

IBM Cloud Pak for Business Automation Demos and Labs 2022

Application Automation using IBM RPA

V 1.5

Bu Feng Hou

houbf@cn.ibm.com

Paul Pacholski

pacholsk@ca.ibm.com

Olaf Hahnl

olaf.hahnl@de.ibm.com

NOTICES

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

TRADEMARKS

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

ITIL is a Registered Trade Mark of AXELOS Limited.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

© Copyright International Business Machines Corporation 2020.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Table of Contents

1	INTRODUCTION.....	5
2	OVERVIEW.....	5
2.1	PRE-REQUISITES	6
2.2	REFERENCES.....	7
3	ACCESSING THE ENVIRONMENT.....	7
3.1	RESERVE ENVIRONMENT.....	7
3.2	ACTIVATE RPA LICENSE	9
3.3	REGISTER YOUR USER ACCOUNT.....	11
4	BUILD IT YOURSELF – STEP-BY-STEP INSTRUCTIONS	13
4.1	EXERCISE 1: JAVA SWING APPLICATION AUTOMATION.....	13
4.1.1	<i>Develop Bot Script</i>	14
4.1.2	<i>Verification Instructions</i>	38
4.2	EXERCISE 2: WEB APPLICATION AUTOMATION.....	40
4.2.1	<i>Develop Bot Script</i>	40
4.2.2	<i>Verification Instructions</i>	51
4.2.3	<i>Publish Script to Tenant</i>	52

1 Introduction

IBM RPA provides a comprehensive set of Robotic Process Automation (RPA) features:

- **Unattended bots**
Use an RPA-driven digital workforce to automate repetitive tasks without human intervention.
- **Attended bots**
Remote Desktop Automation (RDA) enables a human workforce to augment work using bots to perform repetitive tasks on demand.
- **Intelligent Virtual Agent (IVAs) chatbots**
Combine chat and RPA commands to create chatbots through multiple channels that can provide engaging client interactions.
- **Optical Character Recognition (OCR)**
Process documents by extracting structured data from unstructured content.
- **Dashboards**
Gain business insights into business operations.

By bringing RPA in-house, IBM can provide customers with additional benefits:

- **Faster time to value**
Speed and simplicity of purchasing and deploying through easier licensing.
- **A comprehensive platform to automate all types of use cases**
Tighter integrations between RPA and the rest of our platform.
- **Automate business and IT processes**
Expand our automation mission to IT use cases.
- **Operationalize AI**
Fulfill IBM's vision of operationalizing AI in every corner of the business.

You can explore the [Documentation](#) to understand more details about IBM RPA.

2 Overview

The objective of this lab is to learn how to automate business applications using IBM RPA Studio. The lab is composed of 2 exercises:

1. The first exercise shows how to use IBM RPA Studio to automate a stand-alone Java Swing application. The stand-alone Java application named **Client Management System** simulates an enterprise client management application that offers only a user interface but no public API.

2. The second exercise shows how to use IBM RPA Studio to automate a web application. The simple web application named **Services Management System** simulates a web-based enterprise services management system that also does not provide public APIs.

2.1 Pre-requisites

For this lab, you need to reserve **IBM Robotic Process Automation** environment from IBM Technology Zone. All the pre-requisites have been pre-installed/configured in the lab template. The information below is just for information purposes.

IBM Products:

- IBM Robotic Process Automation Studio v20.12.5.

Custom Solutions/Code:

- A Java swing application simulating the backend, third-party system for the Client Management System.
- A web application simulating the backend, third-party Services Management System for managing the services a client has signed up to.

2.2 References

1. [IBM Robotic Process Automation Documentation](#)
2. [IBM Robotic Process Automation Command Documentation](#)

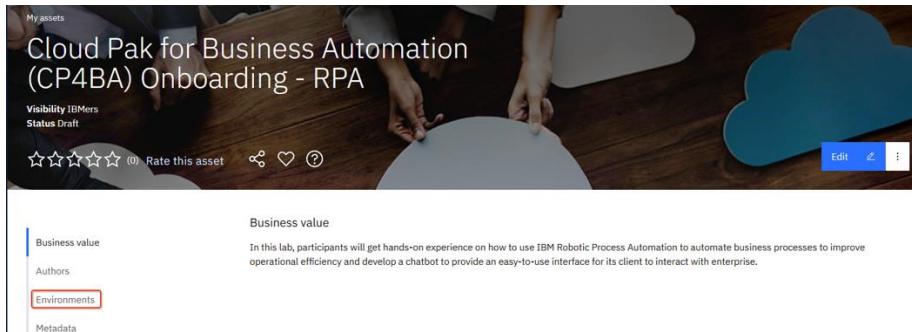
3 Accessing the Environment

If you have already reserved lab environment from IBM Technology Zone and registered your RPA account, please go to [Chapter 4](#) directly.

3.1 Reserve Environment

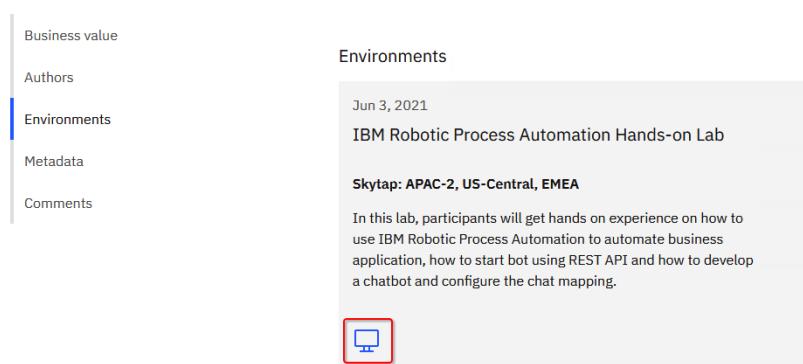
To get started with this lab, please follow below steps to reserve an environment:

1. Click [here](#) to open IBM Technology Zone Reservation portal. You need to use your IBMID to login to the portal.



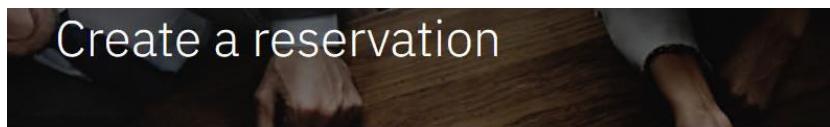
The screenshot shows a web interface for managing assets. At the top, it displays 'My assets' and the title 'Cloud Pak for Business Automation (CP4BA) Onboarding - RPA'. Below the title, there are sections for 'Visibility' (IBMer), 'Status' (Draft), and 'Rating' (0 stars). A sidebar on the left lists categories: Business value, Authors, Environments (which is highlighted with a red box), and Metadata. The main content area shows 'Business value' with a brief description: 'In this lab, participants will get hands-on experience on how to use IBM Robotic Process Automation to automate business processes to improve operational efficiency and develop a chatbot to provide an easy-to-use interface for its client to interact with enterprise.' To the right, there is an 'Edit' button and a blue cloud icon.

2. Click **Environments** on the left panel, then click the  icon.



The screenshot shows the 'Environments' section of the portal. On the left, a sidebar lists 'Business value', 'Authors', 'Environments' (with a red box around it), 'Metadata', and 'Comments'. The main content area shows a single environment entry: 'Jun 3, 2021' and 'IBM Robotic Process Automation Hands-on Lab'. Below this, it says 'Skylab: APAC-2, US-Central, EMEA'. A detailed description follows: 'In this lab, participants will get hands on experience on how to use IBM Robotic Process Automation to automate business application, how to start bot using REST API and how to develop a chatbot and configure the chat mapping.' At the bottom of the entry is a blue monitor icon with a red box around it.

3. Select **Reserve for now**, then click **Submit**.



Purpose: Select Practice/Self-Education.

Purpose description: Enter something like **Self Education**.

End date and time: Select the end date and time that the environment will be deleted.

Preferred Geography: Select the geography where your environment will be created. In order to get better network connection, suggest you to select the same geography as where you located in.

Name: IBM Robotic Process Automation Hands-on Lab
Name this reservation. This will help identify it in your reservation list.

Purpose: Practice / Self-Education
Please enter the purpose for this activity

Purpose description: Self Education

What are you doing? Why do you need this? What are you trying to accomplish?

End date and time

Select a date: 06/17/2021 Select a time: 9:30 PM America/Los_Angeles

Preferred Geography: Cloud Pak for Business Automation (CP4BA) Onboarding - RPA [APAC-2]

Notes: Enter any notes you'd like to attach to this reservation

5. Once you have reserved an environment, you will receive an email with a link to access the management console for the environment including a password (**Desktop URL** and **password**). It also contains a URL to access the IBM RPA Rest Service remotely.

Your environment is ready

Your environment is now available. Please use the following information to access the environment.

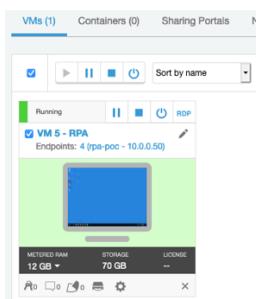
For guidance and support for your environment named **IBM Robotic Process Automation Hands-on Lab**, please refer to these helpful links:

Collection Name: Cloud Pak for Business Automation (CP4BA) Onboarding - RPA [US-Central]
Collection URL:

<https://techzone.ibm.com/collection/60b804b14ca8f000189d567d>

- <http://services-uscentral.skytap.com:10410>
- Desktop URL:
<https://cloud.skytap.com/vms/b211ab242192247f099071618165c79b/desktops>
- Desktop password: S23cm18r
- Environment ID: 121739666
- Environment name: DTE2_2051685_HOUBF_2022-03-07 02:02:38_2022-03-08 10:02:00

6. Click the desktop access link above to open your environment. When you are prompted to enter environment password, please enter the desktop password above. Wait a few minutes, your environment will be started as below.



3.2 Activate RPA license

Before you can start and log into IBM RPA Studio, you need to re-activate the RPA license every time the RPA agent machine (or in our case the VM 5) is restarted. This is caused by special lab infrastructure setup and configuration and only required for this lab.

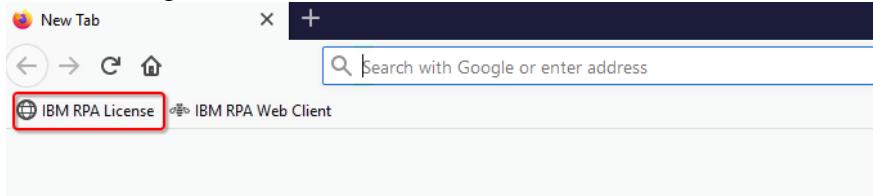
1. Click **VM 5 – RPA** to open the Windows environment in web browser.
2. Click Service from Windows toolbar.



3. Check and ensure that the **IBM Robotic Process Automation Agent** service is in running status through Windows Service Manager.



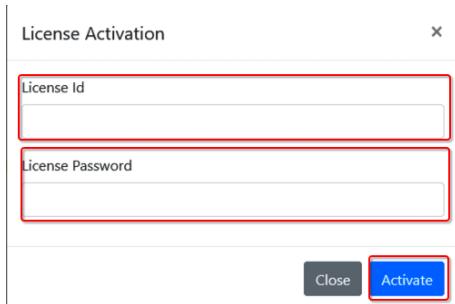
4. Start Firefox, click **IBM RPA license** from the bookmark toolbar to open IBM RPA license manager.



5. You will see the message **Not Licensed**. Click **Activate** button to open the License Activation window.

A screenshot of a web page titled 'License'. At the top, it shows 'Tenant: SWAT Team' and 'Computer: RPA-POC'. Below that, it says 'Edition: Not Licensed' and 'Last Error: [9400] The system could not be identified as the one to which this license was issued.' At the bottom, there are three buttons: 'Activate' (highlighted with a red border), 'Refresh', and 'Deactivate'.

6. Enter the License ID and License Password and click the **Activate** button. You can get the License ID and License Password from [here](#).



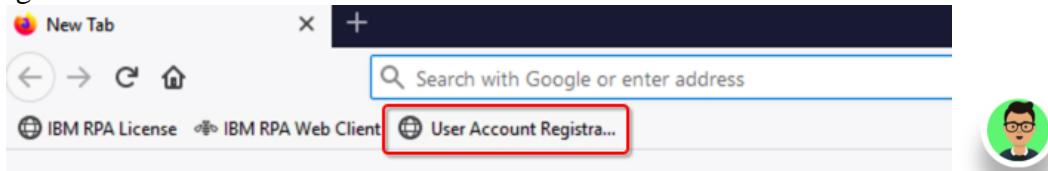
When the license is activated, you should be able to see the number of licenses available for each component.

A screenshot of a web page titled 'License'. At the top, it shows 'Tenant: SWAT Team' and 'Computer: RPA-POC'. Below that, it says 'Edition: Enterprise', 'Id: 5358', and 'Licensed to: SWAT TEAM', all of which are highlighted with a red border. At the bottom, there is a table with rows for Runtime, Studio, Launcher, Vault, IVR, and Bot, each showing '0 of 1' usage. The entire table is highlighted with a red border.

3.3 Register your user account

This lab requires IBM RPA account to grant you access to the IBM RPA tenant and Studio to develop, test and publish bot scripts. Please follow below steps to register your RPA account if you don't have access to the IBM RPA tenant yet.

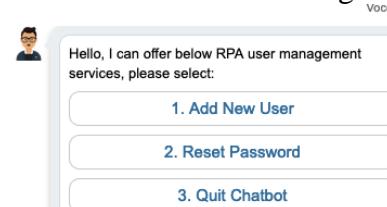
1. Start Firefox, click **User Account Registration** from the bookmark toolbar to open the user account management chatbot. Click the floating robot button at the bottom-right corner.



2. Once you see chatbot message requesting you to select the type of user account, select "RPA User Registration".

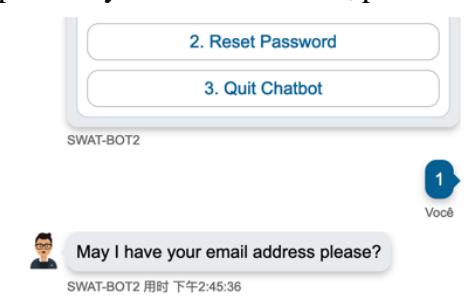


3. Wait for the chatbot message and select "Add New User".



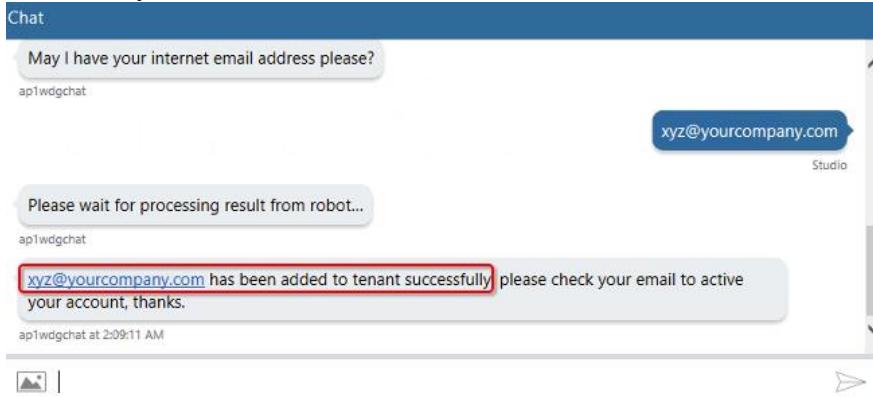
4. Wait for message from the chatbot requesting to enter your email address.

Please note: Your email address is required to grant you access to the IBM RPA tenant only. It will not be used for any other purpose. In case you have any objection to provide your email address, please stop and contact your lab host.



5. Enter your email address. Then wait for the chatbot to register your account. This may take a minute or two. Once the chatbot finishes registering your account, it will

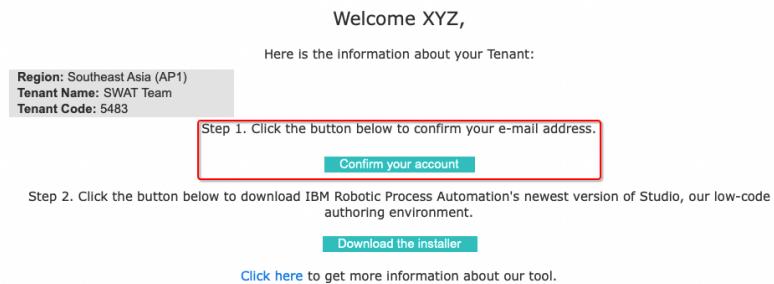
prompt a message to indicate that your account has been added to the RPA tenant successfully.



6. Check your mailbox. You should receive an email from **IBM Robotic Process Automation** as below. Click **Confirm your account** from Step 1.

Since IBM Robotic Process Automation Studio has been installed on the lab environment, Step 2 is not required. You could download the installer and install it on your own machine if you like.

From: IBM Robotic Process Automation <no-reply@wdgautomation.com>
To: xyz@yourcompany.com
Date: 2021/04/22 13:33
Subject: Welcome - Your user has been created



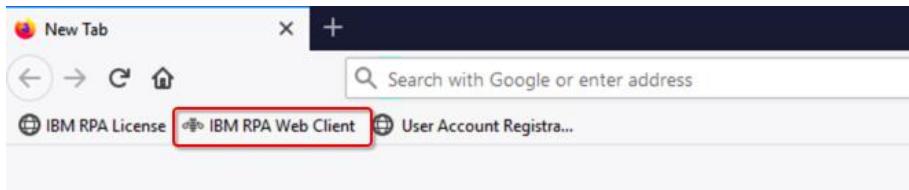
7. Enter new password for your account. Once done, click **Reset password** which will activate your account with the password you set here.

Reset password

A form titled 'Enter a new user for your IBM RPA user houbfx@gmail.com'. It contains fields for 'New password' and 'Confirm password', both of which are highlighted with red boxes. Below these fields is a blue 'Reset password' button, which is also highlighted with a red box.

Please follow below steps to check if your account has been registered and activated successfully:

8. Start Firefox, click **IBM RPA Web Client** from the bookmark toolbar.



9. Enter the **email address** you used to register your account in the web client login page and click the **Continue** button.

A screenshot of the 'Log in to IBM Robotic Process Automation' page. It has a 'User name' input field which is highlighted with a red box. Below it is a 'Continue' button.

10. Enter the **password** you set when activating your account, make sure to select **SWAT Team** tenant. Once done, click the **Login** button.

A screenshot of the same login page. The 'Tenant' dropdown menu is open and shows 'SWAT Team' highlighted with a red box. The 'Password' input field is also highlighted with a red box. At the bottom, there is a 'Login' button which is highlighted with a blue box, and a 'Forgot password' link.

You should now be successfully logged into the web client. In case you can't login, please check if you entered the correct username and password. If so, please contact your lab host.

4 Build it yourself – Step-by-step instructions

4.1 Exercise 1: Java Swing Application Automation

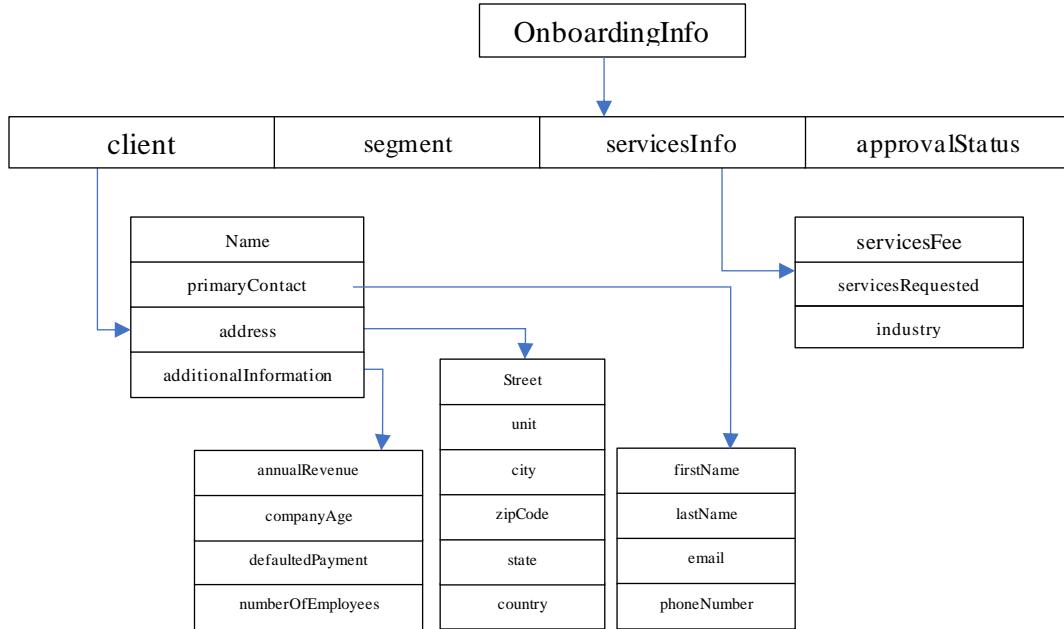
This exercise will be performed on the RPA-VM **VM 5 – RPA** and takes about 1 hour to complete.

As explained above, this is part of an end-to-end client-onboarding solution. At the end of the client-onboarding process, the bot will be automatically started by a Workflow process, the client information will be passed to bot as input parameter in JSON string.

You need to develop a bot script to retrieve the client information from the JSON string first. Then start **Client Management System** Java application to add client information. Once the client information is added into the Java application, it will generate a client ID.

You need to get the client ID and start the **Services Management System** web application to add signed services information which will be performed in exercise 2.

The client information data model is defined as below



An example JSON string looks like below. It will be used to develop and test the bot script in this lab.

```
{
  "servicesInfo": {
    "servicesRequested": "Fibre Internet",
    "servicesFee": 25000,
    "industry": "Telecom"
  },
  "approvalStatus": "Approved",
  "segment": "Segment 1",
  "client": {
    "additionalInformation": {
      "defaultedPayment": true,
      "companyAge": 10,
      "annualRevenue": 5000000,
      "numberOfEmployees": 1200
    },
    "address": {
      "zipCode": "48911",
      "country": "United States of America",
      "unit": "IA",
      "city": "Lansing",
      "street": "3974 Carson St",
      "state": "MI"
    },
    "primaryContact": {
      "firstName": "June Marie",
      "lastName": "Sample",
      "phoneNumber": "517-555-0000",
      "email": "jmarie@example.com"
    },
    "name": "Automation Elite Inc."
  }
}
```

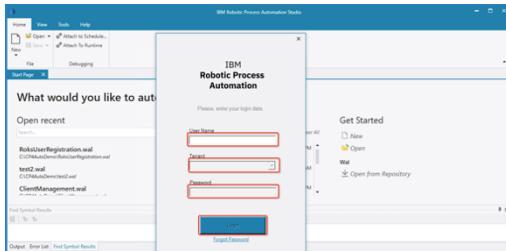
4.1.1 Develop Bot Script

1. Start the **IBM RPA Studio** by clicking the Studio icon on the Windows desktop.

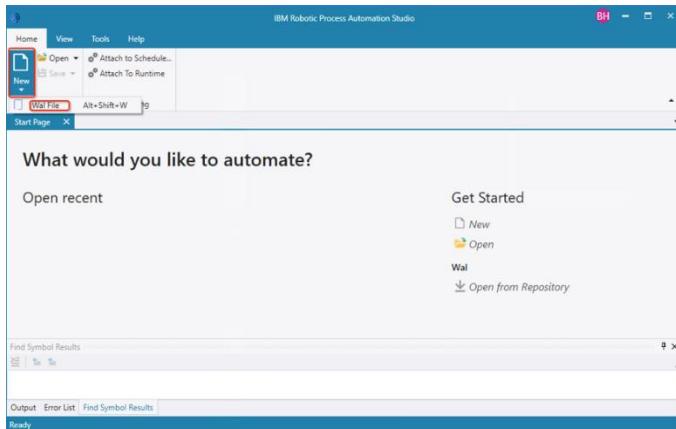


2. Enter your user credential as below, click Login button to start and login Studio.
 1. **User Name:** Enter the email address you used to register your RPA account and press **Enter**. This will enable the Tenant and password fields as well as the Login button.

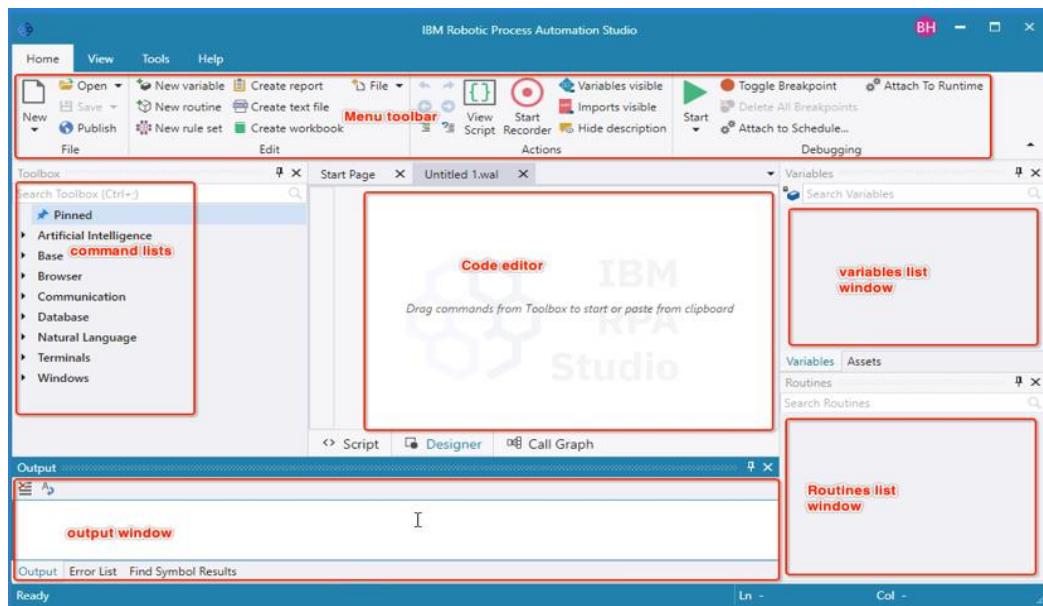
2. **Tenant:** Select **SWAT Team** in case it is not set as default.
3. **Password:** Enter password you set when activating your account.



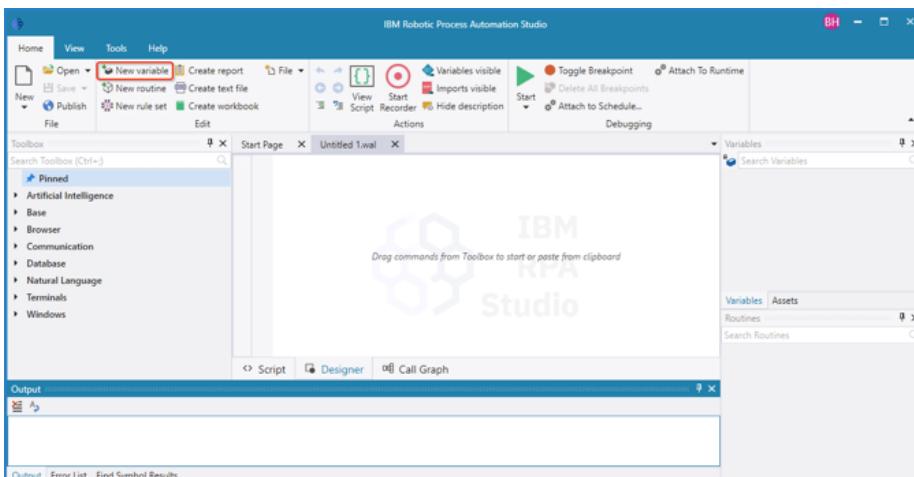
3. Click **New** and select **Wal File** which will start the IBM RPA Studio script development perspective.



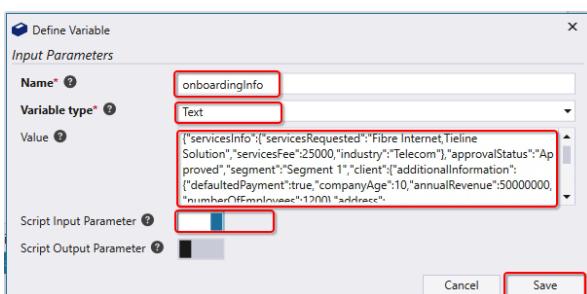
First familiarize yourself with the IBM RPA Studio user interface. The **Commands** are available from the left panel. It lists all available RPA commands which you can drag and drop to develop automation script. The **Code editor** view is in the center. IBM RPA Studio provides 3 types of code views – **Script**, **Designer** and **Call Graph**. On the right, the **Variable** panel shows all defined variables and **Routines** lists all defined routines in the main script. At the bottom, the **Output** and **Error** views display all log message and errors if any.



- Click **New variable** to first define an input parameter to store a JSON string. It will contain the client information as well as signed service information.



- Enter **onboardingInfo** as name of the variable, select **Text** as variable type, copy the sample JSON data from **ClientOnboardingInfo.json**, you can find **ClientOnboardingInfo.json** from lab materials page in **Lab Data** folder. Check **Script Input Parameter**, then click **Save**.

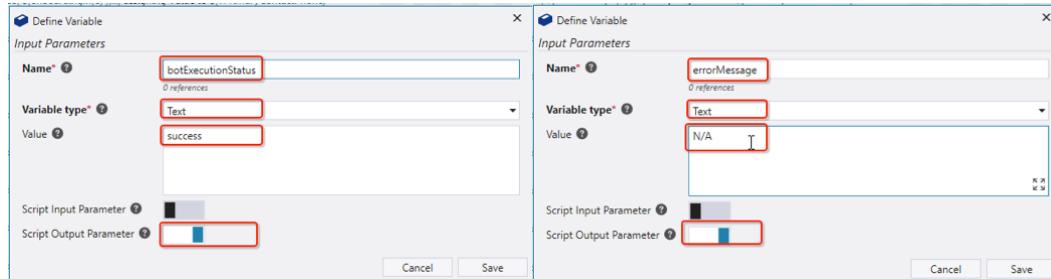


- Follow the same steps to **define two output variables** of type Text:

- **botExecutionStatus** with **default value** “success”
- **errorMessage** with **default value** “N/A” as below

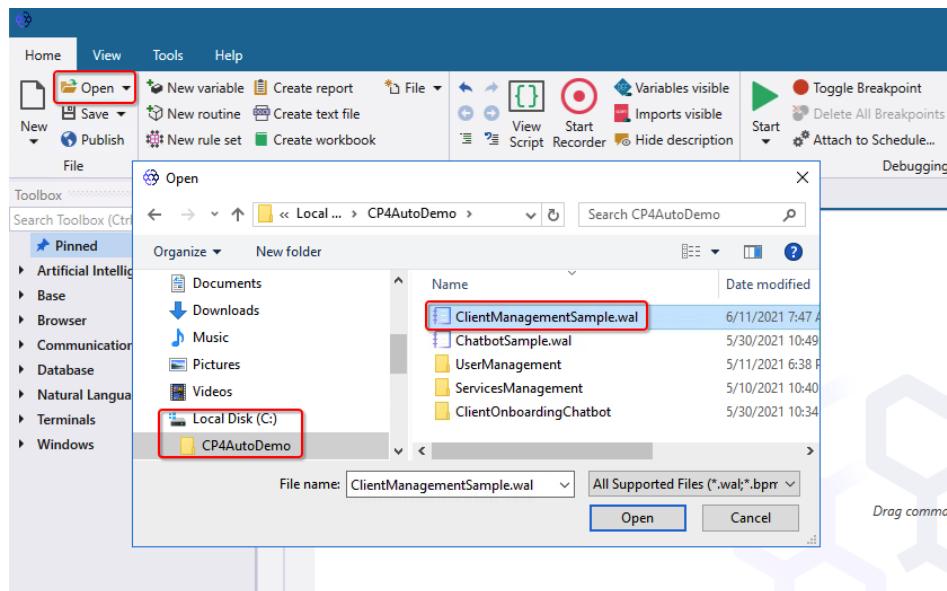
For both check the **Script Output Parameter** switch.

In a real project, the bot needs to handle bot execution failures and exceptions, set appropriate status code and error message accordingly. In this lab, the bot will simply return the default value set here back to caller. You can find more information about handling exceptions in IBM RPA in the [documentation](#).



7. Before you can extract the data from JSON input string, it requires to define the corresponding variables to hold the extracted data. Follow below steps to copy variables definition commands from sample script,

1. Click **Open** and select **c:\CP4AutoDemo\ClientManagementSample.wal**.



2. Click **Script** to switch to Script perspective, select from **line 4** to **line 29**, right click the mouse and select **Copy**.

The screenshot shows the IBM RPA Studio interface in the Script perspective. A context menu is open over a block of WScript code, specifically over the line 'defVar --name vSegment --type String'. The menu options include Format, Toggle Outlining, Collapse to Definitions, Show All Outlining, Cut (Ctrl+X), Copy (Ctrl+C), and Paste (Ctrl+V). The 'Copy' option is highlighted with a red box.

```

1 defVar --name onboardInfo --type String --value "{\"servicesInfo\":{\"servicesRequested\":\"Fibre Internet,TelOne Solution\",\"servicesFee\":25000,\"industry\":\"\"}}
2 defVar --name botExecutionStatus --type String --value success --output
3 defVar --name errorMessage --type String --value "N/A" --output
4 defVar --name vIndustry --type String
5 defVar --name vSignedServices --type String
6 defVar --name vServiceFee --type String
7 defVar --name vSegment --type String
8 defVar --name vApplicantStatus --type String
9 defVar --name vClientName --type String
10 defVar --name vPrimaryContactName --type String
11 defVar --name vPrimaryContactPhone --type String
12 defVar --name vPrimaryContactEmail --type String
13 defVar --name vClientAddressStreet --type String
14 defVar --name vClientAddressUnit --type String
15 defVar --name vClientAddressCity --type String
16 defVar --name vClientAddressState --type String
17 defVar --name vClientAddressZipCode --type String
18 defVar --name vClientAddressCountry --type String
19 defVar --name vCompanyAge --type String
20 defVar --name vEmployeeID --type String
21 defVar --name vEmployeeName --type String
22 defVar --name vDefaultedPayment --type String
23 defVar --name vIndividualService --type String
24 defVar --name vMessage --type String
25 defVar --name vWindow --type Window
26 defVar --name vID --type Numeric
27 defVar --name vSuccess --type Boolean
28 defVar --name vClientID --type Numeric
29 defVar --name vSignedServiceList --type List --intrinsicType String
30 webStart --name web01 --type Firefox --headless
31 webExecuteJavaScript --script "var obj = JSON.parse(\"${onboardingInfo}\")\nvar objAttr = obj.client.name\nreturn objAttr" vClientName=value
32 webExecuteJavaScript --script "var obj = JSON.parse(\"${onboardingInfo}\")\nvar objAttr = obj.client.primaryContact.firstName + \" \" + obj.client.primaryContact.

```

3. Click your script, and then click **Script** to switch to Script perspective, move the cursor to the end of line 3 and press Enter.

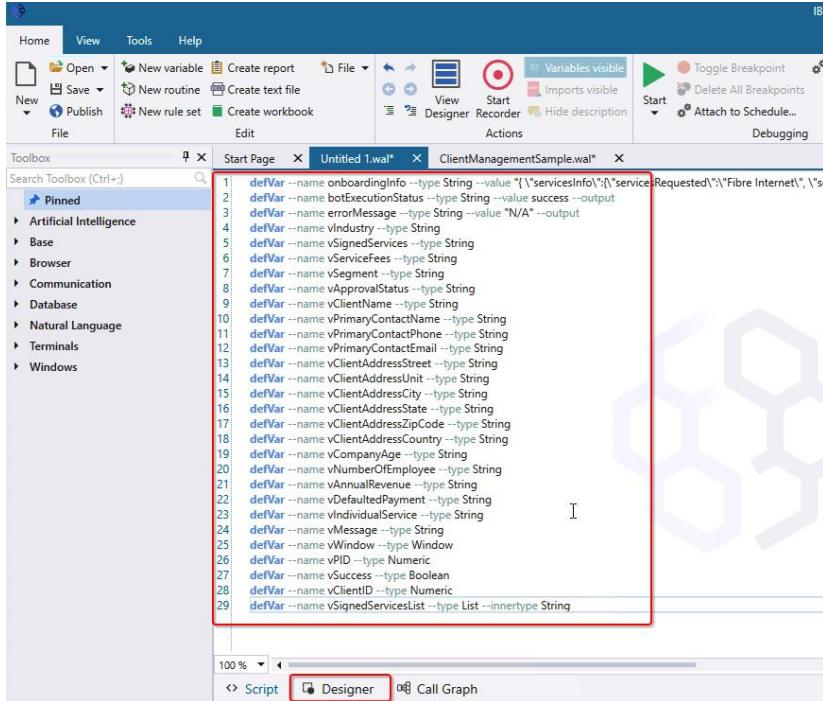
The screenshot shows the IBM RPA Studio interface in the Script perspective. The cursor is positioned at the end of line 3 of the WScript code. The 'Script' tab is highlighted with a red box at the bottom of the interface.

```

1 defVar --name onboardInfo --type String --value "{\"servicesInfo\":{\"servicesRequested\":\"Fibre Intern
2 defVar --name botExecutionStatus --type String --value success --output
3 defVar --name errorMessage --type String --value "N/A" --output
4

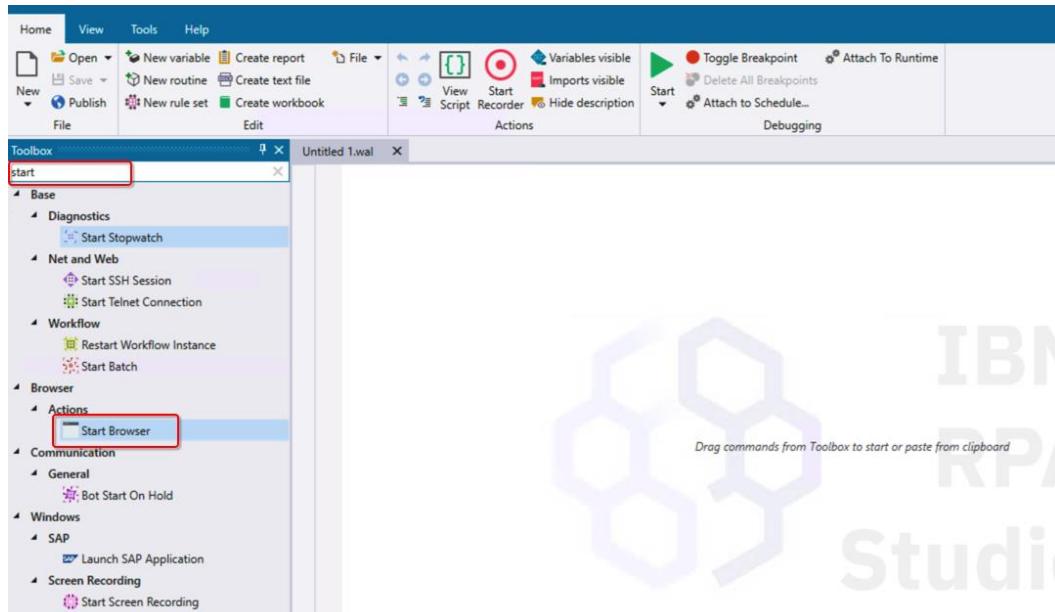
```

4. Paste the variables definition commands there. Once done, your script should be similar as below. Click **Designer** to switch to Designer perspective again.



- Within IBM RPA multiple approaches are possible to extract values from a JSON string. In this exercise we will use JavaScript for that purpose. To be able to execute JavaScript we need to first start a browser.

Type **start** in search box in the Toolbox panel. Find and double click the **Start Browser** command from the **Actions** command category.



- Configure the **Start Browser** command as below. Once done, click **Save**.

Input Parameters:

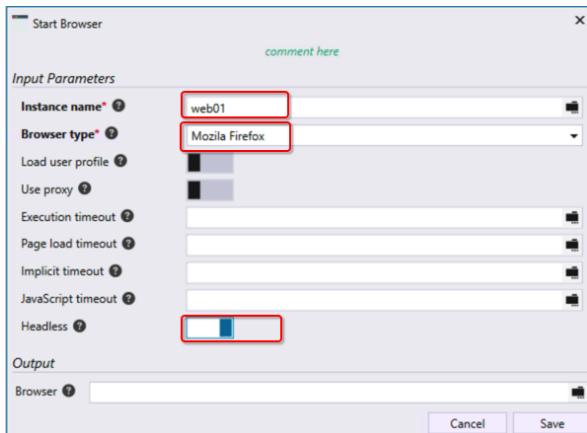
Instance name: Enter web01.

This is the web browser instance name. Once finishing using the browser, it needs to be closed using the instance name.

Browser type: Select Mozilla Firefox.

Headless: Click the switch to turn it on.

Since we just need to execute JavaScript within the browser, it doesn't need open browser window.



- Once you click the **Save** button above, Studio will automatically add **Close Browser** command. This is best practice to close the resources after finish using it. As **Instance name**, enter the same name you entered above – **web01**. Once done, click **Save**.



- Find the **Run JavaScript** command, drag and drop it to before the **Close Brower** command, and configure it as below to extract the client name first. Once done, click **Save**.

Please note if you double click the command, it will put the new command at the end of script, you need manually adjust the command sequence.

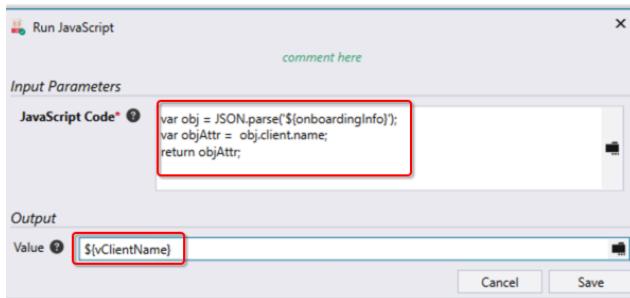
Input Parameters:

JavaScript Code: Enter below JavaScript snippet

```
var obj = JSON.parse(${onboardingInfo});  
var objAttr = obj.client.name;  
return objAttr;
```

Output:

Value: click  icon on the right, select **vClientName** variable.



12. To avoid adding another dozens of **Run JavaScript** commands to retrieve the rest of client information, follow below steps to copy the commands from sample script,

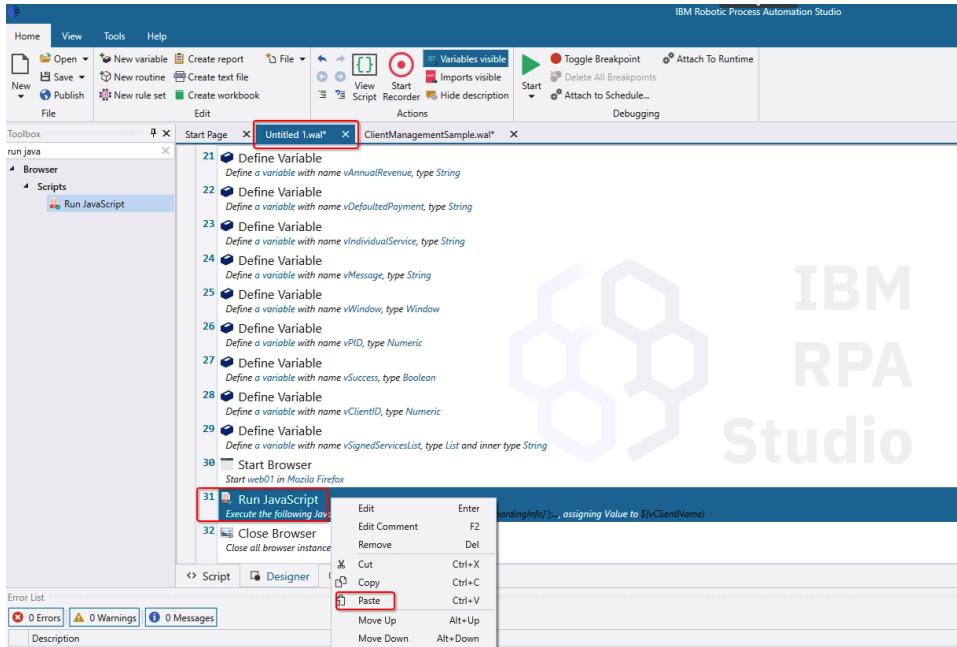
1. Click **ClientManagementSample.wal** script, select from line 32 to line 49, right click the mouse and select **Copy**.

```

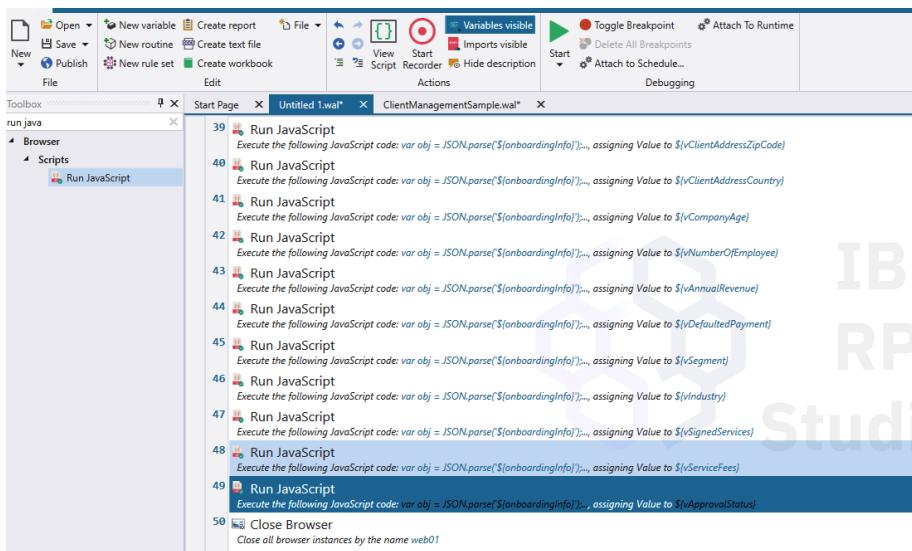
Home View Tools Help
File Open New variable Create report File Designer Recorder Hide description
New Save Publish New routine Create text file Actions Start Debugging
File Toolbox Start Page X Untitled 1xall X ClientManagementSample.wal X
Search Toolbox (Ctrl+J) Pinned Artificial Intelligence Base Browser Communication Database Natural Language Terminals Windows
20 defVar -name vNumberOfEmployees -type String
21 defVar -name vAnnualRevenue -type String
22 defVar -name vDefaultedPayment -type String
23 defVar -name vIndividualService -type String
24 defVar -name vMessage -type String
25 defVar -name vWindow -type Window
26 defVar -name vPID -type Numeric
27 defVar -name vSuccess -type Boolean
28 defVar -name vClientID -type Numeric
29 defVar -name vSignedServicesList --type List --innerType String
30 webStart -name web01 -type Firefox -headless
31 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.client.name\nreturn objAttr'; vClientName=value
32 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.client.name\nobjAttr = obj.client.name\nContact.firstName=v\nreturn objAttr'; vPrimaryContactName=value
33 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.client.name\nobjAttr = obj.client.name\nContact.phoneNumber=v\nreturn objAttr'; vPrimaryContactPhone:value
34 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.client.name\nobjAttr = obj.client.name\nContact.email=v\nreturn objAttr'; vPrimaryContactEmail:value
35 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.client.address\nobjAttr = obj.client.address\nAddress.street=v\nreturn objAttr'; ClientAddressStreet:value
36 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.client.address\nobjAttr = obj.client.address\nAddress.city=v\nreturn objAttr'; ClientAddressCity:value
37 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.client.address\nobjAttr = obj.client.address\nAddress.state=v\nreturn objAttr'; ClientAddressState:value
38 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.client.address\nobjAttr = obj.client.address\nAddress.zipCode=v\nreturn objAttr'; ClientAddressZipCode:value
39 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.client.address\nobjAttr = obj.client.address\ncountry=v\nreturn objAttr'; ClientAddressCountry:value
40 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.client.address\nobjAttr = obj.client.address\nCompany.name=v\nreturn objAttr'; CompanyName:value
41 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.client.address\nobjAttr = obj.client.address\nCompany.address=v\nreturn objAttr'; CompanyAddress:value
42 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.client.address\nobjAttr = obj.client.address\nCompany.zipCode=v\nreturn objAttr'; CompanyZipCode:value
43 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.client.address\nobjAttr = obj.client.address\nCompany.state=v\nreturn objAttr'; CompanyState:value
44 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.client.address\nobjAttr = obj.client.address\nCompany.zipCode=v\nreturn objAttr'; CompanyZipCode:value
45 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.client.address\nobjAttr = obj.client.address\nCompany.state=v\nreturn objAttr'; CompanyState:value
46 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.client.address\nobjAttr = obj.client.address\nCompany.zipCode=v\nreturn objAttr'; CompanyZipCode:value
47 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.client.address\nobjAttr = obj.client.address\nCompany.state=v\nreturn objAttr'; CompanyState:value
48 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.client.address\nobjAttr = obj.client.address\nCompany.zipCode=v\nreturn objAttr'; CompanyZipCode:value
49 webExecuteJavaScript --script 'var obj = JSON.parse('${onboardingInfo}')\nvar objAttr = obj.approvalStatus\nobjAttr = obj.approvalStatus\nApprovalStatus.value=v\nreturn objAttr'; ApprovalStatusValue:value
50 webClose -name web01
51 launchWindow --executablepath "C:\CP4AutoDemo\ClientManagement.war" vWindow=v value vPID=processId

```

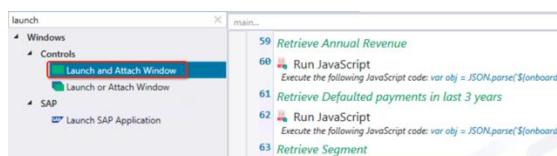
2. Click your script, select the line 31, right click the mouse and select **Paste**.



3. Once done, your script should be similar as below,



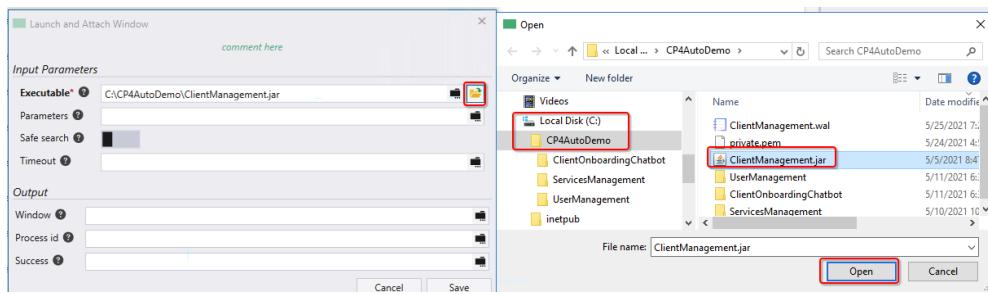
13. Enter **Launch** in search toolbox field, find and double click **Launch and Attach Window** command as we next need to launch Java swing application.



14. Configure the **Launch and Attach Window** command as below. Once done, click **Save**.

Input parameters:

Executable: Click the  icon and select the file **C:\CP4AutoDemo\ClientManagement.jar**



Parameters: Unchanged/leave it blank

Safe Search: Unchanged

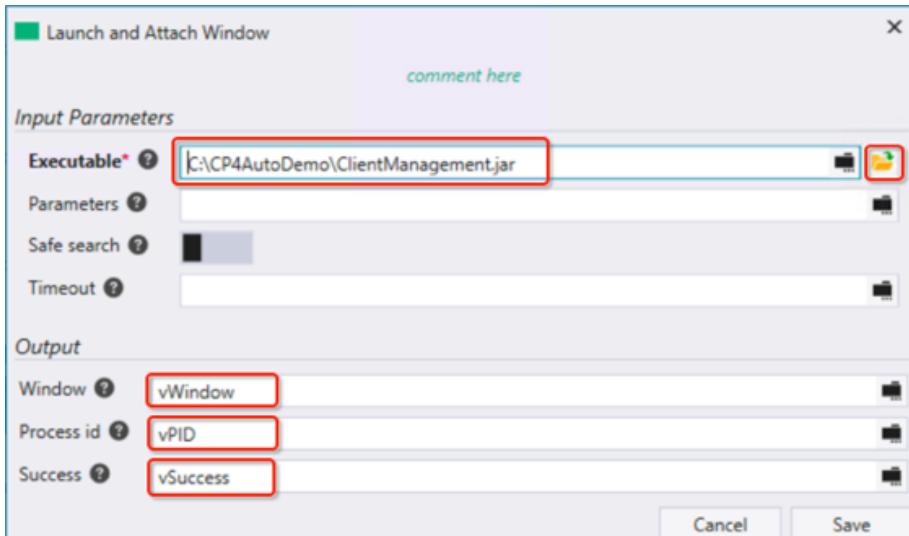
Timeout: Unchanged/leave it blank

Output:

Window: Click  icon on the right, select **vWindow** variable

Process id: Click  icon on the right, select **vPID** variable

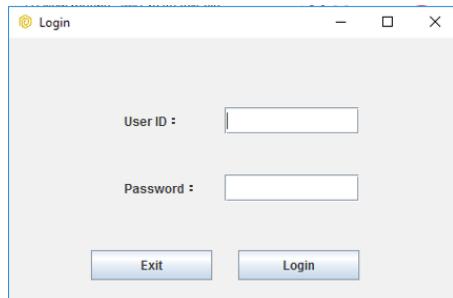
Success: Click  icon on the right, select variable **vSuccess** variable



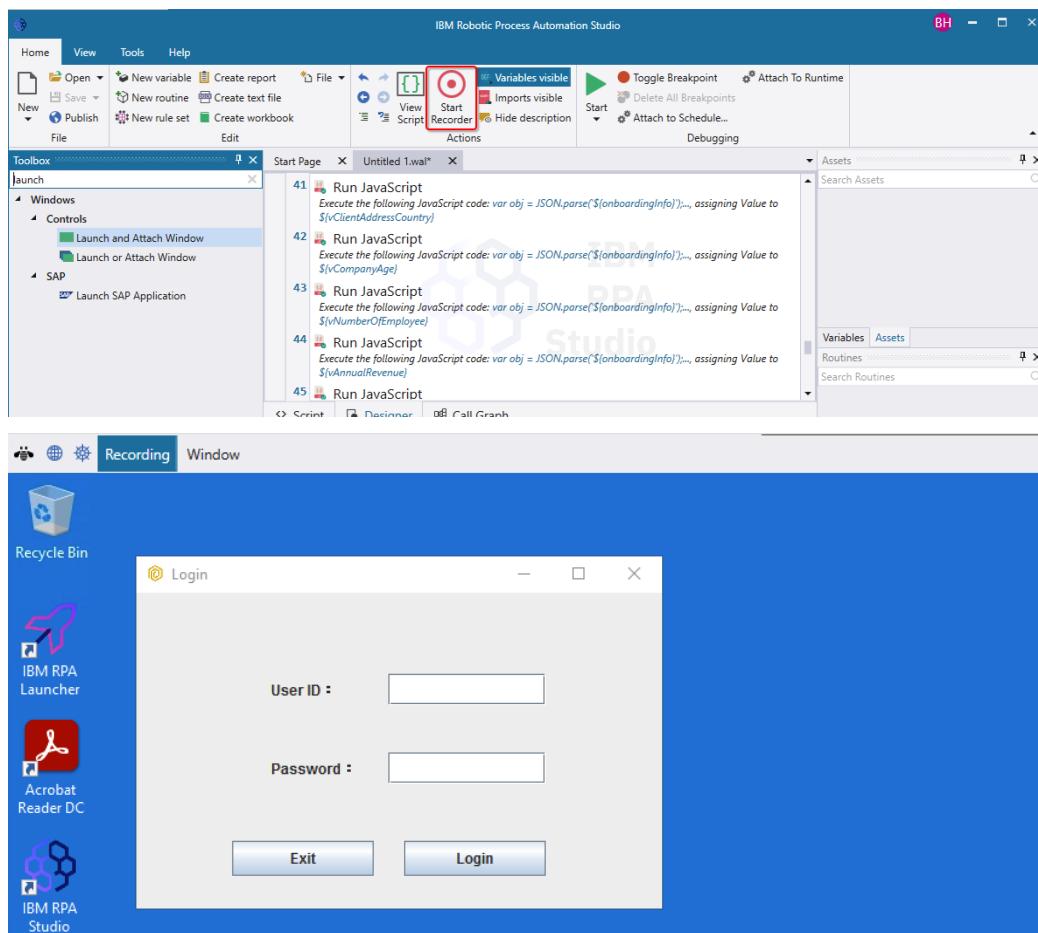
Next, we will use the Recorder to automate the Java application. You need to manually start the Java application first.

15. Double click **ClientManagement.jar** in the **C:\CP4AutoDemo** directory to start the Client Management System application.

Please don't close the Client Management System Login window but go back to the RPA Studio window.



16. Click **Start Recorder** from the Studio toolbar. This will open the Recorder. Once started, you should see the recording window as below.



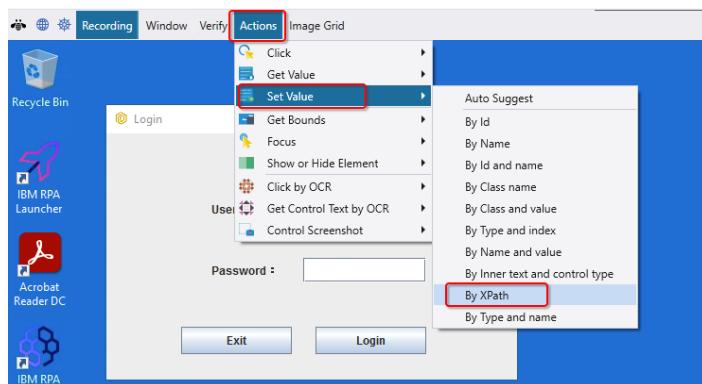
17. Automate the login to the Client Management System application.

1. Press and hold the left CTRL key, move the mouse to the User ID textbox, and wait for a few seconds. The User ID textbox will be captured and marked as light-red color as shown below. Once the User ID textbox is captured, release the CTRL key.

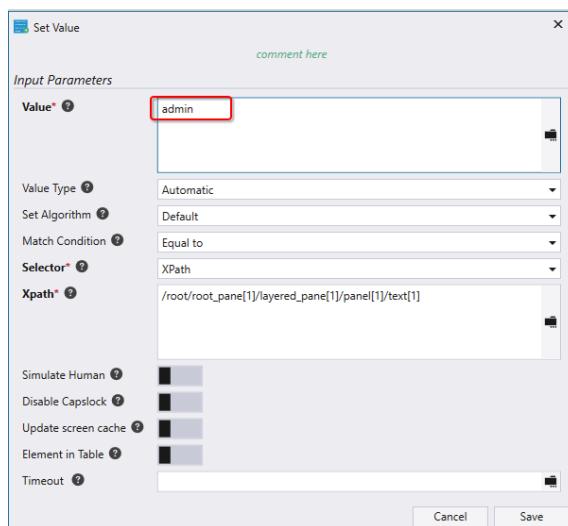


2. Select **Actions** → **Set Value** → **By XPath** from recorder toolbar menu.

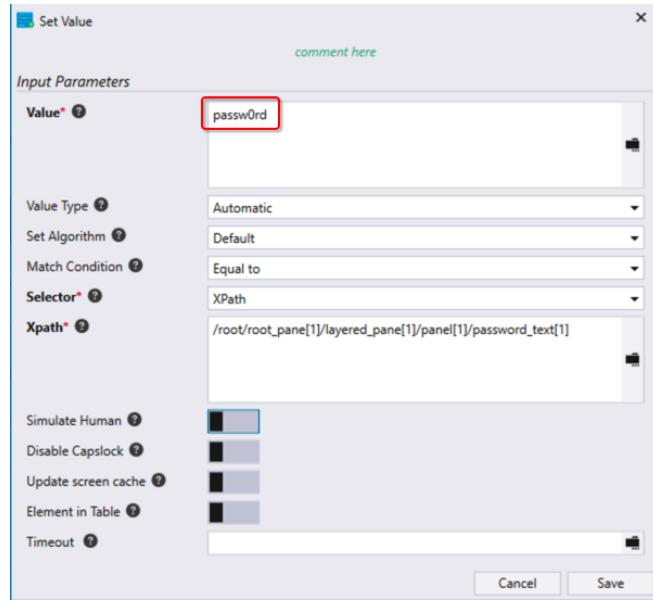
Note: Since the textbox control has been captured, if you notice the red highlighting goes away when going through the menus, this is normal behavior.



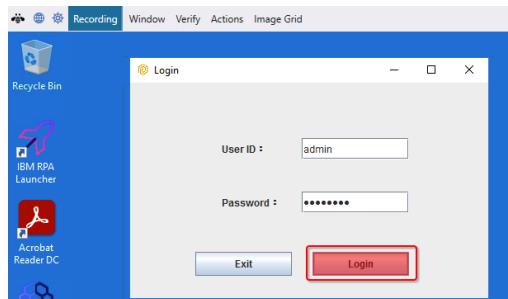
3. Configure **Set Value** command as below. Please enter **admin** which is the only user that can login to the client management system application. Once done, click **Save**.



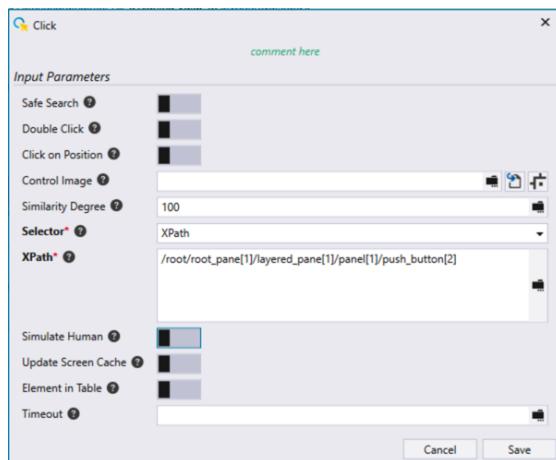
4. Follow the same steps to automate the Password field. Enter **passw0rd** (make sure to use a zero as part of the password) which is the only valid password that can be used to login.



- Follow the same steps to automate the Login button selecting **Actions → Click → By XPath.**



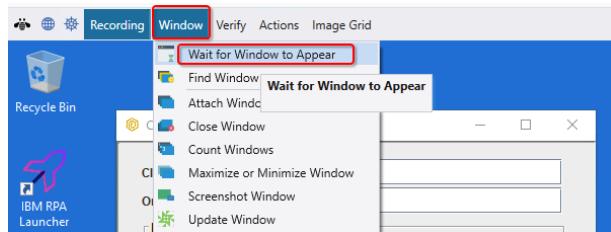
Configure the **Click** command as below. Once done, click **Save** which will finish the Login page automation and enter into Client Management System main window.



IBM RPA searches and captures controls in the current execution context. The execution context will change when switching from one window to another, in this case, switching from the **Login** window to the **Client Management System** main window. It is required

to attach the new window to the current execution context. This can be achieved by using the **Attach Window** command. Considering the performance of the machine, the new window may take some time to appear. A best practice is to use the **Wait for Window to Appear** command to ensure the new window will appear before attaching it to current execution context.

18. Select **Window → Wait for Window to Appear** from recorder toolbar menu.



19. Configure the **Wait for Window to Appear** command as below.

Input parameters

Title: Change it to **Client Management System**.

Clear all other fields and ensure the **switches are turned off**. Otherwise the command may not be able to find the window and cause the script execution to fail in the verification section later.

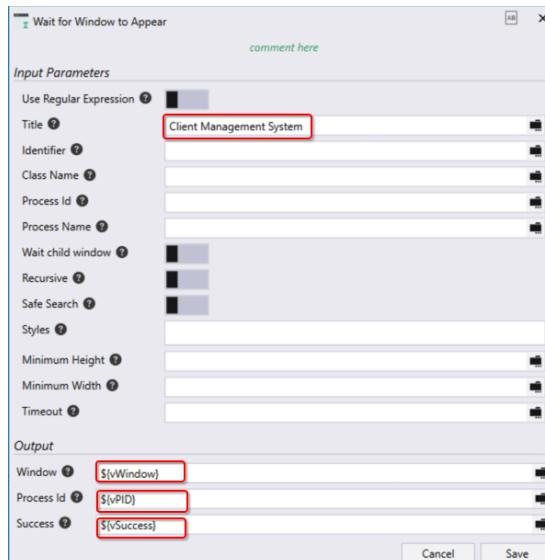
Output

Window: Click icon on the right, select **vWindow** variable

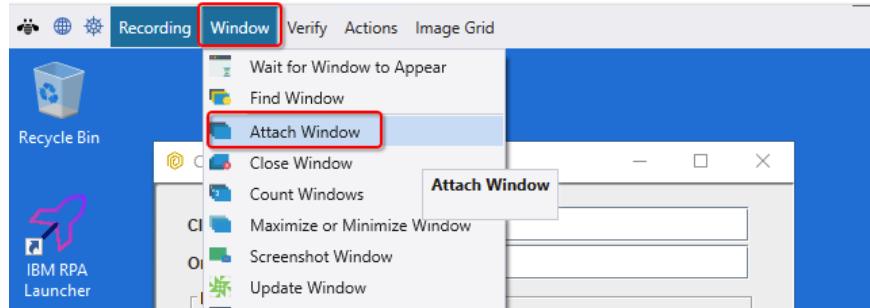
Process Id: Click icon on the right, select **vPID** variable

Success: Click icon on the right, select **vSuccess** variable

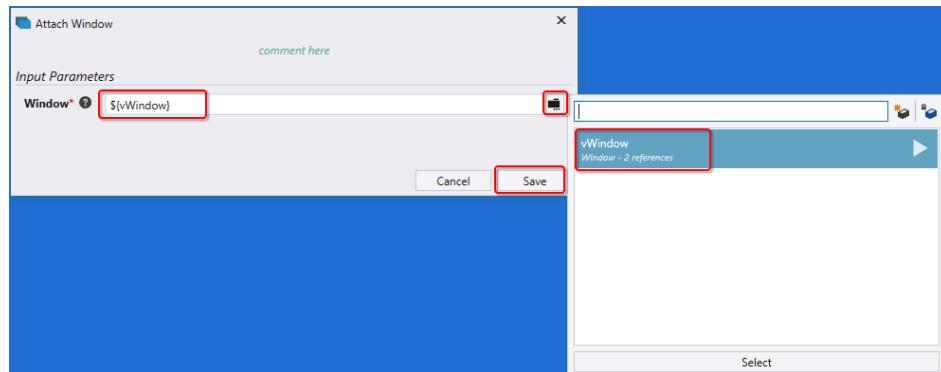
Once done, click **Save**.



20. Select **Window → Attach Window** from recorder toolbar menu.



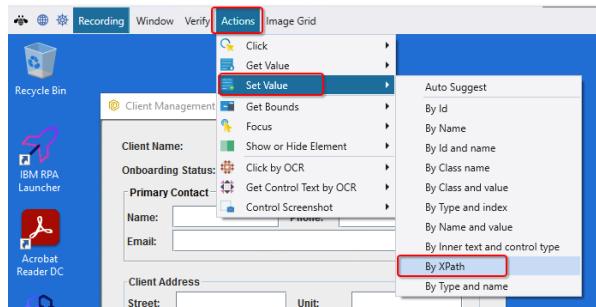
21. Configure the **Attach Window** command as below by selecting the variable **vWindow**. Once done, click **Save**.



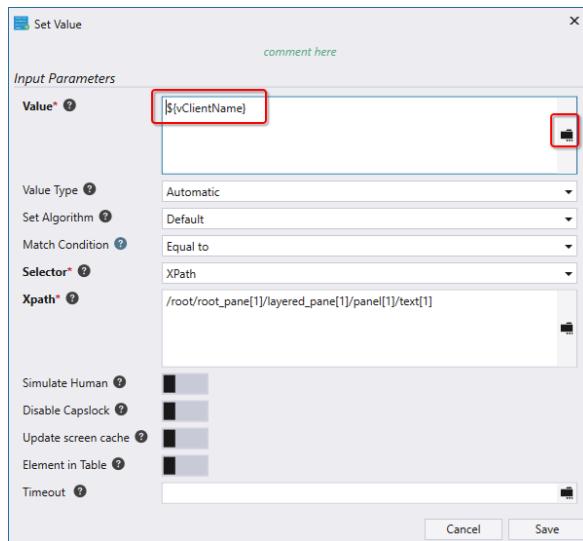
22. Automate the Client Management System application itself.

1. Press and hold the left CTRL key, move the mouse to the Client Name textbox, wait for a few second, the checkbox will be captured and marked as light-red color as below. Once the textbox is captured, release the CTRL key.

2. Select Actions → Set Value → By XPath from the recorder toolbar menu.



3. Configure the **Set Value** command as below. For Input Parameters, click the icon and select the variable **vClientName**. Once done, click **Save**.



4. Next is to automate the rest of the fields. Follow the below steps to copy the commands from sample script,

1. Switch to the RPA Studio window by clicking Studio icon from windows toolbar.



2. Click **ClientManagementSample.wal**, select from line 58 to line 70, right click the mouse and select **Copy**.

```

45 webExecuteJavaScript --script="var obj = JSON.parse('${onboardingInfo}')\n${vSegmentValue} = obj.segment\\'return objAttr' vSegmentValue\n"
46 webExecuteJavaScript --script="var obj = JSON.parse('${onboardingInfo}')\n${vSegmentValue} = obj.segment\\'industry\\' vSegmentValue\n"
47 webExecuteJavaScript --script="var obj = JSON.parse('${onboardingInfo}')\n${vSegmentValue} = obj.segment\\'industry\\' vSegmentValue\n${vSegmentValue} = obj.segment\\'objAttr' \"SignedServices\"=value\n"
48 webExecuteJavaScript --script="var obj = JSON.parse('${onboardingInfo}')\n${vSegmentValue} = obj.segment\\'industry\\' vSegmentValue\n${vSegmentValue} = obj.segment\\'objAttr' \"ServiceFees\"=value\n"
49 webExecuteJavaScript --script="var obj = JSON.parse('${onboardingInfo}')\n${vSegmentValue} = obj.segment\\'industry\\' vSegmentValue\n${vSegmentValue} = obj.segment\\'objAttr' \"objApprovalStatus\"=value\n"
50 webClose --name web01
51 launchWindow --executepath "C:\\CP4AutoDemo\\ClientManagement.jar" --value ${vSegmentValue}
52 setValue --value "admin" --setValType "Automatic" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/layered_panel[1]/panel[1]/text[1]"
53 setValue --value "password" --setValType "Automatic" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/layered_panel[1]/panel[1]/pas
54 click --selector "XPath" --controlsImitating 100 --xpath "/root/root_panel[1]/layered_panel[1]/push_button[2]" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/layered_panel[1]/push_button[2]" --value "Client Management System"
55 setValue --value ${vWindow} --setValType "Automatic" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/push_button[2]" --value ${vWindow}
56 attachNWindow --window ${vWindow}
57 setValue --value "${ClientManagement}" --setValType "Automatic" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/layered_panel[1]/pane
58 setValue --value "${vApprovalStatus}" --setValType "Automatic" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/layered_panel[1]/pane
59 setValue --value "${vPrimaryContactName}" --setValType "Automatic" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/layered_panel[1]/pane
60 setValue --value "${vPrimaryContactPhone}" --setValType "Automatic" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/layered_panel[1]/pane
61 setValue --value "${vPrimaryContactEmail}" --setValType "Automatic" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/layered_panel[1]/pane
62 setValue --value "${vClientAddressStreet}" --setValType "Automatic" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/layered_panel[1]/pane
63 setValue --value "${vClientAddressUnit}" --setValType "Automatic" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/layered_panel[1]/pane
64 setValue --value "${vClientAddressCity}" --setValType "Automatic" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/layered_panel[1]/pane
65 setValue --value "${vClientAddressState}" --setValType "Automatic" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/layered_panel[1]/pane
66 setValue --value "${vClientAddressZipCode}" --setValType "Automatic" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/layered_panel[1]/pane
67 setValue --value "${vClientAddressCountry}" --setValType "Automatic" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/layered_panel[1]/pane
68 setValue --value "${vCompanyLogo}" --setValType "Automatic" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/layered_panel[1]/pane
69 setValue --value "${vNumberOfEmployees}" --setValType "Automatic" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/layered_panel[1]/pane
70 setValue --value "${vAnnualRevenue}" --setValType "Automatic" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/layered_panel[1]/pane
71 if-left "${vApprovalStatus}" --operator "Equal_To" --right true
72     click --selector "XPath" --controlsImitating 100 --xpath "/root/root_panel[1]/layered_panel[1]/pane[1]/checkbox[1]" --algorithm "Default" --matchCondition "Equal" --selector "XPath" --xpath "/root/root_panel[1]/layered_panel[1]/pane[1]/checkbox[1]" --value "true"
73 endif
74 if-left "${vSegment}" --operator "Equal_To" --right "Segment 1"

```

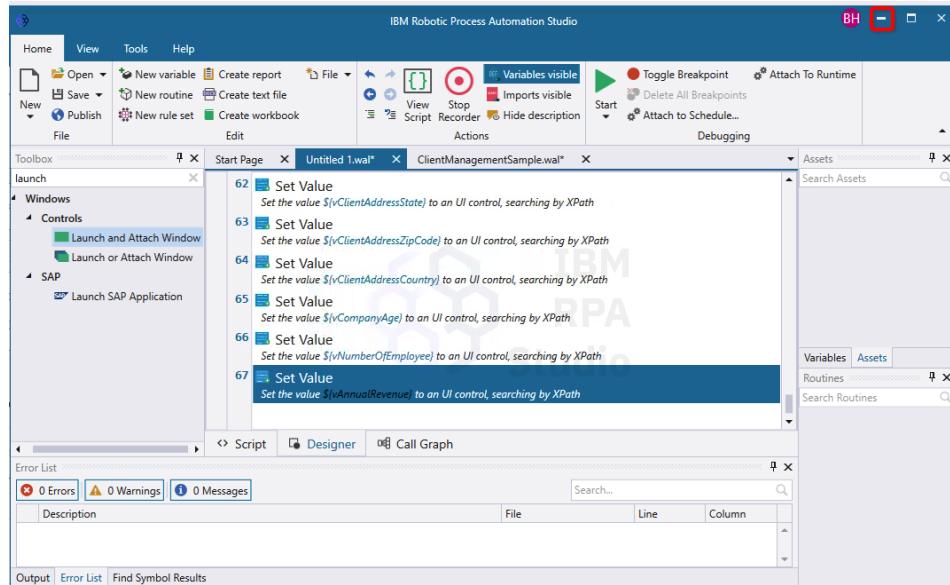
3. Click your script, select last line, right click the mouse and select Paste.

```

47 Run JavaScript
Execute the following JavaScript code: var obj = JSON.parse('${onboardingInfo}')..., assigning Value to ${vSignedServices}
48 Run JavaScript
Execute the following JavaScript code: var obj = JSON.parse('${onboardingInfo}')..., assigning Value to ${vServiceFees}
49 Run JavaScript
Execute the following JavaScript code: var obj = JSON.parse('${onboardingInfo}')..., assigning Value to ${vApprovalStatus}
50 Close Browser
Close all browser instances by the name web01
51 Launch and Attach Window
Launch an application from path C:\\CP4AutoDemo\\ClientManagement.jar and attach the application window to the current execution context, assigning Window
52 Set Value
Set the value admin to an UI control, searching by XPath
53 Set Value
Set the value password to an UI control, searching by XPath
54 Click
Single click a control, searching by XPath in the attached window
55 Wait for Window to Appear
Watch and close windowdefault timeout until a window appears, matching title Client Management System, assigning Window to ${vWindow}, Process Id to ${vProcessId}
56 Attach Window
Attaches the window ${vWindow} to the current execution context
57 Set Value
Set the value ${vClientAddressStreet} to an UI control, searching by XPath

```

4. Once done, your script should be similar as below. Switch back to Recorder windows by clicking the - icon in the top-right of Studio to minimize the Studio window.

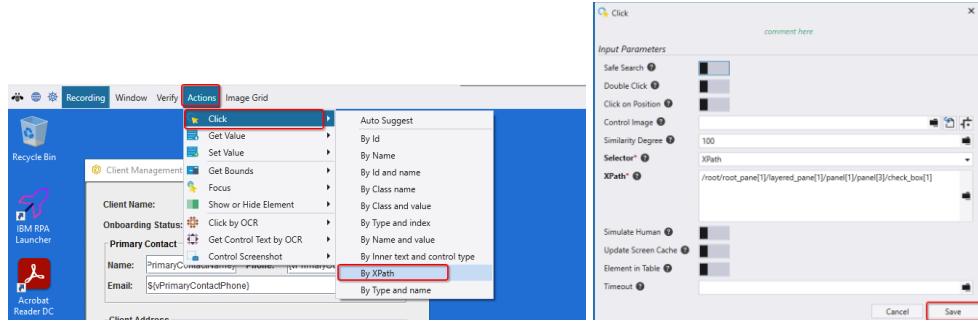


Defaulted Payment is a checkbox. It will only be checked if the client has a defaulted payment. We will use the Recorder to check it first, then add processing logic later.

- Press and hold the left CTRL key, move the mouse to the **Defaulted Payment** checkbox. Once the checkbox is captured, release the CTRL key.

The screenshot shows the 'Client Management System' application window. In the 'Financial History' section, there is a checkbox labeled 'Defaulted Payment' which is checked and highlighted with a red box. The rest of the form contains various input fields for client information like name, address, and financial details.

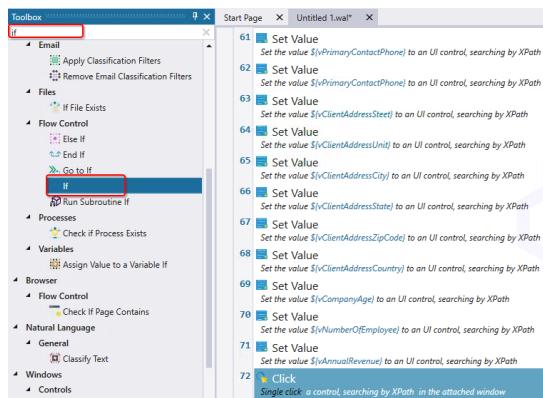
- Select **Actions** → **Click** → **By XPath** from the Recorder toolbar menu. In the Click command configuration window, keep all the default settings and click **Save**.



7. Switch to the RPA Studio window by clicking Studio icon on windows toolbar.



8. As mentioned above, we need to add logic to handle if the **Defaulted Payment** checkbox is to be checked or not. Enter **if** in search toolbox window, find and double click the “If” command.

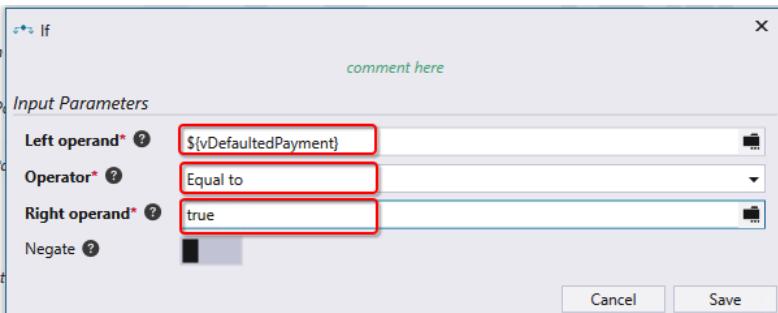


9. Configure the If command as below. Once done, click **Save**.

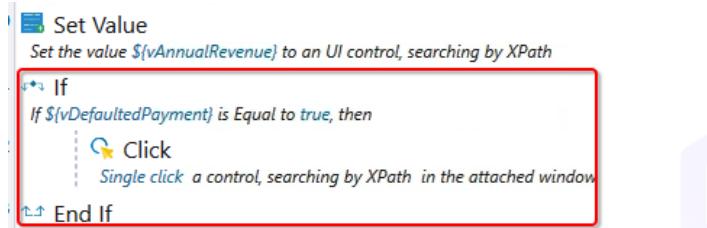
For **Left operand**, click icon on the right, select **vDefaultPayment** variable.

For **Operator**, select **Equal to**.

For **Right operand**, enter **true**.



10. In the Studio Designer editor view, select the last **Click** command added, drag and drop it to the middle of the **If/End If** commands. Once done, it should look like below. This will ensure that the checkbox is only checked if the **vDefaultPayment** variable has the value of **true**.



11. Switch back to Recorder windows by clicking the - icon in the top-right of Studio to minimize the Studio window.



Now let's automate the **segment** field. The segment field is a set of two radio buttons. Similar to the Defaulted Payment checkbox, we will use the Recorder to click the first radio button first, then add processing logic later.

12. Press and hold the **left CTRL key**, move the mouse to the **Segment 1** radio button. Once the radio button is captured, release the CTRL key, then select **Actions → Click → By XPath** from the Recorder toolbar menu.

In the **Click** command configuration window, keep all the default settings and click **Save**.

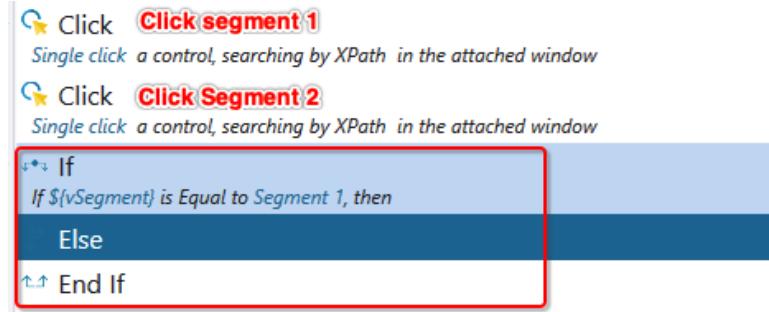
13. Follow the same approach above to click the **Segment 2** radio button.
14. Switch to the Studio window by clicking Studio icon on Windows toolbar.



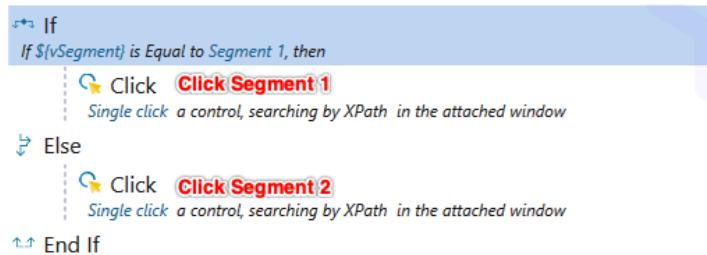
15. Add an **If** command to the end and configure it as below. Once done, your script should look similar as the one below.



16. Add an **Else** command between the **If/End If** commands above. Once done, your scripts should be similar as the one below.



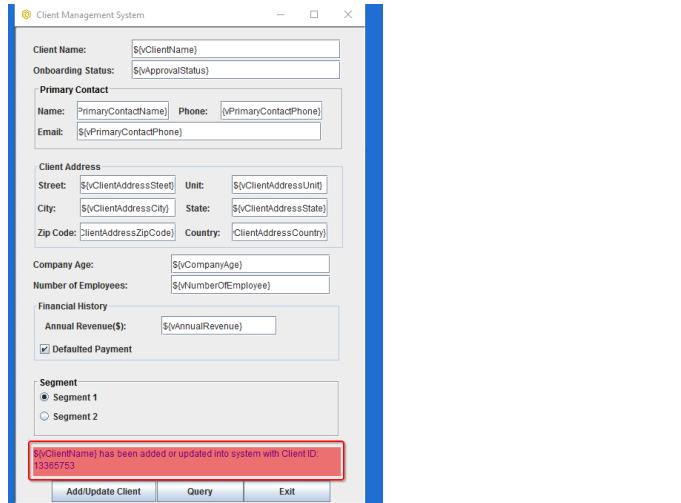
17. Drag and drop the **Click segment 1** radio button command (the first of the two) between the **If/Else** command and the **Click Segment 2** radio button command between the **Else/End If** command. Once done, your scripts should look similar as the one below.



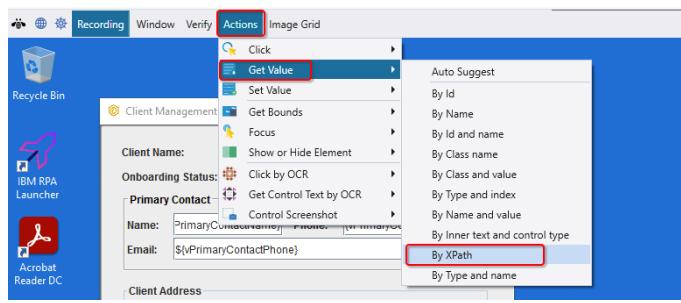
18. Switch back to the Recorder window by minimizing Studio window again.
19. Press and hold the **left CTRL key**, move the mouse to the **Add/Update Client** button. Once the button is captured, release the CTRL key, select **Actions → Click → By XPath** from the Recorder toolbar menu. In the **Click** command configuration window, keep all the default settings and click **Save**.

Once you click the **Add/Update Client** button, the Client Management System application will display a message to indicate the client information has been added or updated including a client ID. You need to capture the client ID and use it in exercise 2 during the automation of the web application.

20. Press and hold the **left CTRL key**, move the mouse to the message box. Once the message box is captured, release the CTRL key.



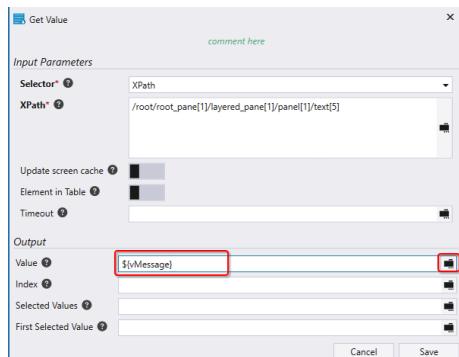
21. Select **Actions** → **Get Value** → **By XPath** from the Recorder toolbar menu.



22. Configure the **Get Value** command as below. Once done, click **Save**.

For **Input parameters**, leave all the defaults setting.

For **Output parameters**, select **vMessage**.



23. Press and hold the **left CTRL key**, move the mouse to the **Exit** button. Once the button is captured, release the CTRL key. Select **Actions** → **Click** → **By XPath** from the recorder toolbar menu.

In the **Click** command configuration window, keep all the default settings and click **Save**.

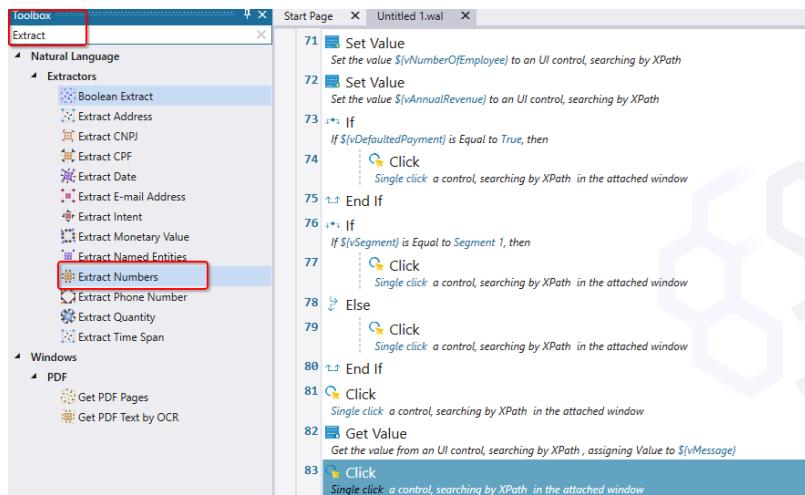
Note: The robot executes very fast, you may want to add a delay before closing the client management application for demo purpose.

24. Close the Recorder by clicking  icon in the top-right corner in the Recorder window and return back to Studio.



25. In order to integrate with the Services Management System web application, the bot needs to retrieve the client ID from the Client Management System Java application. The message retrieved in step 22 is stored in the variable vMessage and includes the client ID. Besides the client ID, it also contains additional information. The bot needs to extract the client ID from vMessage to use it.

Enter **extract** in the search toolbox, find and double click the **Extract Numbers** command.



26. Configure the **Extract Numbers** command as below. Once done, click **Save**.

Input Parameters:

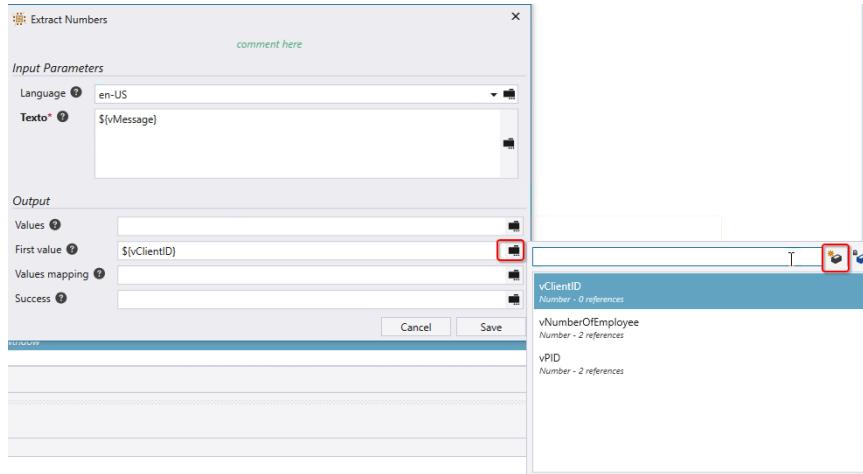
Language: Select en-US

Texto: Click  icon and select variable vMessage

Output:

Values: If the source string contains multiple numbers, the command will retrieve all numbers as a list. In this lab, it only has one number, leave it blank.

First Value: First number in the source string. Click the  icon on the right, and select **vClientID** variable.

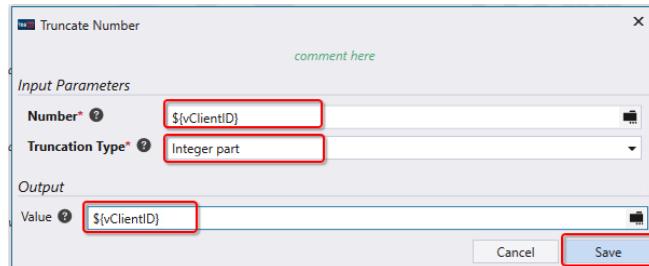


27. The clientID extracted above contains decimals which is not required and needs to be truncated.

Find and double click **Truncate Number** command.



28. Configure the **Truncate Number** command as below. Once done, click **Save**.



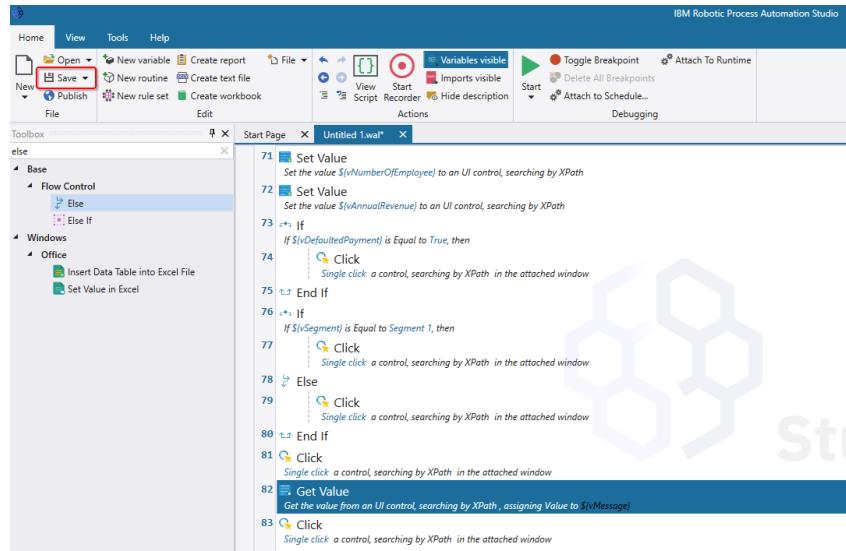
29. Add a **Log Message** command and configure it as below. Once done, click **Save**.

For Message, first type in “Client ID = “, then click the icon and select the variable **vClientID**.



30. Now you have automated the Client Management System Java application. Click the **Save** icon in the Studio menu toolbar and save your script to the **C:\CP4AutoDemo** folder.

Note: Please make sure to use a unique name e.g. including your initials to avoid naming conflicts when publishing your script to the shared tenant in a later exercise.

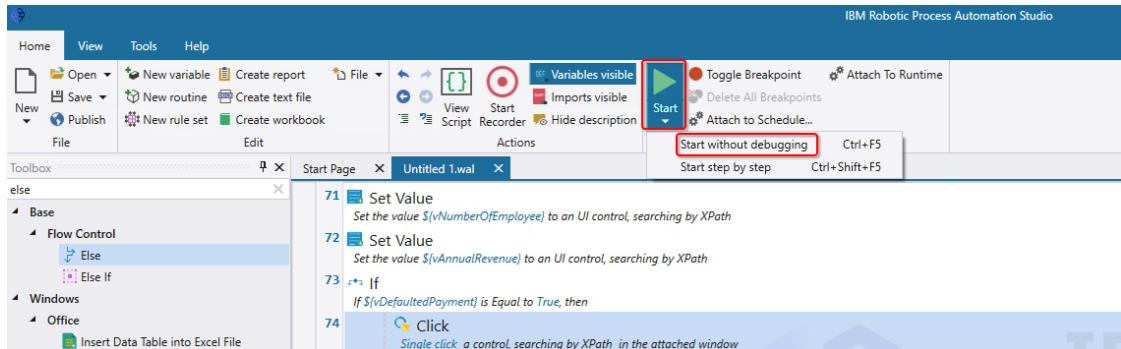


4.1.2 Verification Instructions

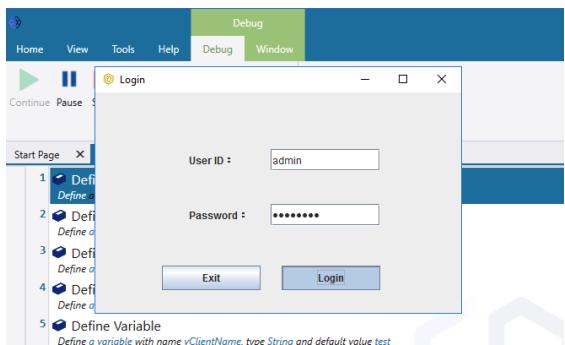
IBM RPA Studio provides two types of debug approaches to validate your bot scripts – **Start without debugging** and **Start step by step**. In this exercise, we will validate the script using the **Start without debugging** approach. You can also choose **Start step by step** if you want to set breakpoints and execute the script step by step.

1. Click the **Start** icon and select **Start without debugging** to execute the bot script.

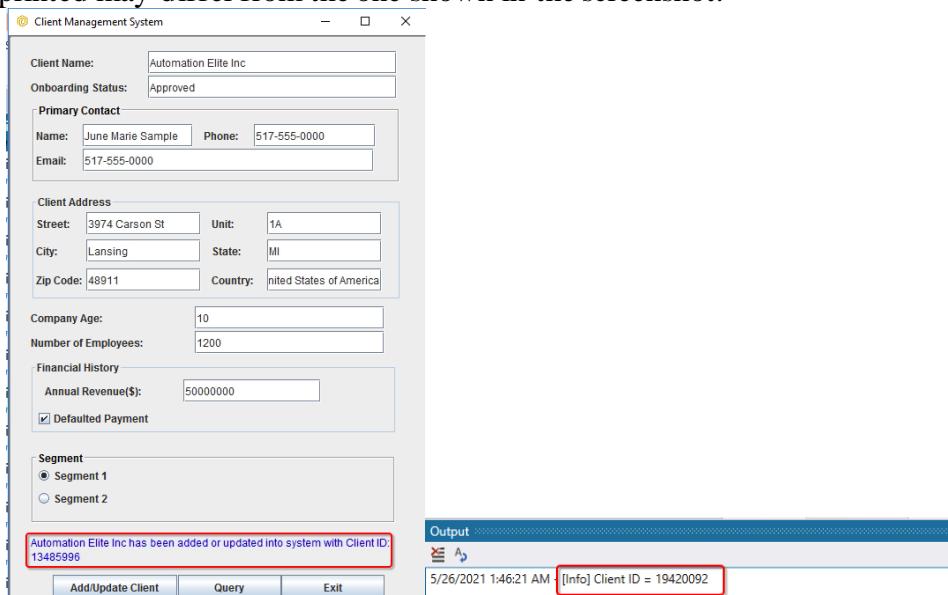
Note: In case you run into the issue of failing to start Firefox, this may be caused by Firefox is applying an update. Just start the bot again which should successfully launch Firefox then. You can also manually start Firefox to check and ensure the update finished.



2. It will retrieve the client information from the onboardingInfo JSON string first, then it launches the Client Management System Java application, and login to it.



- After logging into the Client Management System application, it will populate the client information and add the client into the backend system. Once done, it will retrieve the client ID and print it in the Studio output window. The actual client ID printed may differ from the one shown in the screenshot.



Summary

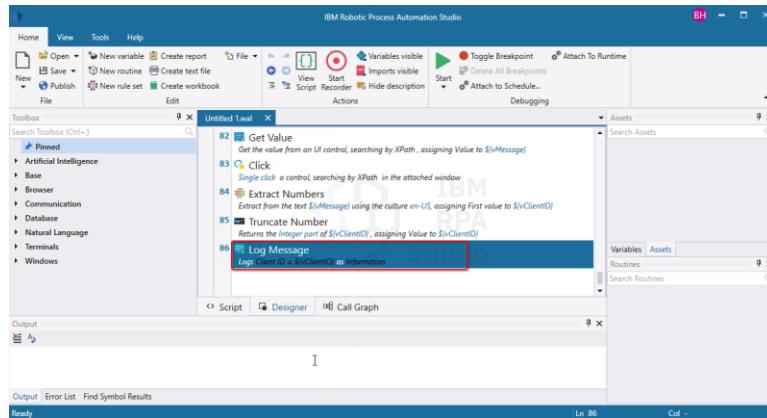
In this exercise, you have learned:

- How to use IBM RPA Studio to develop and test an automation script.
- How to use the IBM RPA Recorder to automate a Java Application by starting it, getting/setting UI control values, and clicking UI controls.
- How to use various other IBM RPA commands to automate your application by for example controlling the execution flow.
- How to extract the number from a given string and to truncate it to just the Integer part.

4.2 Exercise 2: Web Application Automation

This exercise will be performed on the RPA-VM VM 5 – RPA and takes about 45 minutes to complete.

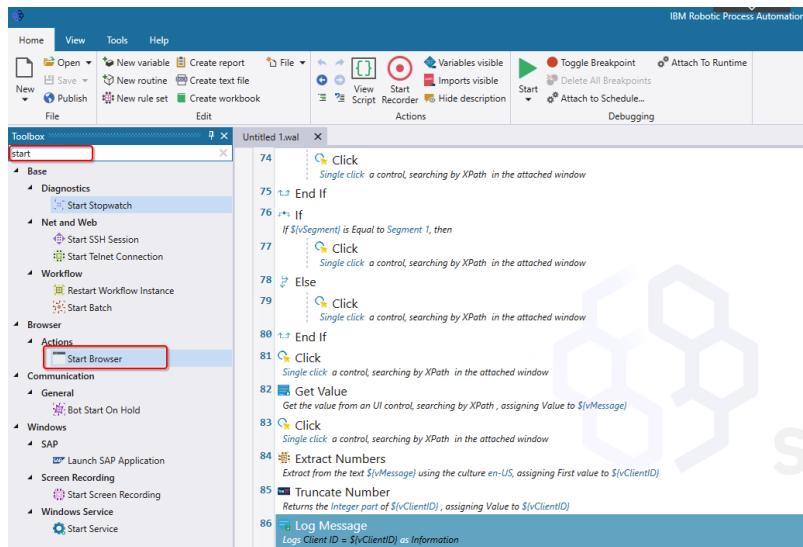
Please open the bot script created in exercise 1 in IBM RPA Studio if not yet open and go to the end of script. You will continue with the web application automation from there.



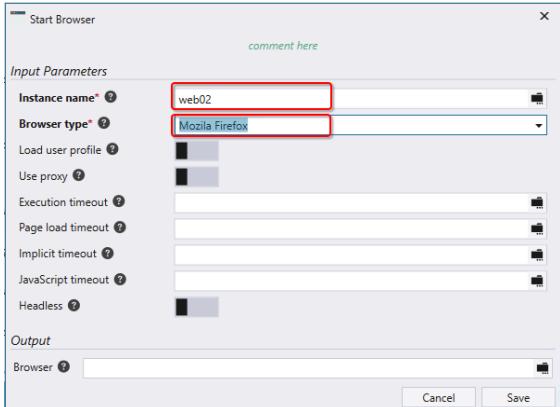
4.2.1 Develop Bot Script

The typical process to automate a web application is to start a web browser first, then navigate to the corresponding web page, capture the controls from the HTML document and take appropriate actions.

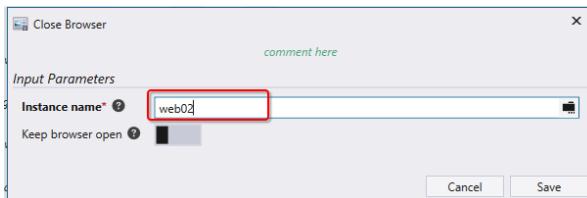
1. Enter **start** in the search toolbox, find and double click the command **Start Browser**.



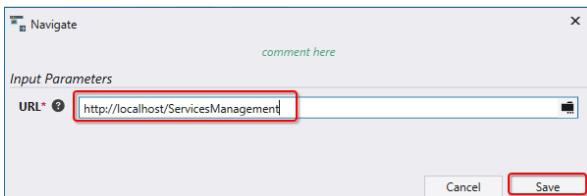
2. Configure the **Start Browser** command as below. For **Instance name**, enter **web02**. For **Browser Type**, select **Mozilla Firefox**. Once done, click **Save**.



- Once you click the **Save** button, Studio will automatically add a **Close Browser** command. For **Instance name**, please enter the same name you entered above – **web02**. Once done, click **Save**.



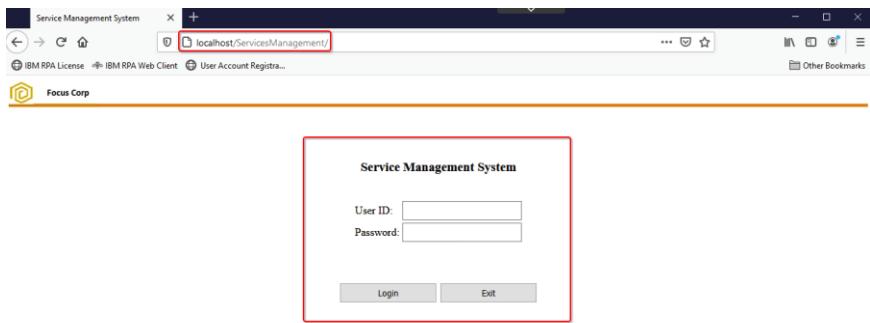
- Find the **Navigate** command, drag and drop it to before the **Close Browser** command and enter <http://localhost/ServicesManagement> in the URL field. Once done, click **Save**.



- Manually start Mozilla Firefox by clicking the  icon from the Windows toolbar.

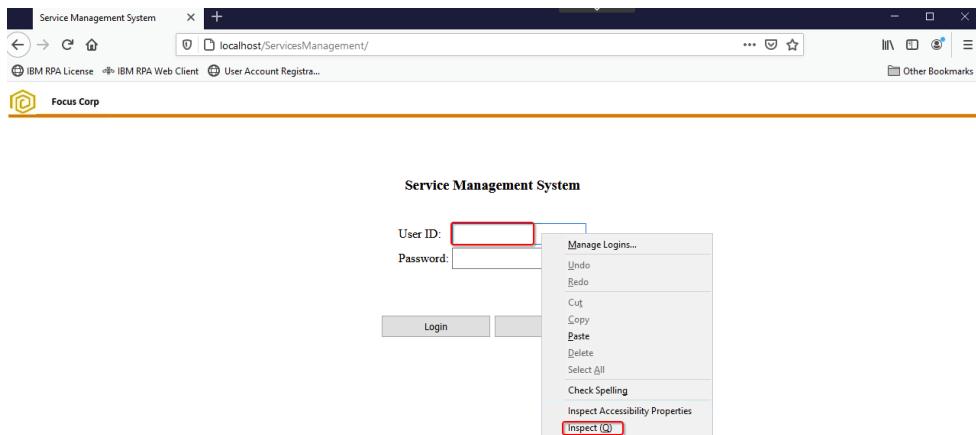


- Enter <http://localhost/ServicesManagement> to open the Service Management System solution login page.

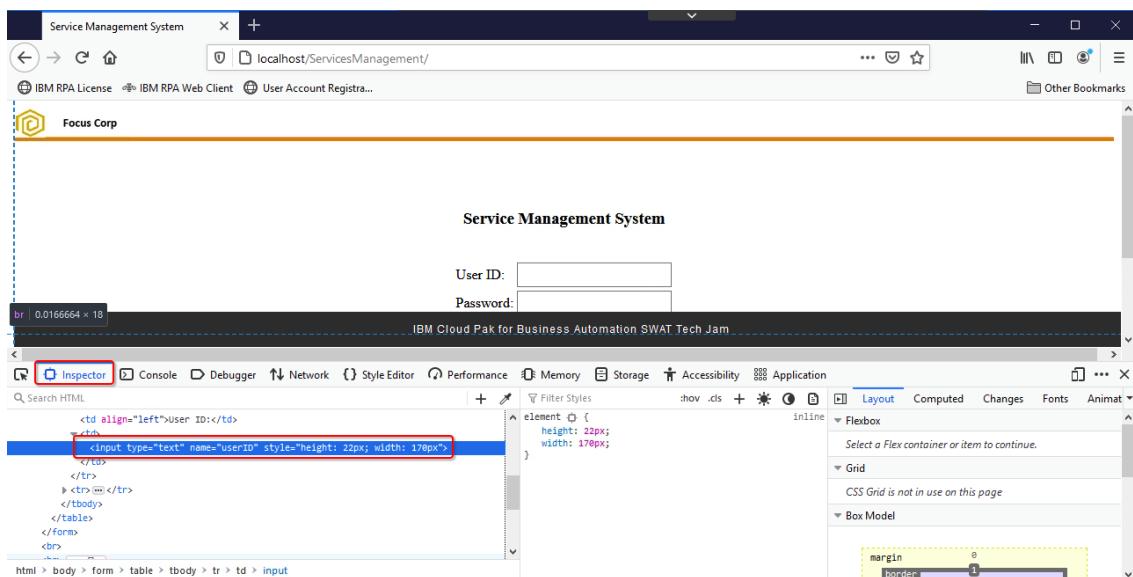


7. Automate the **User ID** field following below steps:

1. In Firefox on the **Service Management System** login page, put cursor focus on the User ID textbox, right click mouse, and select **Inspect** from the context menu.

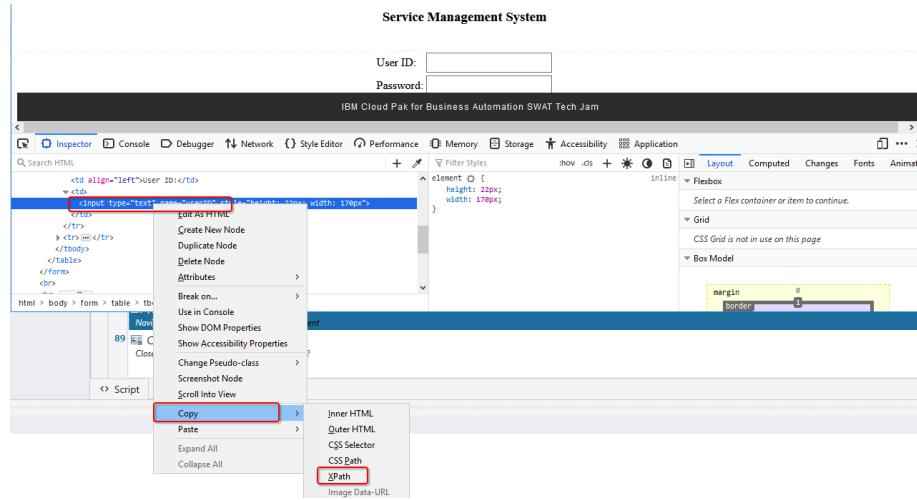


2. Firefox will open the Inspector window and highlight the User ID input textbox as below.

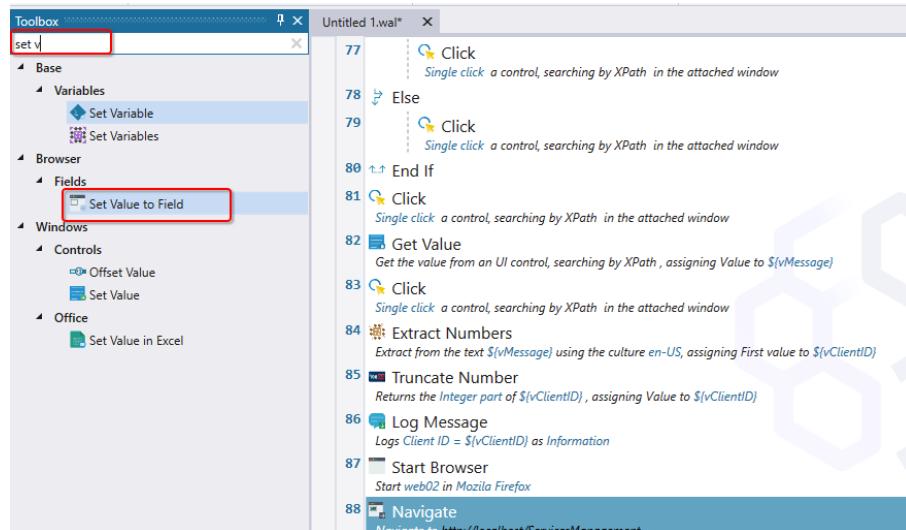


- Right click with the mouse on the User ID input textbox in the Inspector and select **Copy → XPath**. This will copy the XPath of User ID into the Windows clipboard.

Please note IBM RPA supports various different selectors, in a real project, you can choose different types of selectors based on the nature of your application. In this lab, XPath is used for illustration purposes.



- Switch to the IBM RPA Studio, find the **Set Value to Field** command from the **Browser** command category, drag and drop it before **Close Browser** command.



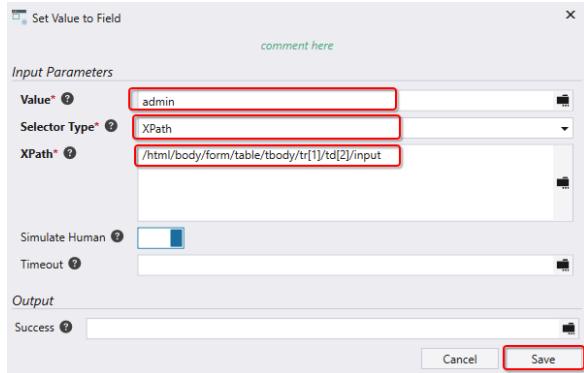
- Configure the **Set Value to Field** command as below. Once done, click **Save**.

Input Parameters:

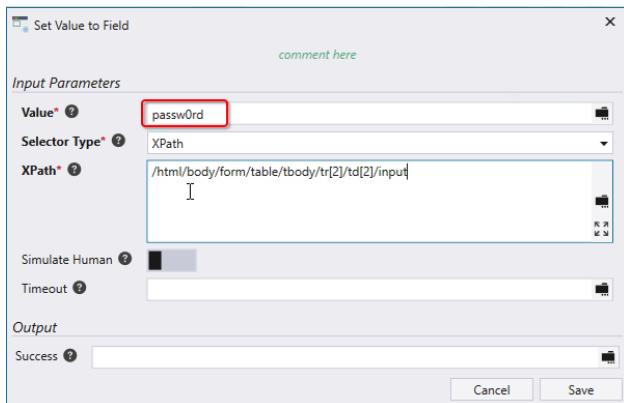
Value: Enter **admin**. This is the only user that can login to the Services Management System.

Select type: Select **XPath**.

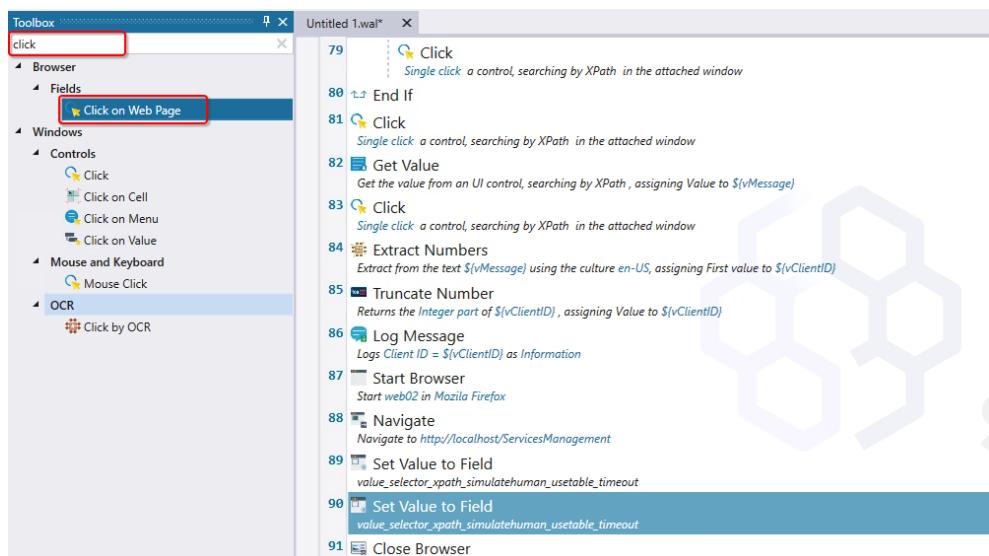
XPath: Paste the XPath in the clipboard that you copied in above step in Firefox.



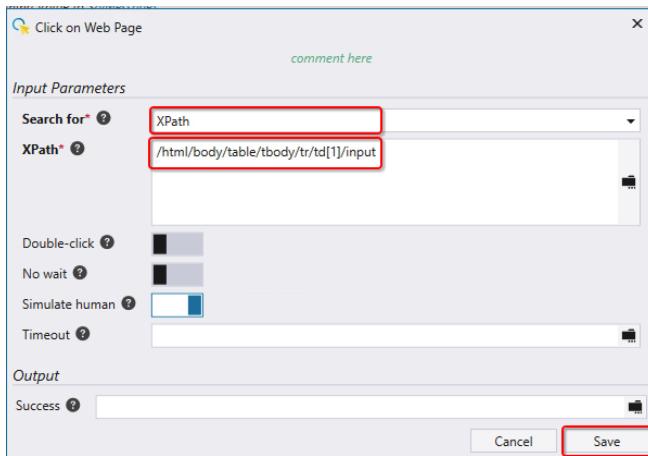
- Follow the same approach above to automate the password field. Please use **passw0rd** (with a zero) in the **Value** field when you configure the **Set Value for Field** command. This is the only password that can log into the Services Management System web application.



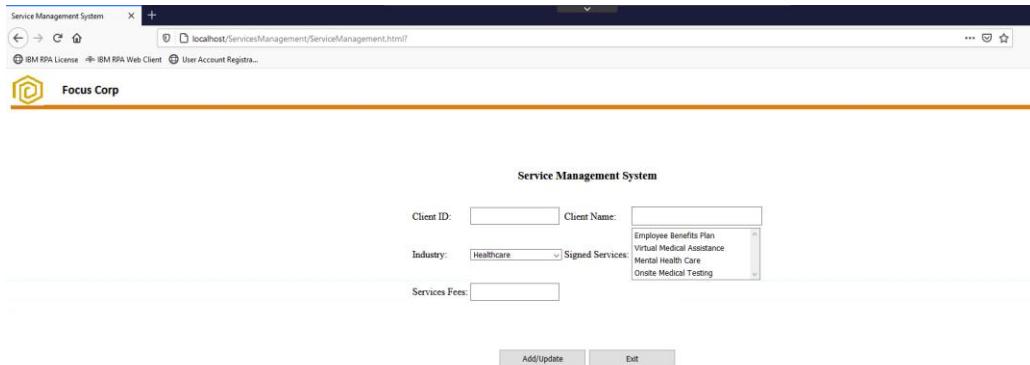
- Automate clicking **Login** button following below steps.
 - Follow the same approach to copy the XPath of the Login button.
 - Switch to the RPA Studio, find the **Click on Web Page** command, drag and drop it before the **Close Browser** command.



3. Configure the **Click on Web Page** command as below. Once done, click **Save**.



10. After logging into the Services Management System, it will navigate to the next page. Switch to the Firefox window, **manually login** to the Services Management System using **admin/passw0rd**. Once done, you should get below page in Firefox.



11. Follow below steps to automate filling in the **Client ID** field. This is the first field you are going to automate after switching to this page. Since the robot executes very fast, it may take some time to load the entire page before you can work with it. A best practice to make your script more robust is to check if the corresponding control is showing up on the web page before you actually work with it.

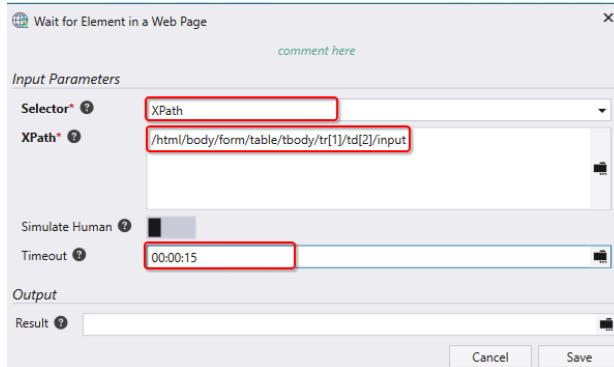
1. Copy **XPath** selector for **Client ID** textbox from Firefox.
2. Find the **Wait for Element in a Web Page** command. Drag and drop it before the **Close Browser** command and configure it as below. Once done, click **Save**.

Input Parameters:

Selector: Select **XPath**.

XPath: Paste the XPath you copied from above step.

Timeout: In this lab, please enter **00:00:15** to wait up to 15 seconds. The command will stop waiting as soon as the control becomes available. The time you have to set depends on the complexity of your web page and the network latency.



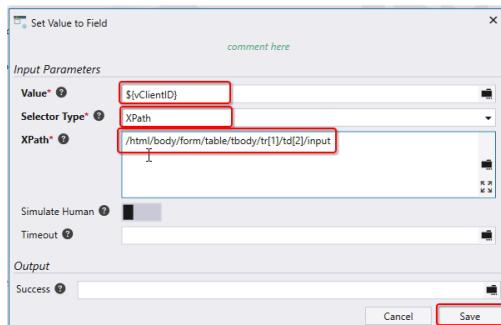
3. Drag and drop the **Set Value to Field** command before the **Close Browser** command and configure it as below. Once done, click **Save**.

Input Parameters:

Value: Select variable **vClientID**.

Select Type: Select **XPath**.

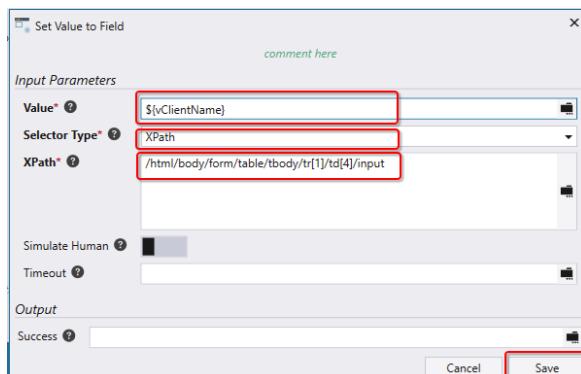
XPath: Paste the XPath you copied above.



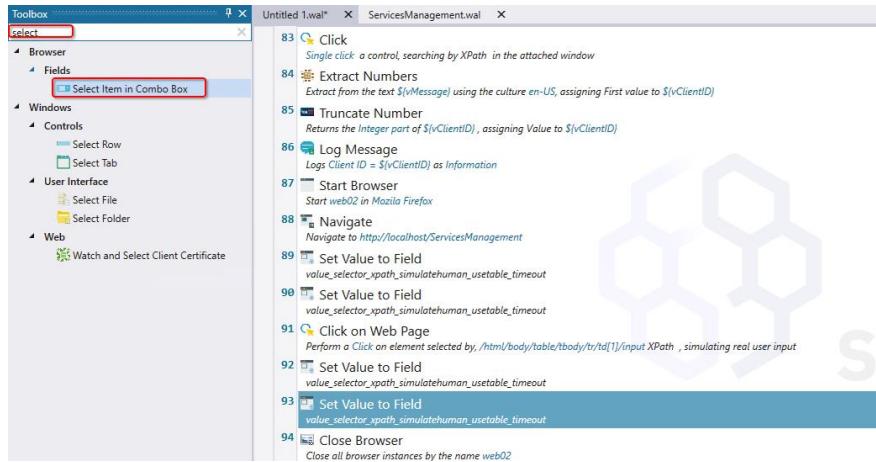
12. Automate the **Client Name** field.

1. Copy the XPath selector for **Client Name** textbox from Firefox.
2. Drag and drop the **Set Value to Field** command before the **Close Browser** command and configure it as below.

For the **Input parameters** select variable **vClientName** in the **Value** field. As **Selector Type** select **XPath** and paste the XPath of the field into the **XPath** field. Once done, click **Save**.



13. Automate the selection of the Industry field,
1. Copy the XPath selector for **Industry** combo box.
 2. Find the **Select Item in Combo Box** command and drag and drop it before the **Close Browser** command.



3. Configure the **Select Item in Combo Box** as below. Once done, click **Save**.

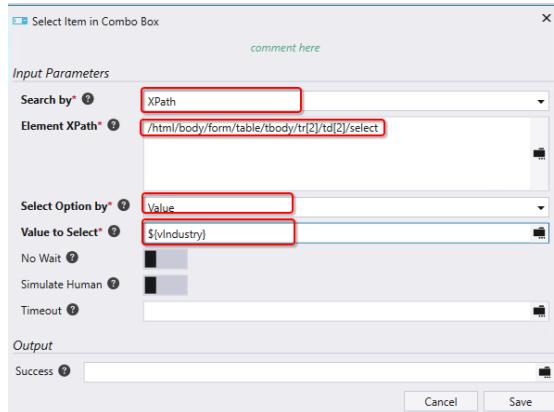
Input Parameters

Search By: Select **XPath**

Element XPath: Paste the XPath you copied above

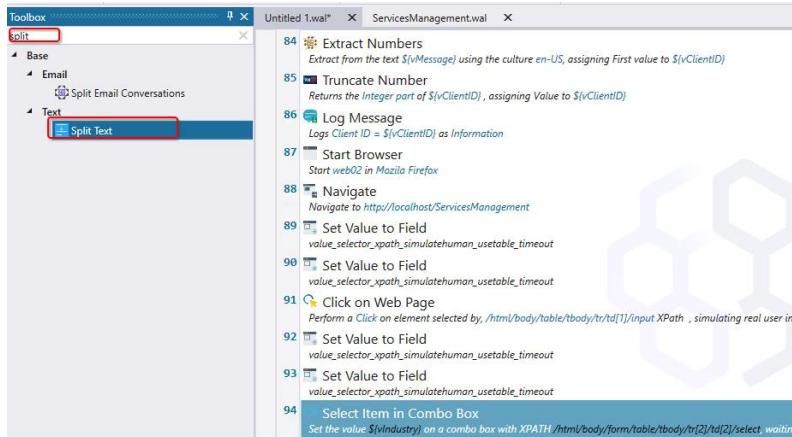
Select Option by: Select **Value**

Value to Select: Select variable **vIndustry**



14. Automate the selection of the Signed Services multi-select field. The client may sign multiple services. Those services are stored as a string in variable **vSignedServices** separated by a comma. You need to first split them into individual services and then make the selection.

1. Find the **Split Text** command and drag and drop it before the **Close Browser** command.



2. Configure Split Text command as below. Once done, click **Save**.

Input Parameters:

Text to split: Select variable **vSignedServices**.

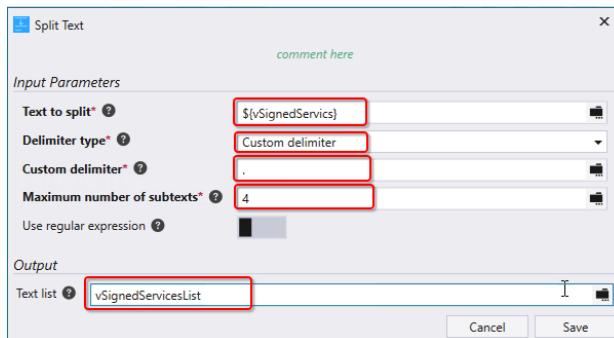
Delimiter type: Select **Custom delimiter**.

Custom delimiter: Enter comma (,).

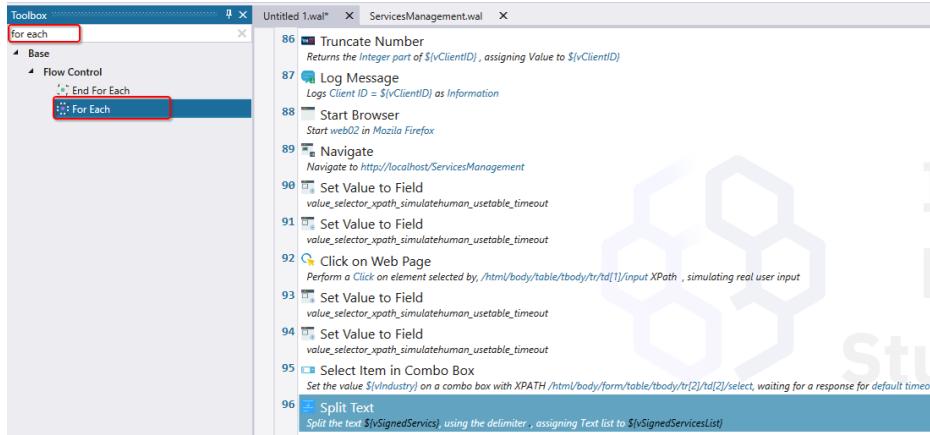
Maximum number of subtexts: Enter 4 since there are up to 4 services for each industry that can be selected in the client onboarding application.

Output:

Text List: Click icon on the right, select **vSignedServicesList** variable.



3. Find the **For Each** command and drag and drop it before the **Close Browser** command.

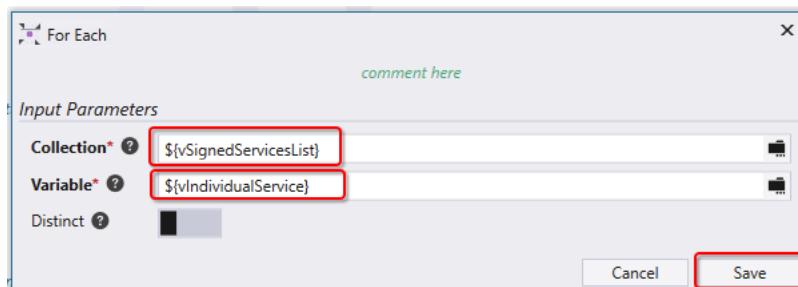


- Configure the **For Each** command as below. Once done, click **Save**.

Input Parameters:

Collection: Select the variable **vSignedServicesList**.

Variable: Select the variable **vIndividualService**.



- Switch to **Firefox** and copy the XPath selector for the **Signed Services** combo box. Make sure to select the combo box itself not one of its options.
- Switch back to IBM RPA Studio. Find the **Select Item in Combo Box** command and drag and drop it between the **For Each** and **End For Each** commands.



- Configure **Select Item in Combo Box** as below. Once done, click **Save**.

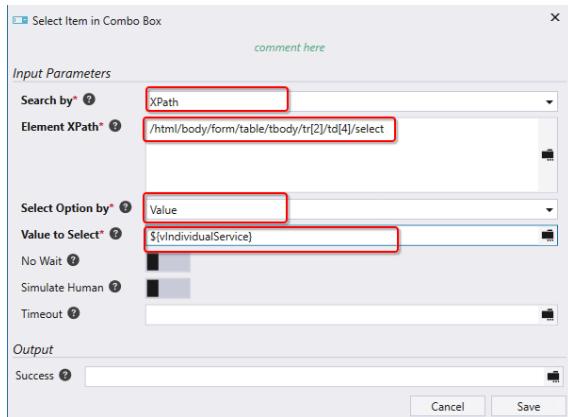
Input Parameters:

Search by: Select **XPath**.

Element XPath: Paste the XPath copied from above step.

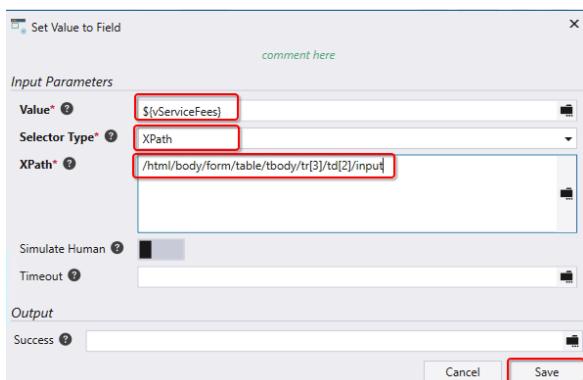
Select Option by: Select **Value**.

Value to Select: Select the variable **vIndividualService**.



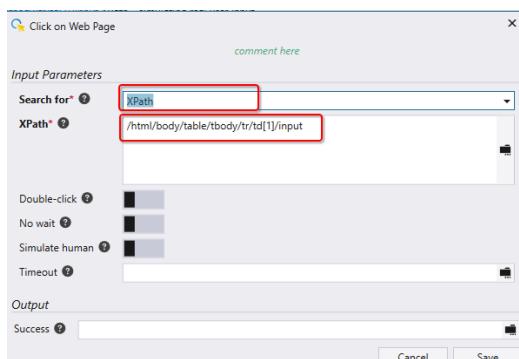
15. Automate the **Services Fees** field following below steps.

1. Copy the XPath selector for **Services Fees** textbox from Firefox.
2. Find the **Set Value to Field** command, drag and drop it before **Close Browser** command.
3. Configure the **Set Value to Field** command as below. Once done, click **Save**.



16. Automate clicking **Add/Update** button to record signed services following below steps.

1. Copy the XPath selector for the **Add/Update** button from Firefox.
2. Find the **Click on Web Page** command and drag and drop it before the **Close Browser** command.
3. Configure the **Click on Web Page** command as below. Once done, click **Save**.



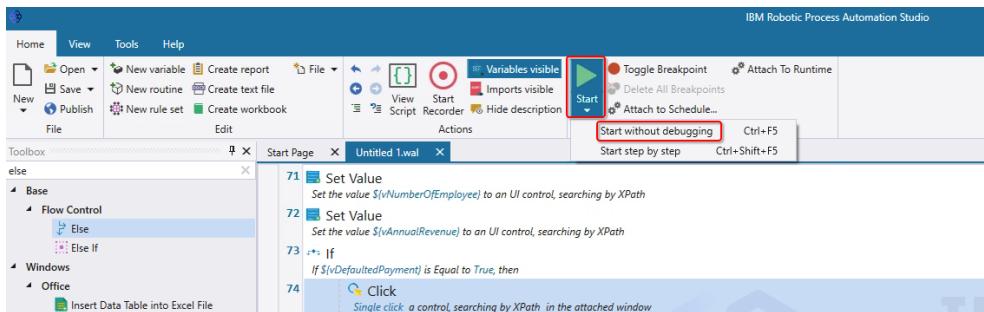
17. Since the bot executes very fast, for demo purpose, you can intentionally add a **Delay** command to the check execution result as below before closing the browser. It will delay the execution for 5 second. Once done, click **Save** button from Studio toolbar to save your script.



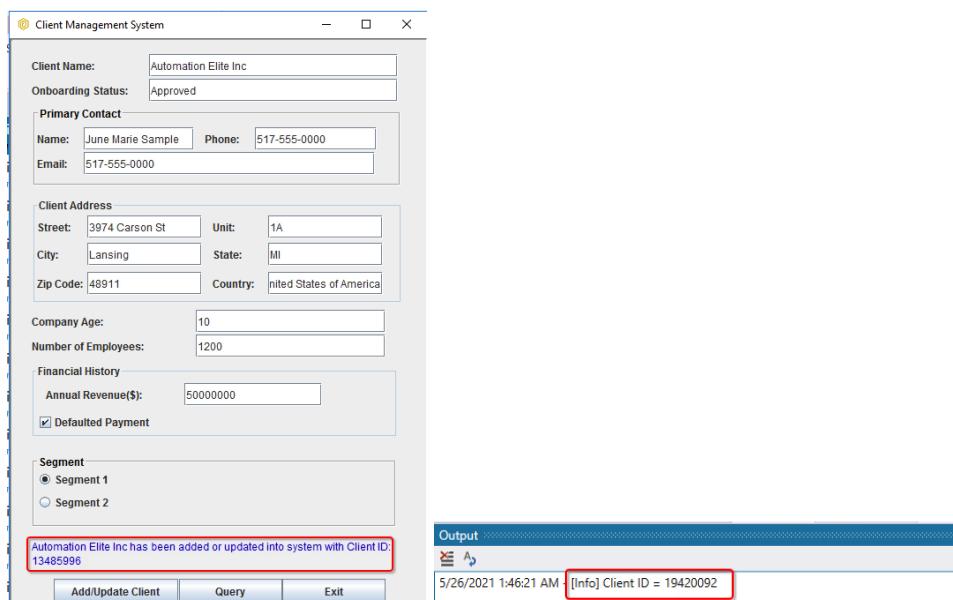
You have successfully automated a web application using IBM RPA. The bot first starts Mozilla Firefox and logs into Services Management System. After that it adds the client information and the signed services information before it saves these changes. Finally, it closes the browser.

4.2.2 Verification Instructions

- Click the **Start** icon and select **Start without debugging** to execute the bot script.



- The bot script will be executed, it will launch the Client Management System Java application and add the client information first. Once done, it will retrieve the client ID from the result message.



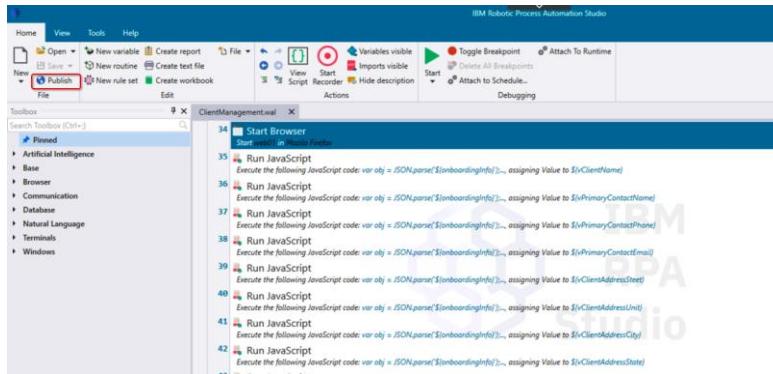
- The bot will then start Mozilla Firefox and login to the Services Management System web application. It will add the client information and the signed services information. Once done, it will close the browser window.

The screenshot shows a form titled "Service Management System". It has fields for "Client ID" (12574039), "Client Name" (Automation Elite Inc), "Industry" (Telecom), "Signed Services" (a dropdown menu showing "Fibre Internet", "Tieline Solution", "Business Devices", and "Communication Equipment", with "Tieline Solution" selected), and "Services Fees" (25000). Below the form, a message says "Signed services have been added or updated successfully." At the bottom are "Add/Update" and "Exit" buttons.

4.2.3 Publish Script to Tenant

You have validated your script, let's publish it to tenant now.

- Click the **Publish** button from the Studio menu toolbar.



- In the Publishing script window, enter appropriate values as below. Once done, click **Publish**.

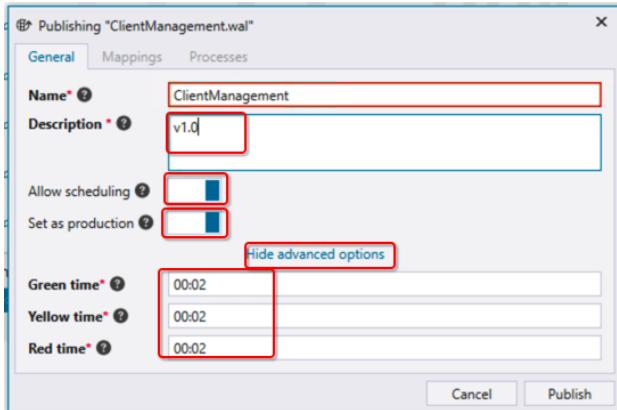
Name: To avoid naming conflict with others on the shared RPA tenant, please use unique name for your script (e.g. including your initials). In case a script with the same name already exists on the tenant, it will show a warning message when you click the Publish button and request you to enter a different name.

Description: This can't be empty. You can enter some meaningful description here, for example – version number etc.

Allow Scheduling: Indicates if you can schedule this script to be run by a scheduler.

Set as production: Publishing a script to a tenant will generate a new version every time. Multiple versions of scripts can exist on tenant, but only one can be the production version. If you execute the script without specifying the version, it will

use the production version by default. Click **Set as production** to set it as production version.



Your script should have been published to the tenant successfully.

Summary

In this exercise, you have learned:

- 1 How to use various IBM RPA commands to automate a web application.
- 2 How to split a string containing separator delimited entries into a collection of individual strings.
- 3 How to loop over all items in a collection.
- 4 How to publish script to tenant.

Congratulations, you have successfully completed this Lab!!!