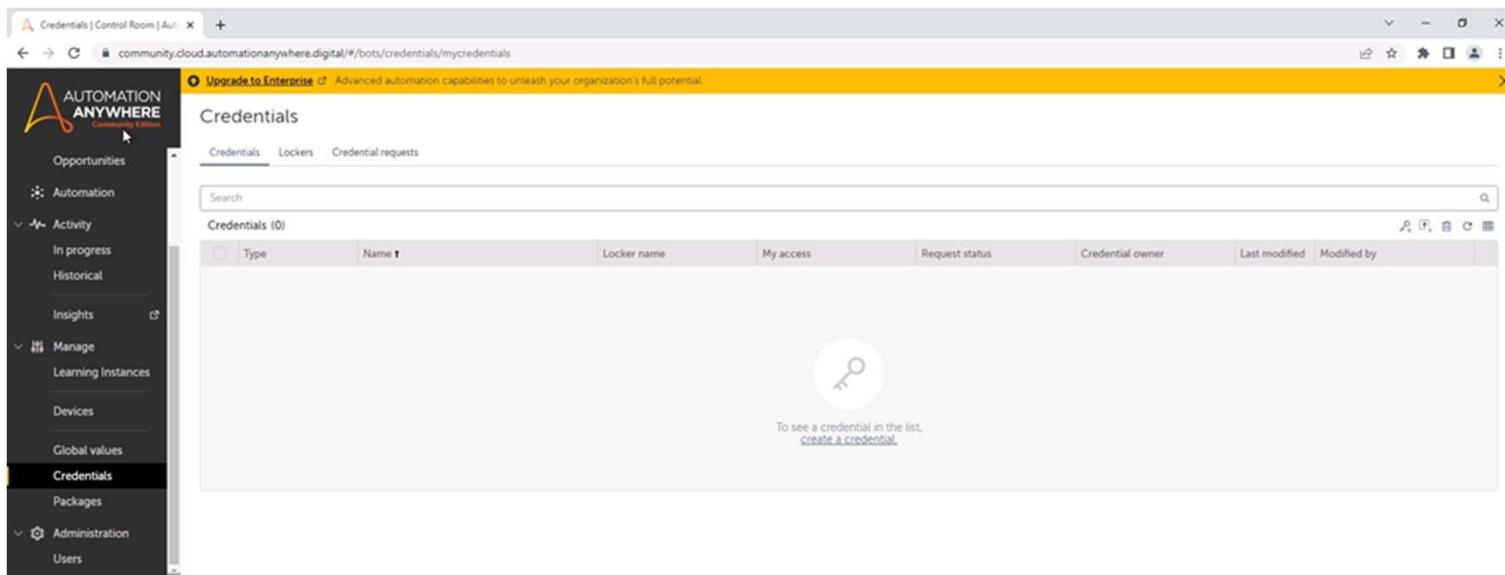
View/Hide columns

Credentials

- Create a credential with the required attributes; that is, a username/password.
- Add this credential to a locker.
- Grant the bot access to the locker.
- The bot can now get this specific credential to access the application.

Credentials

- The final CREDENTIAL REQUESTS tab is an informative tab showing all the requests that have been made for any credentials.
- You can see in the following screenshot the Credentials interface showing the three different tabs:



Credentials

- The options available for credentials are shown in the following screenshot:



Create a credential



Create locker with checked item



Delete checked item



Refresh list



View/Hide columns

Packages

All packages

Name	STA...	NAME ↑	# OF ACTIONS	VERSION	
		Analyze	2	2.1.0-20200204-154550	⋮
		Application	1	2.0.0-20200131-085947	⋮
		AWS Comprehend NLP (Beta)	4	0.1.0-20191001-174008	⋮
		AWS Comprehend NLP (Beta)	4	0.2.0-20191107-111107	⋮
		Boolean	6	2.0.0-20200131-085949	⋮
		Bot Migration	1	1.1.0-20200208-020245	⋮
		Browser	3	2.0.0-20200127-180439	⋮
		Clipboard	3	2.0.0-20200131-085958	⋮

Packages

Email

[**< Back**](#)

This page shows the package details of the selected version. You can also select a different package version and update the installed version.

Versions

2.0.0-20200206-135926 (Default)

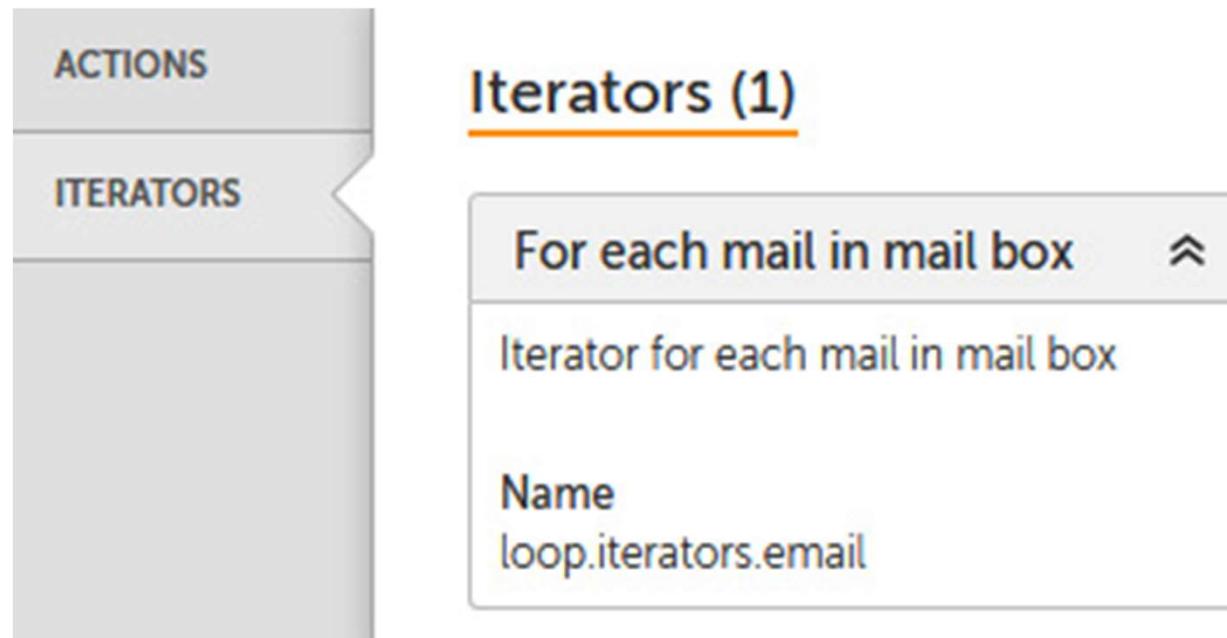


PACKAGE DETAILS				
Name	Description	Version	Status	
Email	Provides actions to perform email operations.	2.0.0-20200206-135926	Default	

ACTIONS	Actions (13)
ITERATORS	<ul style="list-style-type: none">✉ Change status✉ Check if folder exists✉ Delete all✉ Delete✉ Disconnect✉ Connect✉ Forward

Packages

- In the following screenshot, we can see that an iterator is also available to help with our bot:



Managing your devices

- By selecting the MY DEVICES option from the menu pane, you will see the following interface:

The screenshot shows the 'Control Room' interface of the Automation Anywhere Community Edition. The left sidebar has a dark theme with orange highlights for the 'MY DEVICES' option, which is currently selected. The main area is titled 'My devices'. At the top, there's a search bar with 'Device name' and a 'Search' button. Below it, a table lists one device: 'DESKTOP-MDG181K'. The table columns are: STATUS, DEVICE NAME, DEVICE NICKNAME, DEVICE TYPE, DEFAULT USERS, BOT AGENT VERSION, and DEVICE POOL. The device listed is 'Connected' with the status icon showing a green checkmark and a yellow star. The 'DEVICE NAME' column shows 'DESKTOP-MDG181K'. The 'BOT AGENT VERSION' column shows '14.3'. The 'DEVICE POOL' column shows '--'. On the right side of the table, there are several icons for managing the device: a monitor, a gear, an upward arrow, a downward arrow, a refresh symbol, and a grid.

STATUS	DEVICE NAME	DEVICE NICKNAME	DEVICE TYPE	DEFAULT USERS	BOT AGENT VERSION	DEVICE POOL
Connected	DESKTOP-MDG181K	--	Single user	. . .	14.3	--

Managing user administration

The screenshot shows the Automation Anywhere Control Room interface. The left sidebar has a dark theme with orange highlights for the selected 'ADMINISTRATION' item under 'MY PLATE'. The 'Users' option is also highlighted in orange. The main content area is titled 'Control Room' and shows the 'Administration > Users' path. The title 'All users' is displayed above a search bar with 'Username' dropdown and 'Search' input fields. A table lists one user: 'rpa_training'. The columns are: US... (checkbox), USERNAME (rpa_training), FIRST NAME (RPA), LAST NAME (Training), DESCRIPTION (--), ROLES (CE_user, 1 more...), DEVICE LICENSE (Bot creator), USER STATUS (Enabled), and LICENSE STATUS (Verified). There are also icons for upload, refresh, and grid view.

US...	USERNAME	FIRST NAME	LAST NAME	DESCRIPTION	ROLES	DEVICE LICENSE	USER STATUS	LICENSE STATUS
<input type="checkbox"/>	 rpa_training	RPA	Training	--	CE_user 1 more...	Bot creator	Enabled	Verified

Managing user administration

The screenshot shows the Automation Anywhere Control Room interface. The left sidebar menu includes HOME, DASHBOARD, ACTIVITY, BOTS (with sub-options My bots, Credentials, Packages), MY DEVICES, and ADMINISTRATION (with sub-option Users selected). The main content area is titled "Control Room" and shows the "View user" page for the email address "rpa_training@skysoftuk.net". The page displays "USER DETAILS" with fields: First name (RPA), Last name (Training), Description (--); Email (rpa_training), Password (****), User status (Enabled); License (Bot creator), License status (Verified), device (DESKTOP-MDG181K); Auto login (Cannot auto login). Below this is a "Roles (2)" section listing "NAME" (AAE_Bot Insight Expert, CE_user). At the bottom is a "GENERAL DETAILS" section showing Last modified (12:46:13 GMT 2020-02-14), Modified by (rpa_training), Object type (User), and User type (Bot creator).

USER DETAILS			
First name	RPA	Last name	Training
Email	rpa_training	Password	****
License	Bot creator	License status	Verified
Auto login	Cannot auto login	device	DESKTOP-MDG181K

Roles (2)	
NAME	AAE_Bot Insight Expert
NAME	CE_user

GENERAL DETAILS			
Last modified	12:46:13 GMT 2020-02-14	Modified by	rpa_training
Object type	User	User type	Bot creator

Summary

- You should now be comfortable with Automation Anywhere's user interface, having a clearer understanding of all the configurations needed to design and support your bot.
- You will understand why features such as security, reusability, and a simple interface make Automation Anywhere an award-winning RPA tool and one of the industry leaders.
- You should now have an overview of the Control Room interface and its features.

Overview of the Automation Anywhere Development Interface



Automation Anywhere Development Interface

In this lesson, we will cover the following topics:

- Bot development interface
- What can a bot do?
- Programming techniques using Automation Anywhere
- Variables and triggers
- Debugging and dependencies

Technical requirements

- Windows OS version 7 or higher
- A processor with a minimum speed of 3 GHz
- A minimum of 4 GB RAM
- At least 100 MB of hard disk space
- Web browser: Internet Explorer v10 or higher or Chrome v49 or higher
- A minimum screen resolution of 1024*768
- An internet connection with a minimum speed of 10 Mb/sec
- Completed registration with Automation Anywhere Community Edition
- Successful logon to Automation Anywhere Community Edition
- Successful registration of a local device

Bot development interface

- The development interface is where all the magic happens.
- This is where bots are created, edited, and debugged. In order to look at what it has to offer, you need to start by creating a new bot.
- Let's dive straight into creating a bot so that we can explore the development interface.

Creating a new bot

The screenshot shows the Automation Anywhere web interface. The left sidebar has a dark theme with the following navigation items:

- Explore
- Home
- Discovery
- Processes
- Opportunities
- Automation** (selected)
- Activity
- In progress
- Historical
- Insights
- Manage
- Learning Instances

The main content area is titled "Automation". It shows a "Folders" section with a tree view:

- Bots
 - Document Workspace Pr...
 - Sample bots** (selected)

A yellow banner at the top right says "Upgrade to Enterprise" and "Advanced automation capabilities to unleash your organization's full potential". A "Create new" button dropdown menu is open, showing options: Bot, Form, and Process.

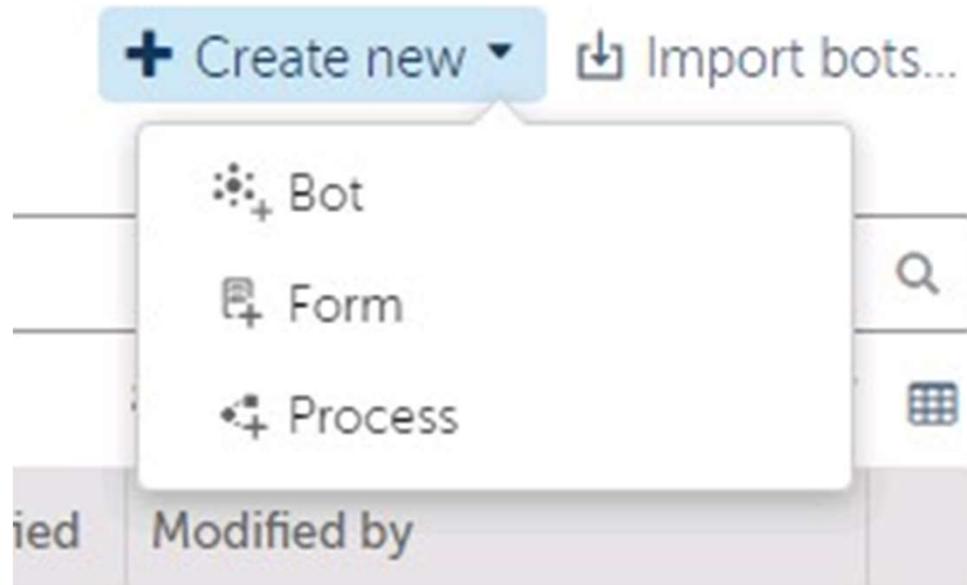
The main table area displays a single row with the following columns:

Name	Type	Name	Status	Size	Last modified	Modified by
	t1	t2				

A large circular icon with a grid pattern is centered below the table. Below it, text reads: "To see bots or folders here, [create a bot](#) or [create a subfolder](#)".

Creating a new bot

- Click on the Create a bot icon from the icon options in the top-right corner:



Creating a new bot

- The following bot properties dialog is presented:

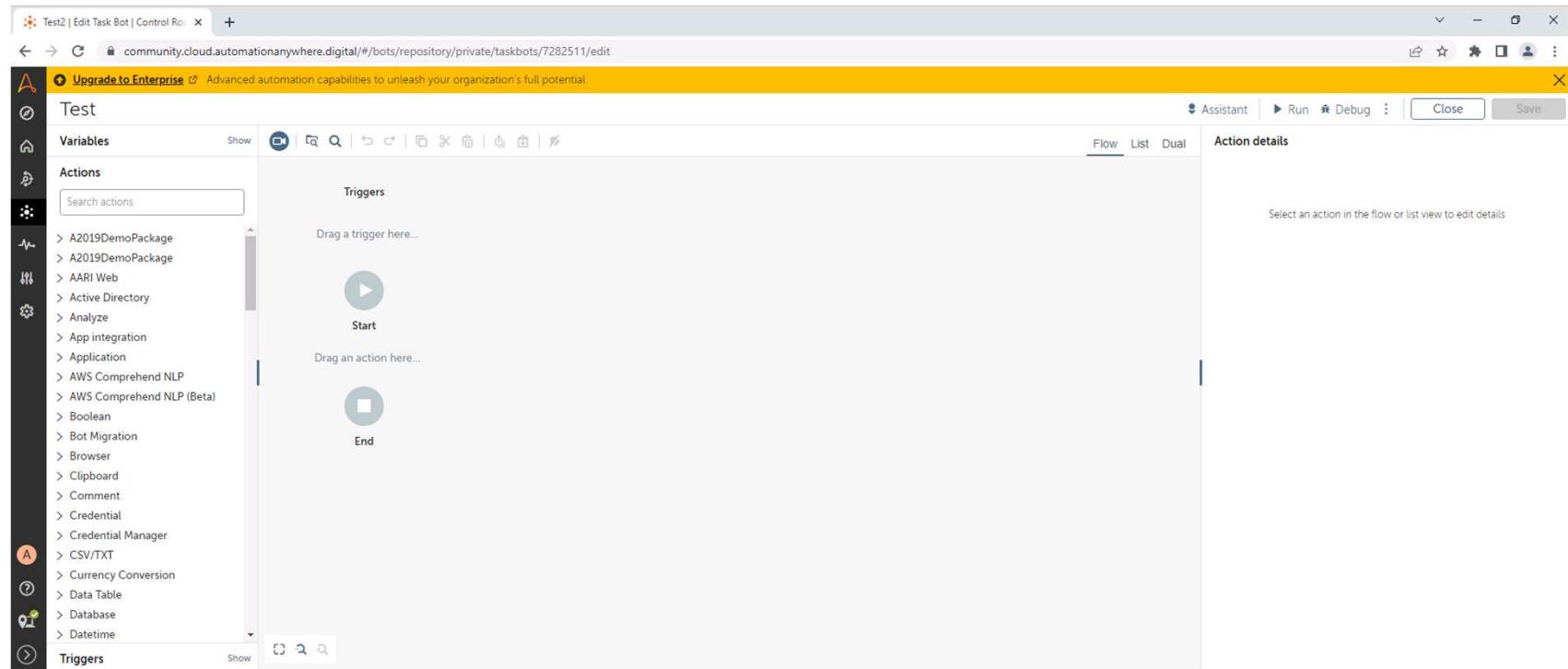
Create Task Bot

Name **Description (optional)**

Max characters = 50 Max characters = 255

Folder **Browse...**

Creating a new bot

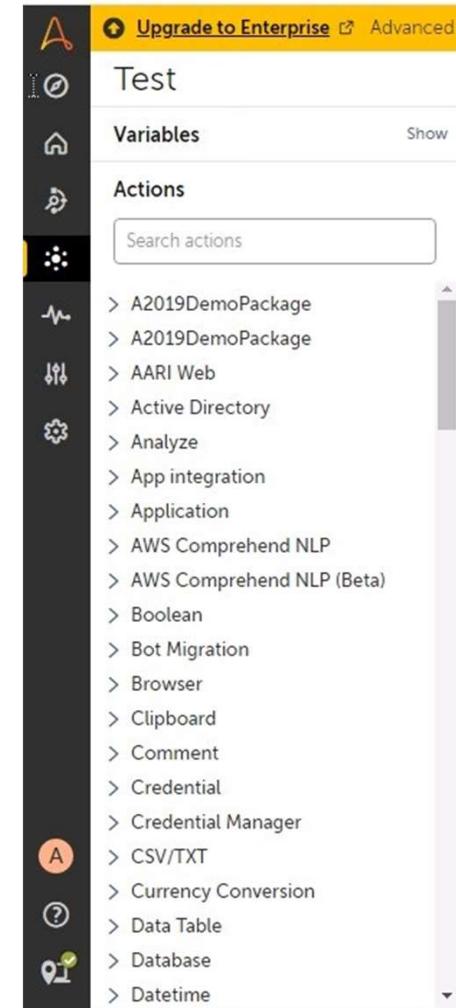


Creating a new bot

The screenshot shows the Automation Anywhere Community Edition Control Room interface. At the top left is the logo and text "AUTOMATION ANYWHERE Community Edition". To the right is a breadcrumb navigation "Control Room" and "My bots > Edit Task Bot". A large blue arrow points upwards from the bottom of the slide towards the "Edit Task Bot" link. On the left, there's a vertical navigation menu with "HOME", "DASHBOARD", "ACTIVITY", and "BOTS". The "BOTS" item has a dropdown menu with "My bots" selected, indicated by an orange dot. On the right, under the heading "Actions", there's a search bar labeled "Search actions" and three collapsed dropdown menus: "Analyze", "Application", and "AWS Comprehend NLP (Beta)".

What can a bot do?

- All the options are listed on the left pane, These actions are grouped into categories known as packages.
- Each package is a collection of individual actions.
- Here you can see the list pane showing all the available packages:



Programming techniques using Automation Anywhere

The screenshot shows the Automation Anywhere Control Room interface. At the top, a dark header bar displays the title "Control Room". Below it, a navigation bar shows the path "Bots > My bots > Edit Task Bot". The main area is titled "Test". At the top of this area, there are three tabs: "Flow", "List" (which is underlined, indicating it is the active tab), and "Dual". On the left, there is a sidebar with the heading "Actions" and a search bar labeled "Search actions". To the right of the sidebar is a toolbar with various icons: a video camera, a laptop, a document, a delete symbol, a file, a lightning bolt, a circular arrow, and a refresh symbol. Below the toolbar, the word "Triggers" is visible.

Programming techniques using Automation Anywhere

The screenshot shows the Control Room interface for a bot named "Edit Task Bot". The main area is titled "Test" and includes buttons for "Run", "Debug", "Analyze", "Close", and "Save". Above the main area are tabs: "Flow", "List" (which is selected), and "Dual". The left side features a sidebar titled "Actions" with a search bar and a list of actions. The "Message box" action is highlighted with a blue arrow labeled "2" pointing to it. A tooltip for "Message box" states: "Displays a message box". The right side of the interface contains a workspace with a toolbar at the top and a central area for dragging triggers and actions. A trigger icon is shown with a blue arrow labeled "1" pointing to it, and a tooltip says "Drag a trigger here...". Below the trigger icon is a placeholder for an action with the text "Drag an action here...".

Programming techniques using Automation Anywhere

The screenshot shows the Automation Anywhere Control Room interface. At the top, it displays 'Control Room', 'Bots > My bots > Private > Edit Task Bot', and a user profile 'rpa_training'. Below the header, there's a title 'Test' and buttons for 'Run', 'Debug', 'Analyze', 'Cancel', and 'Save'. The main area is divided into sections: 'Variables' (with a 'Show' button), 'Actions' (with a 'Search actions' input and a list including 'Legacy automation', 'List', 'Log To File', 'Loop', 'ML - Microsoft Anomaly Dete...', 'MS Word', 'Message box', 'Microsoft LUIS NLP (Beta)', 'Mouse', 'Number', 'NumberUtils', 'OCR', 'Office 365 Calendar', and 'Triggers'), 'Flow' (with tabs for 'List' and 'Dual'), 'Triggers' (with a placeholder 'Drag a trigger here...'), 'Start' (with a selected 'Message box' action), and 'End'. To the right, there's a detailed configuration panel for the 'Message box' action, which includes fields for 'Message box window title' (set to 'Automation Anywhere Enterprise Client'), 'Message to display' (set to 'Hello World'), 'Scrollbar after lines' (set to '# 30'), and a checkbox for 'Close message box after' (unchecked). A 'Seconds' field is also present.

Programming techniques using Automation Anywhere

- Set the following properties for the Message box action:

Message box

Displays a message box

Enter the message box window title

” Test Message Box 1

Enter the message to display

” Hello World

Scrollbar after lines

30

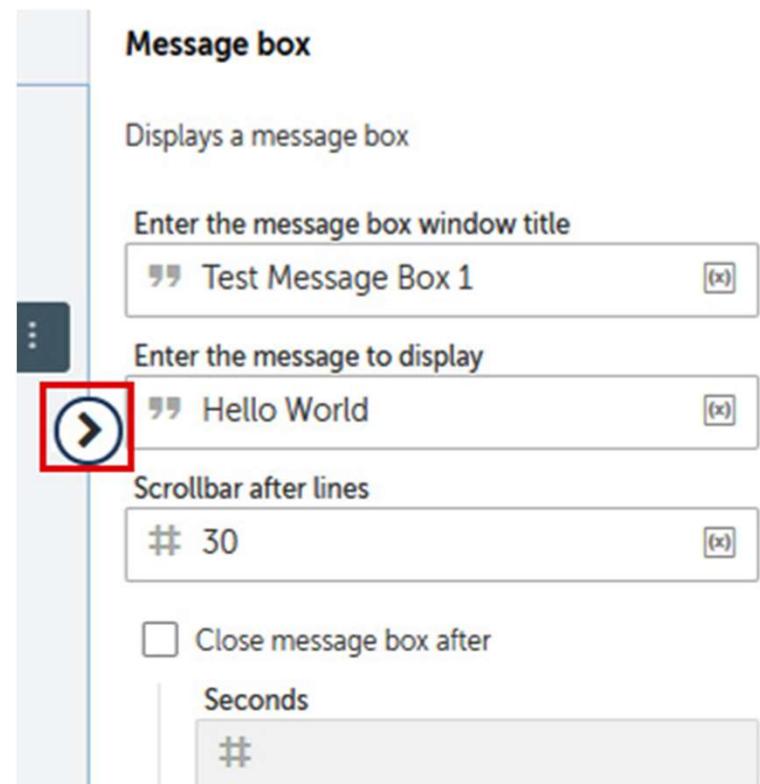
Close message box after

Seconds

2

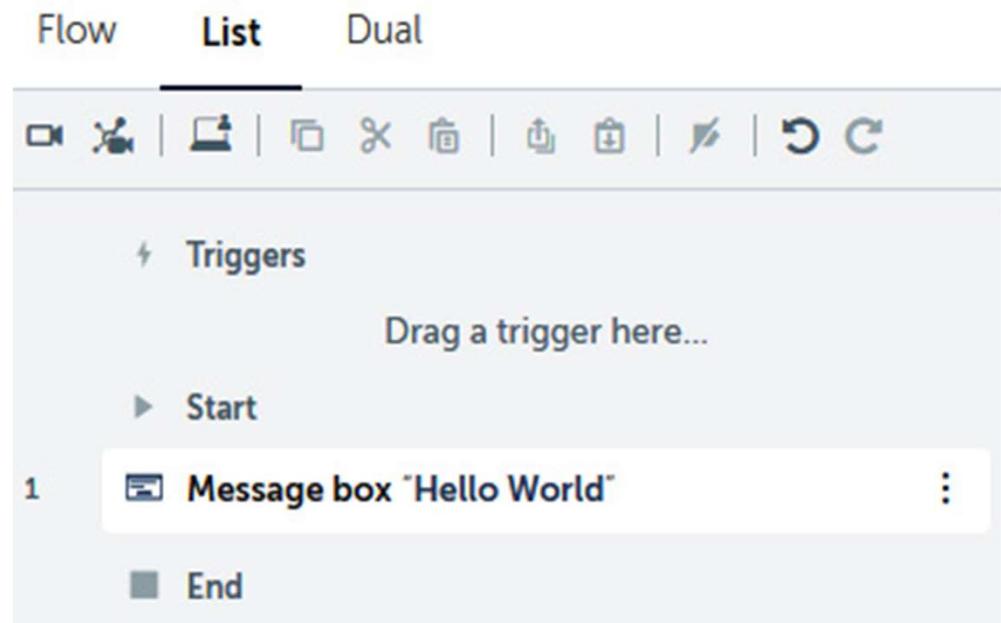
Programming techniques using Automation Anywhere

- You can collapse the properties pane when you are not using it.
- Just hover over and click the collapse icon to hide it:



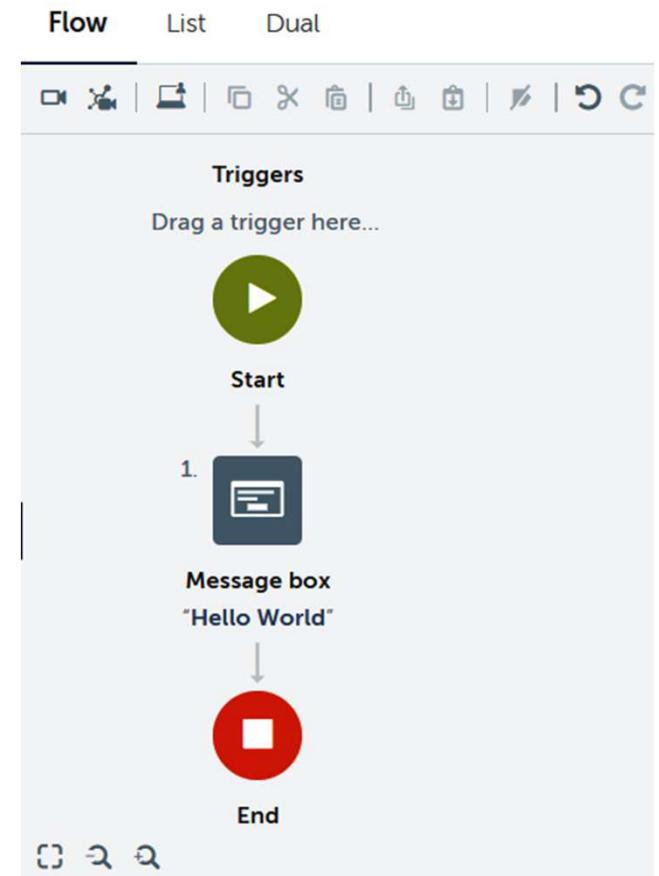
List view

- You have used this so far to add the Message box action.
- In this view, all actions are listed, similar to a script.
- The line number provides the logical workflow of the task:



Flow view

- The Flow view shows your task actions as a flowchart.
- As you can see, this provides a more visual process flow of how your bot works through the workflow.
- The line numbers are omitted as they are not necessary:



Flow view

- You can change the size of the flow diagram using the sizing options on the bottom left.
- These options are as follows:



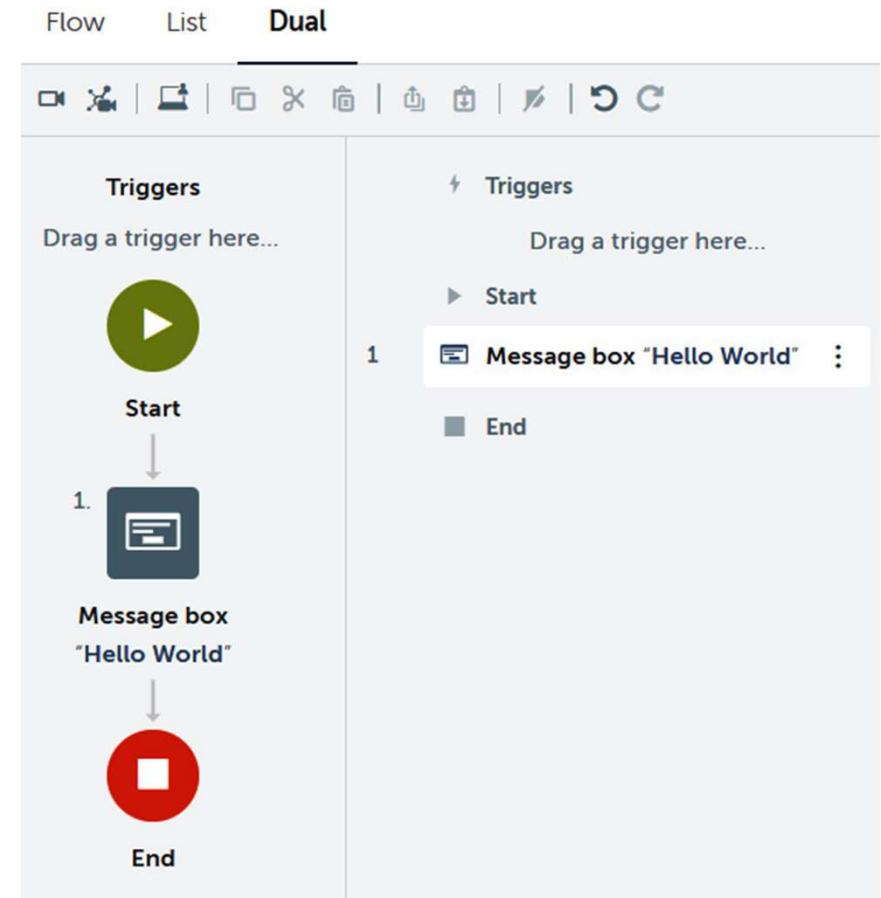
Zoom fit

Zoom in

Zoom out

Dual view

- If you select an action on either view, it will also be selected in the other view.
- This makes navigation and editing much easier:



Dual view

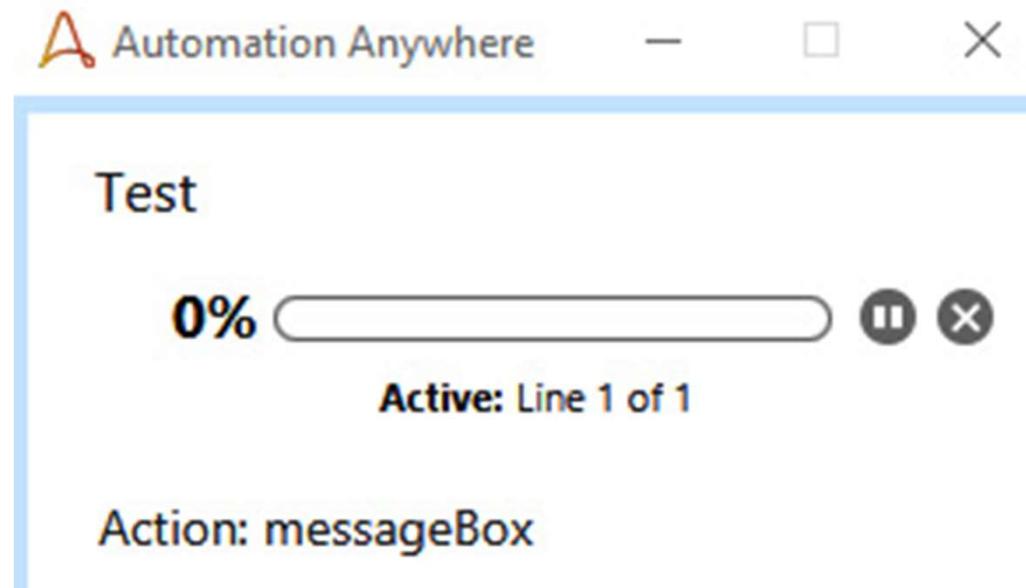
- Click on the Save button in the top-right pane:



- Click on Run to execute the bot.

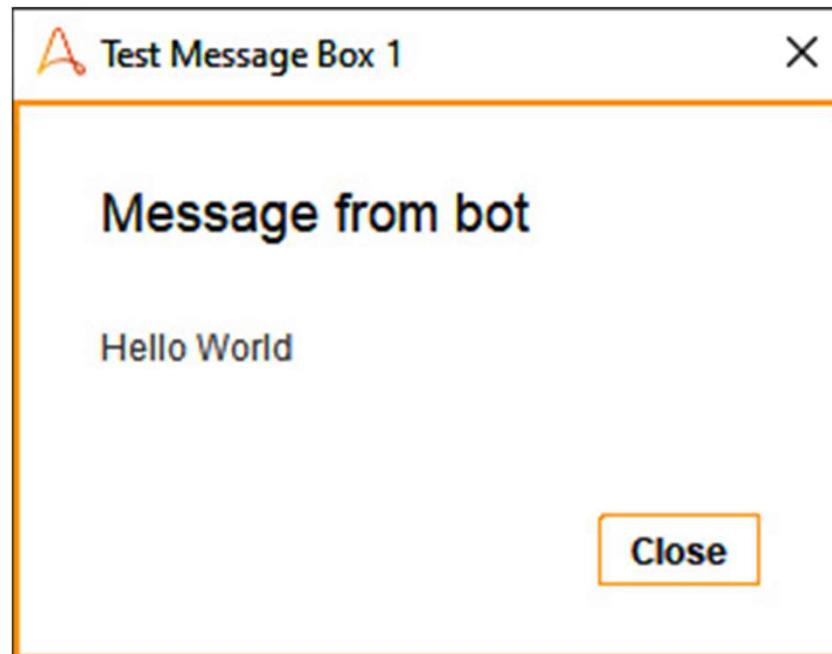
Dual view

- It is useful for testing and debugging your bot:



Dual view

- The Message Box should be displayed as your bot runs the action:



Dual view

- Click on the Close button in Message Box.
- Once the bot has completed, you will be presented with the following:

Your bot has run successfully!



Close

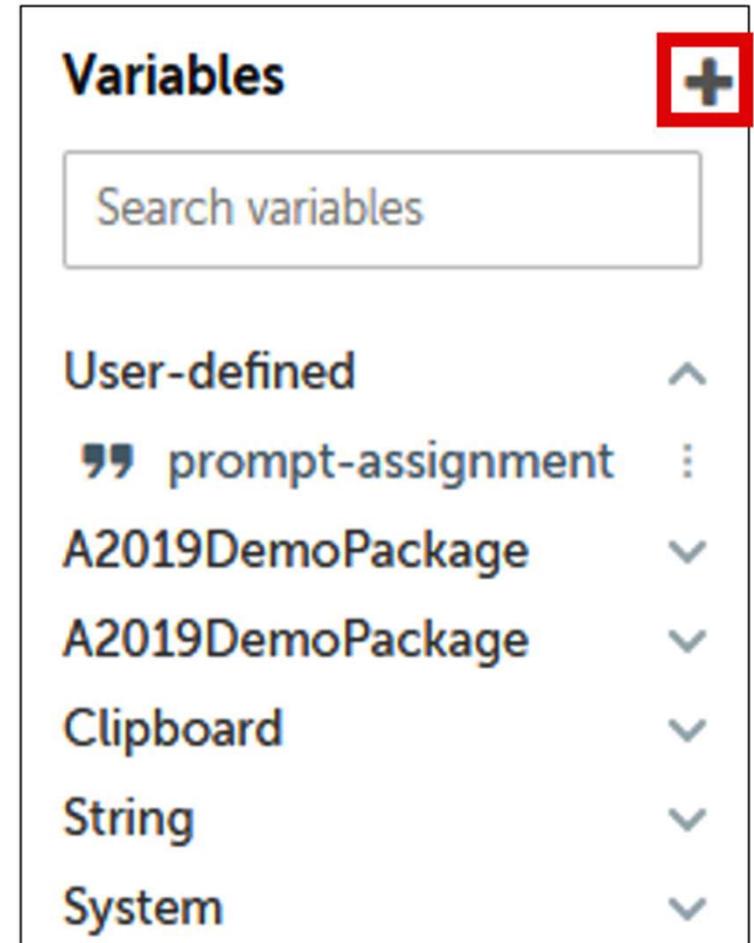
Variables and triggers

- Click on Show for the Variables tab from the option pane on the left:

The screenshot shows the 'Variables' tab selected in the top navigation bar of a software interface. A red box highlights the 'Show' button. Below the tab, the word 'Actions' is displayed. A search bar contains the placeholder text 'Search actions'. To the right of the search bar is a vertical scroll bar. Below the search bar, three items are listed with dropdown arrows: 'Legacy automation', 'List', and 'Log To File'. The 'Legacy automation' item has an upward arrow icon to its right.

Variables and triggers

- Click on the + icon to create a new variable:



The screenshot shows a 'Variables' panel with a search bar at the top. Below the search bar, there is a section titled 'User-defined' with a collapse arrow. Underneath are several variable entries, each with a collapse arrow: 'prompt-assignment', 'A2019DemoPackage', 'Clipboard', 'String', and 'System'. In the top right corner of the panel, there is a red-bordered '+' icon.

User-defined	
prompt-assignment	⋮
A2019DemoPackage	▼
A2019DemoPackage	▼
Clipboard	▼
String	▼
System	▼

Variables and triggers

- Will launch the Create variable dialog box:

Create variable

Cancel

Create

▲ Name

Max characters = 50

Description (optional)

Max characters = 255

Use as input

Use as output

Constant (read-only)

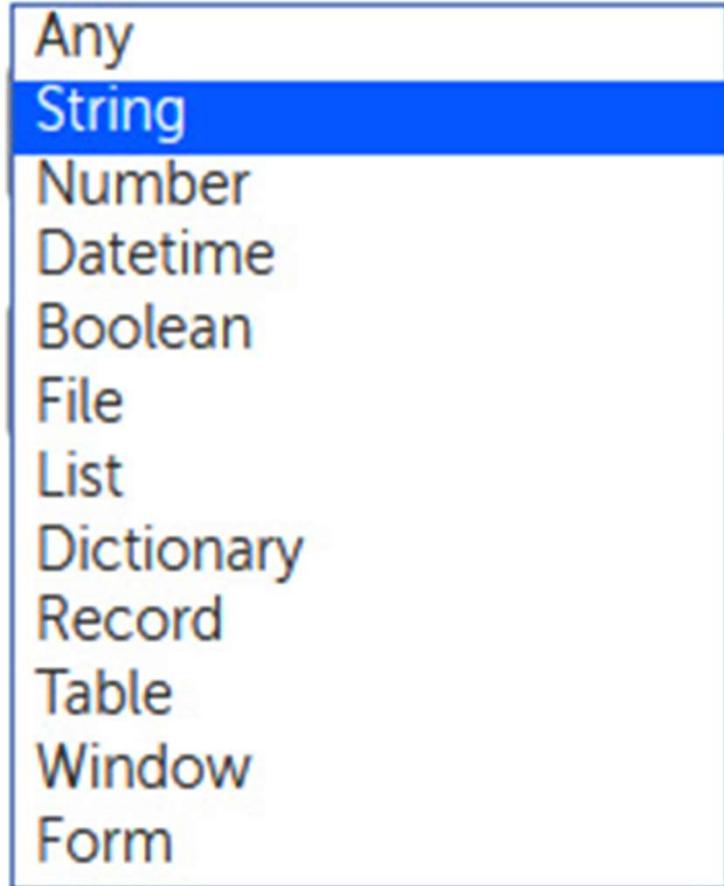
Type

String

Default value

Variables and triggers

- You will notice from the Type variable drop-down list all the different data types that are available for your variables:



Variables and triggers

- The Create variable dialog box should look similar to this:

Create variable

Name
strName
Max characters = 50

Description (optional)
Max characters = 255

Use as input

Use as output

Constant (read-only)

Type
String

Default value
Husan

Cancel **Create**

Variables and triggers

Flow List Dual

The screenshot shows a software interface for configuring triggers. On the left, there's a toolbar with various icons, followed by a list of triggers under the 'Triggers' section. A specific trigger, 'Message box "Hello World"', is selected and highlighted in dark blue. To the right of the trigger list is a detailed configuration panel for the selected trigger.

Message box

Displays a message box

Enter the message box window title
"Test Message Box"

Enter the message to display
"Hello World"

Scrollbar after lines
30

Close message box after
Seconds
#

Variables and triggers

- Modify the following property value:

Enter the message to display: Hello \$strName\$

- The property value should look like this:

Enter the message to display

” Hello \$strName\$



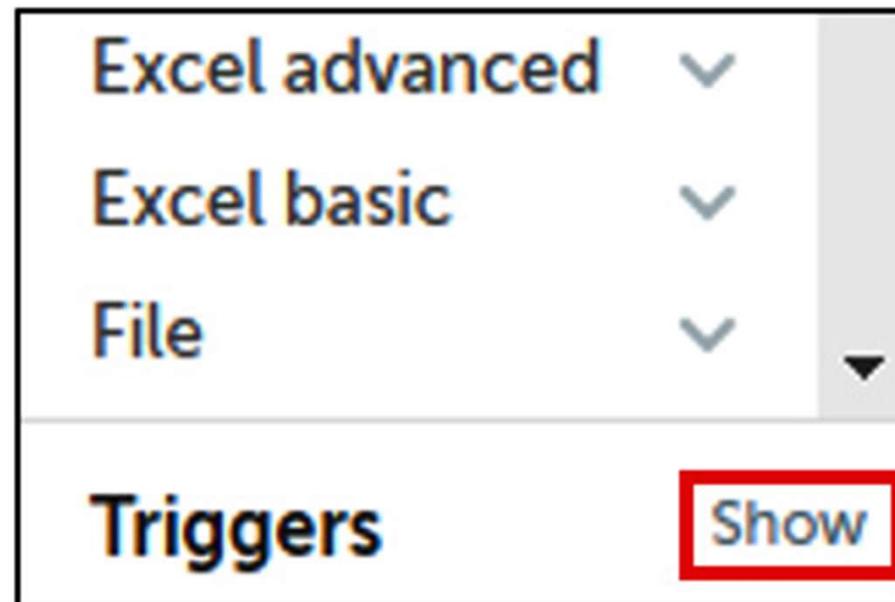
Variables and triggers

- Now, let's run the bot by clicking on Run.
- You should get a Message Box with your name:



Variables and triggers

- Click on Show on the Triggers tab from the option pane on the left:



Variables and triggers

- Expand the Files & folders trigger group:

The screenshot shows a 'Triggers' interface with a search bar at the top. Below the search bar, there is a list of trigger categories. The 'Files & folders' category is highlighted with a red border. Underneath it, two sub-options are listed: 'File trigger' and 'Folder trigger'. Other categories shown include 'A2019DemoPackage' (expanded), 'Email trigger', 'Hot key', and 'Interface trigger'.

Triggers	
Search triggers	
A2019DemoPackage	▼
A2019DemoPackage	▼
Email trigger	▼
Files & folders	▲
File trigger	
Folder trigger	
Hot key	▼
Interface trigger	▼

Variables and triggers

Flow List **List** Dual

Variables Show

Actions Show

Triggers

Search triggers

A2019DemoPackage

A2019DemoPackage

Email trigger

Files & folders

- File trigger
- Folder trigger

Hot key

Interface trigger

Triggers

Files & folders: File trigger when a file is created

Start

1 Message box "Hello \$strName\$"

End

The screenshot shows the Microsoft Power Automate interface in 'List' mode. On the left, there's a sidebar with sections for 'Variables', 'Actions', and 'Triggers'. Under 'Triggers', there's a search bar and a list of trigger types: A2019DemoPackage, A2019DemoPackage, Email trigger, Files & folders (which is expanded, showing 'File trigger' and 'Folder trigger'), Hot key, and Interface trigger. On the right, the main area shows a trigger flow. It starts with a 'File trigger' (highlighted in dark blue) labeled 'Files & folders: File trigger when a file is created'. This is followed by a 'Start' action, then a single step 'Message box "Hello \$strName\$"', and finally an 'End' action.

Variables and triggers

Set the following properties for the new
Files & folders: File trigger:
File: C:\RPA\TriggerFile.txt

Start the bot when the file is...: modified

- The trigger properties should look similar to this:

Files & folders: File trigger

Triggers on a file event.

File

C:\RPA\TriggerFile.txt

Browse...

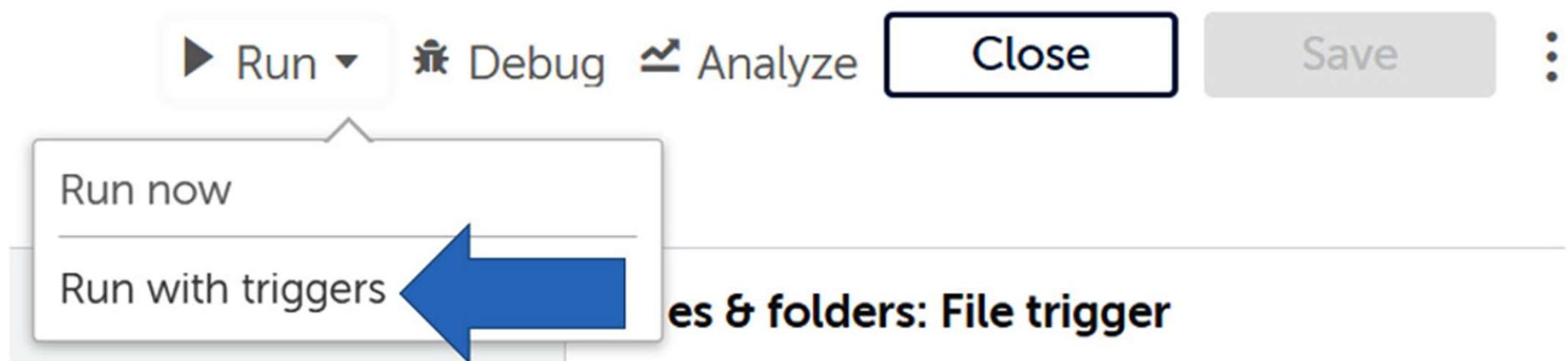
Start the bot when the file is...

modified



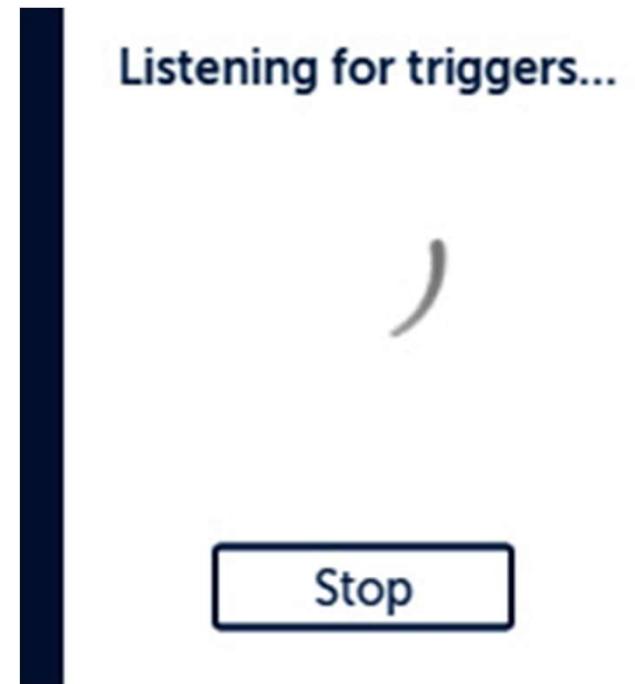
Variables and triggers

- To run the bot with triggers, click on Run and, from the dropdown, select Run with triggers:



Variables and triggers

- The bot will be deployed to your device. It will then wait for the trigger event to be true.
- This is known as listening for triggers. The following message box should appear as it listens for the trigger:



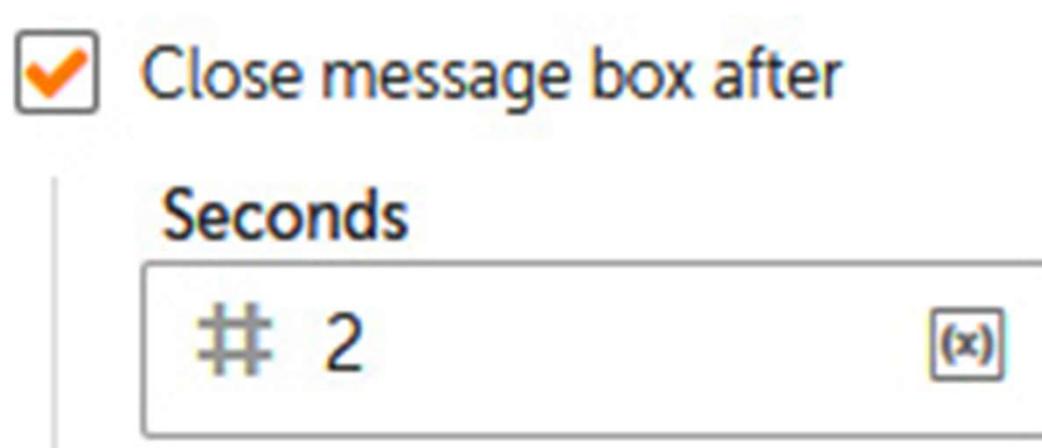
Variables and triggers

- Open, edit, and save the TriggerFile.txt trigger file that we created.
- You will notice the bot start and execute once you have saved the file:

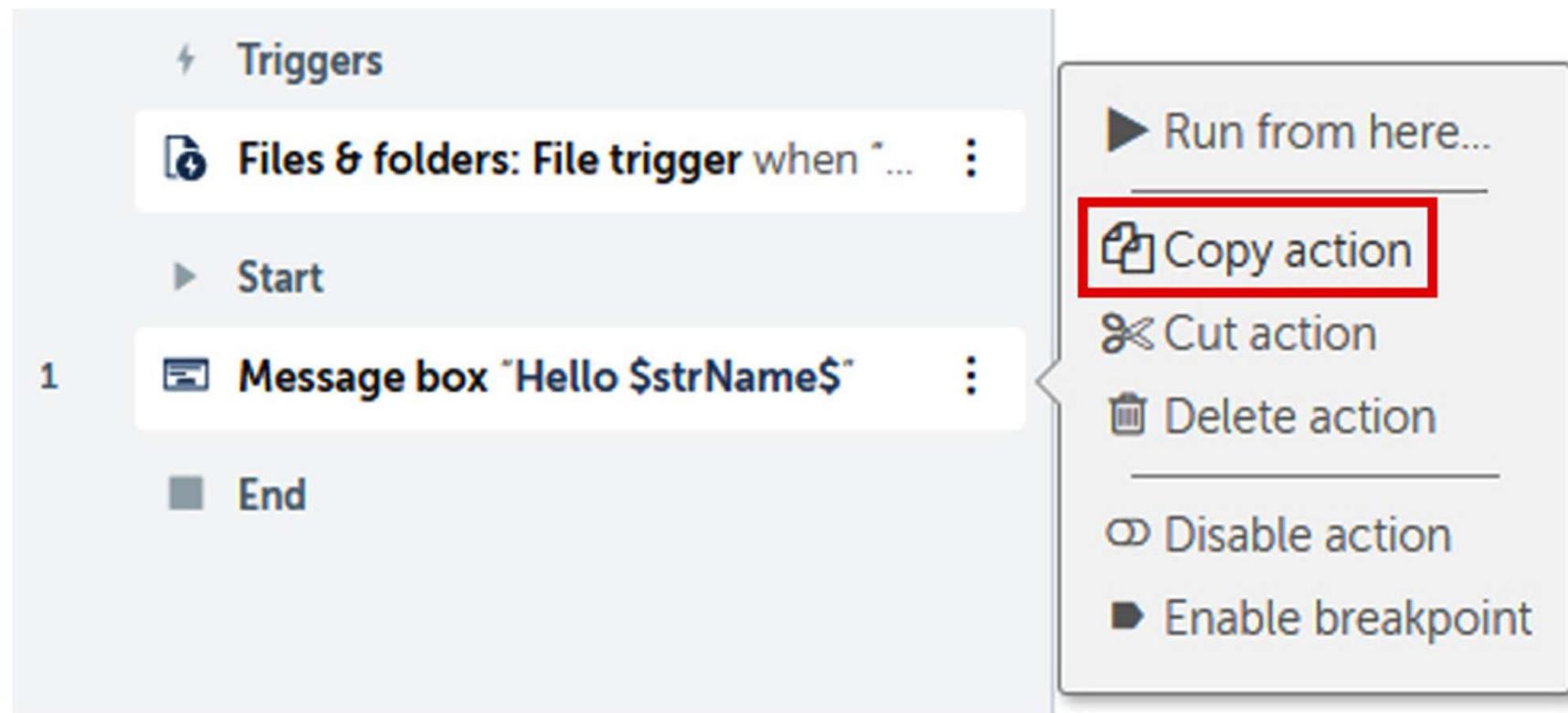


Debugging and dependencies

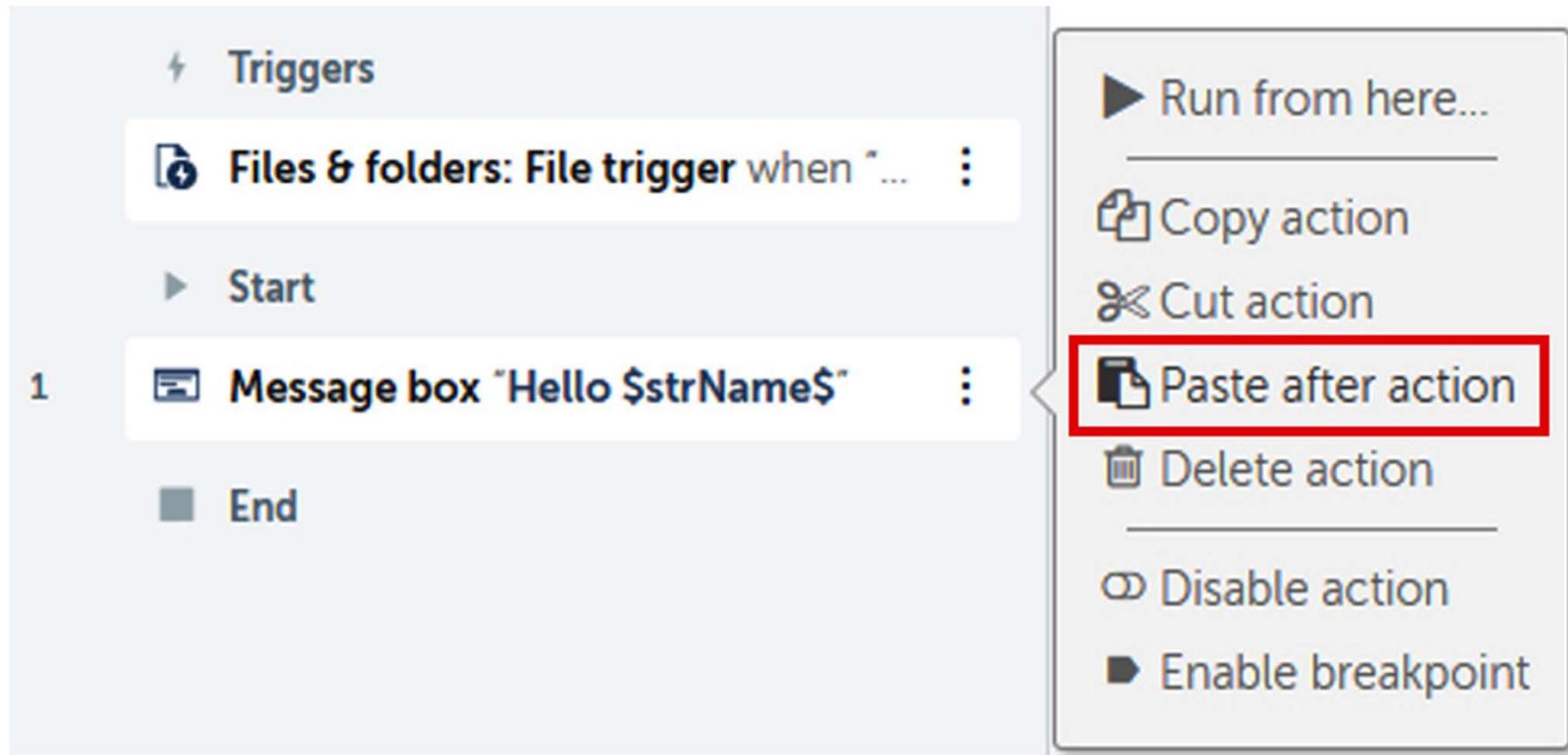
- Navigate to the properties of our Message box action on line 1 and update the following properties:



Debugging and dependencies

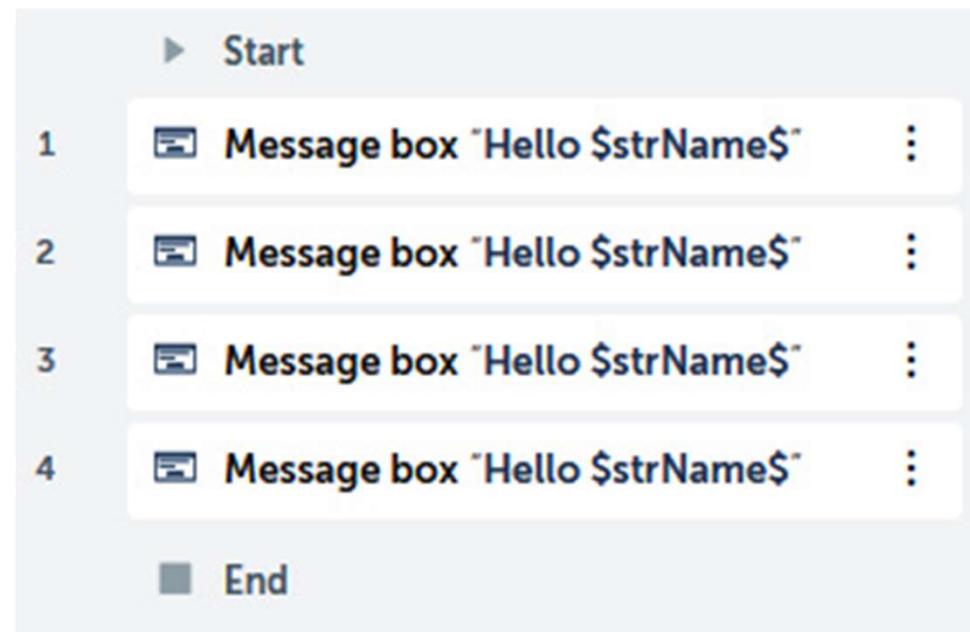


Debugging and dependencies



Debugging and dependencies

- Repeat last step two more times.
- The development interface should look like this:



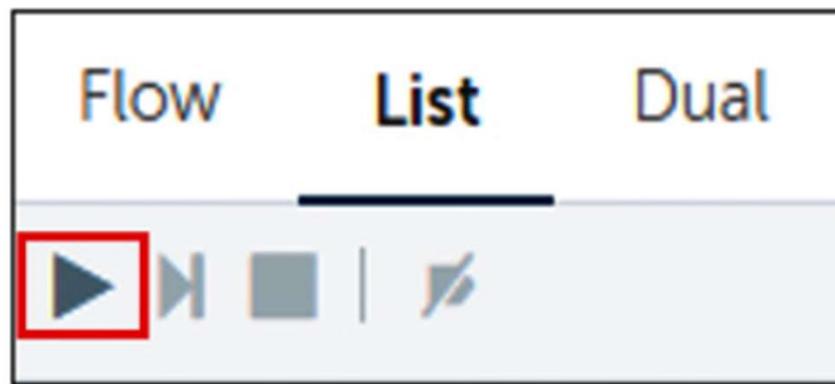
Switch to debug mode by clicking on **Debug** from the top-right menu:



Figure 4.39 – Switching to debug mode

Debugging and dependencies

Once in debug mode, you can run the bot from the start icon above the development window, as shown in the following screenshot:



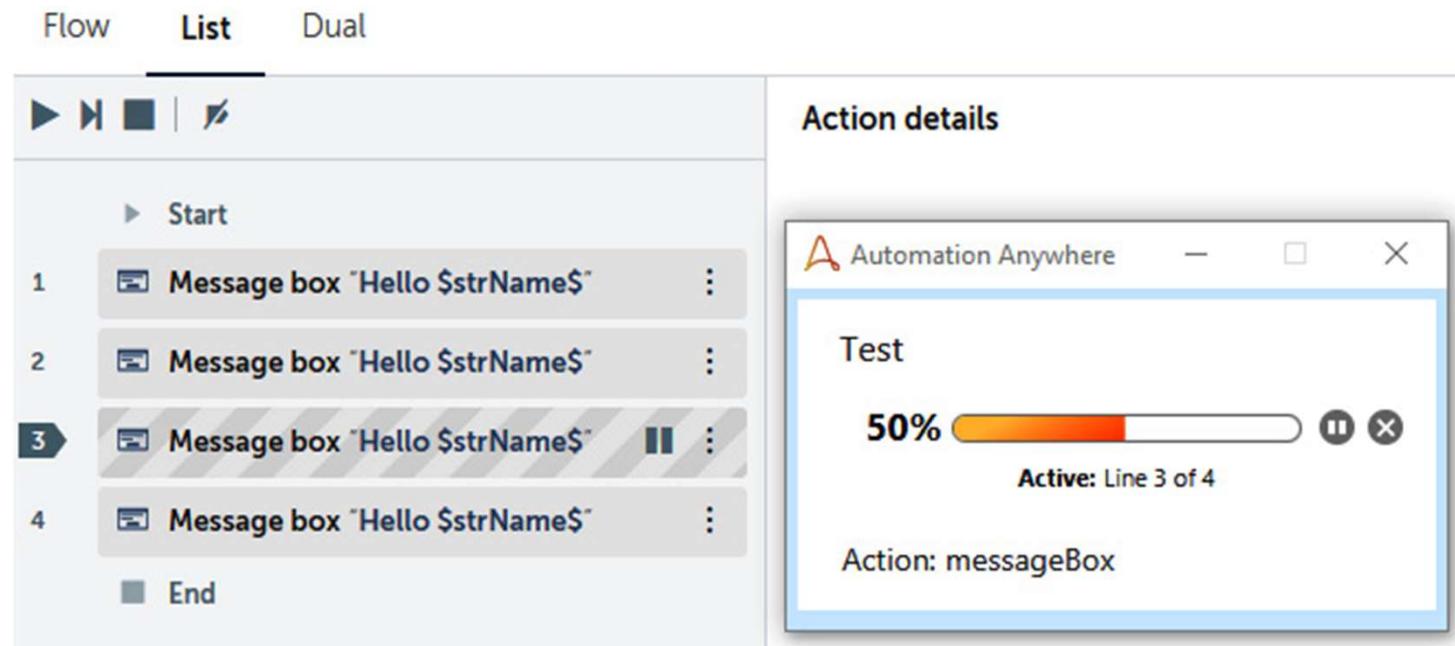
Debugging and dependencies

- We will add a breakpoint to the Message box action on line 3. To do this, select Enable breakpoint from the action line options:



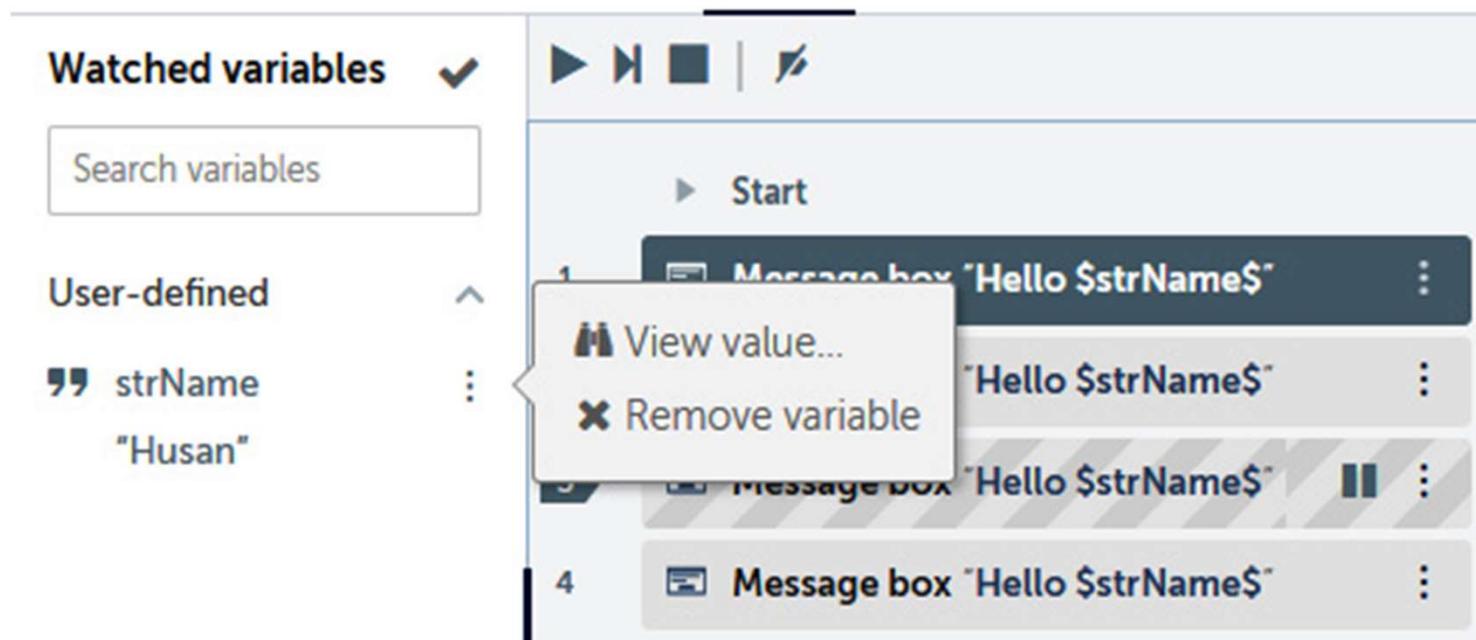
Debugging and dependencies

- Now run the bot and you will notice the bot pause at the breakpoint action:



Debugging and dependencies

- The breakpoint also allows you to view any values assigned variables during runtime:

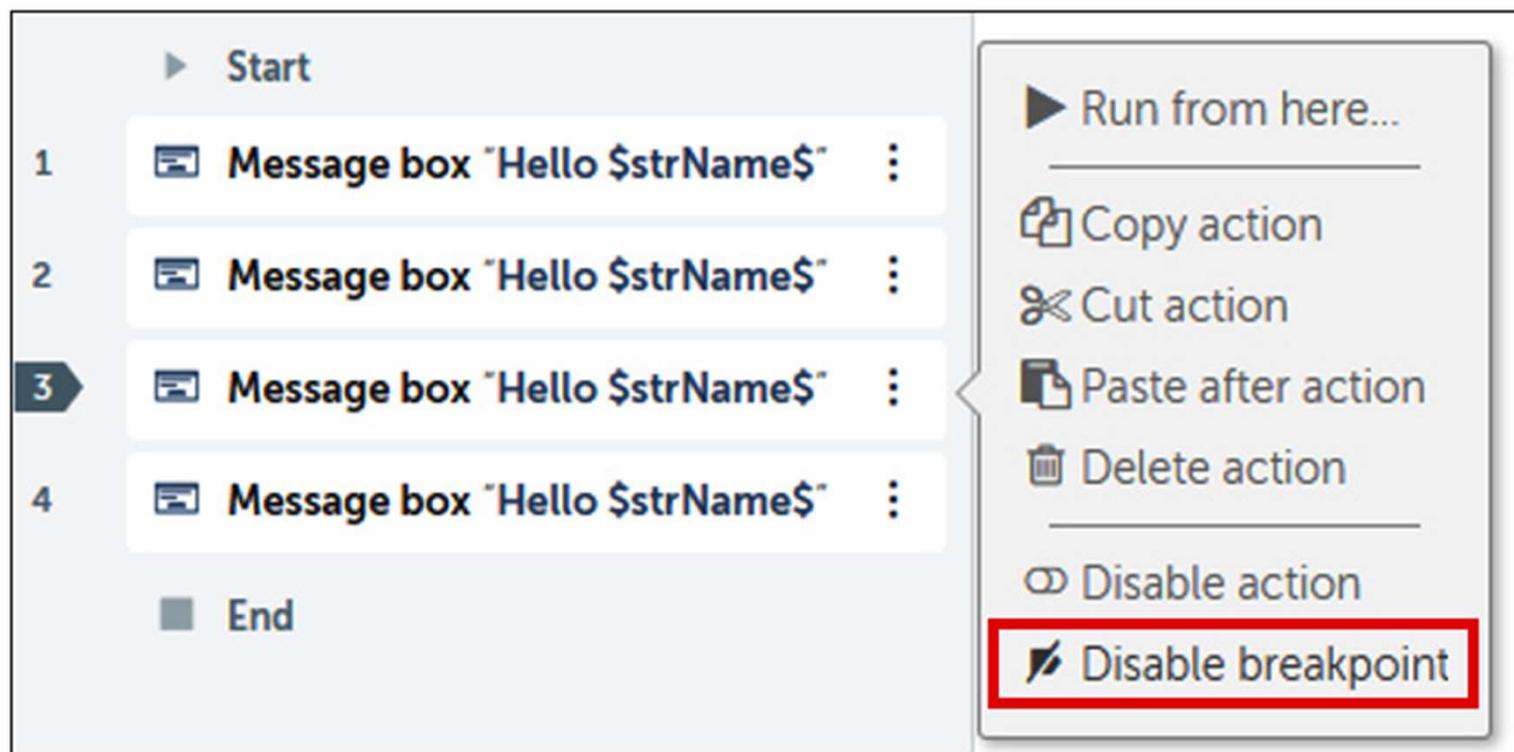


Debugging and dependencies

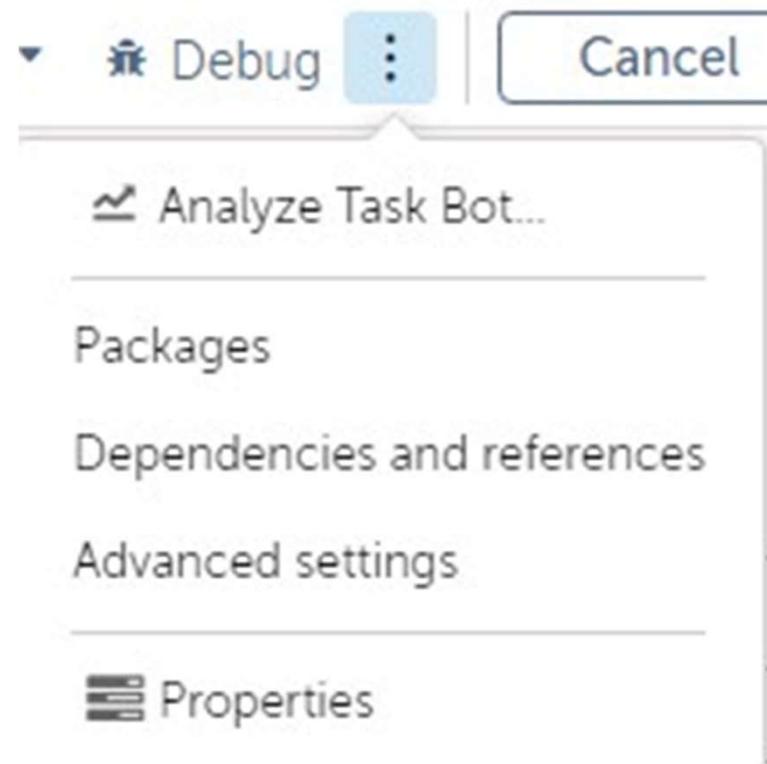
- Once the bot has completed, we can exit debug mode by clicking on Exit debug to take you back to the normal development interface:



Debugging and dependencies



Debugging and dependencies



Debugging and dependencies

The screenshot shows the Control Room interface with the following details:

- Control Room Header:** Includes a menu icon, user status (green checkmark), and email (rpa_training@skysoftuk.net).
- Breadcrumbs:** Bots > My bots > Edit Task Bot.
- Title:** Test.
- Buttons:** Close, Save, and a three-dot menu.
- Folders:** A tree view showing a 'Bots' folder expanded, with a 'Sample bots' subfolder selected.
- Available files:** A list titled 'Available files (0)' with columns: Name (dropdown), Search, and a refresh icon.
- Selected files:** A list titled 'Selected (0)' with columns: Name (dropdown), Search, and a refresh icon.
- GENERAL DETAILS:** A summary section with the following data:

Last modified 12:47:10 BST 2020-04-01	Modified by rpa_training@skysoftuk.net	Object type Task Bot
---	---	-------------------------

Summary

- You should now be comfortable with the Automation Anywhere development interface.
- We have gone through creating a simple bot. You have created variables and triggers for your bot.
- In addition, you have also acquired some hands-on experience of using the debugging tool for Automation Anywhere.
- Having created your initial bot and gotten an overview of all the other actions available, you must be keen to start implementing more complex actions in your bot.

Building Your First Bot



Building Your First Bot

In this lesson, we will cover the following topics:

- Downloading sample data from GitHub
- Understanding your automation task
- Creating and reading a CSV file
- Performing basic arithmetic calculations
- Appending records to a CSV file

Technical requirements

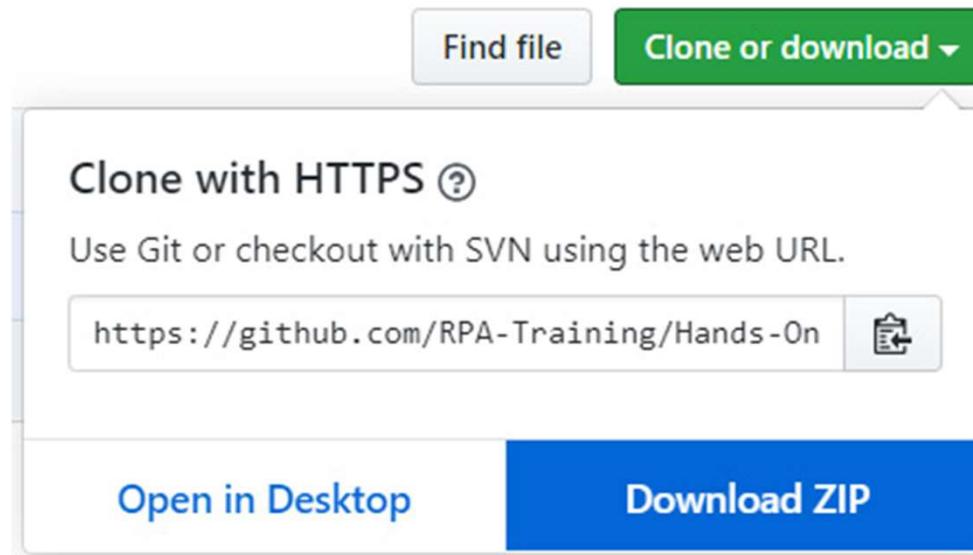
- Windows OS version 7 or higher
- A processor with a minimum speed of 3 GHz
- A minimum of 4 GB RAM
- At least 100 MB of hard disk space
- Internet Explorer v10 or higher OR Chrome v49 or higher
- A minimum screen resolution of 1024*768
- An internet connection with a minimum speed of 10 Mb/sec
- Completed registration with Automation Anywhere Community Edition
- Successful login to Automation Anywhere Community Edition
- Successful registration of a local device

Downloading sample data from GitHub

The screenshot shows a GitHub repository page. At the top, there's a navigation bar with links for Why GitHub?, Team, Enterprise, Explore, Marketplace, Pricing, and a search bar. On the right of the bar are 'Sign in' and 'Sign up' buttons. Below the bar, the repository path 'RPA-Training / Hands-On-RPA-with-AA' is shown, along with 'Watch 1', 'Star 0', and 'Fork 0' buttons. A main navigation menu below the path includes 'Code' (which is selected), 'Issues 0', 'Pull requests 0', 'Actions', 'Projects 0', 'Security', and 'Insights'. A prominent 'Join GitHub today' banner is displayed in the center, featuring a 'Sign up' button. Below the banner, it says 'GitHub is home to over 40 million developers working together to host and review code, manage projects, and build software together.' To the left of the banner, there's a small icon of a person with a checkmark. At the bottom of the page, there's a summary section with metrics: '5 commits', '2 branches', '0 packages', '0 releases', and '1 contributor'. A dropdown menu for the branch 'Sample-Data' is open. Below the summary, a message states 'This branch is 4 commits ahead, 1 commit behind master.' There are buttons for 'Find file', 'Clone or download', 'Pull request', and 'Compare'. A file list shows 'RPA-Training Add files via upload' (latest commit 23e206a 4 minutes ago) and 'Chapter05_InputData.csv' (Add files via upload, 4 minutes ago).

Downloading sample data from GitHub

- To download, click on the green Clone or download button. This will open a small dialog box, and then click on Download ZIP:

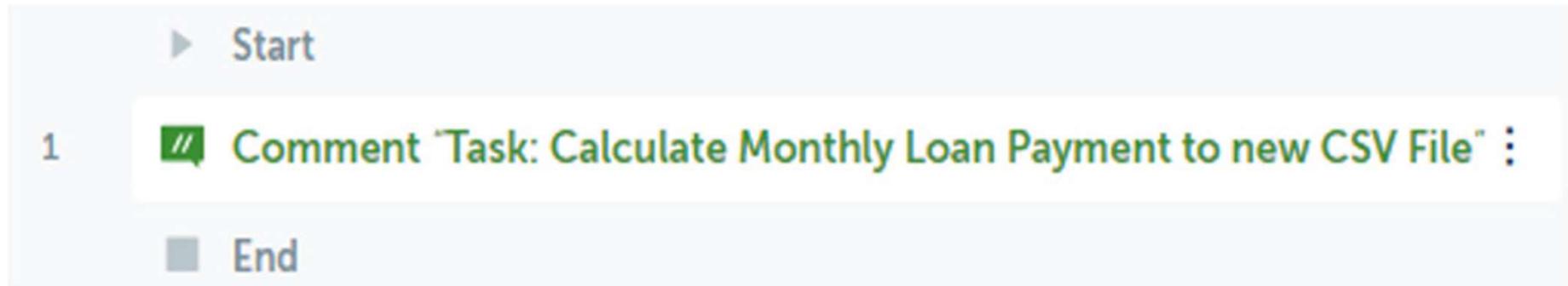


Creating your first bot

- Log in to Control Room.
- Create a new bot and call it lesson 5 - FirstBot in the \Bot\ folder.
- Add a Comment action as line 1; we will use this as our bot description comment.
- Set the Comment property's text as "Task: Calculate Monthly Loan Payment to new CSV File".

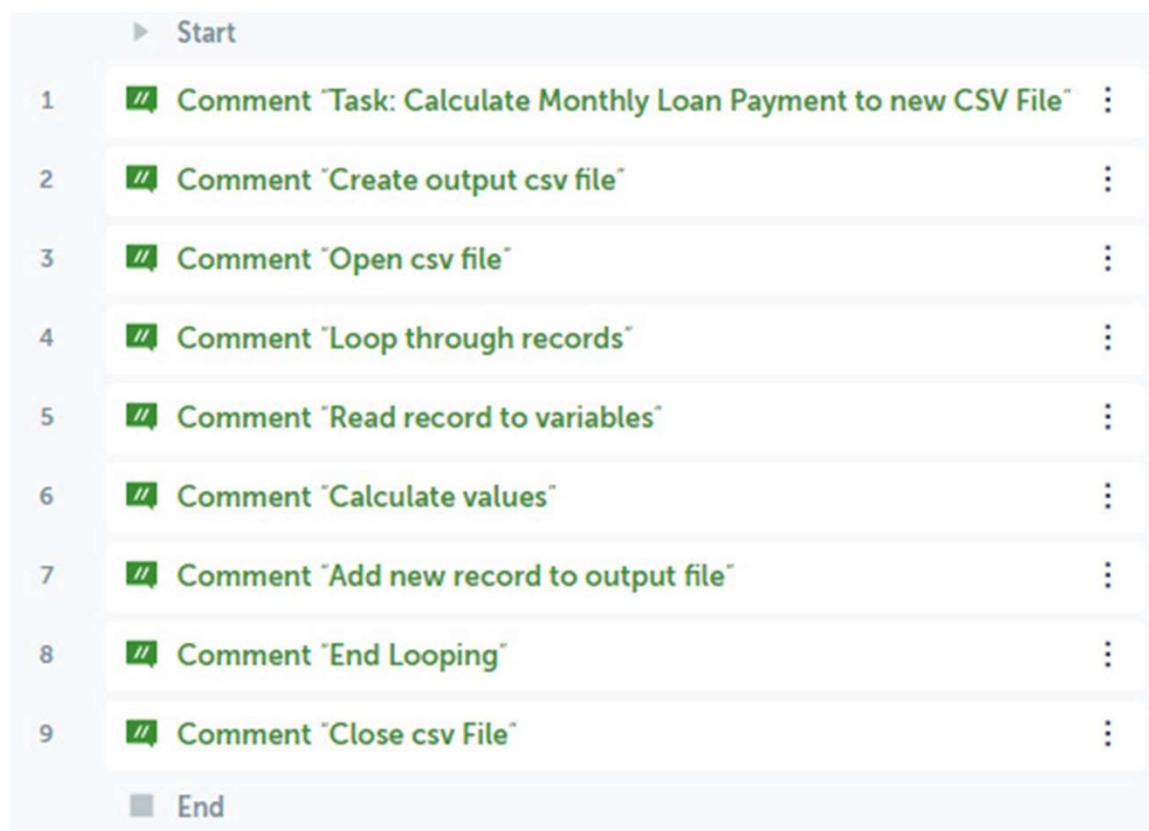
Creating your first bot

- Click on Save. The development interface should look like this:



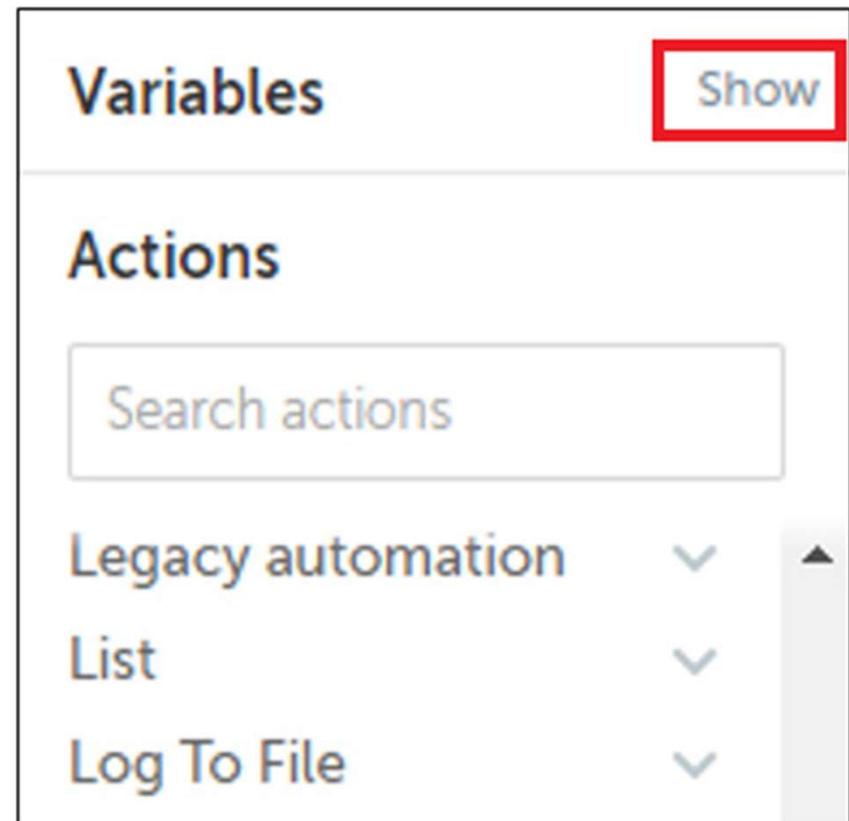
Creating your first bot

- Add a new Comment action as "Close csv file" on line 9 and click on Save.
- Your bot should look something like this:



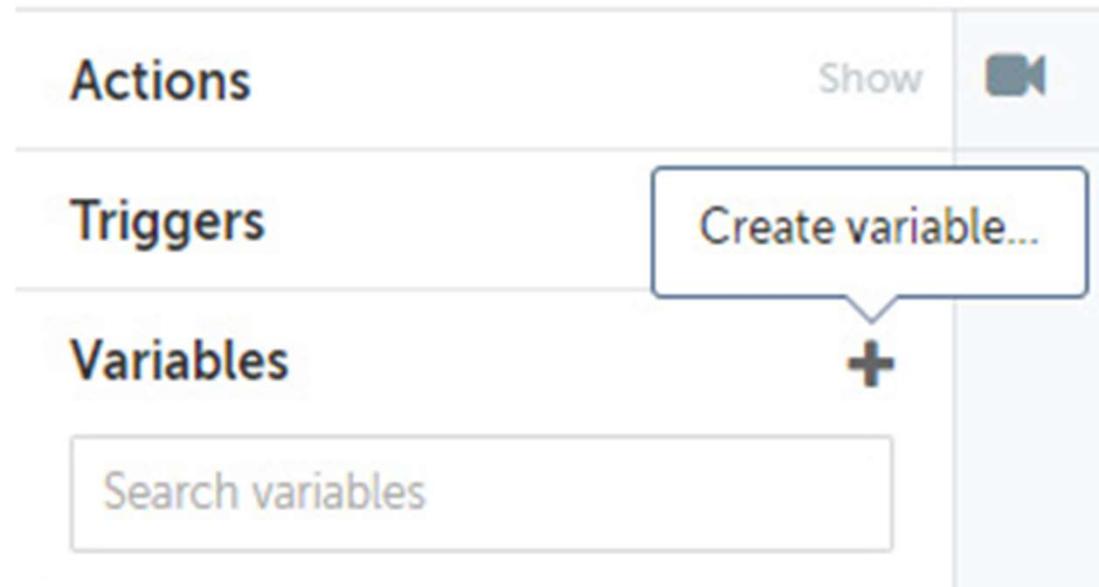
Creating bot variables

- Click on the Show option for the Variables tab from the option pane on the left



Creating bot variables

- Click on the + icon to create a variable:



Creating bot variables

- The Create variable dialog will appear.
- Give your new variable the name strRef, set it as a String type, and then click on Create:

Create variable

Name
strRef

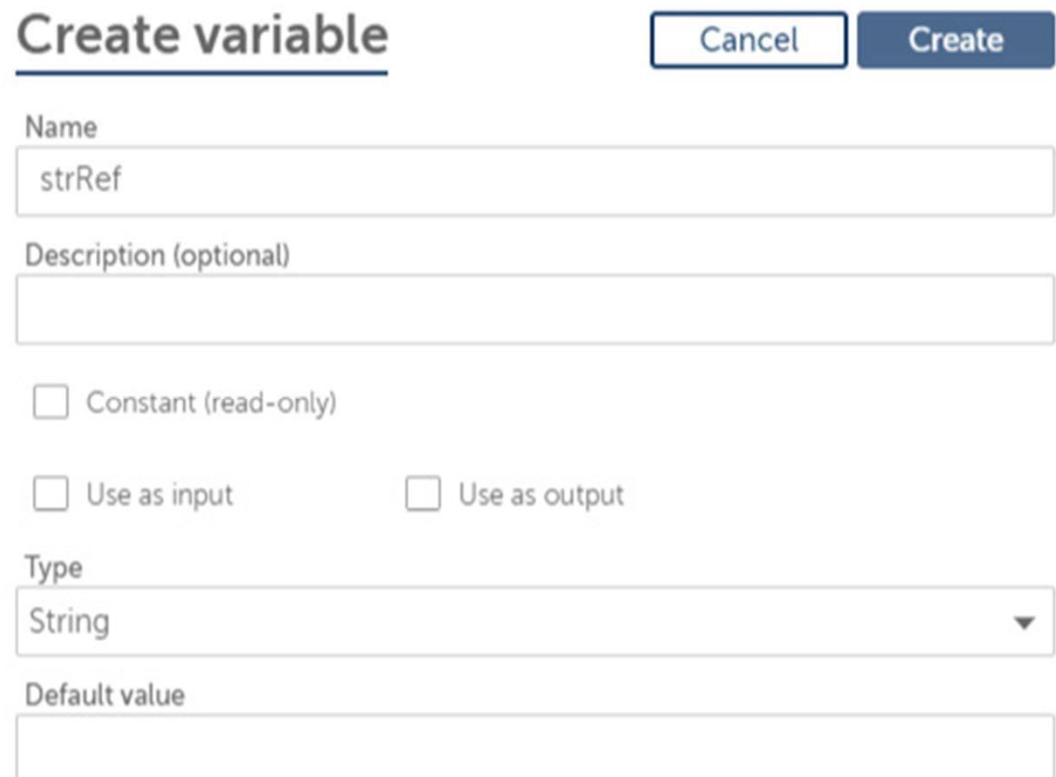
Description (optional)

Constant (read-only)

Use as input Use as output

Type
String

Default value



Creating bot variables

- Create another new variable named recLoan as the Record type.
- You should now have created the following seven variables:

Variables	
Search variables	
	User-defined ^
”	strRef :
#	numAmount :
#	numYears :
#	numInterest :
#	numMonthly :
■	recLoan :
”	strMonthly :

Creating and reading a CSV file

The screenshot shows the Microsoft Power Automate interface. At the top, there are three tabs: Flow, List (which is selected), and Dual. Below the tabs is a toolbar with icons for Video, Share, Run, Stop, Pause, and other workflow controls.

The main area is divided into two columns. The left column, titled "Actions", contains a search bar and a list of actions categorized by provider:

- IQ Bot
- JavaScript
- Legacy automation
- List
- Log To File
 - Log to file
- Loop
- Message box
- Microsoft LUIS NLP (Beta)
- Mouse
- Number
- NumberUtils
- OCR
- Office 365 Calendar

The right column, titled "Triggers", has a section for "Start" and a list of triggers numbered 1 to 10:

- Comment "Task: Calculate Monthly Loan Payment to new CSV File"
- Comment "Create output csv file"
- Log to file** (This trigger is currently selected, indicated by a dark gray background.)
- Comment "Open csv file"
- Comment "Loop through records"
- Comment "Read record to variables"
- Comment "Calculate values"
- Comment "Add new record to output file"
- Comment "End Looping"
- Comment "Close csv File"

A placeholder text "Drag a trigger here..." is visible above the trigger list.

Creating and reading a CSV file

Log to file

Logs any text into a file

File path

“ C:\Hands-On-RPA-with-AA-Sample-Data

Enter text to log

“ Reference,Monthly Amount

Append timestamp

When logging

Append to existing log file

Overwrite existing log file

Encoding

ANSI

Opening and closing a CSV file

	▶ Start	
1	〃 Comment 'Task: Calculate Monthly Loan Payment to new CSV File'	⋮
2	〃 Comment 'Create output csv file'	⋮
3	log Log to file 'Reference,Monthly Amount' to 'C:\Hands-On-RPA-with-AA-Sample-Data\'	⋮
4	〃 Comment 'Open csv file'	⋮
5	CSV/TXT: Open	⚠ ⋮
6	〃 Comment 'Loop through records'	⋮
7	〃 Comment 'Read record to variables'	⋮
8	〃 Comment 'Calculate values'	⋮
9	〃 Comment 'Add new record to output file'	⋮
10	〃 Comment 'End Looping'	⋮
11	〃 Comment 'Close csv File'	⋮
12	CSV/TXT: Close csv/txt "Default"	⋮
	▀ End	

Opening and closing a CSV file

CSV/TXT: Open

Opens a CSV/TXT file

Session name

InputData1

File path

Control Room file Desktop file Variable

C:\Hands-On-RPA-with-AA-Sample-Data\

[x]

Browse...

Required extensions: ".csv", ".txt", ".tsv"

Contains header

Delimiter

Comma

Tab

Regional list separator

Newline

Other

Specific Delimiter (optional)

[x]

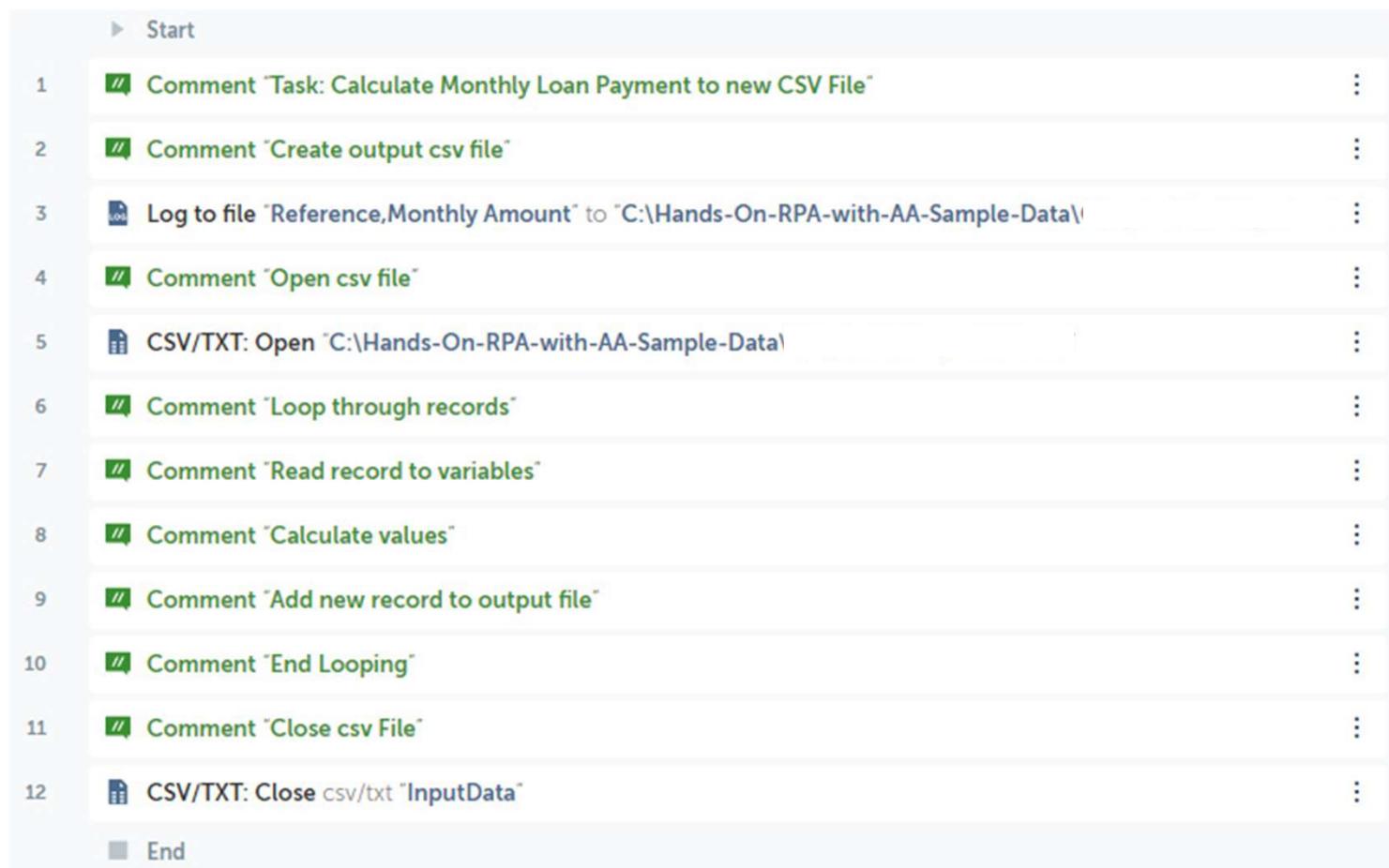
Trim leading spaces

Trim trailing spaces

Encoding

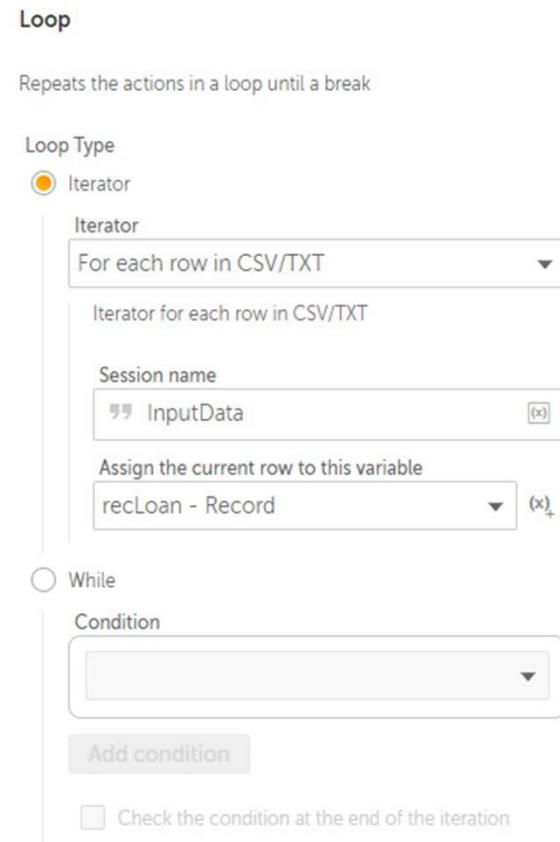
UTF-8

Opening and closing a CSV file



Looping through rows in a CSV file

- Assign the current row to this variable: recLoan – Record
- The action properties dialog should look like this:



Looping through rows in a CSV file

```
> Start
1  # Comment 'Task: Calculate Monthly Loan Payment to new CSV File'
2  ## Comment 'Create output csv file'
3  ! Log to file 'Reference,Monthly Amount' to 'C:\Hands-On-RPA-with-AA-Sample-Data'
4  ## Comment 'Open csv file'
5  ! CSV/TXT: Open 'C:\Hands-On-RPA-with-AA-Sample-Data'
6  ## Comment 'Loop through records'
7  -> Loop for each row in csv/txt
8    ## Comment 'Read record to variables'
9    ## Comment 'Calculate values'
10   ## Comment 'Add new record to output file'
11   ## Comment 'End Looping'
12   ## Comment 'Close csv File'
13   ! CSV/TXT: Close csv/txt 'InputData'
      End
```

Looping through rows in a CSV file

- Select the destination string variable: strRef - String
- The properties should look like the following screenshot:

String: Assign

Assign or Concatenate the given strings

Select the source string variable(s)/ value (optional)

” \$recLoan[0]\$

(x)

Select the destination string variable

strRef - String

▼

(x)

Looping through rows in a CSV file

- Drag the Number: Assign action just below line 9, ensuring it is within the Loop action on line 7.
- Set the following properties for the Number: Assign action on line 10:Select the source string variable/value: \$recLoan[1]\$
- Select the destination number variable: numAmount - Number
- The properties should look like the following screenshot:

Number: Assign

Assigns user specified number to number variable

Select the source string variable/ value

\$recLoan[1]\$

Specify value to assign to number

Select the destination number variable

numAmount - Number



(x)₊

Looping through rows in a CSV file

- Set the following properties for the Number: Assign action on line 11:Select the source string variable/value: \$recLoan[2]\$
- Select the destination number variable: numYears - Number
- The properties should look like the following screenshot:

Number: Assign

Assigns user specified number to number variable

Select the source string variable/ value

\$recLoan[2]\$ (x)

Specify value to assign to number

Select the destination number variable

numYears - Number ▼ (x) +

Looping through rows in a CSV file

- Set the following properties for the Number: Assign action on line 12:
- Select the source string variable/value: \$recLoan[3]\$
- Select the destination number variable: numInterest - Number
- The properties should look like the following screenshot:

Number: Assign

Assigns user specified number to number variable

Select the source string variable/ value

\$recLoan[3]\$

Specify value to assign to number

Select the destination number variable

numInterest - Number



Looping through rows in a CSV file

```
> Start
1  // Comment 'Task: Calculate Monthly Loan Payment to new CSV File'
2  // Comment 'Create output csv file'
3  ↗ Log to file 'Reference,Monthly Amount' to 'C:\Hands-On-RPA-with-AA-Sample-Data'
4  // Comment 'Open csv file'
5  ↗ CSV/TXT: Open 'C:\Hands-On-RPA-with-AA-Sample-Data\' 
6  // Comment 'Loop through records'
7  ↘ Loop for each row in csv/txt
8  // Comment 'Read record to variables'
9  # String: Assign $recLoan[0]$ to $strRef$
10 # Number: Assign $recLoan[1]$ to $numAmount$
11 # Number: Assign $recLoan[2]$ to $numYears$
12 # Number: Assign $recLoan[3]$ to $numInterest$
13 // Comment 'Calculate values'
14 // Comment 'Add new record to output file'
15 // Comment 'End Looping'
16 // Comment 'Close csv File'
17 ↗ CSV/TXT: Close csv/txt "InputData"
█ End
```

Performing basic arithmetic calculations

- Set the following properties for the Number: Assign action on line 14:Select the source string variable/value: (\$numAmount\$ + \$numInterest\$)/(12 * \$numYears\$)
- Select the destination number variable: numMonthly - Number
- The action properties dialog should look like this:

Number: Assign

Assigns user specified number to number variable

Select the source string variable/ value

(\$numAmount\$ + \$numInterest\$) / (12 * \$numYears\$)

Specify value to assign to number

Select the destination number variable

numMonthly - Number

Appending records to a CSV file

- Set the following properties for the String: Assign action on line 16:
- Enter a number: \$numMonthly\$
- Enter number of digits after decimal: 2
- Assign the output to variable: strMonthly - String
- The action properties dialog should look like this:

Number: To string

Converts a user specified number to a string

Enter a number

\$numMonthly\$ (x)

Specify number to convert to string e.g. 35

Enter number of digits after decimal (number format)

2 (x)

e.g for number 35.265, enter the number of digits after decimal as 3

Assign the output to variable

strMonthly - String (x) (+)

Appending records to a CSV file

Log to file

Logs any text into a file

File path

“ C:\Hands-On-RPA-with-AA-Sample-Data\



Browse...

Enter text to log

“ \$strRef\$, \$strMonthly\$



Append timestamp

When logging

Append to existing log file

Overwrite existing log file

Encoding

ANSI



Appending records to a CSV file

```
> Start
1  # Comment 'Task: Calculate Monthly Loan Payment to new CSV File'
2  # Comment 'Create output csv file'
3  Log to file 'Reference,Monthly Amount' to 'C:\Hands-On-RPA-with-AA-Sample-Data\
4  # Comment 'Open csv file'
5  CSV/TXT: Open 'C:\Hands-On-RPA-with-AA-Sample-Data\
6  # Comment 'Loop through records'
7  -  Loop for each row in csv/btx
8    # Comment 'Read record to variables'
9    String: Assign $recLoan[0]$ to $strRef$
10   # Number: Assign $recLoan[1]$ to $numAmount$
11   # Number: Assign $recLoan[2]$ to $numYears$
12   # Number: Assign $recLoan[3]$ to $numInterest$
13   # Comment 'Calculate values'
14   # Number: Assign '$numAmount$ + $numInte...' to $numMonthly$
15   # Comment 'Add new record to output file'
16   # Number: To string convert $numMonthly$ to a string datatype and assign output to $strMonthly$
17   Log to file '$strRef$', $strMonthly$ to 'C:\Hands-On-RPA-with-AA-Sample-Data\
18   # Comment 'End Looping'
19   # Comment 'Close csv File'
20   CSV/TXT: Close csv/btx 'InputData'
  End
```

Appending records to a CSV file

- You have done some great work, Go ahead and run your first bot.
- Your first bot should create the output CSV file containing the calculated Monthly Payments.
- The output file should look like this in Excel:

	A	B
1	Reference	Monthly Amount
2	508-001	176.66
3	523-679	147.22
4	524-602	110.41
5	534-001	265.00
6	302-170	298.33
7	230-614	53.00
8	550-205	176.66

Summary

- You have built your first bot using Automation Anywhere.
- You are now a lot closer to becoming an RPA developer, having gained some valuable knowledge with the help of a real-life business process.
- You should be very comfortable with using actions to make your bot perform tasks.
- We have already looked at comma-separated files, in other words, reading, creating, and appending.

Introducing Variables



Introducing Variables

In this lesson, we will cover the following topics:

- Working with different variable types
- Using message boxes and prompts
- Converting data types

Working with different variable types

- There are other data types available in and each type is represented by a specific icon.
- Here, you can see the icons for each type of variable:

 Boolean	 List
 Credential	 Number
 Datetime	 Record
 Dictionary	 String
 File	 Table
 Form	 Window

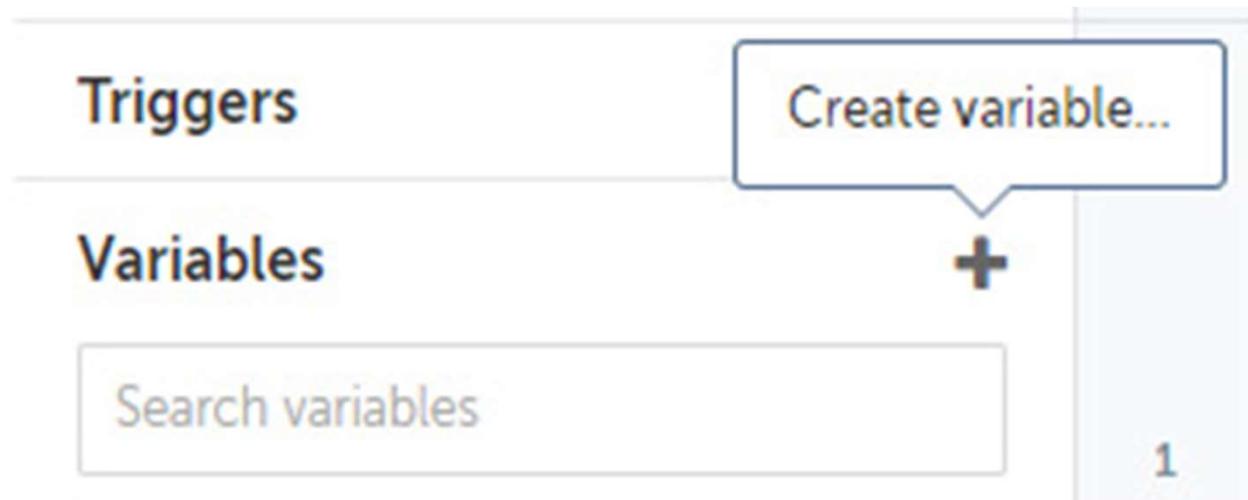
Working with different variable types

- Like most development platforms, it is good practice to use a naming convention when creating variable names.
- This course will be using the following variable naming notation:

Data Type	Variable name Prefix
Number	num
String	str
Boolean	bln
Date\Time	dte
Record	rec
List	lst
Table	tbl
Dictionary	dct

Using the String variable type

- Log in to the Control Room.
- Create a new bot and call it lesson 6 - Variables in the \Bot\ folder.
- Expand the Variables pane from the options on the left and select + to create a new variable:



Using the String variable type

- The Create variable dialog will appear.
- Call this variable strFirstName and set it as a String type.
- Once the details are entered, click on Create, The dialog should look like this:

Create variable

Name
strFirstName
Max characters = 50

Description (optional)
Max characters = 255

Use as input
 Use as output
 Constant (read-only)

Type
String

Default value

Using the String variable type

- Create another new variable named strFullscreen as a String type.
- Your variable list should appear as follows:

The screenshot shows a user interface for managing variables. At the top, there is a header labeled "Variables" with a plus sign icon for adding new variables. Below the header is a search bar with the placeholder text "Search variables". The main area displays a list of variables under the heading "User-defined". The list contains three entries, each consisting of a double quotes icon, a variable name, and a more options icon. The variable names are "strFirstName", "strFullscreen", and "strSurname".

User-defined
” strFirstName
” strFullscreen
” strSurname

Using the String variable type

- Add a new Comment action as "-----" on line 4 and click on Save.
- Your bot should now look like this:



Using the String variable type

- Set the following properties for the String: Assign action on line 2:
- Select the source string variable(s)/ value (optional): John
- Select the destination string variable: strFirstName - String
- The action properties dialog should look like this:

String: Assign

Assign or Concatenate the given strings

Select the source string variable(s)/ value (optional)

“ John

(x)

Select the destination string variable

strFirstName - String

▼

(x)
+

Using the String variable type

- Set the following properties for the String: Assign action on line 3:
- Select the source string variable(s)/ value (optional): Smith
- Select the destination string variable: strSurname - String
- The action properties dialog should look like this:

String: Assign

Assign or Concatenate the given strings

Select the source string variable(s)/ value (optional)

“ Smith (x)

Select the destination string variable

strSurname - String ▼ (x)

Using the String variable type

- Set the following properties for the String: Assign action on line 5:
- Select the source string variable(s)/ value (optional): \$strFirstName\$ \$strSurname\$
- Select the destination string variable: strFullname - String
- The action properties dialog should look like this:

String: Assign

Assign or Concatenate the given strings

Select the source string variable(s) / value (optional)

“ ” \$strFirstName\$ \$strSurname\$

Select the destination string variable

strFullname - String

Using the String variable type

- Set the following properties for the Message box action on line 7:
- Enter the message box window title: Merged variables
- Enter the message to display: \$strFullscreen\$
- The action properties dialog should look like this:

Message box

Displays a message box

Enter the message box window title

” Merged Variables

Enter the message to display

” \$strFullscreen\$

Using the String variable type

- Click on Save, The development interface should look something like this:

1	// Comment "String Variables"	:
2	" String: Assign "John" to \$strFirstName\$:
3	" String: Assign "Smith" to \$strSurname\$:
4	// Comment "Merge variables"	:
5	" String: Assign "\$strFirstName\$ \$strSurname\$" to \$strFullname\$:
6	// Comment "Show Output"	:
7	Message box \$strFullname\$:
8	// Comment "-----"	:

Using the Datetime variable type

Create variable

Cancel

Create

Name

dteChristmas

Max characters = 50

Description (optional)

Max characters = 255

Use as input

Use as output

Constant (read-only)

Type

Datetime

Default value (optional)

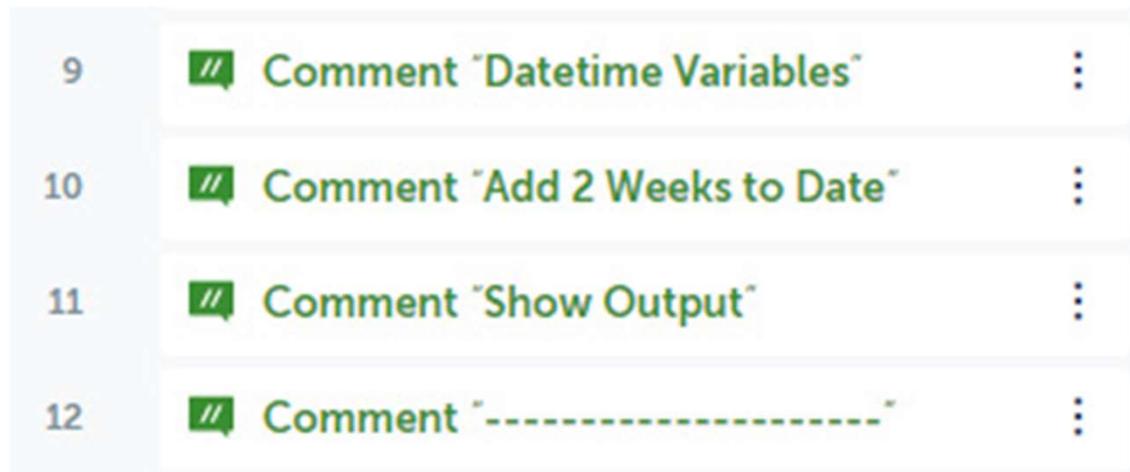
 12/25/2019

 12:00 AM

BST (UTC+1:00) London, Europe

Using the Datetime variable type

- Add a new Comment action as "-----" on line 12 and click on Save.
- Your bot should look like this:



Using the Datetime variable type

Datetime: Add

Adds a specified time unit to a given date and time

Source date and time variable

dteChristmas - Datetime ▾ (x) 

Time value to add

2 

Maximum value for addition is "69948627" hours

Time unit to add

Weeks ▾

Assign the output to a variable

dteChristmasPlus2Weeks - Datetime ▾ (x) 

Using the Datetime variable type

Datetime: To string

Converts a datetime value to a string value and assigns it to a string variable

Source date and time variable

dteChristmasPlus2Weeks - Datetime ▾ (x)₊

Select date time format

Formats

▼

Custom format

▼ DD MMM YYYY (x)₊

Assign the output to a variable

strDate - String ▾ (x)₊

Using the Datetime variable type

- Set the following properties for the Message box action on line 14:
Enter the message box window title: Datetime variables
- Enter the message to display: \$strDate\$
- The action properties dialog should look like this:

Message box

Displays a message box

Enter the message box window title

” Datetime Variables

Enter the message to display

” \$strDate\$

Using the Datetime variable type

- 9 Comment "Datetime Variables" : ...
- 10 Comment "Add 2 Weeks to Date" : ...
- 11 Datetime: Add 2 Weeks to \$dteChristmas\$ and assign result to \$dteChristmasPlus2Weeks\$: ...
- 12 Comment "Show Output" : ...
- 13 Datetime: To string Convert \$dteChristmasPlus2Weeks\$ and assign result to \$strDate\$: ...
- 14 Message box \$strDate\$: ...
- 15 Comment "-----" : ...

Using the Boolean variable type

- The Create variable dialog will appear. Set the following values:

Name: blnLeapYear

Type: Boolean

Default value: True

The dialog should look like this:

Create variable Cancel **Create**

Name Max characters = 50

Description (optional) Max characters = 255

Use as input

Use as output

Constant (read-only)

Type ▾

Default value False **True**

Using the Boolean variable type

- Add another Comment action as "-----" as line 20 and click on Save.
- Your bot should look like this:

16	// Comment "Boolean Variables"	:
17	// Comment "Assign Boolean Value"	:
18	// Comment "Invert Boolean Value"	:
19	// Comment "Show Output"	:
20	// Comment "-----"	:

Using the Boolean variable type

- Set the following properties for the Boolean: Assign action on line 18:
- Select the source Boolean variable/ value: Constant values
- Constant values: True
- Select the destination Boolean variable: blnLeapYear - Boolean
- The action properties should look like this:

Boolean: Assign

Assigns the source boolean variable's value or the user defined value to the destination boolean variable

Select the source boolean variable/ value

Constant values

True

False

Variable value

Select the destination boolean variable

blnLeapYear - Boolean



Using the Boolean variable type

- Set the following properties for the Boolean: Invert action on line 20:

Select the Boolean variable to be inverted: Variable

Value: \$blnLeapYear\$

Assign the output: blnLeapYear - Boolean

The Boolean: Invert action properties should look like this:

Boolean: Invert

Inverts a boolean variable's value i.e. converts True to False and False to True and assigns the output to a variable (same or different)

Select the boolean variable to be inverted

False True Variable

 \$blnLeapYear\$ (x)

Assign the output to

blnLeapYear - Boolean (x)

Using the Boolean variable type

- Set the following properties for the Boolean: To string action on line 22:

Select Boolean variable:

blnLeapYear - Boolean

Select the string variable to store the result: strLeapYear - String

The action properties should look like this:

Boolean: To string

Converts a boolean value to string and assigns it to a string variable

Select boolean variable

blnLeapYear - Boolean

Select the string variable to store the result

strLeapYear - String

Using the Boolean variable type

- Set the following properties for the Message box action on line 23:
- Enter the message box window title: Boolean variables
- Enter the message to display: \$strLeapYear\$
- The action properties dialog should look like this:

Message box

Displays a message box

Enter the message box window title

” Boolean Variables

Enter the message to display

” \$strLeapYear\$

Using the Boolean variable type

16	#[Comment "Boolean Variables"	:
17	#[Comment "Assign Boolean Value"	:
18	FLAG Boolean: Assign True to \$blnLeapYear\$:
19	#[Comment "Invert Boolean Value"	:
20	FLAG Boolean: Invert value of boolean variable \$blnLeapYear\$ and assign result to \$bl...	:
21	#[Comment "Show Output"	:
22	FLAG Boolean: To string \$blnLeapYear\$ and assign result to a \$strLeapYear\$:
23	INFO Message box \$strLeapYear\$:
24	#[Comment "-----"	:

Using the Number variable type

- Add another Comment action as "-----" as line 29 and click on Save. Your bot should look like this:

25	#[Comment "Number Variables"	:
26	#[Comment "Assign Random Value"	:
27	#[Comment "Apply Formula"	:
28	#[Comment "Show Output"	:
29	#[Comment "-----"	:

Using the Number variable type

- Set the following properties for the Number: Random action on line 27:
- Beginning of range: 1
- End of range: 100
- Save the outcome to a number variable: numRandom - Number
- The action properties should look like this:

Number: Random

Assigns a random number to a number variable

Beginning of range:

Accepts decimal and negative value.

End of range:

Must be larger than beginning of range.

Save the outcome to a number variable

Using the Number variable type

- Set the following properties for the Number: Assign action on line 29:
- Select the source string variable: `($numRandom$/2) + 25`
- Select the destination number variable: numResult - Number
- The action properties should look like this:

Number: Assign

Assigns user specified number to number variable

Select the source string variable/ value

`($numRandom$/2) + 25` (x)

Specify value to assign to number

Select the destination number variable

`numResult - Number` ▼ (x) +

Using the Number variable type

- Set the following properties for the Number: To string action on line 31:
- Enter a number: \$numResult\$
- Enter number of digits after decimal: 2
- Assign the output to variable: strResult - String
- The action properties should look like this:

Number: To string

Converts a user specified number to a string

Enter a number

\$numResult\$

Specify number to convert to string e.g. 35

Enter number of digits after decimal (number format)

2

e.g for number 35.265, enter the number of digits after decimal as 3

Assign the output to variable

strResult - String

Using the Number variable type

- Set the following properties for the Message box action on line 32:
- Enter the message box window title: Number variables
- Enter the message to display: \$strResult\$
- The action properties dialog should look like this:

Message box

Displays a message box

Enter the message box window title

” Number Variables

Enter the message to display

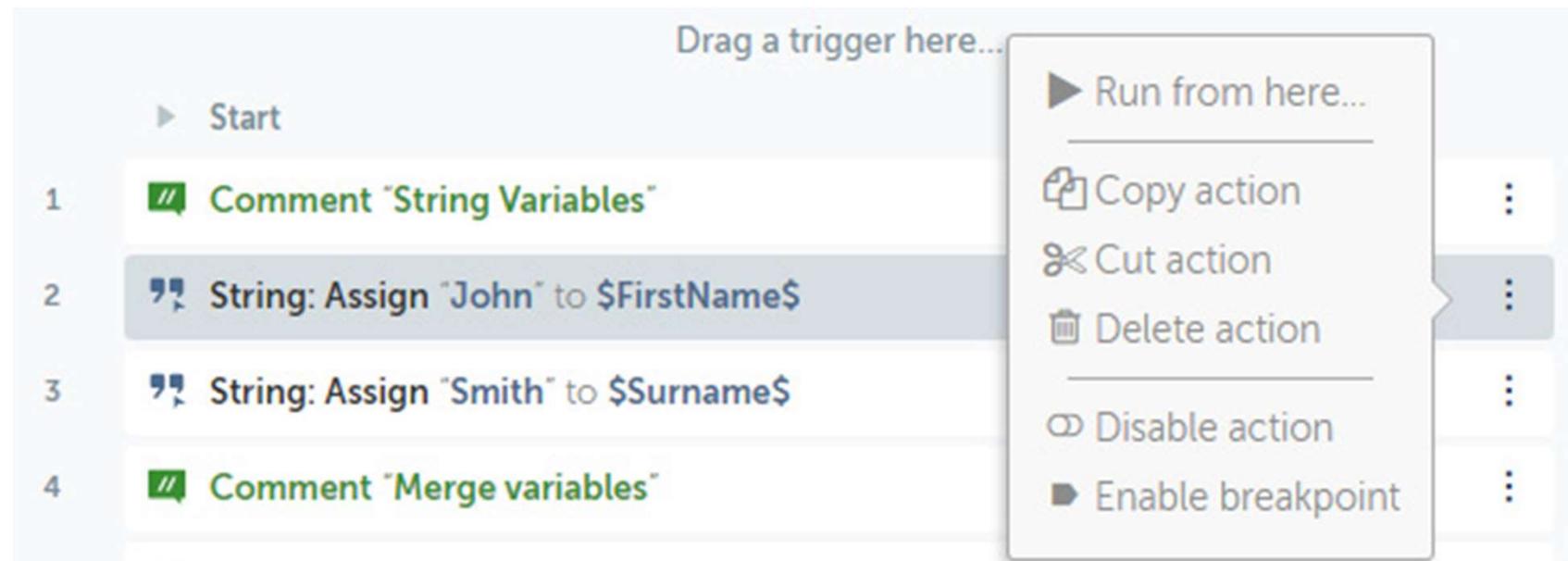
” \$strResult\$

Using the Number variable type

25	Comment "Number Variables"	⋮
26	Comment "Assign Random Value"	⋮
27	Number: Random Assign a random number from beginning of range 1 to end of range 100 to \$numRandom\$	⋮
28	Comment "Apply Formula"	⋮
29	Number: Assign <code>"(\$numRandom\$/2) + 25"</code> to \$numResult\$	⋮
30	Comment "Show Output"	⋮
31	Number: To string convert \$numResult\$ to a string datatype and assign output to \$strResult\$	⋮
32	Message box \$strResult\$	⋮
33	Comment "-----"	⋮

Using message boxes and prompts

- Select the options menu for line 2:



Using message boxes and prompts

- Set the following properties for the Prompt: For value action on line 3:
- Prompt window caption: Prompt for String
- Prompt message: Enter Firstname:
- Assign the value to a variable: strFirstName - String
- The action properties should look like this:

Prompt: For value

Prompts user for entering a value

Prompt window caption

” Prompt for String (x)

Prompt message

” Enter Firstname: (x)

Mask keystroke

Assign the value to a variable

strFirstName - String ▼ (x) +

Using message boxes and prompts

- 1 Comment "String Variables" : ...
- 2 String: Assign "John" to \$strFirstName\$ ∅ :
- 3 Prompt: For value : ...
- 4 String: Assign "Smith" to \$strSurname\$ ∅ :
- 5 Prompt: For value : ...
- 6 Comment "Merge variables" : ...
- 7 String: Assign "\$strFirstName\$ \$strSurname\$" to \$strfullname\$: ...
- 8 Comment "Show Output" : ...
- 9 Message box \$strfullname\$: ...
- 10 Comment "-----" : ...

Converting data types

Let's look at the data conversions performed by the current bot. The first data conversion the bot performs is converting a **Datetime** data type to a **String** data type. Take a look at line **15** from the development interface:

```
15  📅 Datetime: To string Convert $dteChristmasPlus2Weeks$ and assign result to $strDate$ :
```

- Converting a Datetime data type to String

The next data conversion performed is from a **Boolean** data type to a **String** data type. This is demonstrated on line **24** of the development interface:

```
24  🏟 Boolean: To string $blnLeapYear$ and assign result to a $strLeapYear$ :
```

- Converting a Boolean data type to String

Finally, the bot converts a **Number** data type to a **String** data type. This is on line **33** of the development interface:

```
33  # Number: To string convert $numResult$ to a string datatype and assign output to $strResul... :
```

Summary

- To recap, you have learned how to create different types of variables from numbers, Booleans, and dates.
- Your knowledge has been expanded by assigning values to these variables as well as performing calculations.
- We also explored the reasons for using, and how to convert, different data types.
- Having also created prompts and message boxes, you are well on your way to start learning some more exciting Automation Anywhere packages and actions.

Interacting with Applications



Interacting with Applications

In this lesson, we will cover the following topics:

- Automating web applications
- Automating desktop applications
- Simulating keystrokes

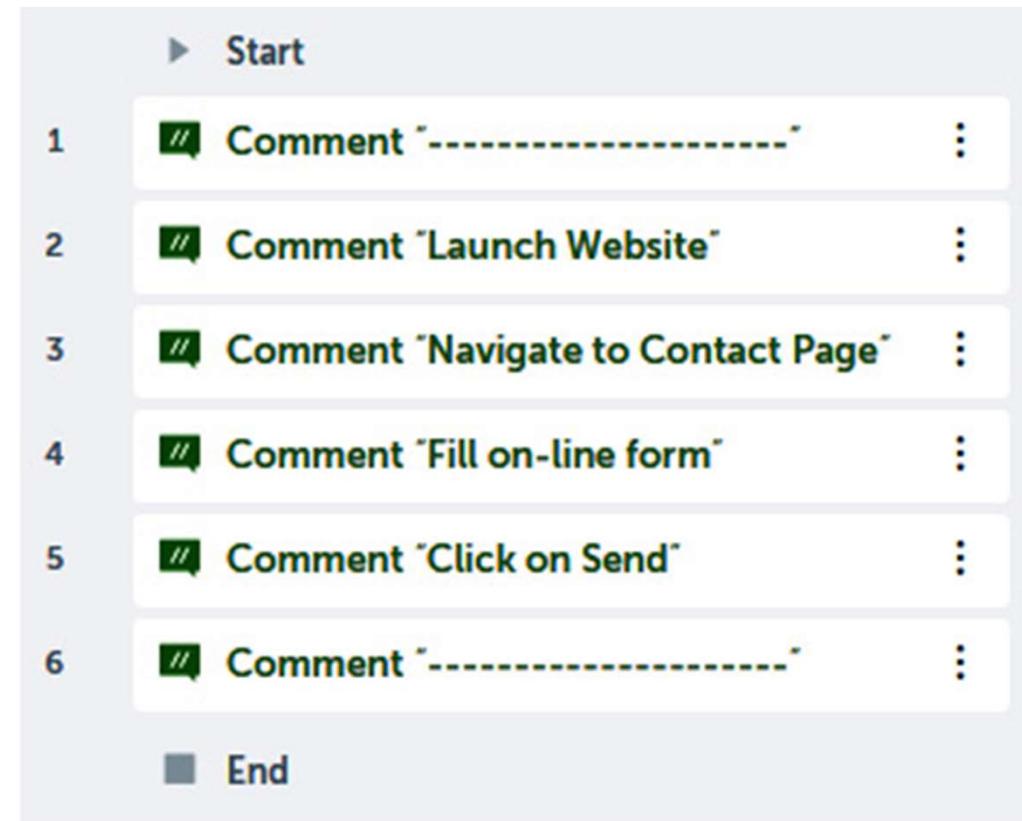
Interacting with Applications

- By the end of this lesson, you will have the skills needed to build bots that can launch applications, navigate through various application interfaces, interact with buttons and checkboxes, and read and enter data.
- This lesson will be using the following packages:

<input checked="" type="checkbox"/> Application	▼
<input checked="" type="checkbox"/> Browser	▼
<input checked="" type="checkbox"/> Comment	▼
<input checked="" type="checkbox"/> Message box	▼
<input checked="" type="checkbox"/> Recorder	▼
<input checked="" type="checkbox"/> Simulate keystrokes	▼
<input checked="" type="checkbox"/> Window	▼

Automating web applications

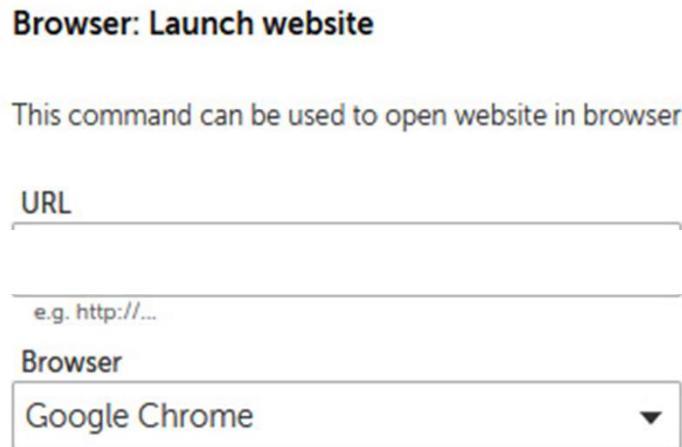
- Add a new Comment action as "-----" on line 6 and click on Save.
- Your bot should look like this:



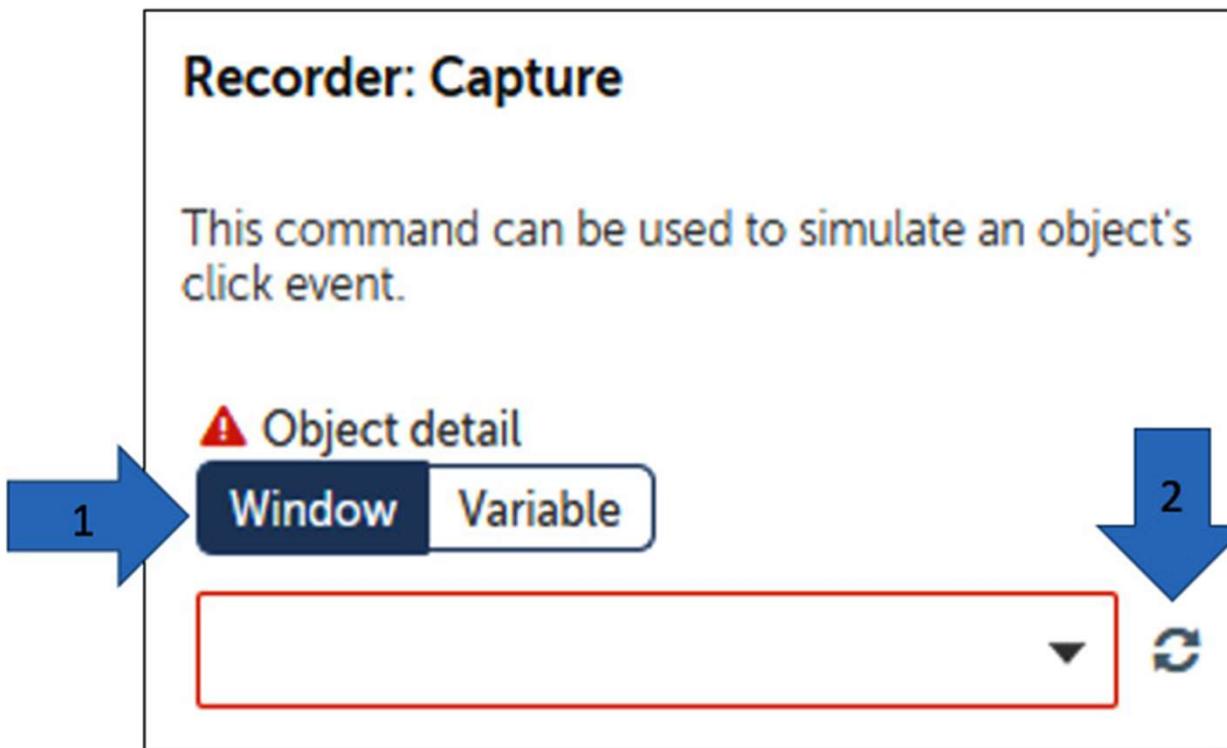
Automating web applications

Set the following properties for the Browser: Launch website action on line 3:

- URL: <http://fenago.net/>
- Browser: Google ChromeThe Browser: Launch website action properties should look like this:



Automating web applications



Automating web applications

- The drop-down list will show all windows that are currently open.
- Select Home - Google Chrome.
- The Recorder: Capture action properties should look like this:

Recorder: Capture

This command can be used to simulate an object's click event.

Object detail

Window Variable

Home - Google Chrome

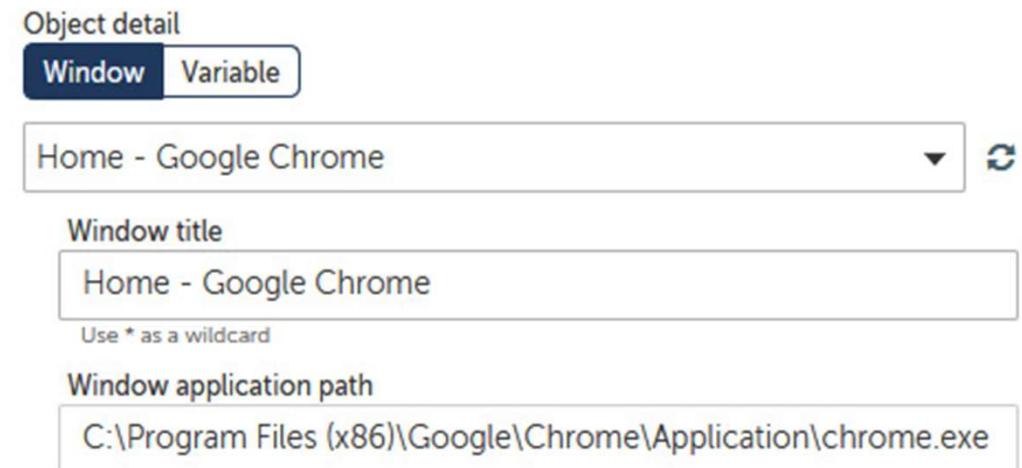
Window title

Home - Google Chrome

Use * as a wildcard

Window application path

C:\Program Files (x86)\Google\Chrome\Application\chrome.exe



Automating web applications



The Bots Are Coming...



Are you ready for the Digital Workforce?

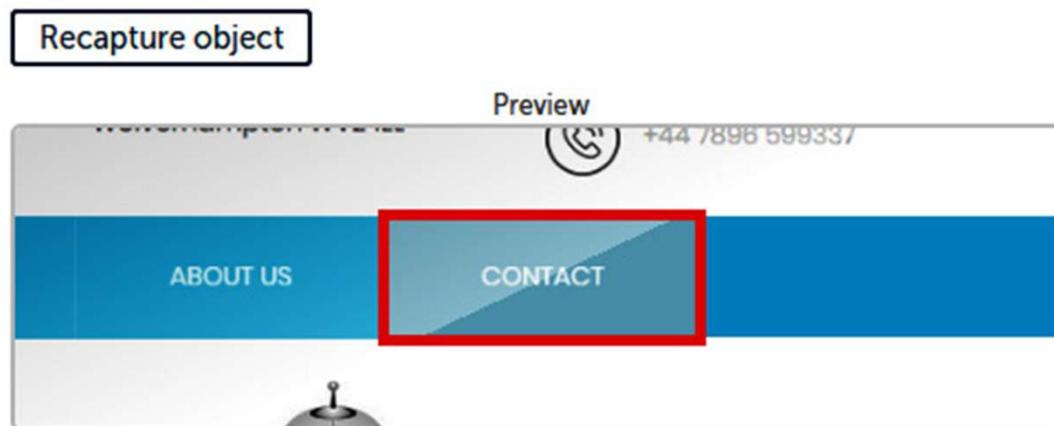
Robotic Processing Automation (RPA) is currently one of the fastest growing technologies around the world. Automation of repetitive, tedious and boring tasks can help your business get most out of your workforce.

What is Robotic Process Automation?

Commonly known as RPA it enables organisations and businesses to 'teach' computers to execute repetitive and time consuming tasks. Being a software based tool, it is easily able to connect and integrate with most desktop applications as well as integration with back-end systems via API's.

Automating web applications

- Once the correct object has been identified with the red border, click to select it.
- Once clicked, the bot will capture all the attributes it needs.
- You will see the selected object in the preview section of the properties, It should look something like this:



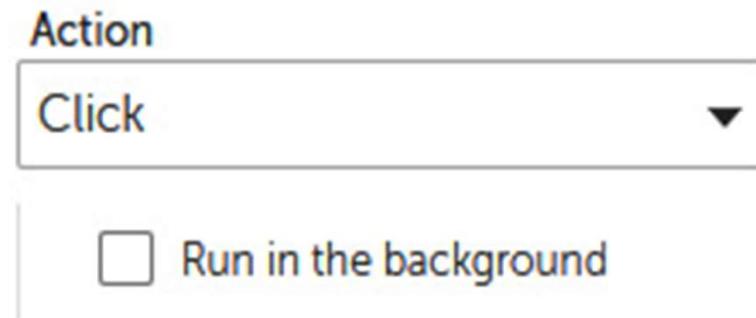
Automating web applications

- The next step is to look at the attributes used to identify the object.
- If we collapse the Object properties list, you will see only the checked attributes used to identify the object.
- The properties should look like this:

Object properties (8 of 48)	
Name	Value
Control Type	LINK
Technology Type	HTML
HTML Tag	” A
DOMXPath	” /html/body/div[4]/div[3]/div[1]/div[1]/div[1] ◀ ▶
HTML HasFrame	” false
HTML InnerText	” CONTACT
HTML Href	”
Path	” 5 3 1 1 1 1 1 1 1 1 -1 1

Automating web applications

- Now that the bot has found the CONTACT tab, the next step is to get the bot to click on it.
- This is where we set the Action property, This should be set to Click, as shown here:



Automating web applications

- We have set the first bot interaction with the website.
- The development interface for this step should look like this:

2	Comment 'Launch Website'	:
3	Browser: Launch website	
4	Comment 'Navigate to Contact Page'	:
5	Recorder: Capture Click on link HTML InnerText "CONTACT" in window "Home - Google Chrome"	:

Automating web applications

HOME SERVICES TRAINING ABOUT US CONTACT

Get In Touch With Us

We are more than happy to help however we can so, please do not hesitate to contact us

(location pin)

(phone receiver)

(envelope)

Name *

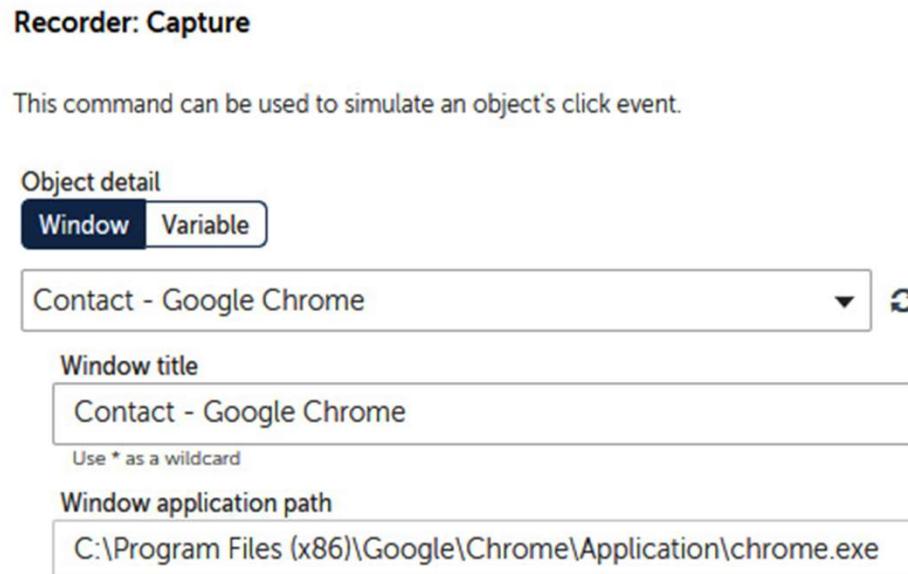
Email *

Message *

Send

Automating web applications

- Refresh the windows drop-down list and select Contact - Google Chrome.
- The action properties should look like this:



Automating web applications

- When the Contact web page appears, hover the mouse over the Name textbox until it has a red border around it, as shown in the following screenshot:

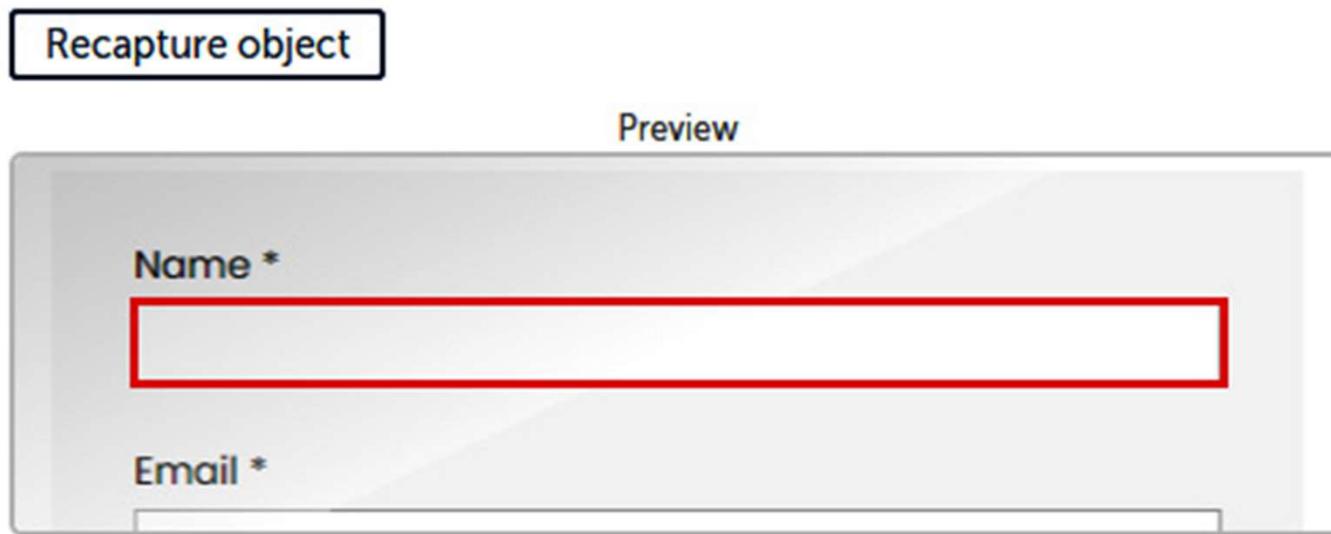
A screenshot of a contact form. It consists of three input fields: 'Name *' (highlighted with a red border), 'Email *', and 'Message *'. Below the inputs is a blue 'Send' button.

Name *
Email *
Message *

Send

Automating web applications

- Click in the red border to capture it.
- Once captured, check the preview to ensure that the correct object has been captured, as shown in the following screenshot:



Automating web applications

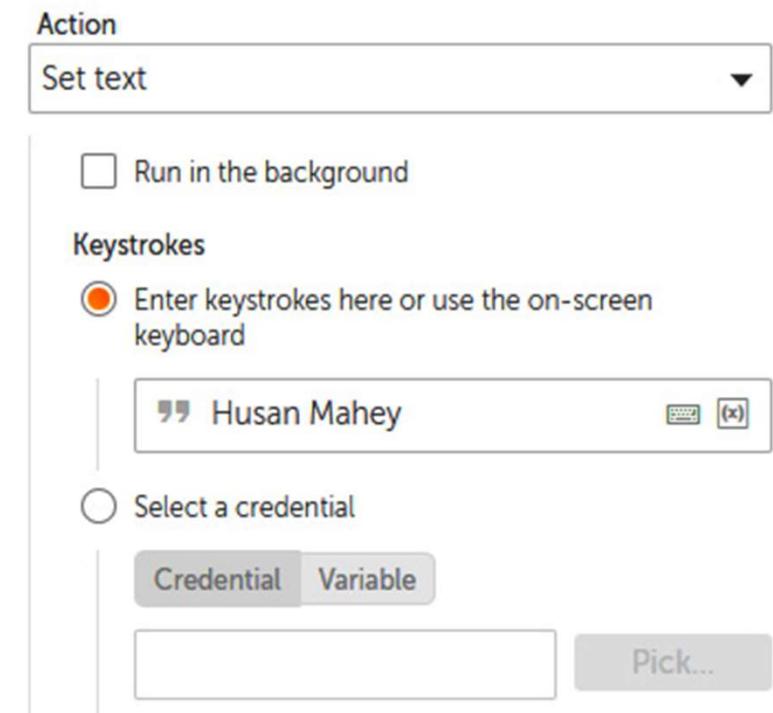
- This time we want to populate the textbox with your name. Set the following properties for the Recorder:
Capture action on line 7:

Action: Set text

Keystrokes: Enter keystrokes here or use the on-screen keyboard

Value: (Enter your name)

The properties should look like the following screenshot:



Automating web applications

- When the Contact web page appears, hover the mouse over the Email textbox until it has a red border around it, as shown in the following screenshot:

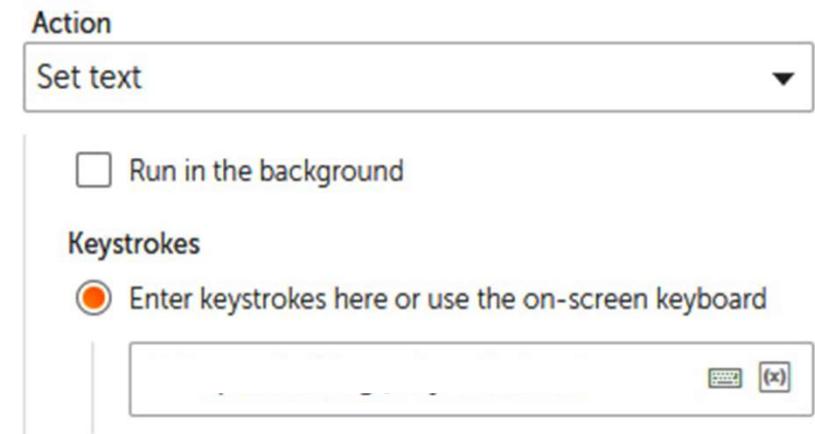
The screenshot shows a contact form with three fields: 'Name *' (empty), 'Email *' (highlighted with a red border), and 'Message *' (empty). A blue 'Send' button is at the bottom right.

Name *	<input type="text"/>
Email *	<input type="text"/>
Message *	<input type="text"/>

Send

Automating web applications

- This time we want to populate the textbox with your email address.
Set the following properties for the Recorder: Capture action on line 8:
- Action: Set text
- Keystrokes: Enter keystrokes here or use the on-screen keyboard
- Value: (Enter your email address)
- The properties should look similar to the following screenshot:



Automating web applications

- When the Contact web page appears, hover the mouse over the Message textbox until it has a red border around it, as shown in the following screenshot:

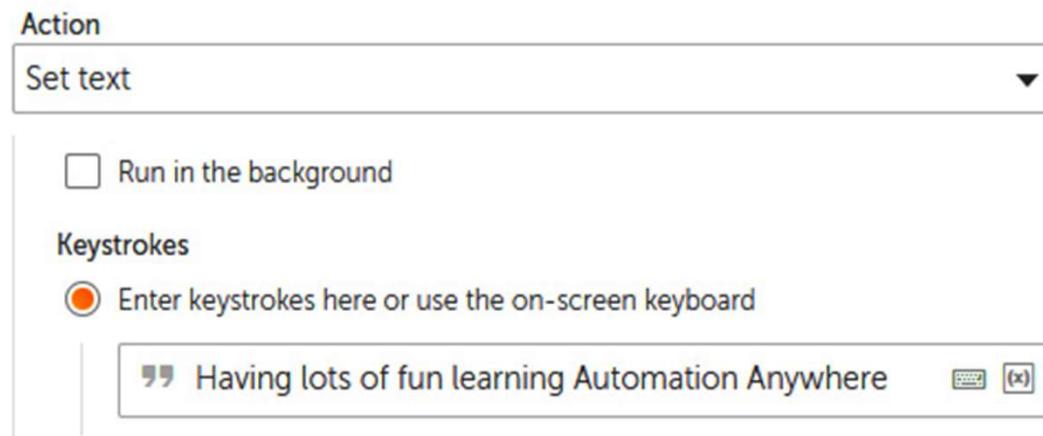
The screenshot shows a contact form with three fields: 'Name *', 'Email *', and 'Message *'. The 'Message' field is highlighted with a red border, indicating it is the current target for the mouse cursor. A blue 'Send' button is located at the bottom right.

Name *	<input type="text"/>
Email *	<input type="text"/>
Message *	<input type="text"/>

Send

Automating web applications

- This time we want to populate the textbox with your email address. Set the following properties for the Recorder: Capture action on line 9:
- Action: Set text
- Keystrokes: Enter keystrokes here or use the on-screen keyboard
- Value: Having lots of fun learning Automation Anywhere
- The properties should look similar to the following screenshot:



Automating web applications

- Click on Save. Your development interface for this section should look like this

6	Comment 'Fill on-line form'	:
7	Recorder: Capture Set text on textbox HTML Name "Name" in window "Contact - Google ..."	:
8	Recorder: Capture Set text on textbox HTML Name "Email" in window "Contact - Google C..."	:
9	Recorder: Capture Set text on textbox HTML Name "Message" in window "Contact - Googl..."	:

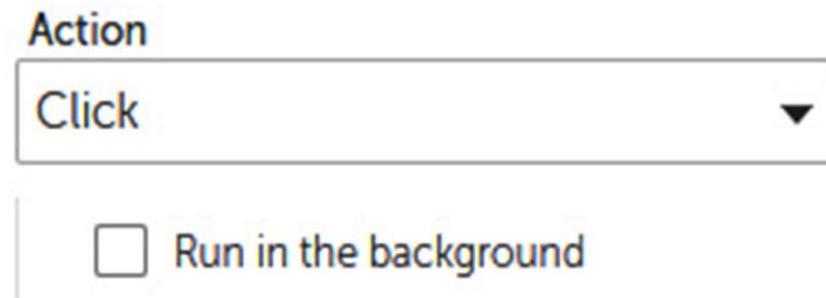
Automating web applications

- When the Contact web page appears, hover the mouse over the Send button until it has a red border around it, as shown in the following screenshot:

A screenshot of a contact form. It consists of three input fields: 'Name *' (empty), 'Email *' (empty), and 'Message *' (empty). Below the message field is a large text area. At the bottom right is a blue rectangular button labeled 'Send'. A red border surrounds the 'Send' button, indicating it is the current target for the mouse cursor.

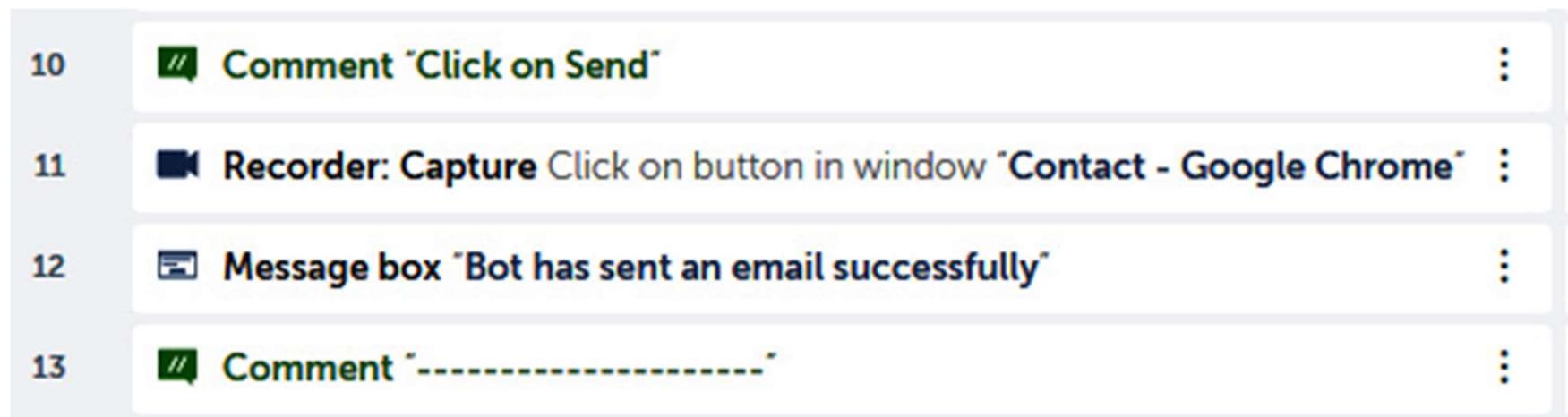
Automating web applications

- This time we want the bot to click the button.
- To do this, set the following properties for the Recorder: Capture action on line 11:
- Action: Click
- The properties should look similar to the following screenshot:



Automating web applications

- Click on Save.
- The development interface for this section should look like this:

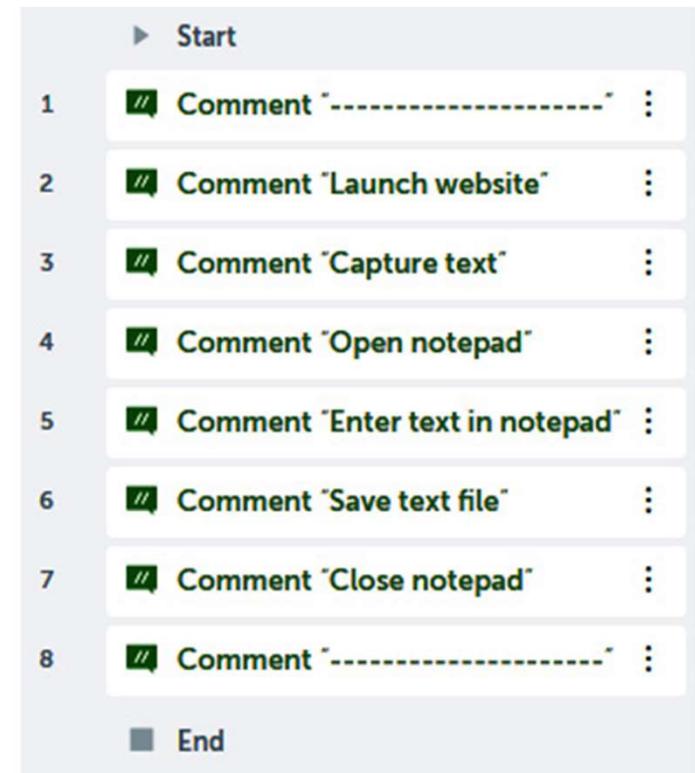


The screenshot shows a development interface with a list of steps numbered 10 through 13. Each step is represented by a card with an icon, a description, and a three-dot menu icon on the right.

- 10 Comment "Click on Send"
- 11 Recorder: Capture Click on button in window "Contact - Google Chrome"
- 12 Message box "Bot has sent an email successfully"
- 13 Comment "-----"

Automating desktop applications

- Add a new Comment action as "-----" on line 8 and click on Save.
- Your bot should look like this:



Automating desktop applications

- Set the following properties for the Browser: Launch website action on line 3:URL: <http://fenago.net/>
- Browser: Google Browser
- The Browser: Launch website action properties should look like this:

Browser: Launch website

This command can be used to open website in browser

URL

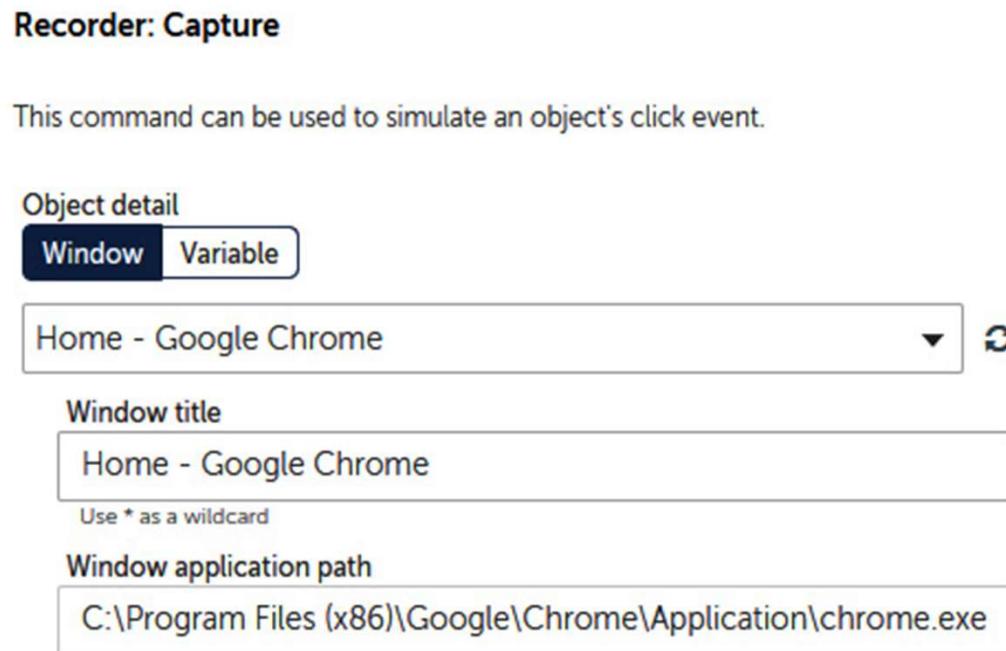
e.g. <http://>...

Browser

Google Chrome ▾

Automating desktop applications

- Refresh the windows drop-down list and select Home - Google Chrome.
- The action properties should look like this



Automating desktop applications



The Bots Are Coming...



Are you ready for the Digital Workforce?

Robotic Processing Automation (RPA) is currently one of the fastest growing technologies around the world. Automation of repetitive, tedious and boring tasks can help your business get most out of your workforce.

What is Robotic Process Automation?

Commonly known as RPA it enables organisations and businesses to 'teach' computers to execute repetitive and time consuming tasks. Being a software based tool, it is easily able to connect and integrate with most desktop applications as well as integration with back-end systems via API's.

Automating desktop applications

As we want to extract the text, we need to identify which property contains this. If we look through the object properties, we can identify the correct property. Once identified, make a note of it. When I captured this, the text was in the **HTML InnerText** property, as you can see in the following screenshot:

HTML InnerText  Commonly known as RPA it enables organisations

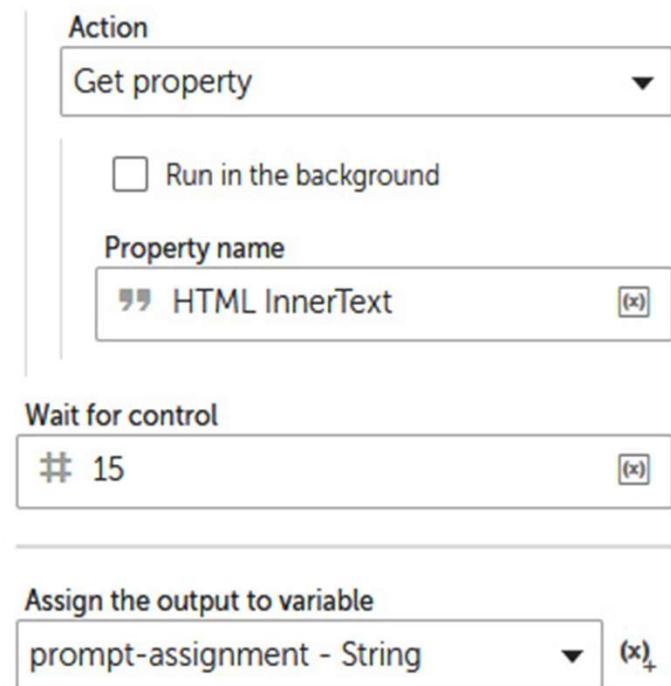
– HTML InnerText property

When you capture the paragraph, you may not necessarily see the text in the same attribute. In some cases, it may be in the **Name** attribute, as shown in the following screenshot:

Name  Commonly known as RPA it enables organisations

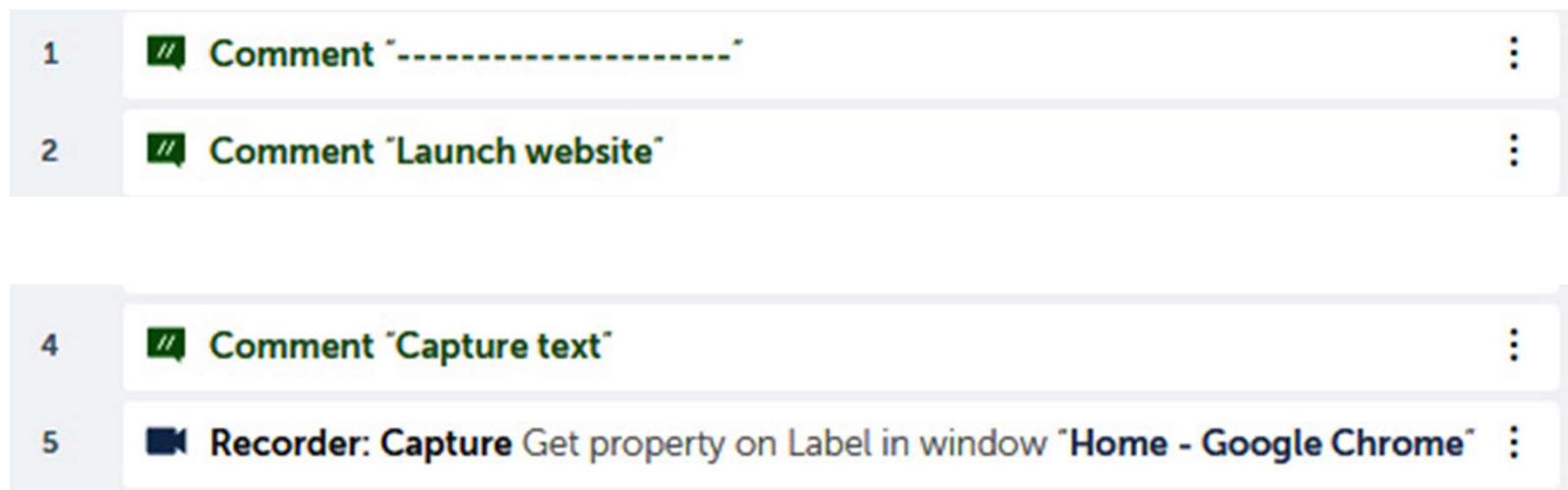
Automating desktop applications

- Set the following properties for the Recorder:
Capture action on line 5:
- Action: Get property
- Property name: HTML InnerText (if your property was Name, then set as Name)
- Assign the output to variable: prompt-assignment - String
- The properties should look similar to the following screenshot:



Automating desktop applications

- Click on Save, Your development interface should look something like this:



Working with Notepad

Application: Open program/file

Opens program/file

Location of the program/file

“ C:\Windows\System32\notepad.exe

e.g.“...\\excel.exe”

Start in path (optional)

“

e.g.“C:\\My Folder”

Parameters (optional)

“

e.g. /r “E:\\My Folder\\test.xls”

Working with Notepad

Recorder: Capture

This command can be used to simulate an object's click event.

Object detail

Window Variable

Untitled - Notepad ▾ 

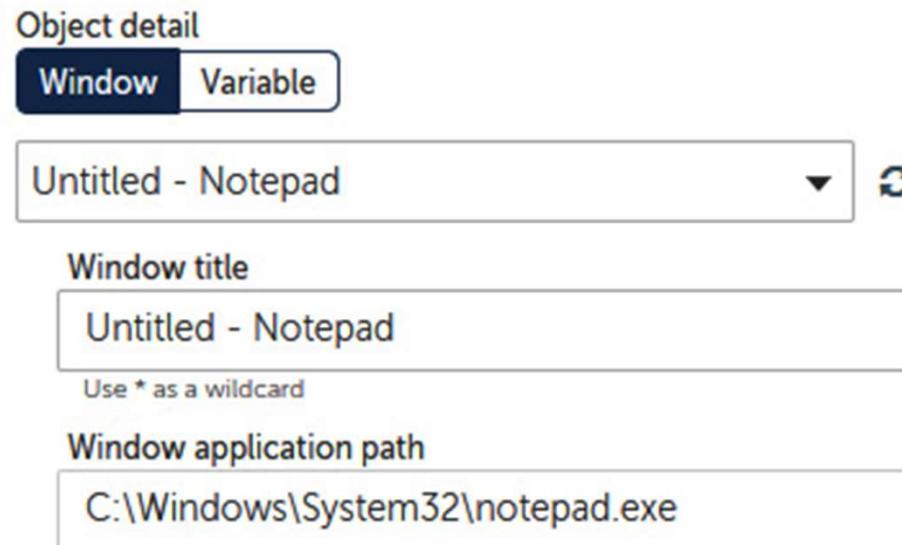
Window title

Untitled - Notepad

Use * as a wildcard

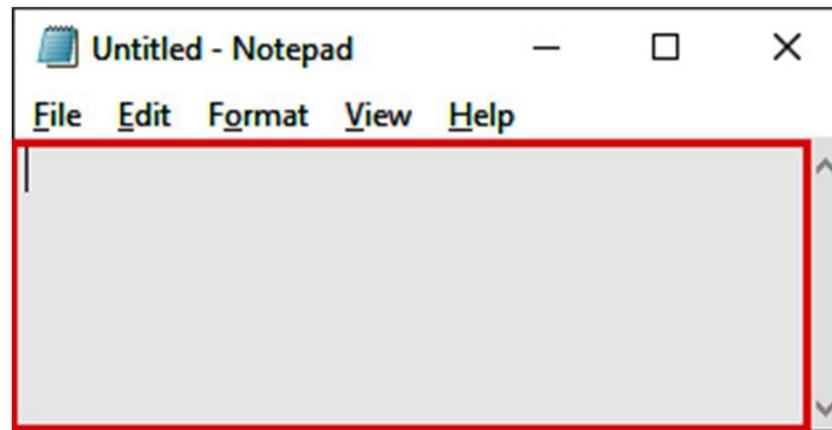
Window application path

C:\Windows\System32\notepad.exe



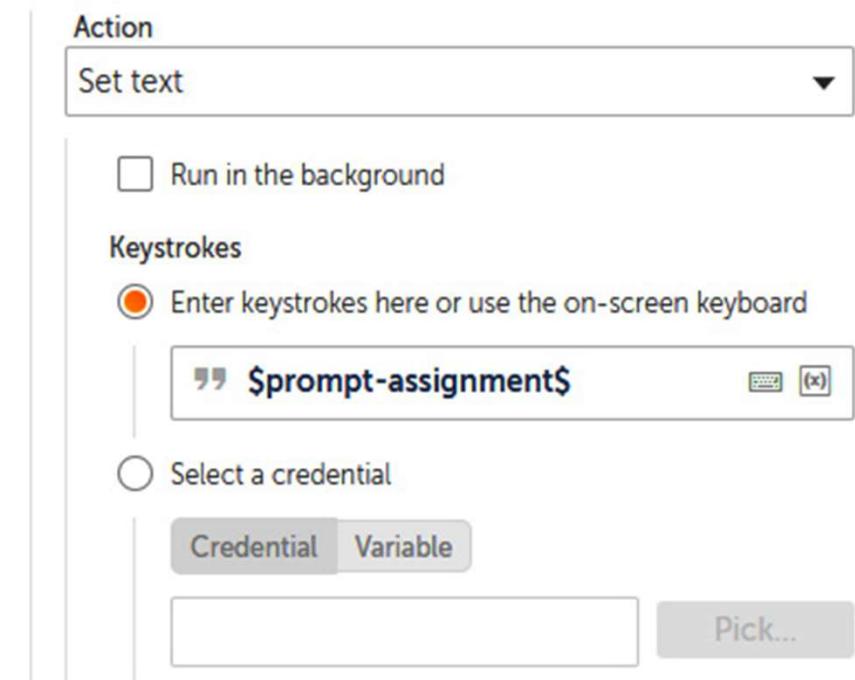
Working with Notepad

- To capture the text entry pane, click on Capture object.
- When Notepad appears, hover the mouse over the text entry pane until it has a red border around it, as follows:



Working with Notepad

- We want to enter the contents of our variable in this pane, Set the following properties for the Recorder: Capture action on line 9:
- Action: Set text
- Keystrokes: \$prompt-assignment\$
- The properties should look similar to the following screenshot:

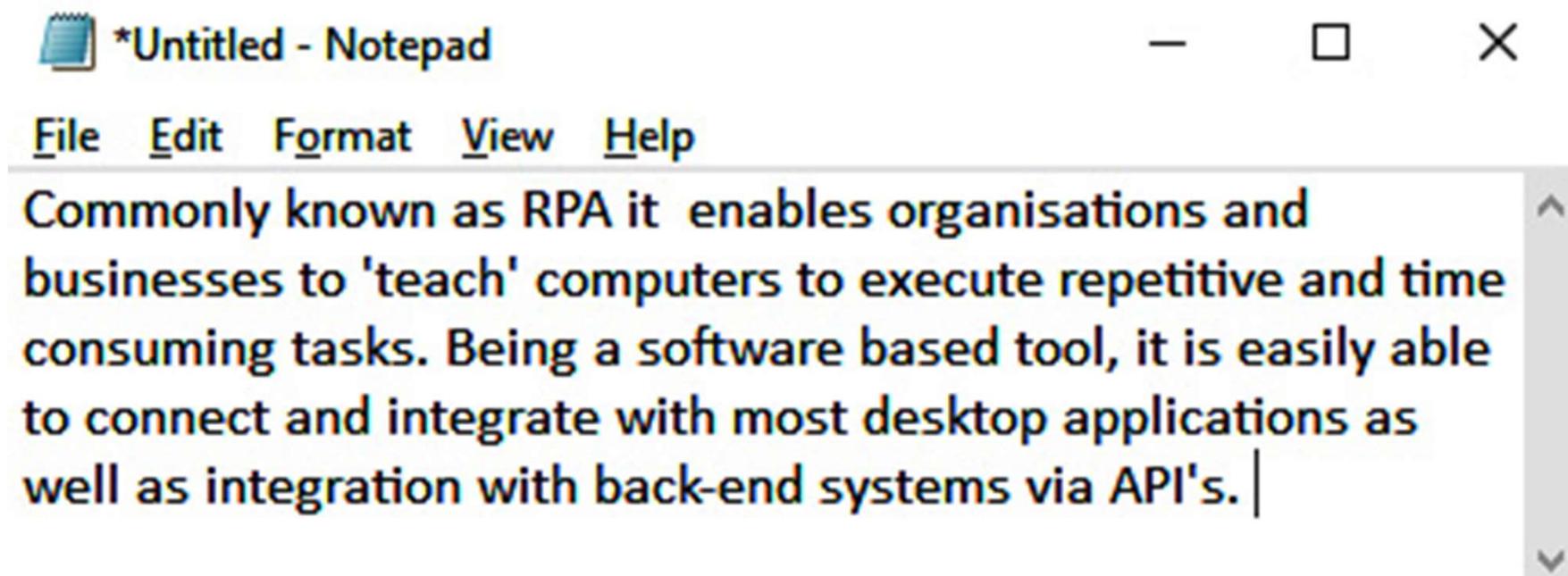


Working with Notepad

- Click on Save.
- Your development interface should look something like this:

6	Comment "Open notepad"	:
7	Application: Open program/file "C:\Windows\System32\notepad.exe"	:
8	Comment "Enter text in notepad"	:
9	Recorder: Capture Set text on textbox in window "Untitled - Notepad"	:

Simulating keystrokes



Simulating keystrokes

Simulate keystrokes

Inserts keystrokes into a selected window

Select window

Window Variable

*Untitled - Notepad



Window title

*Untitled - Notepad

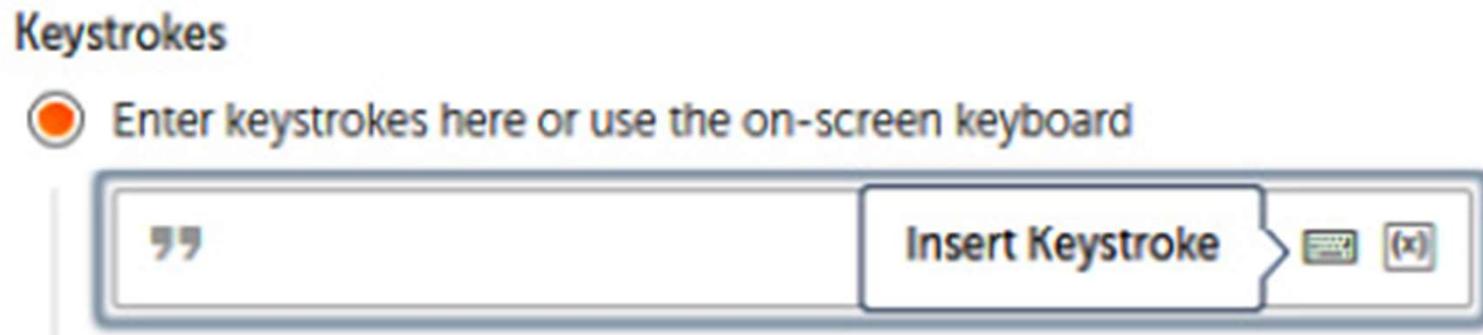
Use * as a wildcard

Window application path

C:\Windows\System32\notepad.exe

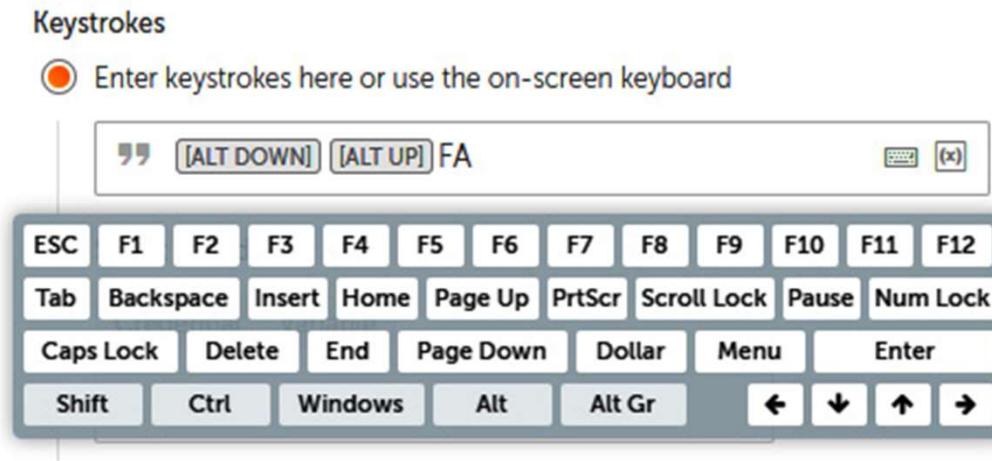
Simulating keystrokes

- To assign the keystroke sequence of Alt + F + A, set the Keystrokes property to Enter keystrokes here or use the on-screen keyboard.
- Click on the keyboard icon, as shown in the following screenshot:

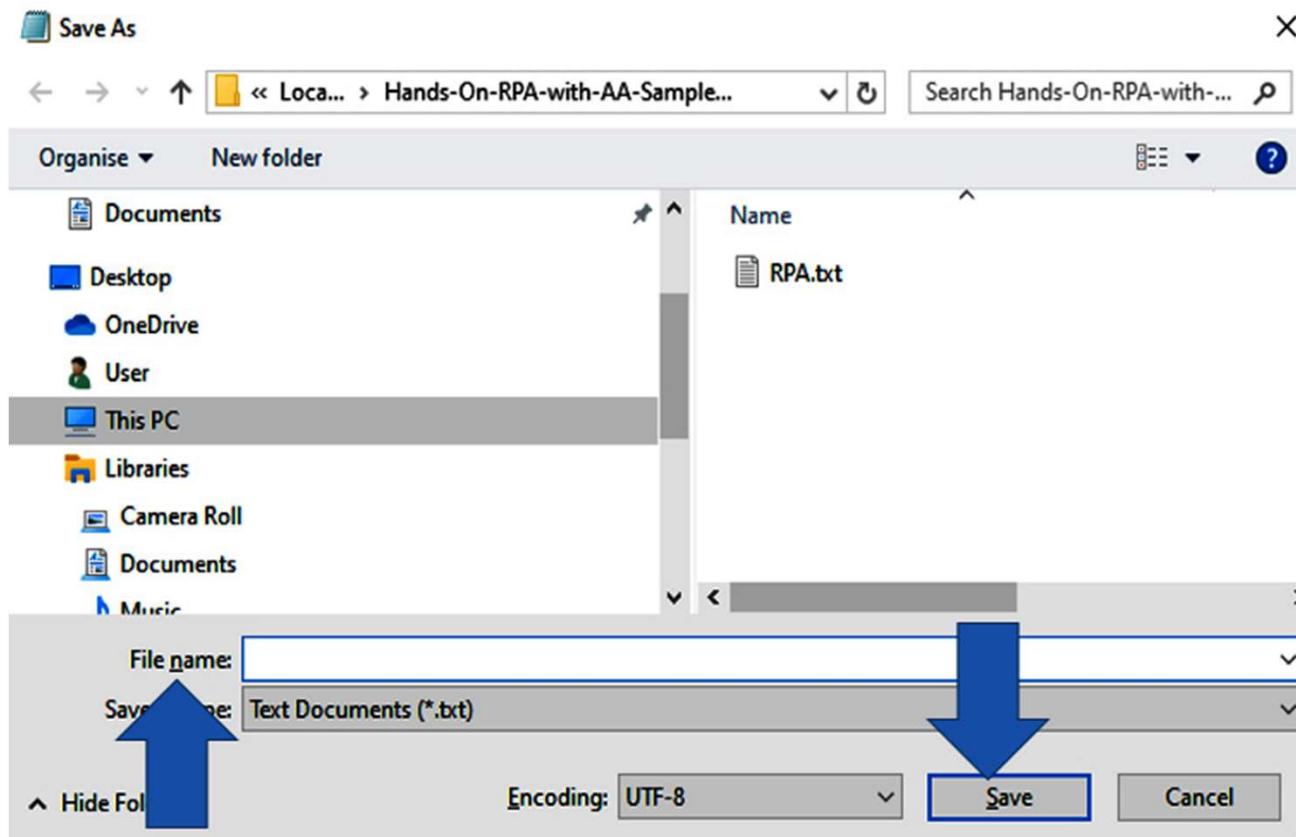


Simulating keystrokes

- The keyboard will appear with all the special keys.
- Any alphanumeric keys can just be typed in the desired case, Select the sequence Alt + F + A.
- This property should look like this:



Simulating keystrokes



Simulating keystrokes

- Refresh the windows drop-down list and select Save As.
- The action properties should look like this:

Simulate keystrokes

Inserts keystrokes into a selected window

Select window

Window Variable

Save As



Window title

Save As

Use * as a wildcard

Window application path

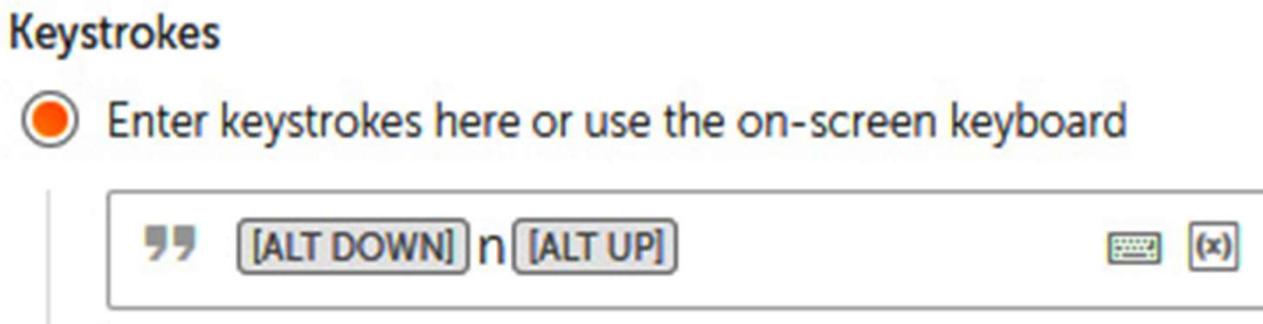
C:\Windows\System32\notepad.exe

Window type

Dialog

Simulating keystrokes

- To assign the keystroke sequence of Alt + n, set the Keystrokes property to Enter keystrokes here or use the on-screen keyboard.
- Once the keyboard appears, select the sequence Alt + n, This property should look like this:



Simulating keystrokes

- Once the keyboard appears, enter C:\Hands-On-RPA-with-AA-Sample-Data\lesson07.txt, as shown in the following screenshot:

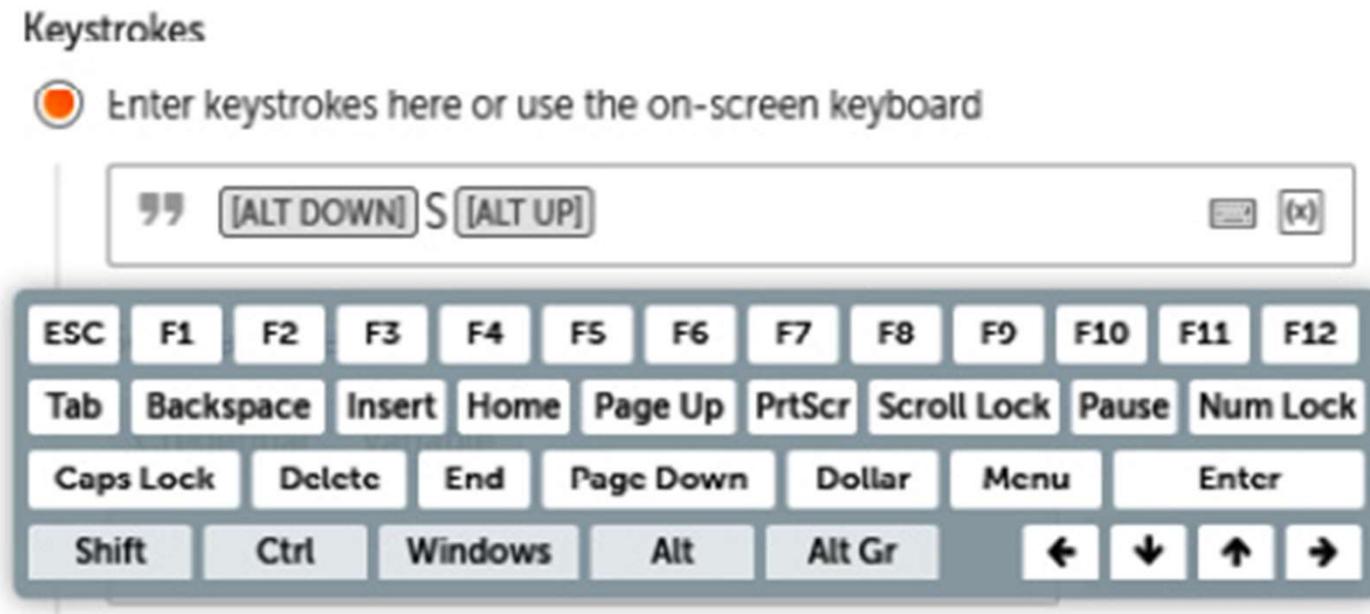
Keystrokes

- Enter keystrokes here or use the on-screen keyboard



Simulating keystrokes

- Once the keyboard appears, select the sequence Alt + S.
- This property should look like this:



Simulating keystrokes

- Click on Save. Your development interface for this section should look something like this:

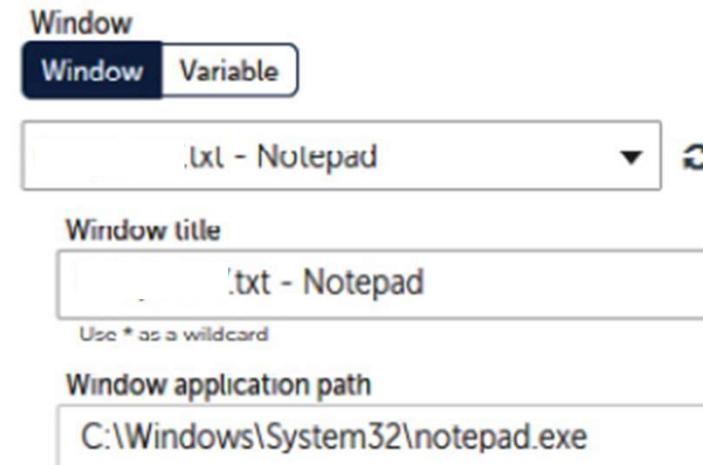
10	Comment "Save text file"	:
11	Simulate keystrokes "[ALT DOWN][ALT UP]FA" on window "*Untitled - Notepad"	:
12	Simulate keystrokes "[ALT DOWN]n[ALT UP]" on window "Save As"	:
13	Simulate keystrokes "C:\Hands-On-RPA-with-AA..." on window "Save As"	:
14	Simulate keystrokes "[ALT DOWN]S[ALT UP]" on window "Save As"	:

Simulating keystrokes

- Refresh the windows drop-down list and select lesson07.txt –
- Notepad. The properties should look like this:

Window: Close

Closes a window



Simulating keystrokes

▶ Start	
1	〃 Comment "-----"
2	〃 Comment "Launch website"
3	↗ Browser: Launch website
4	〃 Comment "Capture text"
5	■ Recorder: Capture Get property on Label in window "Home - Google Chrome"
6	〃 Comment "Open notepad"
7	⌚ Application: Open program/file "C:\Windows\System32\notepad.exe"
8	〃 Comment "Enter text in notepad"
9	■ Recorder: Capture Set text on textbox in window "Untitled - Notepad"
10	〃 Comment "Save text file"
11	■ Simulate keystrokes "[ALT DOWN][ALT UP]FA" on window "*Untitled - Notepad"
12	■ Simulate keystrokes "[ALT DOWN]n[ALT UP]" on window "Save As"
13	■ Simulate keystrokes "C:\Hands-On-RPA-with-AA..." on window "Save As"
14	■ Simulate keystrokes "[ALT DOWN]S[ALT UP]" on window "Save As"
15	〃 Comment "Close notepad"
16	□ Window: Close the "Untitled - Notepad" window
17	〃 Comment "-----"
	▀ End

Summary

- This lesson has covered some key elements of implementing RPA in relation to your daily tasks.
- Understanding how to interact with web and desktop applications is what we humans do.
- All the tasks that we perform while sitting in front of a computer involve interacting with an application of some sort.
- This interaction may involve selecting or clicking on objects as well as entering inputs using the keyboard.
- The walk-throughs in this lesson have given you the practical knowledge to enable you to create bots that navigate through applications, as well as read and enter text.

String Manipulation and List Variables



String Manipulation and List Variables

In this lesson, we will cover the following topics:

- Manipulating strings
- Creating and looping through List variables
- Applying simple conditional logic

String Manipulation and List Variables

 CSV/TXT	 Log To File
 Comment	 Loop
 If	 Message box
 List	 String

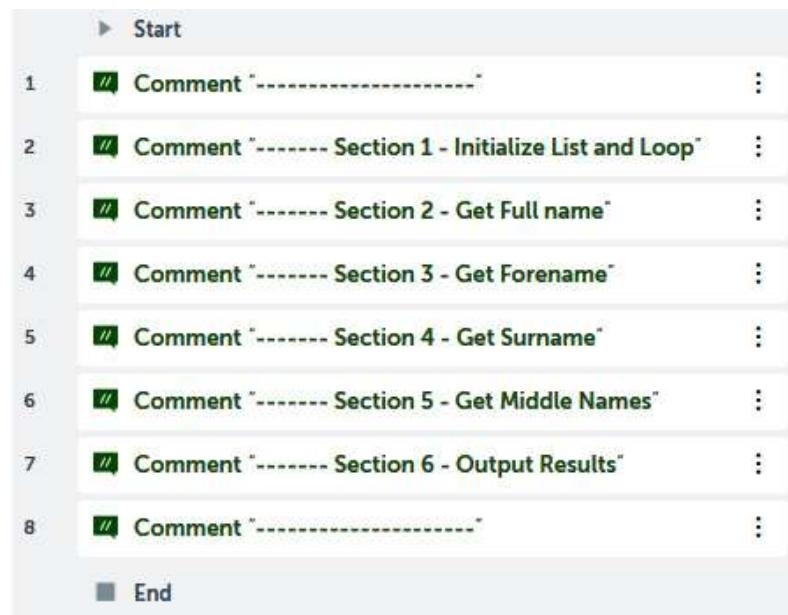
Manipulating strings

While working through this walk-through, we will look at the following string manipulation actions:

- Replace
- Find
- Substring
- Split
- Trim
- Uppercase
- Lowercase

Manipulating strings

- Add a new Comment action of "-----" on line 8 and click on Save.
- Now, your bot should look like this:



Section 1 – Initializing lists and loops

Create variable

Cancel

Create

Name

tblSourceText

Max characters = 50

Description (optional)

Max characters = 255

Use as input

Use as output

Constant (read-only)

Type

Table

Default value (optional)

Edit table (1x1)

Section 1 – Initializing lists and loops

Create variable

Cancel

Create

Name

lstSourceList

Max characters = 50

Description (optional)

Max characters = 255

Use as input

Use as output

Constant (read-only)

Type

List

Subtype

String

Default value (optional)

This list is empty



Section 1 – Initializing lists and loops

- To store each full name from the list, create a String type variable called strFullName.
- The initial variable list should look like this:

Variables +

Search variables

User-defined ^

- list lstSourceList
- String strFullName
- Table tblSourceText

Section 1 – Initializing lists and loops

- The source list of names that we will be using is stored in the lesson08_InputData.txt file in our root folder, which can be found at C:\Hands-On-RPA-with-AA-Sample-Data.
- The content of this file is a single string containing the following data:

"husan lal mahey, priya mahey, sonam mahey, ravinder raj lal mahey,
sunita kumari mahey, manisha Mahey"

Section 1 – Initializing lists and loops

- Set the following properties for the CSV/TXT: Open action on line 3:

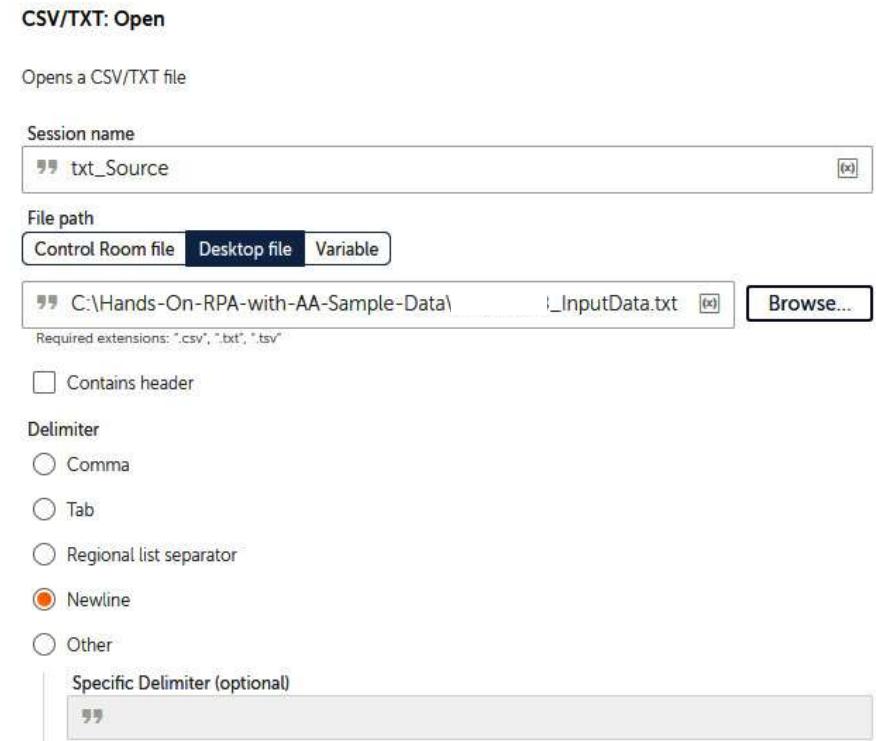
Session name: txt_Source

File path: Desktop file – C:\Hands-On-RPA-with-AA-Sample-Data\lesson08_InputData.txt

Contains header: Unchecked

Delimiter: Newline

The properties should look like this:



Section 1 – Initializing lists and loops

- Set the following properties for the CSV/TXT: Read action on line 4:

Session name: txt_Source

Assign value to the variable:
tblSourceText - Table

The properties should look like this:

CSV/TXT: Read

Reads the entire content of a CSV file

Session name

 (x)

Assign value to the variable

 ▼ (x)

Section 1 – Initializing lists and loops

Set the following properties for the CSV/TXT: Close action on line 5:

- Session name: txt_Source
- The properties should look like this:

CSV/TXT: Close

Closes CSV/TXT session

Session name

 txt_Source <input type="button" value="X"/>
--

Applying the Split action

- When a string needs to be separated into a List variable, we can apply the Split action.
- We have already created our List type variable; that is, `IstSourceList`.
- To split a variable, it is essential we know what character to use for the split, Let's take a look at our source string:

husan lal mahey, priya mahey, sonam mahey, ravinder raj lal mahey,
sunita kumari mahey, manisha mahey

Applying the Split action

- Set the following properties for the String: Split action on line 6:
- Source string: \$tblSourceText[0][0]\$
- Delimiter: ,
- Delimiter is: Not case sensitive
- Split into substrings: All possible
- Assign the output to list variable: lstSourceList – List of Strings
- The properties should look like this:

String: Split

Splits the source string into multiple strings using a delimiter.

Source string
\$tblSourceText[0][0]\$

Delimiter
,

Delimiter is
 Case sensitive
 Not case sensitive

Split into substrings
 All possible
 Only
#

Assign the output to list variable
lstSourceList - List of Strings

Applying the Split action

- Click on Save. Your development interface for this section should look like this:

2	Comment "----- Section 1 - Initialize List and Loop"	:
3	CSV/TXT: Open "C:\Hands-On-RPA-with-AA-Sample-Data\"	:
4	CSV/TXT: Read data and assign to \$tblSourceText\$:
5	CSV/TXT: Close csv/txt "txt_Source"	:
6	String: Split \$tblSourceText[0][0]\$ with delimiter ";" and assign the result to \$lstSourceList\$:

Looping through lists

- Drag the Loop action from the Loop package just below line 6.
- Set the following properties for the Loop action on line 7:
 - Loop Type: Iterator
 - Iterator: For each item in the list
 - list: lstSourceList - List
 - For: All items in the list
 - Assign the current value to variable: strFullName - String
 - The properties should look like this:

Loop

Repeats the actions in a loop until a break

Loop Type

Iterator

Iterator

For each item in the list

iterate list

list

lstSourceList - List

(x)

For

All items in the list

Range

From index (optional)

#

To index (optional)

#

Assign the current value to variable

strFullName - String

(x)

Looping through lists

Message box

Displays a message box

Enter the message box window title

String Manipulation



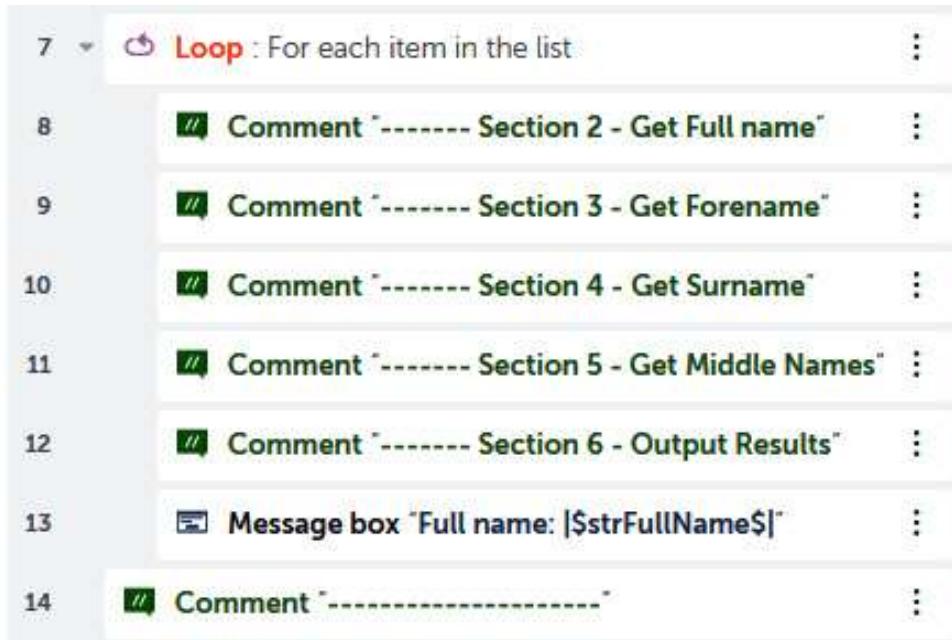
Enter the message to display

Full name: |\$strFullName\$|



Looping through lists

- Click on Save.
- Your development window for this section should look like this:



The screenshot shows a software development environment with a code editor. The code is a sequence of numbered steps (7 to 14) representing a loop structure. Step 7 is a 'Loop' action, and steps 8 through 13 are 'Comment' actions describing different sections of the loop. Step 14 is another 'Comment' action at the end.

7	Loop : For each item in the list
8	Comment ----- Section 2 - Get Full name
9	Comment ----- Section 3 - Get Forename
10	Comment ----- Section 4 - Get Surname
11	Comment ----- Section 5 - Get Middle Names
12	Comment ----- Section 6 - Output Results
13	Message box 'Full name: \$strFullName\$'
14	Comment -----

Section 2 – Getting full names

- Set the following properties for the String: Trim action on line 9:
- Source String: \$strFullName\$
- Trim from the beginning: Checked
- Trim from the end: Checked
- Assign the output to variable: strFullName - String
- The properties should look like this:

String: Trim

Trims blanks and whitespaces from a given string.

Source string

“ \$strFullName\$ ” (x)

Trim from the beginning

Trim from the end

Assign the output to variable

strFullName - String ▼ (x)

Applying uppercase to a string

- Set the following properties for the String: Uppercase action on line 10:Source
String: \$strFullName\$
- Assign the output to variable: strFullName - String
- The properties should look like this:

String: Uppercase

Converts the source string to upper case.

Source string

” \$strFullName\$ (x)

Assign the output to variable

strFullName - String ▼ (x)

Applying uppercase to a string

- Click on Save.
- Your development interface for this section should look like this:

```
8  // Comment "----- Section 2 - Get Full name"
9  ?? String: Trim $strFullName$ and assign the result to $strFullName$
10 ?? String: Uppercase Convert $strFullName$ to upper case and assign the result to $strFullName$
```

Section 3 – Getting forenames

- It would be a good idea to add these variables to our message box to help us with progress testing.
- Edit the message display property of the message box on line 15 to the following:

Message box

Displays a message box

Enter the message box window title

” String Manipulation

Enter the message to display

” Full name: |\$strFullName\$|
Initial: |\$strInitial\$|
Forename:|\$strForename\$|

Section 3 – Getting forenames

String: Extract text

Extracts a sub-string between two given strings specified by 'Before' and 'After'.

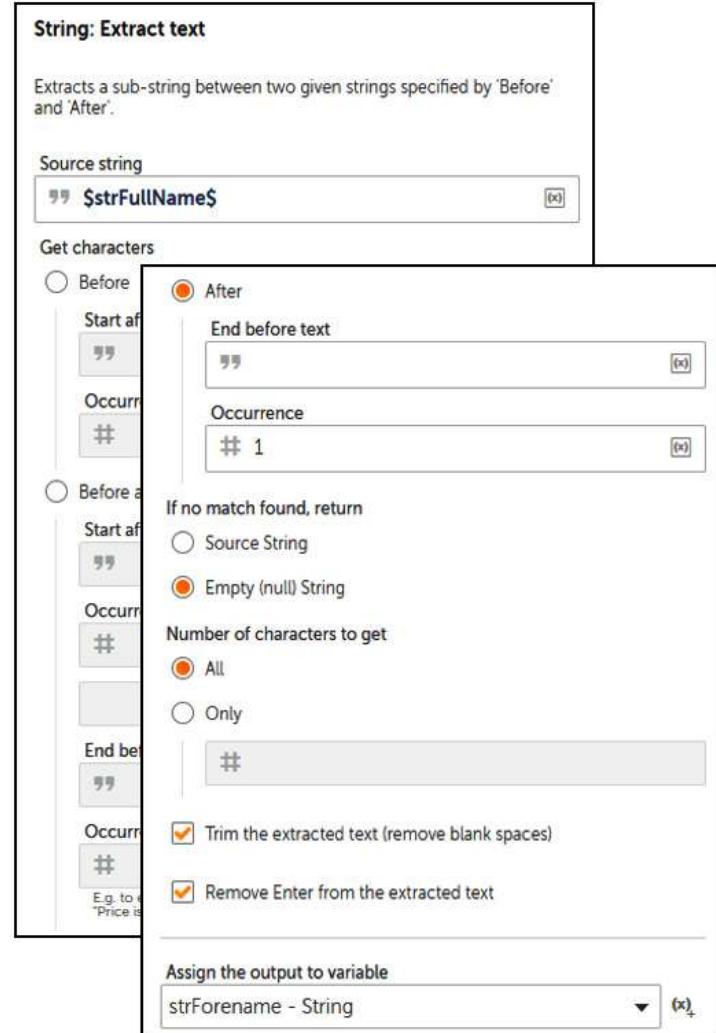
Source string
\$strFullName\$

Get characters

After
Start after
Occurrences
End before text
Occurrence
If no match found, return
Source String
Empty (null) String

Before a
Start after
Occurrences
End before
Occurrences
E.g. to extract "Price is \$100" from "The price is \$100."
Number of characters to get
All
Only
Trim the extracted text (remove blank spaces)
Remove Enter from the extracted text

Assign the output to variable
strForename - String



Section 3 – Getting forenames

String: Substring

Extracts a sub-string from a given string.

Source string

“ \$strForename\$

(x)

Start from

1

(x)

Length (optional)

1

(x)

Assign the output to variable

strInitial - String

▼

(x)

Section 3 – Getting forenames

String: Substring

Extracts a sub-string from a given string.

Source string

” \$strForename\$

(x)

Start from

2

(x)

Length (optional)

#

(x)

Assign the output to variable

strForename - String

▼

(x)

Section 3 – Getting forenames

String: Lowercase

Converts the source string to lower case.

Source string

“ \$strForename\$ ”

(x)

Assign the output to variable

strForename - String

▼

(x)

Section 3 – Getting forenames

String: Assign

Assign or Concatenate the given strings

Select the source string variable(s)/ value (optional)

“ \$strInitial\$\$strForename\$



Select the destination string variable

strForename - String



(x)₄

Section 3 – Getting forenames

- | | | |
|----|---|---|
| 11 | Comment "----- Section 3 - Get Forename" | : |
| 12 | ?? String: Extract text Source string \$strFullName\$: Extract sub-string after " | : |
| 13 | ?? String: Substring : Extract substring from the \$strForename\$ string | : |
| 14 | ?? String: Substring : Extract substring from the \$strForename\$ string | : |
| 15 | ?? String: Lowercase : Convert the \$strForename\$ to lowercase | : |
| 16 | ?? String: Assign "\$strInitial\$strForenam..." to \$strForename\$ | : |

Section 4 – Getting surnames

- Set the following properties for the String: Find action on line 18:
- Source string variable: \$strFullName\$
- Find string: (\w+)\$\$
- When finding: Do not match case
- The "find string" is: A regular expression
- Start from: 1
- Assign the output to variable: numLoc - Number
- The properties should look like this:

String: Find

Locates a given string within the source string.

Source string

\$strFullName\$

(x)

Find string

(\w+)\$\$

(x)

When finding

Match case

Do not match case

The "find string" is

A regular expression

Not a regular expression

Start from (optional)

1

(x)

Assign the output to variable

numLoc - Number

(x)

Section 4 – Getting surnames

- Set the following properties for the String: Substring action on line 19:Source String: \$strFullName\$
- Start from: \$numLoc\$
- Assign the output to variable: strSurname - String
- The properties should look like this:

String: Substring

Extracts a sub-string from a given string.

Source string

” \$strFullName\$

Start from

\$numLoc\$

Length (optional)

#

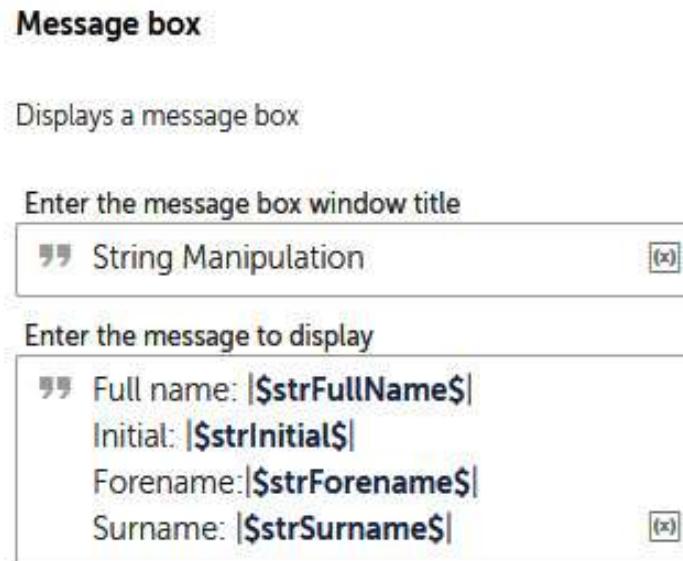
Assign the output to variable

strSurname - String

(x) +

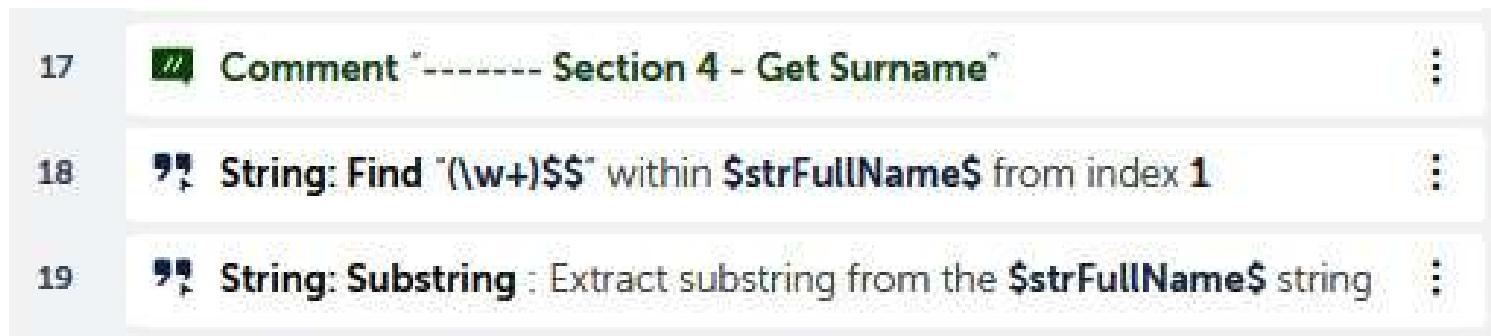
Section 4 – Getting surnames

- Now that the surname has been extracted, add it to the final message box on line 22.
- Edit the message display property of the message box so that it includes the following:



Section 4 – Getting surnames

- Great progress! That's Section 4 – Getting surnames complete. Run the bot to test it.
- We now have the forename and the surname in the correct format. The development interface for this section should look like this:



```
17  Comment ----- Section 4 - Get Surname
18  String: Find '(\w+)$' within $strFullName$ from index 1
19  String: Substring : Extract substring from the $strFullName$ string
```

Section 4 – Getting surnames

String: Replace

Replaces specified part of a 'Source string' with a 'Replacement string'

Source string

\$strFullName\$

(x)

Find string

\$strForename\$

(x)

When finding

- Match case
- Do not match case

The "find string" is

- A regular expression
- Not a regular expression

Start from (optional)

1

(x)

Count (optional)

-1

(x)

Replace with (optional)

##

(x)

Assign the output to variable

strMiddleNames - String

(x)

▼

Section 4 – Getting surnames

- Set the following properties for the String Replace action on line 22:

Source String: \$strMiddleNames\$

Find string: \$strSurname\$

When finding: Do not match case

The "find string" is: Not a regular expression

Start from: 1

Replace with: (enter space)

Assign the output to variable: strMiddleNames

- String

The properties should look like this:

String: Replace

Replaces specified part of a 'Source string' with a 'Replacement string'

Source string

\$strMiddleNames\$

Find string

\$strSurname\$

When finding

Match case

Do not match case

The "find string" is

A regular expression

Not a regular expression

Start from (optional)

1

Count (optional)

-1

Replace with (optional)

" "

Assign the output to variable

strMiddleNames - String

Section 4 – Getting surnames

- Set the following properties for the String: Trim action on line 23:
- Source String: \$strMiddleNames\$
- Trim from the beginning: Checked
- Trim from the end: Checked
- Assign the output to variable: strMiddleNames - String
- The properties should look like this:

String: Trim

Trims blanks and whitespaces from a given string.

Source string

“ \$strMiddleNames\$

Trim from the beginning

Trim from the end

Assign the output to variable

strMiddleNames - String ▼ (x)

Section 4 – Getting surnames

- The middle names have now been extracted to a variable.
- Let's add this variable to the final message box on line 25.
- Edit the message display property of the Message box so that it includes this:

Message box

Displays a message box

Enter the message box window title

” String Manipulation

Enter the message to display

” Full name: |\$strFullName\$|
Initial: |\$strInitial\$|
Forename:|\$strForename\$|
Middle Names: |\$strMiddleNames\$|
Surname: |\$strSurname\$|

Assigning a null value to a string

- Set the following properties for the String: Assign action on line 24:
- Select the source string variable: (leave empty)
- Select the destination string variable: strInitial - String
- The properties should look like this:

String: Assign

Assign or Concatenate the given strings

Select the source string variable(s)/ value (optional)

 (x)

Select the destination string variable

 (x)

Applying a simple logical condition

- Set the following properties for the If action on line 25:
- Condition: String condition
- Source value: \$strMiddleNames\$
- Operator: Not equal to
- Target value: (leave blank)
- The properties should look like this:

If

Runs a sequence of actions if a condition is true

Condition

String condition

Checks the string variable condition.

Source value (optional)
\$strMiddleNames\$

Operator
Not equal to(≠)

Target value (optional)
''

Match case

Add condition

Implementing a nested loop

- Create two String type variables called strCurrentMiddleName and strCurrentMiddleNameInitial.
- Your variables list should now look similar to this:

User-defined

```
■ lstMiddleNames
■ lstSourceList
# numLoc
„ strCurrentMiddleName
„ strCurrentMiddleNameInitial
„ strForename
„ strFullName
„ strInitial
„ strMiddleNames
„ strSurname
■■■tblSourceText
```

Implementing a nested loop

- Set the following properties for the String: Split action on line 26:
- Source string: \$strMiddleNames\$
- Delimiter: (enter a space)
- Delimiter is: Not case sensitive
- Split into substrings: All possible
- Assign the output to list variable: lstMiddleNames - List of Strings
- The properties should look like this:

String: Split

Splits the source string into multiple strings using a delimiter.

Source string

\$strMiddleNames\$

(x)

Delimiter

" "

(x)

Delimiter is

- Case sensitive
 Not case sensitive

Split into substrings

- All possible
 Only

#

Assign the output to list variable

lstMiddleNames - List of Strings

▼

(x)

Implementing a nested loop

- Set the following properties for the Loop action on line 27:
- Loop Type: Iterator
- Iterator: For each item in the list
- List: lstMiddleNames - List
- For: All items in the list
- Assign the current value to variable: strCurrentMiddleName - String
- The properties should look like this:

Loop

Repeats the actions in a loop until a break

Loop Type

Iterator

Iterator

For each item in the list

iterate list

list

lstMiddleNames - List

(x)

For

All items in the list

Range

From index (optional)

#

To index (optional)

#

Assign the current value to variable

strCurrentMiddleName - String

(x)

(x)

Implementing a nested loop

- Set the following properties for the String: Substring action on line 28:
- Source String:
\$strCurrentMiddleName\$
- Start from: 1
- Length: 1
- Assign the output to variable:
strCurrentMiddleNameInitial - String
- The properties should look like this:

String: Substring

Extracts a sub-string from a given string.

Source string

\$strCurrentMiddleName\$ (x)

Start from

1 (x)

Length (optional)

1 (x)

Assign the output to variable

strCurrentMiddleNameInitial - String (x)

Implementing a nested loop

- Set the following properties for the String: Assign action on line 29:
- Select the source string variable(s)/ value (optional):
\$strInitial\$\$strCurrentMiddleName
Initial\$
- Select the destination string variable: strInitial - String
- The properties should look like this:

String: Assign

Assign or Concatenate the given strings

Select the source string variable(s)/ value (optional)

\$strInitial\$\$strCurrentMiddleNameInitial\$

Select the destination string variable

strInitial - String

Implementing a nested loop

```
20  // Comment ----- Section 5 - Get Middle Names
21  " String: Replace $strForename$ with "" in $strFullName$ and assign the result to $strMiddleNames$ ...
22  " String: Replace $strSurname$ with "" in $strMiddleNames$ and assign the result to $strMiddleNames$ ...
23  " String: Trim $strMiddleNames$ and assign the result to $strMiddleNames$ ...
24  " String: Assign "" to $strInitial$
25  ◇ If string $strMiddleNames$ Not equal to(≠) ""
26  " String: Split $strMiddleNames$ with delimiter "" and assign the result to $lstMiddleNames$ ...
27  - ◇ Loop : For each item in the list
28      " String: Substring : Extract substring from the $strCurrentMiddleName$ string ...
29      " String: Assign "$strInitial$$strCurrent..." to $strInitial$ ...
```

Section 6 – Outputting the results

- A CSV file with headers will need to be created just before the primary loop as we only want this to be created once.
- The record should be added while we're within the primary loop so that it's created once per individual.
- Once we've done this, we will need to identify the sequence of our name items as required. The output should be in the following format:

Surname in uppercase, Forename in Proper case, Middle name initials in uppercase

Section 6 – Outputting the results

Log to file

Logs any text into a file

File path

(x)Browse...

Enter text to log

(x)

Append timestamp

When logging

Append to existing log file

Overwrite existing log file

Encoding

▼

Section 6 – Outputting the results

Log to file

Logs any text into a file

File path

C:\Hands-On-RPA-with-AA-Sample-Data\

(x)

Browse...

Enter text to log

\$strSurname\$, \$strForename\$ \$strInitial\$

(x)

Append timestamp

When logging

Append to existing log file

Overwrite existing log file

Encoding

ANSI

▼

Section 6 – Outputting the results

- Finally, delete the Message box action on line 33 and click on Save.
- Congratulations – you have completed your bot! The development interface for this final section should look like this:

2	#[Comment "----- Section 1 - Initialize List and Loop"	:
3	Log to file "Surname, Forename/Initi..." to "C:\Hands-On-RPA-with-AA-Sample-Data\Output.csv"	:
31	#[Comment "----- Section 6 - Output Results"	:
32	Log to file "\$strSurname\$, \$strForen..." to "C:\Hands-On-RPA-with-AA-Sample-Data\Output.csv"	:

Section 6 – Outputting the results

- Now, it's time to test the bot. When it's executed, you will get the lesson08_Output.csv file at C:\Hands-On-RPA-with-AA-Sample-Data\lesson08_Output.csv.
- The contents should look like this:

	A	B
1	Surname	Forename/Initials
2	MAHEY	Husan L
3	MAHEY	Priya
4	MAHEY	Sonam
5	MAHEY	Ravinder RL
6	MAHEY	Sunita K
7	MAHEY	Manisha

Summary

- To recap, we have covered a number of useful actions, including extracting a specific substring from a string via locations, extracting a substring from a string via specific text, finding a specific substring within a string, replacing parts of a string
- Converting strings into upper or lowercase, trimming leading/trailing spaces from strings, using regular expressions to find string patterns, splitting strings with a specific delimiter, concatenating strings, creating list variables, looping through list variables, and finally, understanding simple logical conditions.

Working with Conditional Logic, Loops, and the Filesystem



Working with Conditional Logic, Loops, and the Filesystem

In this lesson, we will cover the following topics:

- Applying different types of loops
- Applying logical conditions
- Working with the filesystem

Working with Conditional Logic, Loops, and the Filesystem

 Comment	 List
 CSV/TXT	 Log To File
 File	 Loop
 Folder	 String

Applying different types of loops

- Add a new Comment action called "-----" on line 9 and click on Save, Our initial development interface should look like this:

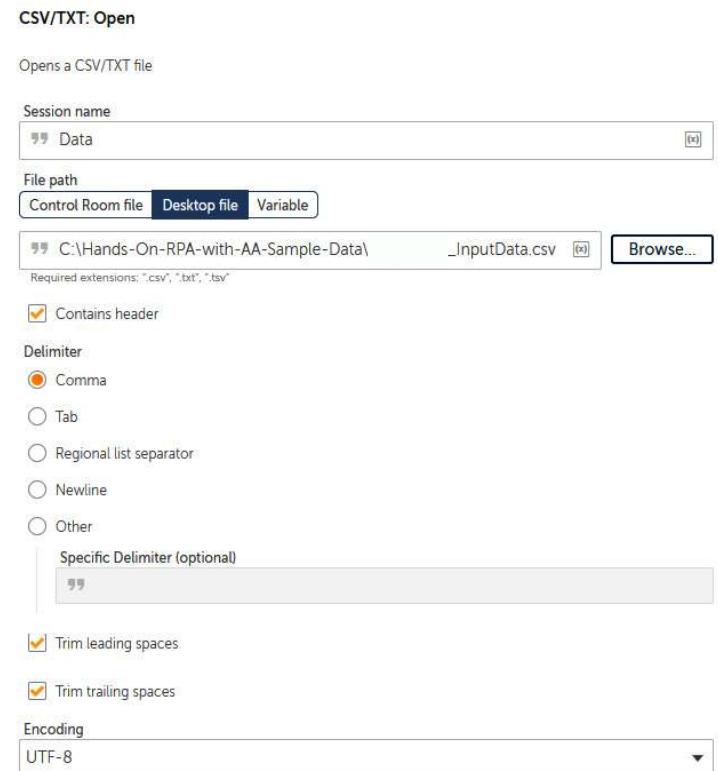
1	#[Comment "-----"	:
2	#[Comment "----- Section 1 - Open the Source file"	:
3	#[Comment "----- Section 2 - Loop through each row"	:
4	#[Comment "----- Section 3 - Get Surname initial & identify group"	:
5	#[Comment "----- Section 4 - Check if sub folder exists"	:
6	#[Comment "----- Section 4a - Create sub folder & output file if it doesn't exist"	:
7	#[Comment "----- Section 5 - Update output file"	:
8	#[Comment "----- Section 6 - Close the Source file"	:
9	#[Comment "-----"	:

Section 1 – Opening the source file

Set the following properties for the CSV/TXT:

Open action on line 3:

- Session name: Data
- File path: Desktop file – C:\Hands-On-RPA-with-AA-Sample-Data\lesson09_InputData.csv
- Contains header: Checked
- Delimiter: Comma
- Trim leading spaces: Checked
- Trim trailing spaces: Checked
- The properties window should look like this:



Section 1 – Opening the source file

- Click on Save.
- The development window for this section should look like this:



Section 2 – Looping through each row

- Create a new variable named recSource as a Record type.
- The Create variable dialog should look like this:

Create variable

Name
recSource
Max characters = 50

Description (optional)

Max characters = 255

Constant (read-only)

Use as input Use as output

Type
Record

Default value (optional)
Record

+

Section 2 – Looping through each row

Set the following properties for the Loop action on line 5:

- Loop Type: Iterator
- Iterator: For each row in CSV/TXT
- Session name: Data
- Assign the current row to this variable: recSource - Record
- The properties window should look like this:

Loop

Repeats the actions in a loop until a break

Loop Type

Iterator

Iterator

For each row in CSV/TXT

Iterator for each row in CSV/TXT

Session name

” Data

Assign the current row to this variable

recSource - Record

Section 2 – Looping through each row

- Click on Save.
- The completed development window should look like this:

1	Comment "-----"	:
2	Comment "----- Section 1 - Open the Source file"	:
3	CSV/TXT: Open "C:\Hands-On-RPA-with-AA-Sample-Data_\InputData.csv"	:
4	Comment "----- Section 2 - Loop through each row"	:
5	Loop for each row in csv/txt	:
6	Comment "----- Section 3 - Get Surname initial & identify group"	:
7	Comment "----- Section 4 - Check if sub folder exists"	:
8	Comment "----- Section 4a - Create sub folder & output file if it doesn't exist"	:
9	Comment "----- Section 5 - Update output file"	:
10	Comment "----- Section 6 - Close the Source file"	:
11	Comment "-----"	:

Section 3 – Getting the surname initial and identifying the group

- Set the following properties for the String:
Substring action on line 7:Source String
name: \$recSource[1]\$
- Start from: 1
- Length: 1
- Assign the output to variable: strRefInitial - String
- The properties window should look like this:

String: Substring

Extracts a sub-string from a given string.

Source string

\$recSource[1]\$

(x)

Start from

1

(x)

Length (optional)

1

(x)

Assign the output to variable

strRefInitial - String

▼

(x)

Applying logical conditions

A2019DEMPACKAGE
Window Exists demo
False condition demo
A2019DEMPACKAGE
Window Exists demo
False condition demo
APPLICATION
Application is not running
Application is running
BOOLEAN
Boolean condition
DATA TABLE
Data table is empty
Number of columns
Number of rows
DATETIME
Date Condition
DICTIONARY
Check key
Check for a single value

VBSCRIPT
Script is successful
Script is unsuccessful
WINDOW
Window exists
Window does not exist

FILE
File date
File exists
File does not exist
File size
FOLDER
Folder does not exist
Folder exists
IMAGE RECOGNITION
Image file is NOT found in image file
Image file is NOT found in window
Window is NOT found in image file
Window is NOT found in window
Image file is found in image file
Image file is found in window
Window is found in image file
Window is found in window

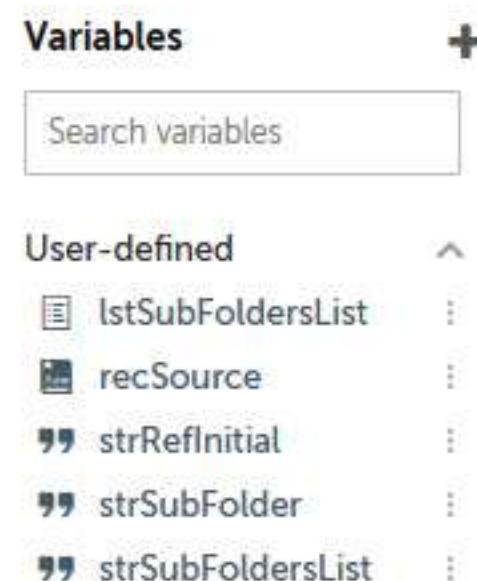
STRING
String condition
TASK BOT
Task successful
Task unsuccessful
UTILS
Not Empty String
Empty String
Compare Record Value by Index
List contains Record value

JAVASCRIPT
Script is successful
Script is unsuccessful
LEGACY AUTOMATION
Web control exists
Web control does not exist
Window control is active
Window control does not exist
Window control exists
Window control is not active
Script is unsuccessful
Script is successful
Child window does not exist
Child window exists
LIST
List variable
NUMBER
Number condition

PING
Ping is successful
Ping is unsuccessful
PROCESS DISCOVERY
Object
RECORDER
Object
SERVICE
Service is not running
Service is running

Applying different types of loops

- Create a String type variable named strSubFolder for storing the allocated subfolder.
- The variables list should look like this:



Applying different types of loops

Set the following properties for the String: Assign action on line 8:

- Select the source string variable/value:
ABCD,EFGH,IJKL,MNOP,QRST,UVW
X,YZ
- Select the destination string variable: strSubFoldersList - String
- The properties window should look like this:

String: Assign

Assign or Concatenate the given strings

Select the source string variable(s)/ value (optional)

“ ABCD,EFGH,IJKL,MNOP,QRST,UVWX,YZ

Select the destination string variable

strSubFoldersList - String

Applying different types of loops

Set the following properties for the String: Assign action on line 9:

- Source string: \$strSubFoldersList\$
- Delimiter: ,
- Assign the output to list variable: lstSubFoldersList - List of Strings
- The properties window should look like this:

String: Split

Splits the source string into multiple strings using a delimiter.

Source string
\$strSubFoldersList\$

Delimiter
,

Delimiter is
 Case sensitive
 Not case sensitive

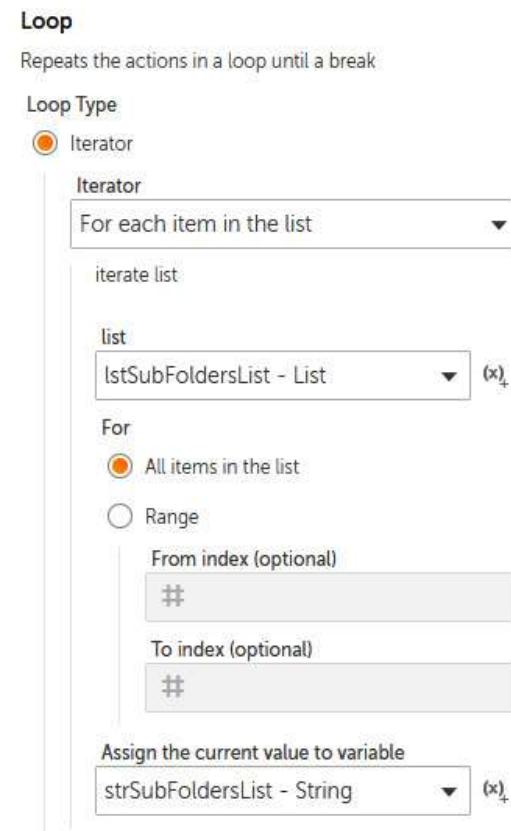
Split into substrings
 All possible
 Only
#

Assign the output to list variable
lstSubFoldersList - List of Strings

Applying different types of loops

Set the following properties for the Loop action on line 10:

- Loop Type: Iterator
- Iterator: For each item in the list
- List: lstSubFoldersList - List
- For: All items in the list
- Assign the current value to variable: strSubFoldersList - String
- The properties window should look like this:



Applying different types of loops

Set the following properties for the If action on line 11:

- Condition: String condition
- Source value: strSubFoldersList
- Operator: Includes
- Target value: \$strReflinitial\$
- The properties window should look like this:

If

Runs a sequence of actions if a condition is true

Condition

String condition

Checks the string variable condition.

Source value (optional)
\$strSubFoldersList\$

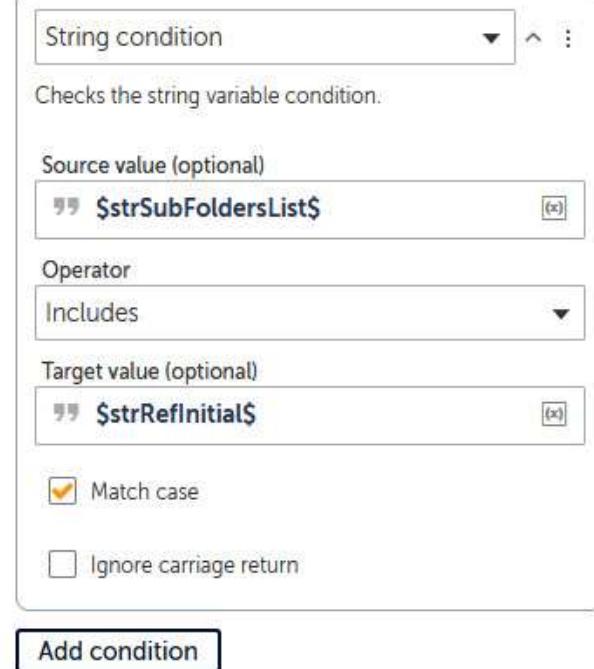
Operator
Includes

Target value (optional)
\$strReflinitial\$

Match case

Ignore carriage return

Add condition



Applying different types of loops

Set the following properties for the String: Assign action on line 12:

- Select the source string variable(s)/value (optional): \$strSubFoldersList\$
- Select the destination string variable: strSubFolder - String
- The properties window should look like this:

String: Assign

Assign or Concatenate the given strings

Select the source string variable(s)/ value (optional)

” \$strSubFoldersList\$ (x)

Select the destination string variable

strSubFolder - String ▼ (x)

Applying different types of loops

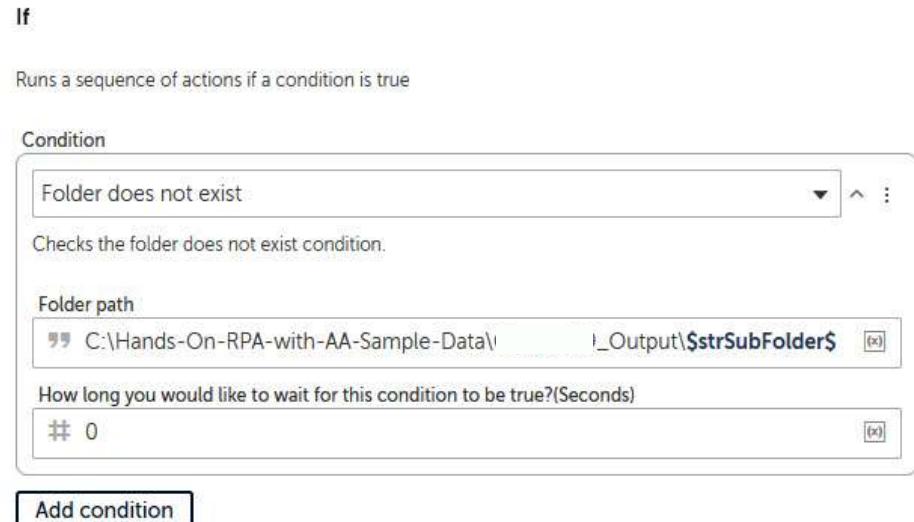
- The bot is at a stage now where it has identified the subfolder group for the working record using the surname initial.
 - This brings Section 3 – Getting the surname initial and identifying the group, to an end.
 - The development interface for this section should look like this:

6	Comment ----- Section 3 - Get Surname initial & identify group	:
7	;; String: Substring : Extract substring from the \$recSource[1]\$ string	:
8	;; String: Assign "ABCD,EFGH,IJKL,MNOP,QRS..." to \$strSubFoldersList\$:
9	;; String: Split \$strSubFoldersList\$ with delimiter ";" and assign the result to \$lstSubFold...	:
10	Loop : For each item in the list	:
11	If string \$strSubFoldersList\$ Includes \$strRefInitial\$:
12	String: Assign \$strSubFoldersList\$ to \$strSubFolder\$:

Section 4 – Checking or creating a subfolder

Set the following properties for the If action on line 14:

- Condition: Folder does not exist
- Folder path: C:\Hands-On-RPA-with-AA-Sample-Data\lesson09_Output\\$strSubFolder\$
- The properties window should look like this:



Section 4 – Checking or creating a subfolder

- The properties window should look like this:

Folder: Create

Creates a folder

Folder

„ C:\Hands-On-RPA-with-AA-Sample-Data\

_Output\\$strSubFolder\$

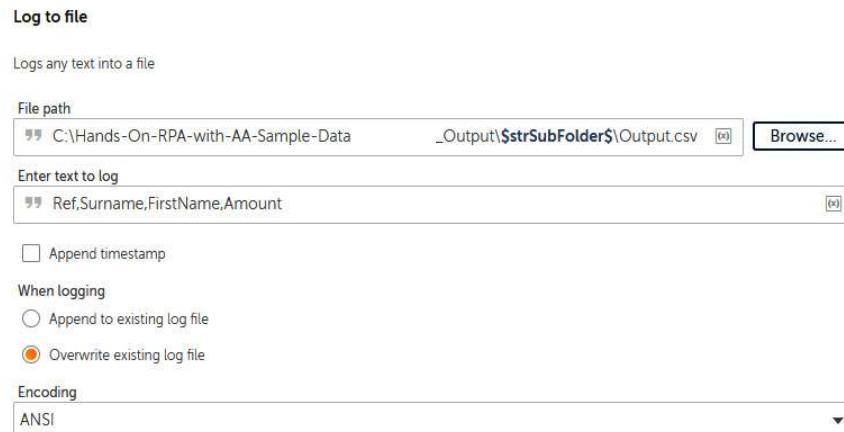
e.g. C:\MyDoc\MyNewFolder

Overwrite an existing folder

Section 4 – Checking or creating a subfolder

Set the following properties for the Log to file action on line 17:

- File path: C:\Hands-On-RPA-with-AA-Sample-Data\lesson09_Output\\$strSubFolder\$\Output.csv
- Enter text to log: Ref,Surname,FirstName,Amount
- When logging: Overwrite existing log file
- The properties window should look like this:



Section 4 – Checking or creating a subfolder

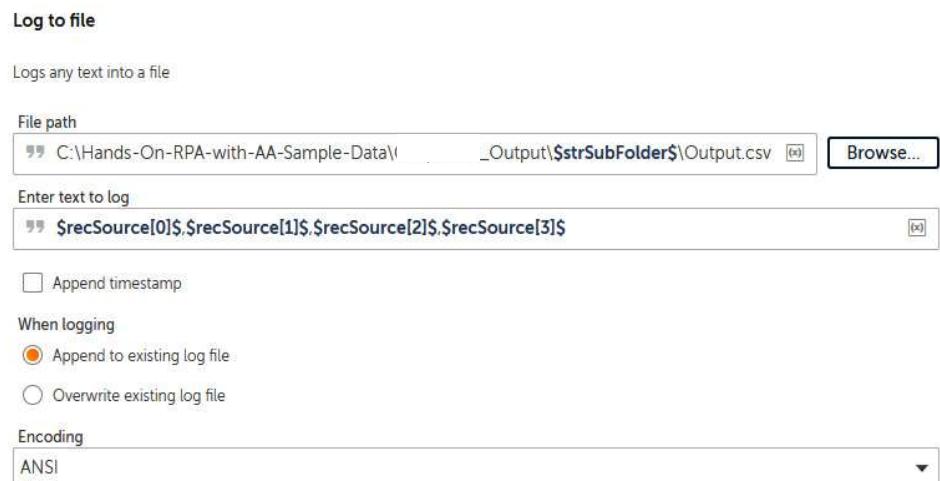
- Checking or creating a subfolder, complete! Our bot will create the necessary subfolder and file if needed.
- The development window for this section should look like this:

```
13  ┌─ Comment "----- Section 4 - Check if sub folder exists"
14  ▀ └─ ◊ If folder does not exist at "C:\Hands-On-RPA-with-AA-Sample-Data\"
15    ┌─ Comment "----- Section 4a - Create sub folder & output file if it doesn't exist"
16    └─ ┌─ Folder: Create "C:\Hands-On-RPA-with-AA-Sample-Data\"
17    └─ ┌─ Log to file "Ref,Surname,FirstName,A..." to "C:\Hands-On-RPA-with-AA-Sample-Data\"
      └─ └─
```

Section 5 – Updating the output file

Set the following properties for the Log to file action on line 19:

- File path: C:\Hands-On-RPA-with-AA-Sample-Data\lesson09_Output\\$sSubFolder\$\Output.csv
- Enter text to log:
\$recSource[0]\$, \$recSource[1]\$, \$recSource[2]\$, \$recSource[3]\$
- When logging: Append existing log file
- The properties window should look like this:



Section 5 – Updating the output file

- And that's it – great work! Our bot is now complete.
- The development interface for this section should look like this

```
18  // Comment "----- Section 5 - Update output file" :  
19  Log to file "$recSource[0]$,${recSour...}" to "C:\Hands-On-RPA-with-AA-Sample-Data\  
           _Output\$strSubFolder$\Output.csv" :
```

Section 6 – Closing the source file

- Set the following properties for the CSV/TXT: Close action on line 21:Session name: Data
- The properties window should look like this:



Section 6 – Closing the source file

- And that's it – great work! Our bot is now complete.
- The development interface for this section should look like this:



Working with the filesystem

- There may be instances where we must create, delete, move, or rename files and folders.
- Automation Anywhere has two packages: one for files and one for folders.
- In the following screenshot, you can see the actions that are available for managing the filesystem in Automation Anywhere:

File	Folder
📄 Assign	📁 Zip
📄 Copy	📁 Copy
📄 Create	📁 Create
📄 Delete	📁 Unzip
📄 Download CR file	📁 Delete
📄 Open	📁 Open
📄 Print	📁 Rename
📄 Print multiple files	
📄 Rename	

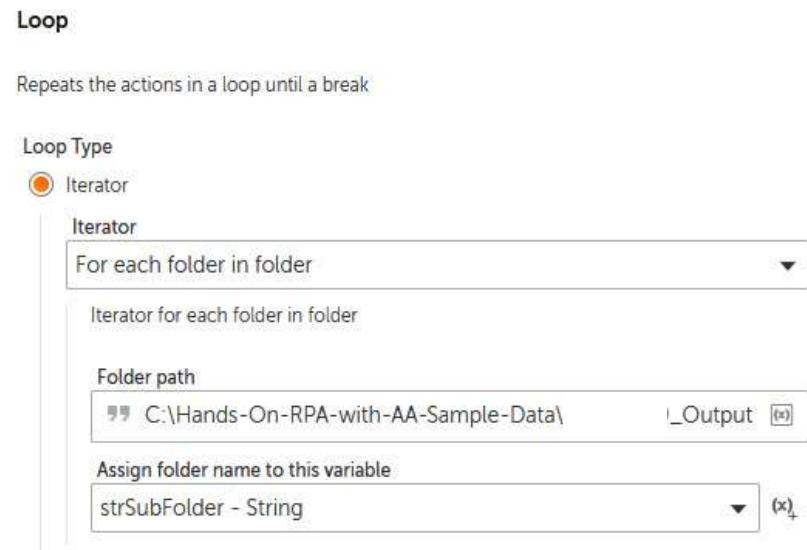
Working with the filesystem

- Add a new Comment action called "-----" on line 7 and click on Save.
- Your bot should look like this:

1	#[Comment -----	:
2	#[Comment ----- Section 1 - Loop through Sub Folders	:
3	#[Comment ----- Section 2 - Rename output file	:
4	#[Comment ----- Section 3 - Copy output file	:
5	#[Comment ----- Section 4 - Delete file	:
6	#[Comment ----- Section 5 - Delete Sub Folder	:
7	#[Comment -----	:

Section 1 – Looping through subfolders

- Set the following properties for the Loop action on line 3:Loop Type: For each folder in folder
- Folder path: C:\Hands-On-RPA-with-AA-Sample-Data\lesson09_Output
- Assign folder name to this variable: strSubFolder - String
- The properties window should look like this:



Section 1 – Looping through subfolders

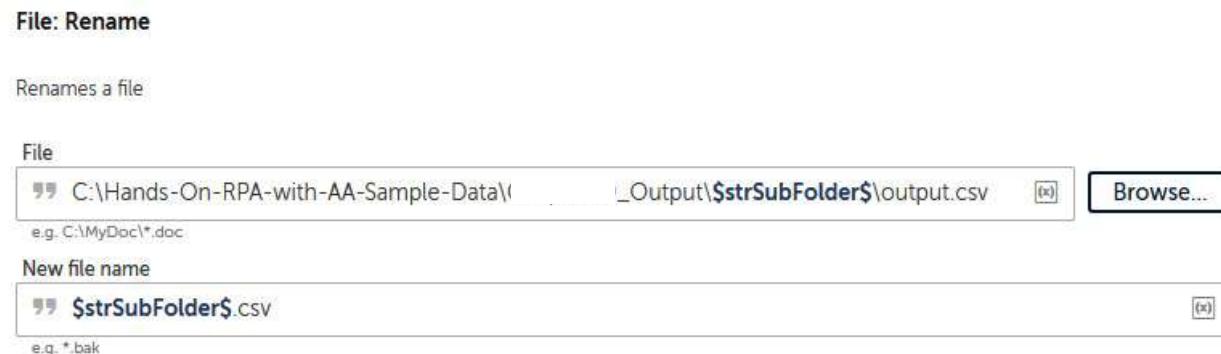
- Since all the other sections will be completed within this Loop, drag Comment lines 4 to 7 so that they are within the Loop element on line 3.
- The development interface should look like this:

```
1  // Comment "-----"
2  // Comment "----- Section 1 - Loop through Sub Folders"
3  ↗ Loop for each folder and assign folder name to $sSubFolder$ :
4  // Comment "----- Section 2 - Rename output file"
5  // Comment "----- Section 3 - Copy output file"
6  // Comment "----- Section 4 -Delete file"
7  // Comment "----- Section 5 - Delete Sub Folder"
8  // Comment "-----"
```

Section 2 – Renaming the output file

Set the following properties for the File: Rename action on line 5:

- File: C:\Hands-On-RPA-with-AA-Sample-Data\lesson09_Output\\$strSubFolder\$\output.csv
- New file name: \$strSubFolder\$.csv
- The properties window should look like this:



Section 3 – Copying the output file

- Set the following properties for the File: Copy action on line 7:
- Source file: C:\Hands-On-RPA-with-AA-Sample-Data\lesson09_Output\\$sSubFolder\$\\$strSubFolder\$.csv
- Destination file/folder: C:\Hands-On-RPA-with-AA-Sample-Data\lesson09_Output\\$strSubFolder\$.csv
- The properties window should look like this:



Section 4 – Deleting the output file

Set the following properties for the File: Delete action on line 9:

- File: C:\Hands-On-RPA-with-AA-Sample-Data\lesson09_Output\\$strSubFolder\$\\$strSubFolder\$.csv
- The properties window should look like this:

File: Delete

Deletes a file

File

C:\Hands-On-RPA-with-AA-Sample-Data\

_Output\\$strSubFolder\$\\$strSubFolder\$.csv



Browse...

e.g. C:\MyDoc*.doc

Section 5 – Deleting the subfolder

- Set the following properties for the Folder: Delete action on line 11:
- Folder: C:\Hands-On-RPA-with-AA-Sample-Data\lesson09_Output\\$strSubFolder\$
- The properties window should look like this:

Folder: Delete

Deletes a folder



Section 5 – Deleting the subfolder

- Now that the bot is complete, you can go ahead and run it. The development interface should look like this:

The screenshot shows a sequence of 12 steps in a development interface:

1. Comment "-----"
2. Comment "----- Section 1 - Loop through Sub Folders"
3. Loop for each folder and assign folder name to \$strSubFolder\$
 - 4. Comment "----- Section 2 - Rename output file"
 - 5. File: Rename "C:\Hands-On-RPA-with-AA-Sample-Data\1_O..." to "\$strSubFolder\$.csv"
 - 6. Comment "----- Section 3 - Copy output file"
 - 7. File: Copy "C:\Hands-On-RPA-with-AA-Sample-Data\1_O..." to "C:\Hands-On-RPA-with-AA-Sample-Data\1_Output\\$strSubFolder\$.csv"
 - 8. Comment "----- Section 4 - Delete file"
 - 9. File: Delete "C:\Hands-On-RPA-with-AA-Sample-Data\1_Output\\$strSubFolder\$\\$strSubFolder\$.csv"
 - 10. Comment "----- Section 5 - Delete Sub Folder"
 - 11. Folder: Delete "C:\Hands-On-RPA-with-AA-Sample-Data\1_Output\\$strSubFolder\$"
 - 12. Comment "-----"

End

Summary

- We have covered a lot of packages. Just to recap once more, first, we built some nested loops that went through lists, records, and folders. We also discovered a new Record data type variable.
- You were then given some exposure to all the other types of looping available, all of which we will cover in later lessons.
- The walk-throughs also included building some rule-based decisions with conditional statements, using variables, and checking if files exist.

Working with XML Files



Working with XML Files

- To change things up a little, we will not be building a single bot to demonstrate XML actions. Instead, we will go through the individual actions within the XML package.
- Each walk-through will provide a step-by-step guide that shows you how to use each action. We will be using the following packages:

<input checked="" type="checkbox"/> Comment	<input checked="" type="checkbox"/> NumberUtils
<input checked="" type="checkbox"/> If	<input checked="" type="checkbox"/> Step
<input checked="" type="checkbox"/> Loop	<input checked="" type="checkbox"/> String
<input checked="" type="checkbox"/> Message box	<input checked="" type="checkbox"/> XML

Working with XML Files

In this lesson, we will cover the following topics:

- Starting, validating, and ending XML sessions
- Reading and updating XML nodes
- Inserting and deleting XML nodes
- Executing XPath functions