IBM Cloud Pak for Business Automation Demos and Labs

Application Automation using IBM RPA

V 3.0

Bu Feng Hou
houbf@cn.ibm.com
Paul Pacholski
pacholsk@ca.ibm.com
Olaf Hahnl
olaf.hahnl@de.ibm.com
Vinicius Dutra
v.dutra@ibm.com
Aldo Justiniano
aldo.justiniano@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 referenceto 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

Table of Contents

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.

With IBM RPA, IBM can provide customers with additional benefits:

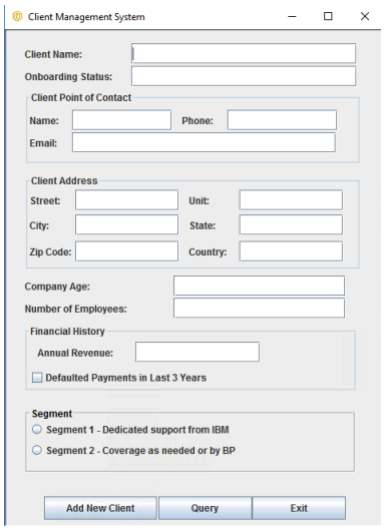
- **Faster time to value**
  Speed and simplicity of purchasing and deploying through easier licensing.
- **A comprehensive platform to automate all types of use cases**
  Tighter integrations between RPA and the rest of IBM business automation platform.
- **Automate business and IT processes**
  Expand the IBM business automation mission to IT use cases.
- **Operationalize AI**
  Fulfill IBM's vision of operationalizing AI in every aspect 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** simulating 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** simulating a web-based enterprise services management system that does not provide public APIs.

2.1 Pre-requisites

For this lab, you need to reserve an **IBM Robotic Process Automation** environment from IBM Technology Zone (see chapter 3). 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 v23.0.x.

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

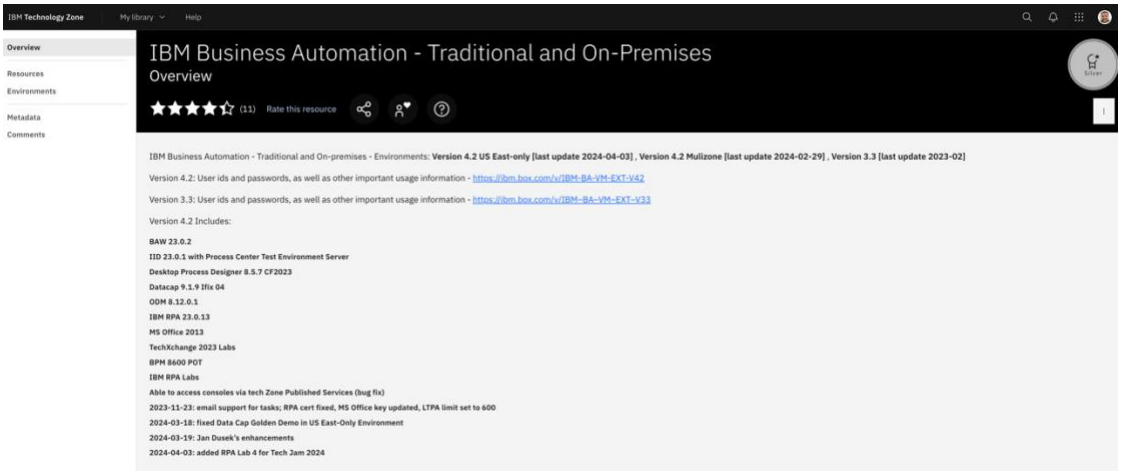
3 Accessing the Environment

If you have already reserved a lab environment from IBM Technology Zone, please go to Chapter 5 directly.

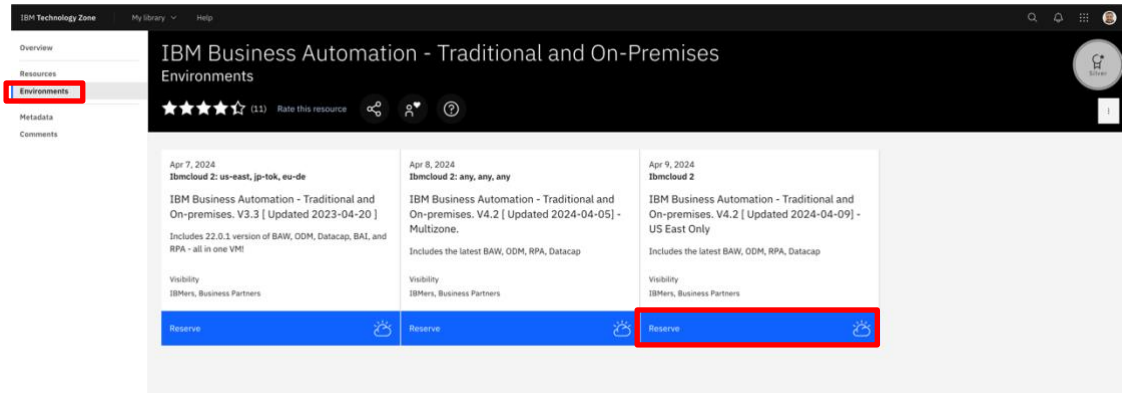
Reserve Environment

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

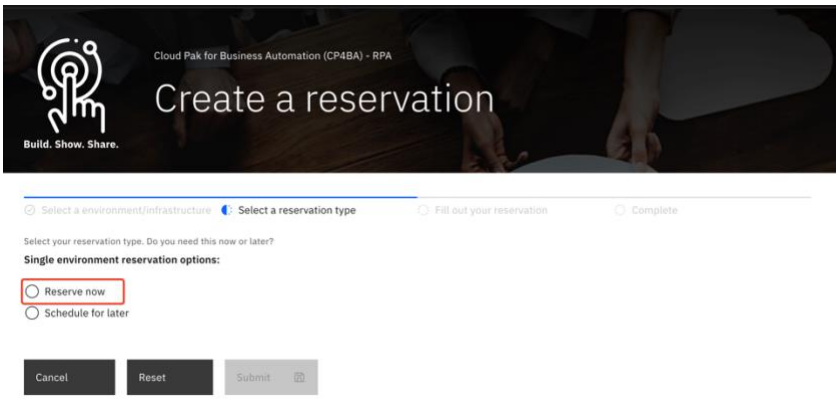
1. Click here to open IBM Technology Zone Reservation portal. You need to use your IBMID to login to the portal.



2. Click **Environments** on the left panel, and then reserve the last environment on click the blue button.



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



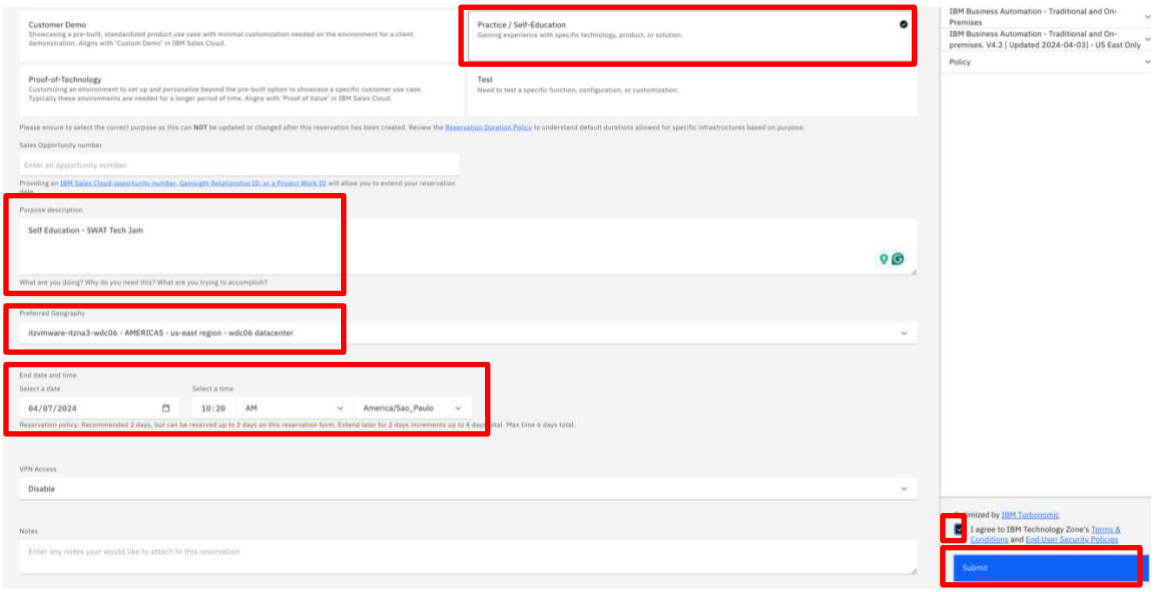
4. On the reservation page, make the appropriate selections as below. Once done, click **Submit**.

   Purpose: Select Practice/Self-Education.

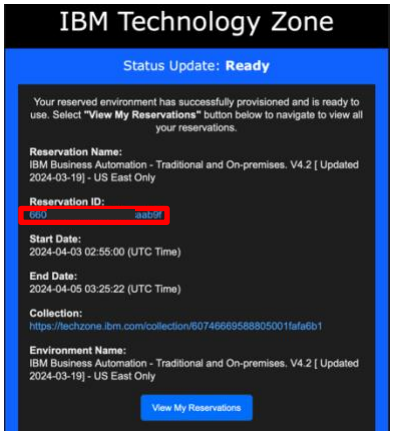
   Purpose description: Enter something like **Self Education**.

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

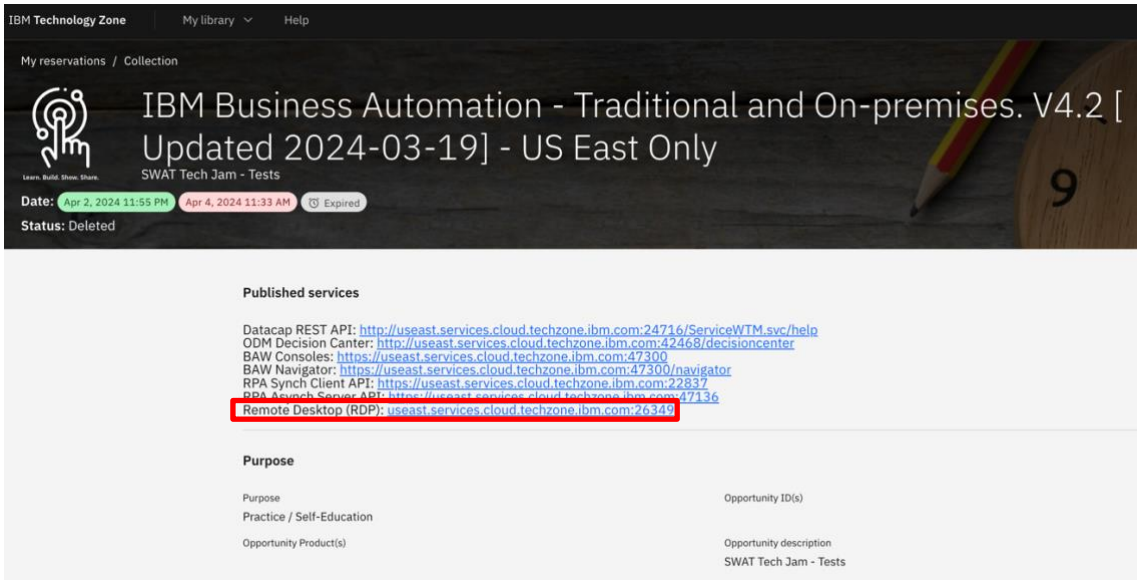
   Preferred Geography: Select the geography where your environment will be created. To get a better network connection, select the same geography as where you are located in.

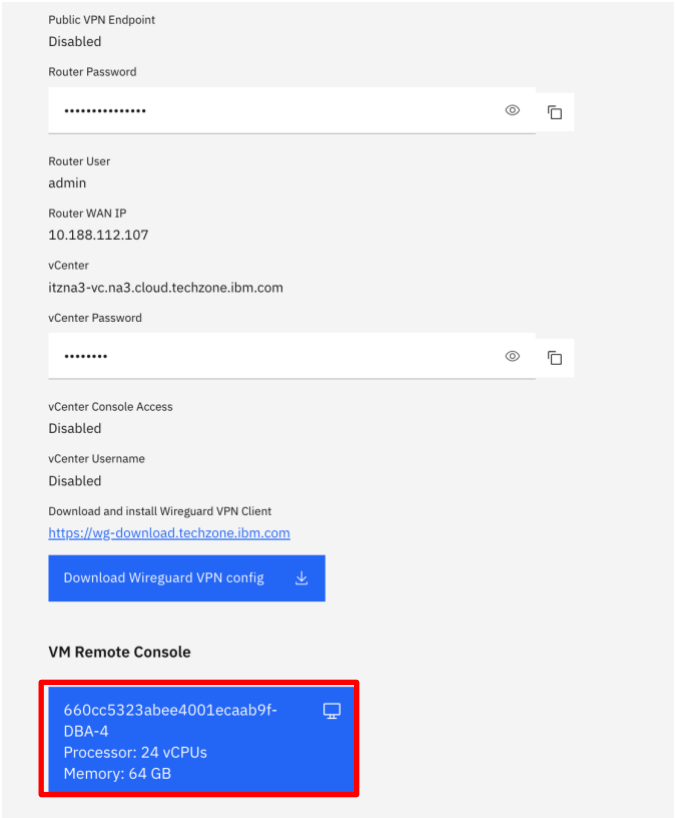


5. Once you have reserved an environment, you will receive an email with a link to access the environment's management console, click on Reservation ID.

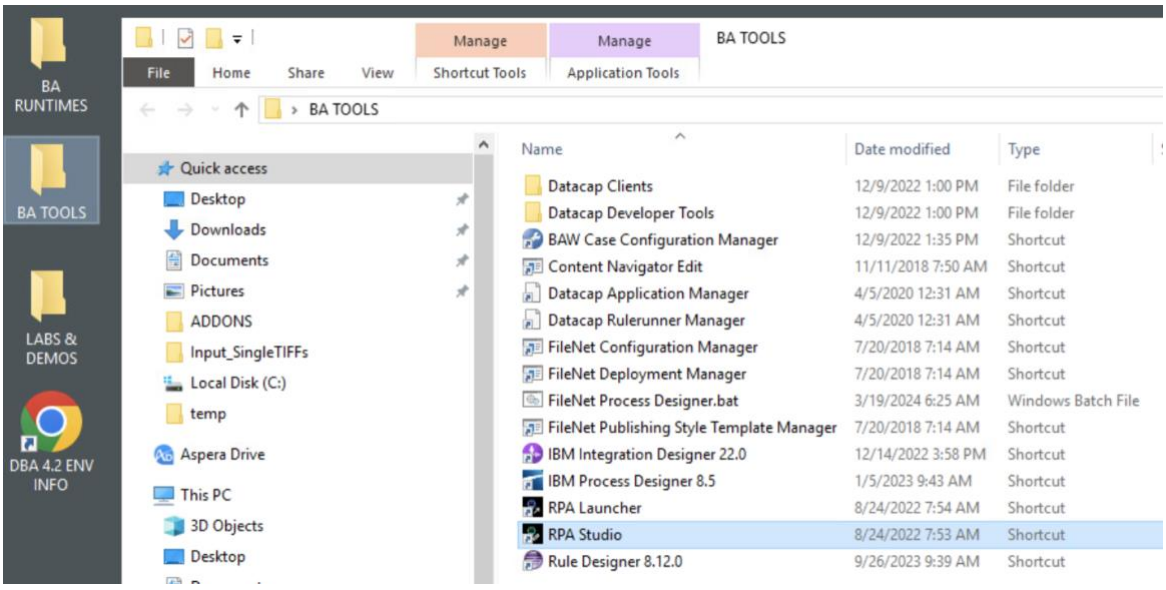


6. You can access the environment using Remote Desktop (RDP) or Remote Console (Web). Our recommendation is to use Remote Console (Web) for practicality. If you prefer to use the RPD, use the Remote Desktop (RDP) link, or keep rolling the page to access the Remote Console (Web).





7. After waiting for the VM to load, open the folder BA TOOLS on the Desktop to access the IBM RPA Studio.



4 Build it yourself – Step-by-step instructions

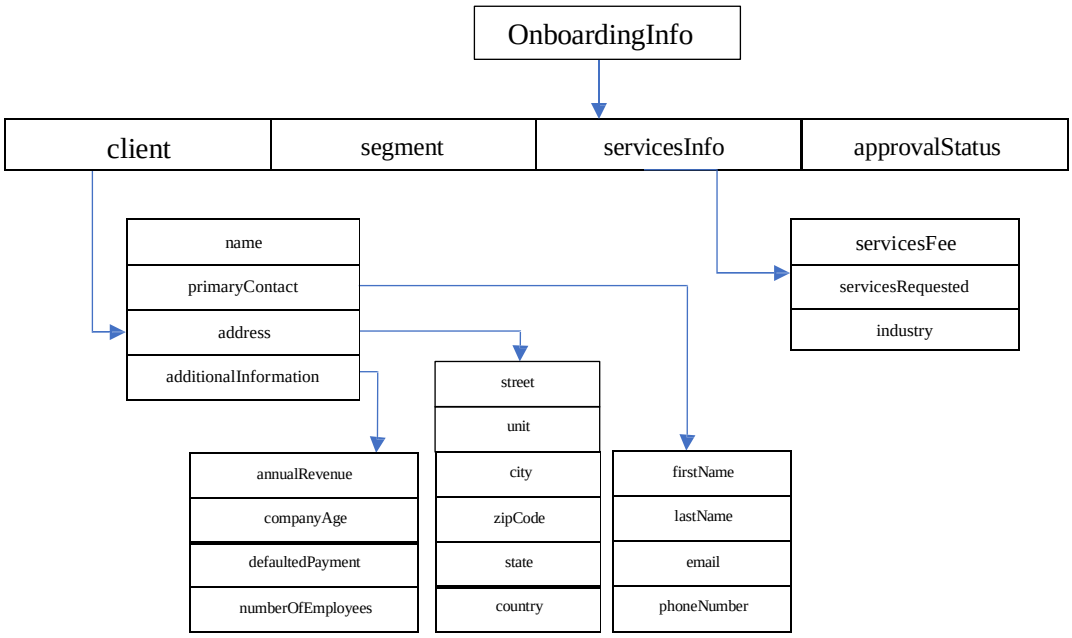
4.1 Exercise 1: Java Swing Application Automation

This exercise will take 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 the bot as an input parameter as a JSON string.

You need develop a bot script to retrieve the client information from the JSON string first. Then start the **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 information about the signed services. You will do this as part of exercise 2.

The client information data model is defined as below

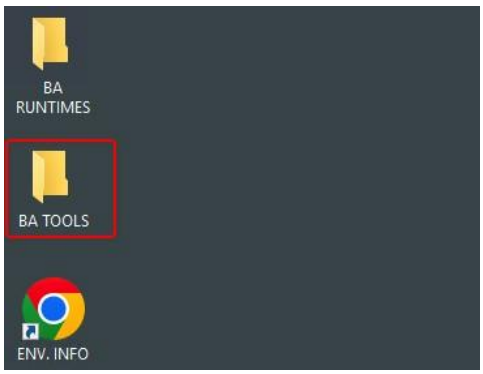


An example JSON string looks like below. It will be used to develop and test the bot script in this lab.

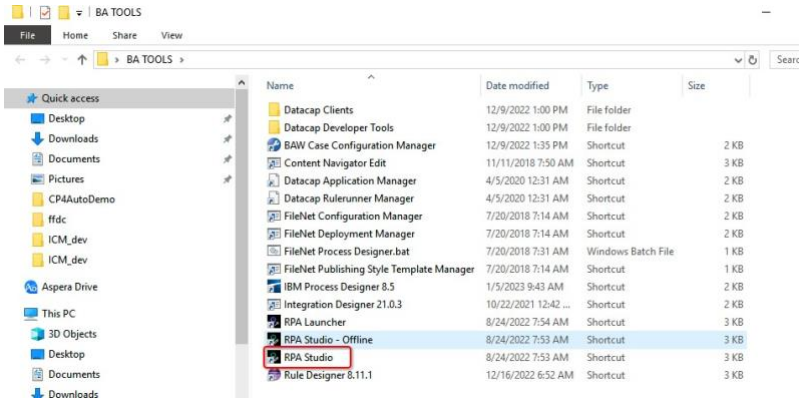
*{ "**servicesInfo**":{"servicesRequested":"Fibre Internet", "servicesFee":25000, "industry":"Telecom"}, "**approvalStatus**":"Approved", "**segment**":"Segment 1", "**client**":{"additionalInformation":{"defaultedPayment":true,"companyAge":10,"annualRevenue":500000 00,"numberOfEmployees":1200},"address":{"zipCode":"48911","country":"United States of America", "unit":"1A","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

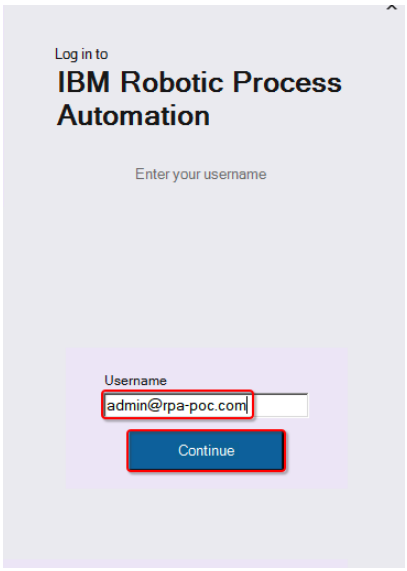
1. Access your environment VM using remote desktop if not yet, click **BA TOOLS** icon on the windows desktop.



2. **Start** the **IBM RPA Studio** by clicking the **RPA Studio** shortcut.

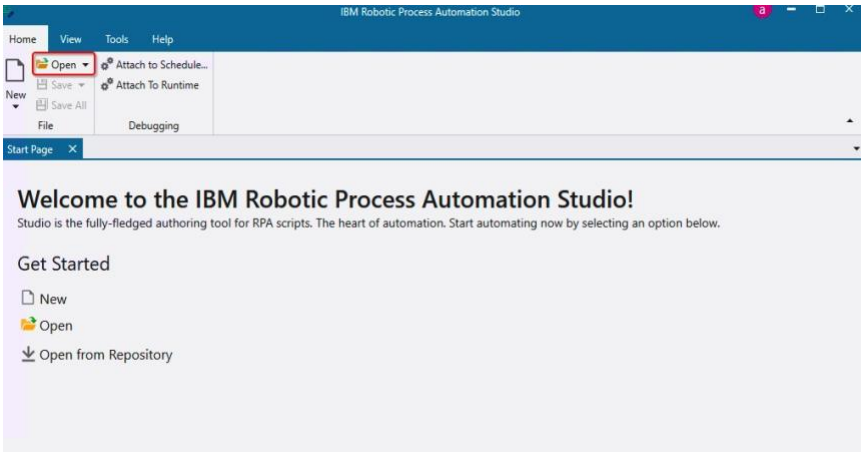


3. Enter **admin@rpa-poc.com** as Username and click **Continue**.

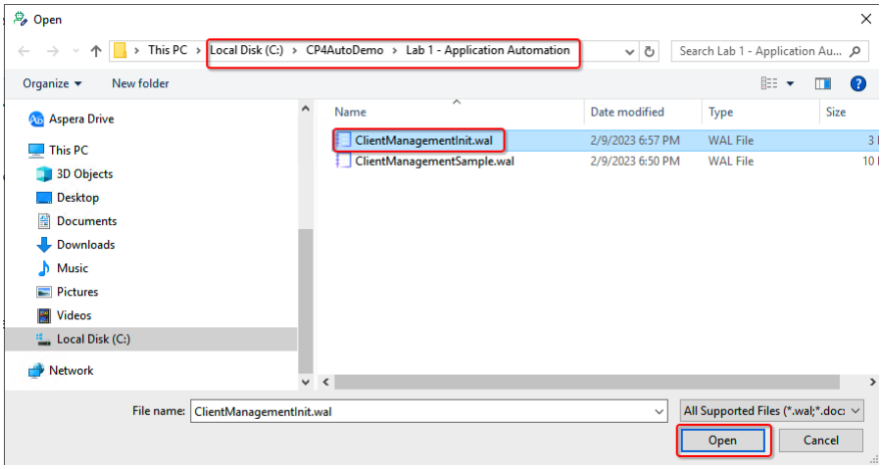


4. Since there is only one tenant **rpa-poc** in the environment, it is set as default automatically to Tenant. Enter **passw0rd (use a zero not a capital o)** as password. Click **Login** to start and login to Studio.

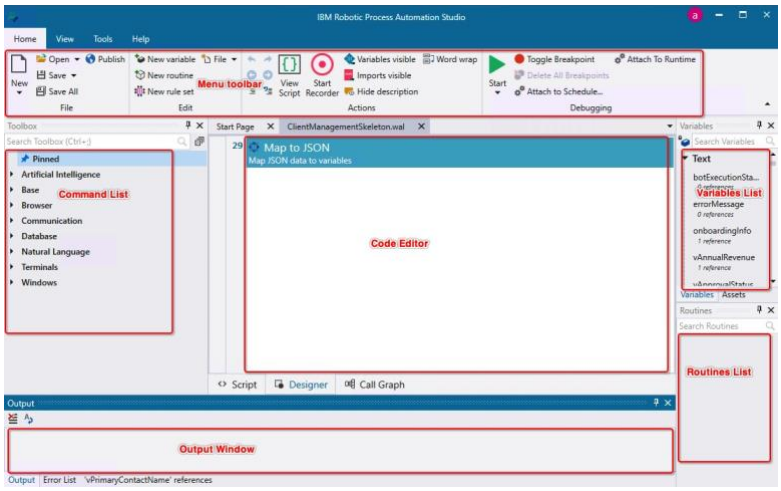
5. We will start from an init script which has all the variables required by this lab defined. Click **Open**.



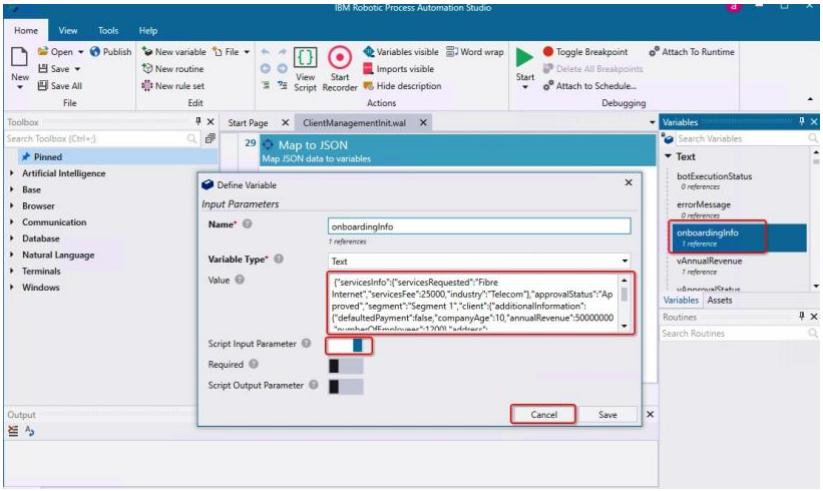
6. Select **C:\CP4AutoDemo\Lab 1 – Application Automation\ClientManagementInit.wal** and click **Open**, which will open the IBM RPA Studio script development perspective.



7. 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 an 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 messages and errors, if any.



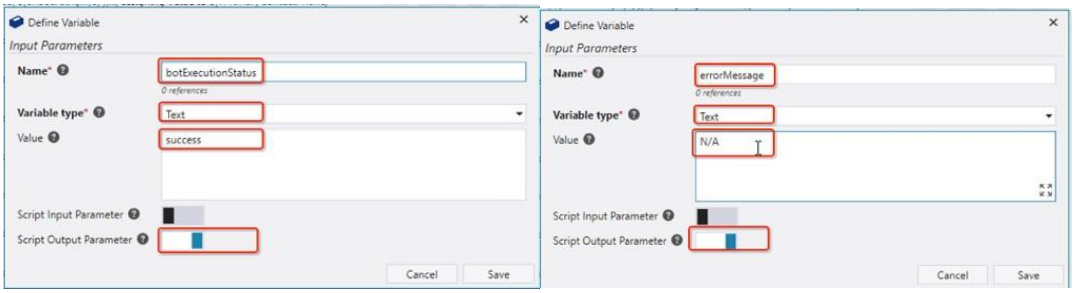
8. Init script has defined all the variables that are required in this lab, let's review a few key variables first. Find **onboardingInfo** from right Variables windows, double click to open it. It is defined as **Script Input Parameter** with a default value of client onboarding information in JSON format, then click **Cancel** to close it.



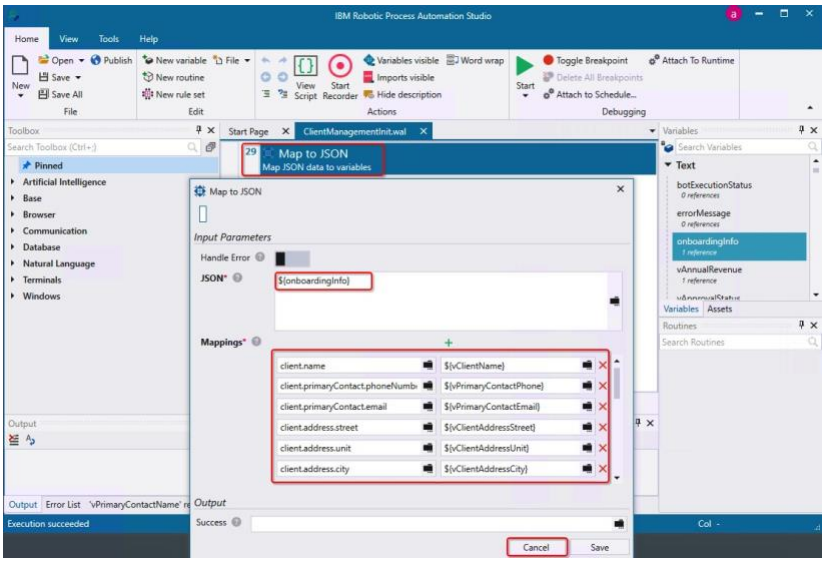
9. Follow the same steps to review another **two output variables**:

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

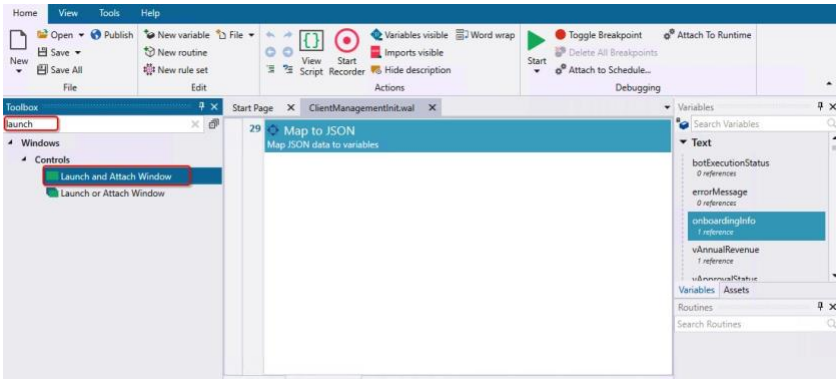
   In an real project, the bot must handle its execution failures and exceptions and set appropriate status code and error message accordingly. In this lab, the bot will simply return the default value set here back to the caller. You can find more information about handling exceptions in IBM RPA in the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



10. Client onboarding information is stored in **onboardingInfo** variable in JSON string, **Map to JSON** command is used to retrieve property value and store to corresponding variables. Double click **Map to JSON** command to review its configuration. Once done, click **Cancel** to close the window.



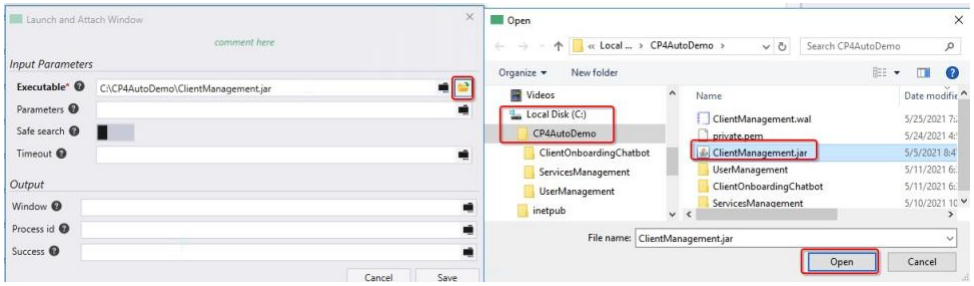
11. Enter **Launch** in search toolbox field, find and double click **Launch and Attach Window** command as we next need to launch Java swing application first.



12. 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 blank

Safe Search: Unchanged

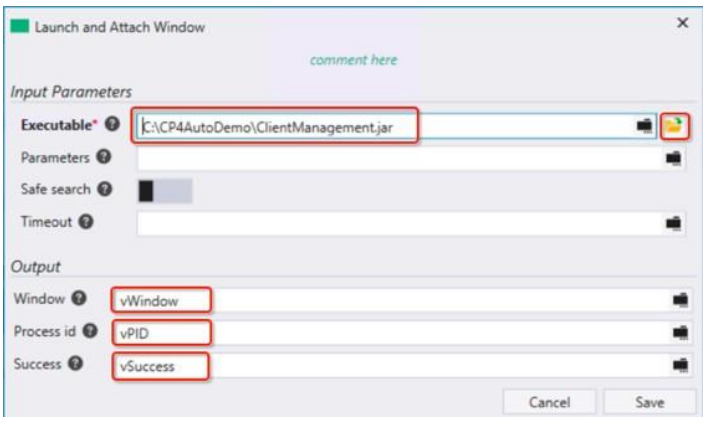
Timeout: Unchanged/leave blank

Output:

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

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

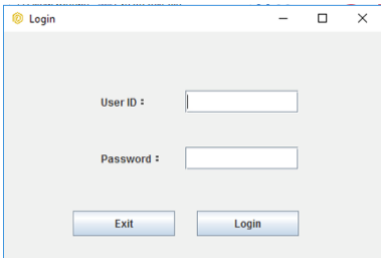
Success: Click the icon on the right, select the **vSuccess** variable



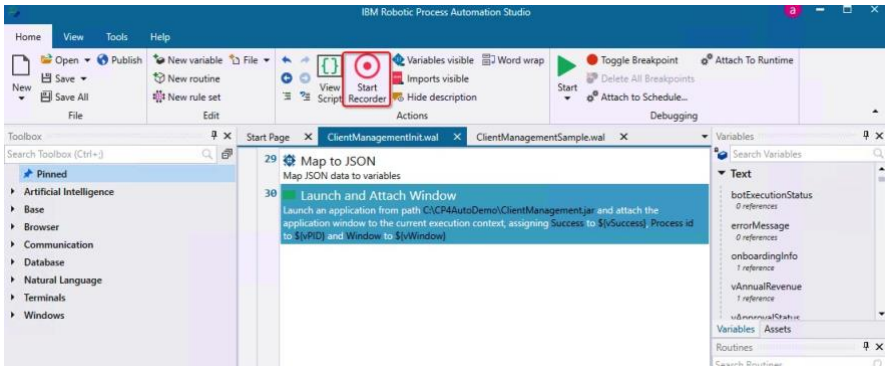
Next, we will use the Recorder to automate the Java application. First, you need to manually start the Java application.

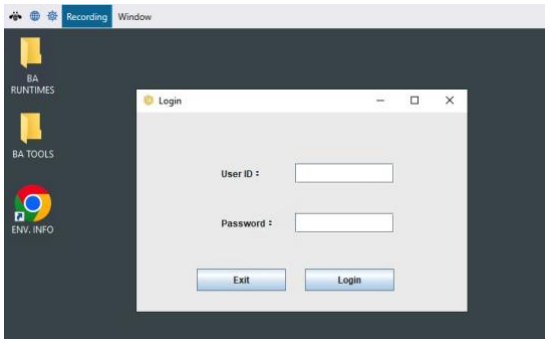
13. Double click **ClientManagement.jar** in the **C:\CP4AutoDemo** directory to start the Client Management System application.

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



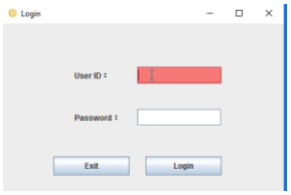
14. Click **Start Recorder** from the Studio toolbar. This will open the recorder. Once started, you should see the recording window as below.





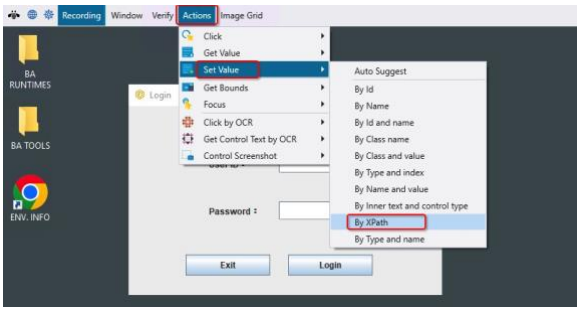
15. Automate the login to the Client Management System application

    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**.

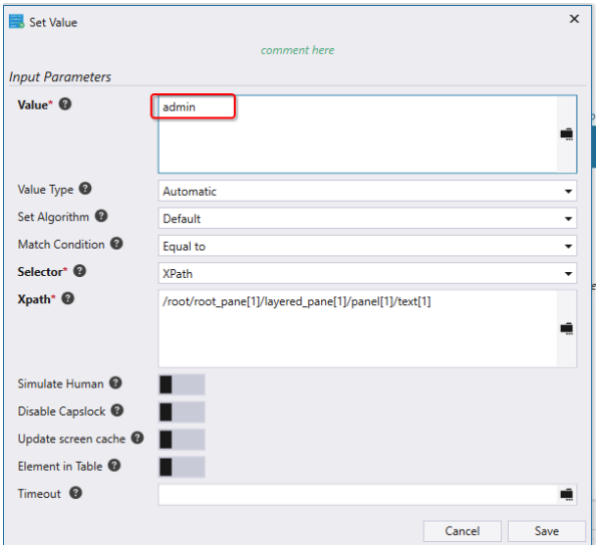


    2. Select **Actions ➔ Set Value ➔ By XPath** from the Recorder toolbar menu.

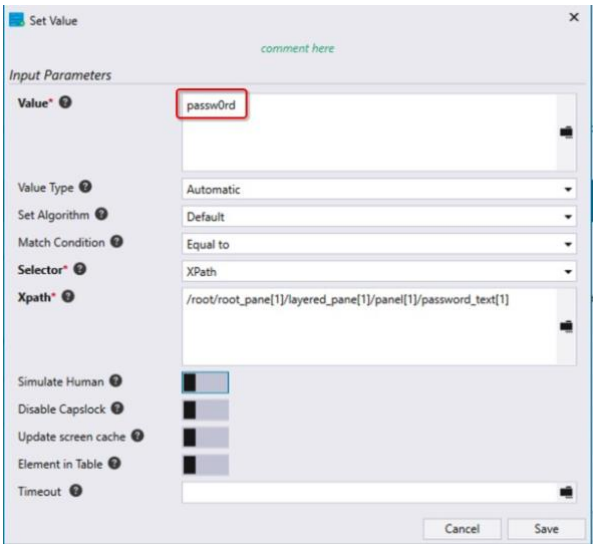
       Note: Since the textbox control has been captured, if you notice the red highlighting goes away when going through the menus, this is the expected behavior.



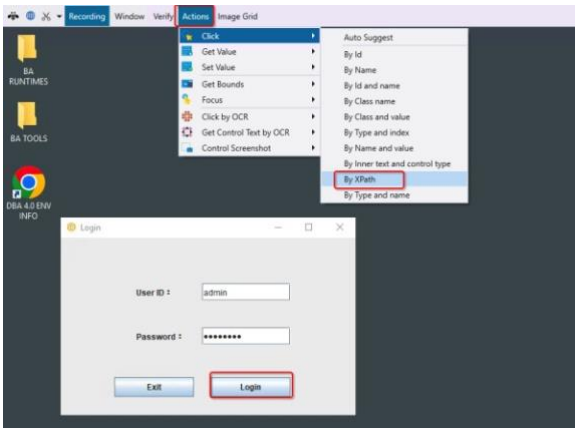
3. Configure the **Set Value** command as below. Please enter **admin**, which is the only user that can log in to the client management system application. Once done, click **Save**.



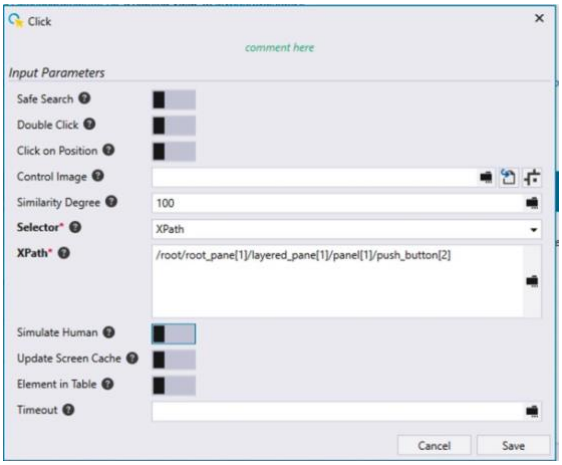
4. Follow the same steps to automate the Password field. Enter **passw0rd** (make sure to use a zero as part of the password), which is the only valid password that can be used to login.



5. Follow the same steps to automate the Login button by selecting **Actions ➔ Click ➔ By XPath.**

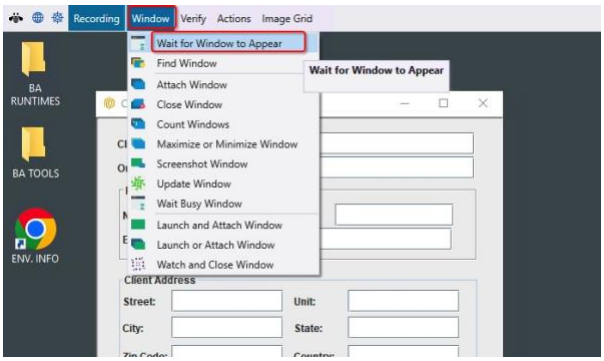


You needn't make change on **Click** command configuration, just click **Save,** which will finish the Login page automation, and result in the main window of the Client Management System to be shown.



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 machine's performance, 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 the current execution context.

16. Select **Window ➔ Wait for Window to Appear** from the recorder toolbar menu.



17. 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.

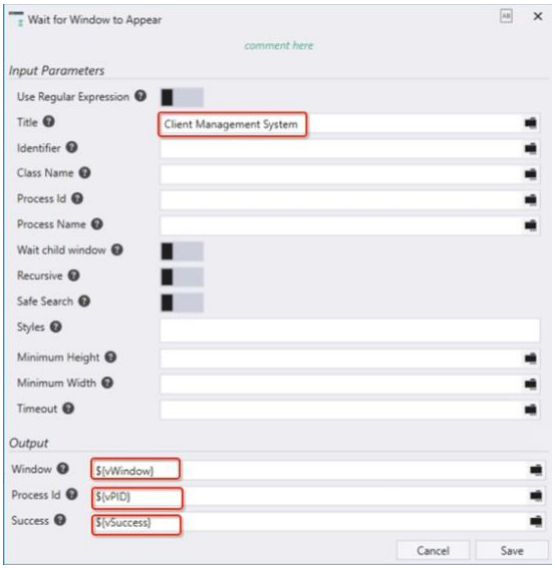
    Output

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

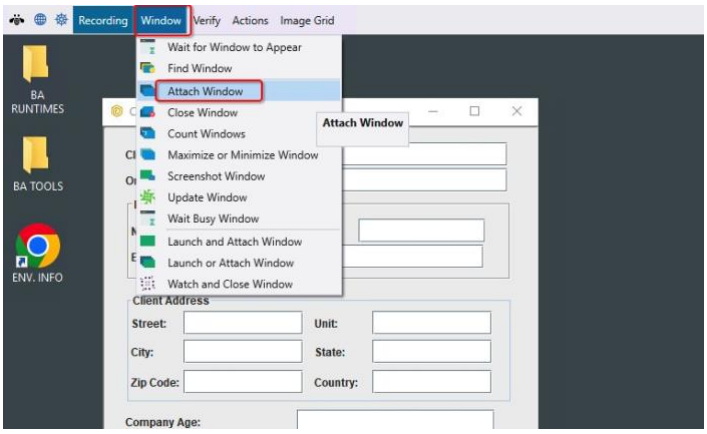
    Process Id: Click the icon on the right, select the **vPID** variable

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

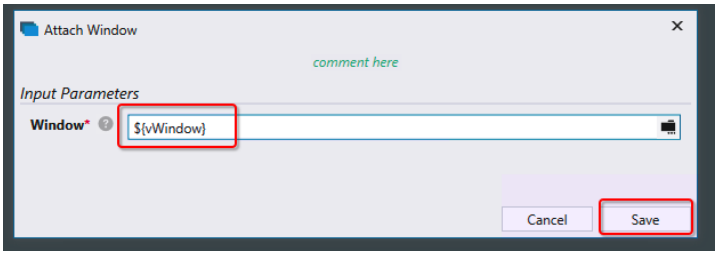
    Once done, click **Save**.



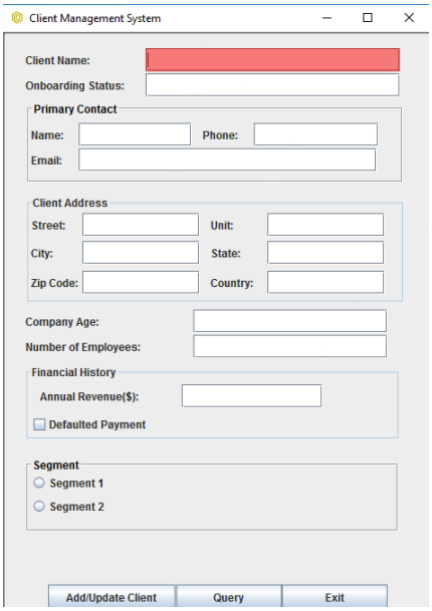
18. Select **Window ➔ Attach Window** from the recorder toolbar menu.



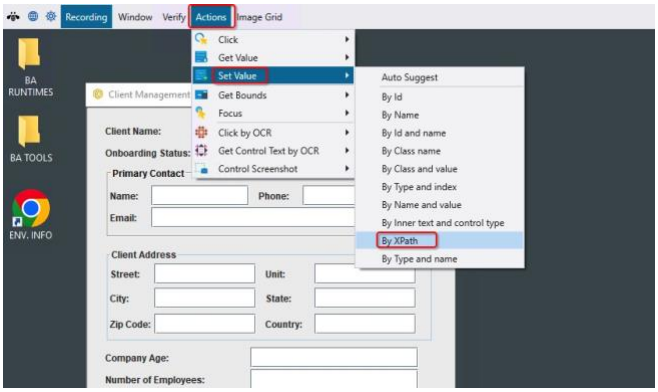
19. Configure the **Attach Window** command as below by selecting the variable **vWindow**. Once done, click **Save**.



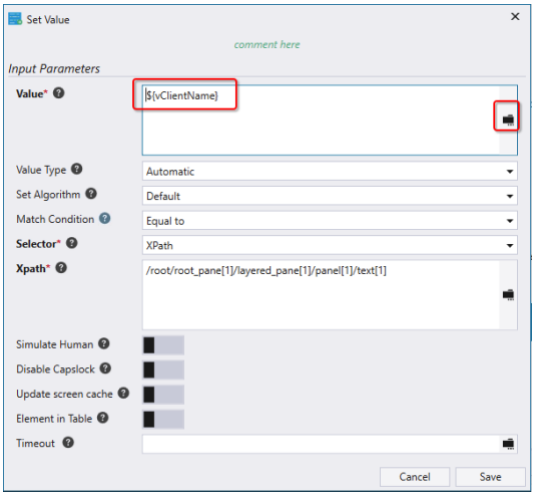
22. Automate the Client Management System application itself.
    1. **Press and hold the left CTRL** key, **move the mouse to the Client Name textbox** and wait for a few seconds. The checkbox will be captured and marked as light-red color as below. **Once the textbox is captured, release the CTRL** key.



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

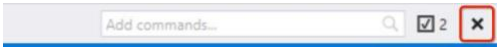


3. Configure the **Set Value** command as below. For Input Parameters, click the icon and select the variable **vClientName**. Once done, click **Save.**

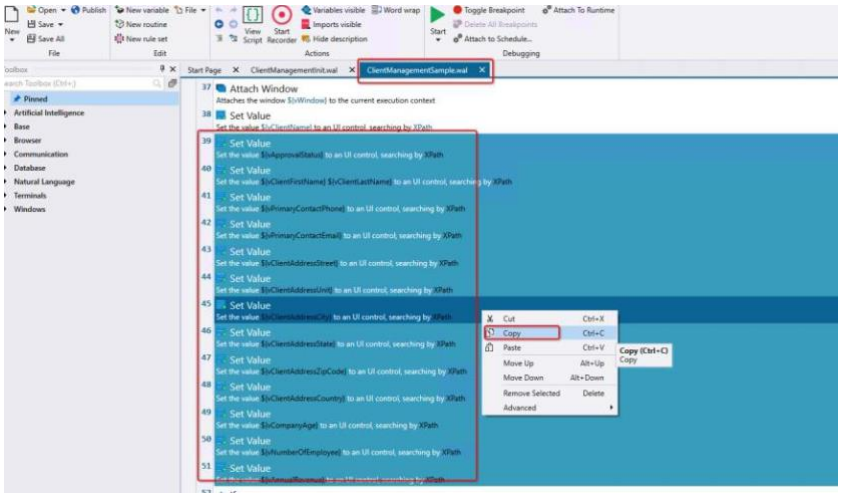


4. Next is to automate the rest of the fields. Follow the steps below to copy the commands from the sample script:

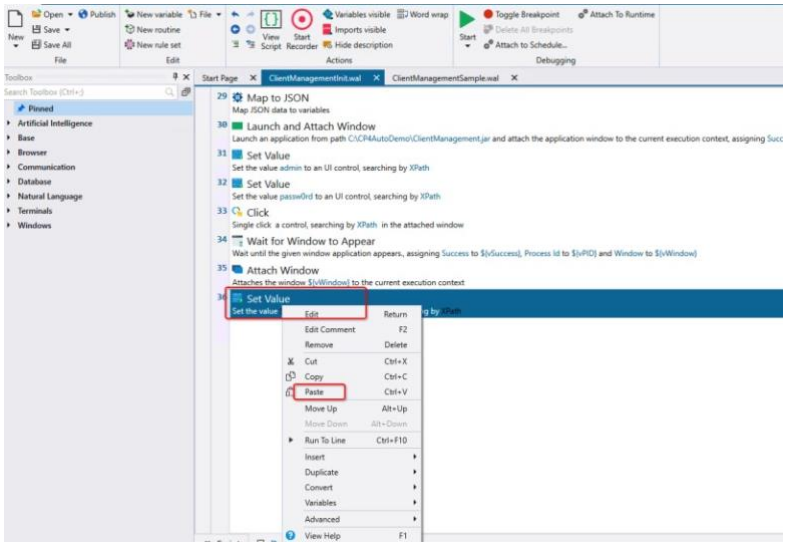
   1. Switch to the RPA Studio by clicking in the Recorder window's top-right corner and return to Studio.



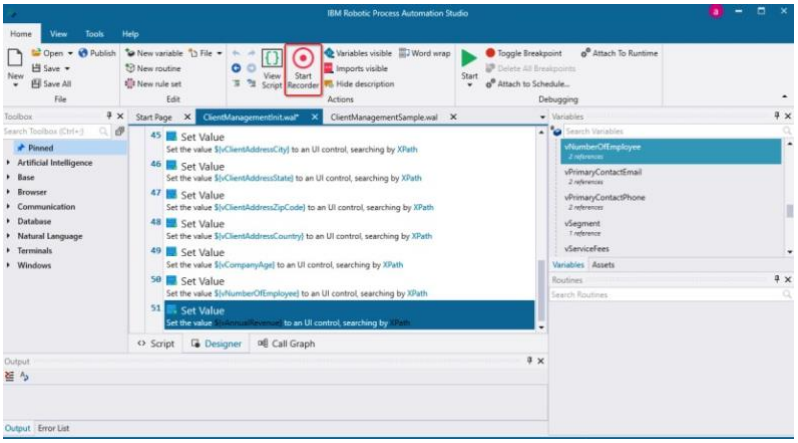
   2. **Open** the **C:\CP4AutoDemo\ClientManagementSample.wal**, select lines 39 to 51, right-click the mouse, and select **Copy**.



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

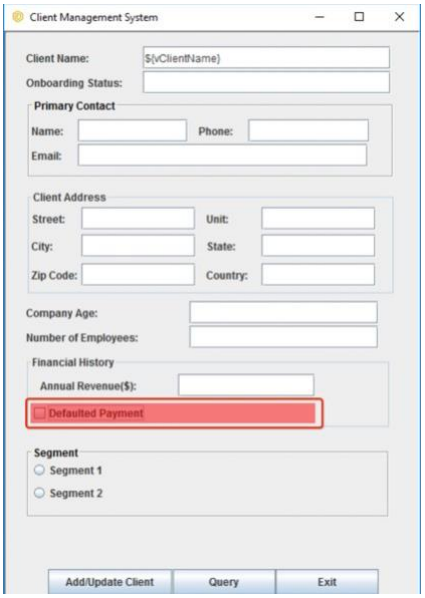


4. Once done, your script should be similar as the one below. **Start** the Recorder again by **clicking** the icon from.

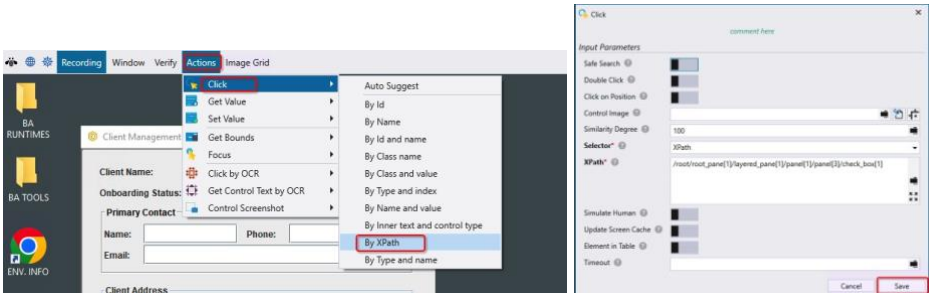


Defaulted Payment is a checkbox. It will only be checked if the client has a defaulted payment. We will use the Recorder to check it first, then add processing logic later.

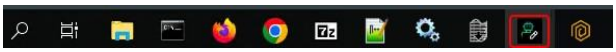
5. Press and hold the left CTRL key and move the mouse to the **Defaulted Payment** checkbox. Once the checkbox is captured, release the CTRL key.



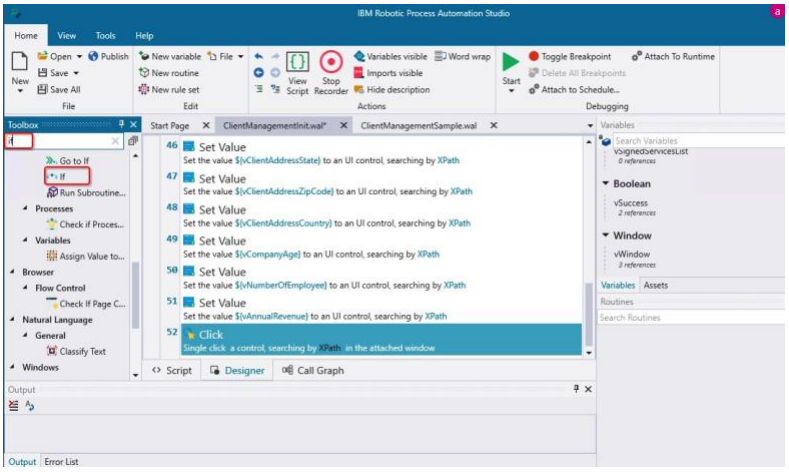
6. Select **Actions ➔ Click ➔ By XPath** from the Recorder toolbar menu. In the **Click** command configuration window, keep all the default settings and click **Save**.



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



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

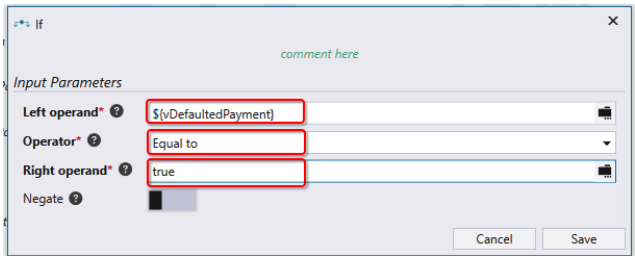


9. Configure the **If** command as below. Once done, click **Save**.

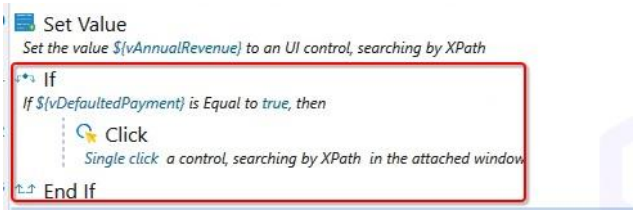
   For **Left operand**, click the icon on the right, select the **vDefaultPayment** variable.

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

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



10. In the Studio Designer editor view, select the last **Click** command, 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 the Recorder windows by clicking the icon in the top-right of Studio to minimize the Studio window.

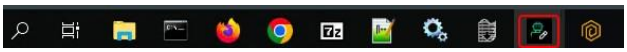
Now let's automate the **segment** field. The segment field is a set of two radio buttons. Similar to the Defaulted Payment checkbox, we will use the Recorder to click the radio button first, then add processing logic later.

12. Press and hold the **left CTRL key**, and 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.

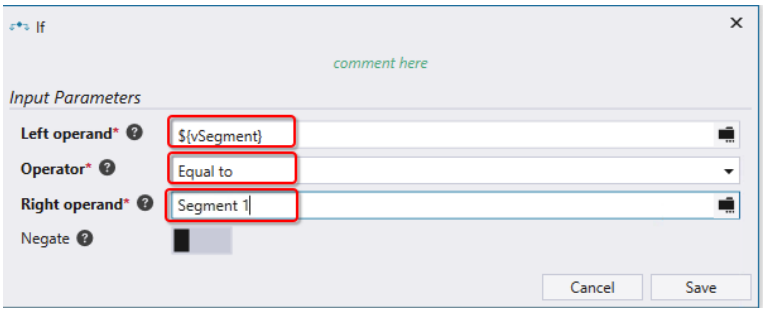
    Keep all the default settings in the Click command configuration window and click **Save**.

13. Follow the same approach above to click the **Segment 2** radio button.

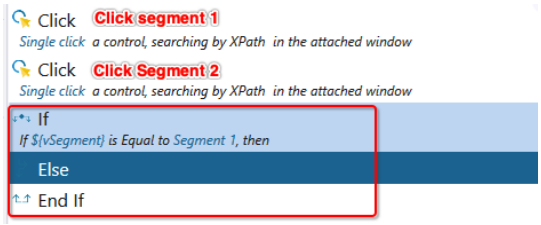
14. Switch to the Studio window by clicking the Studio icon on the Windows toolbar.



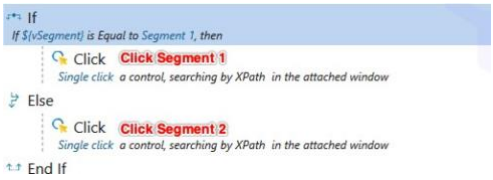
15. Add the **If** command to the end and configure it as below. Once done, your script should look similar as the one below.



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



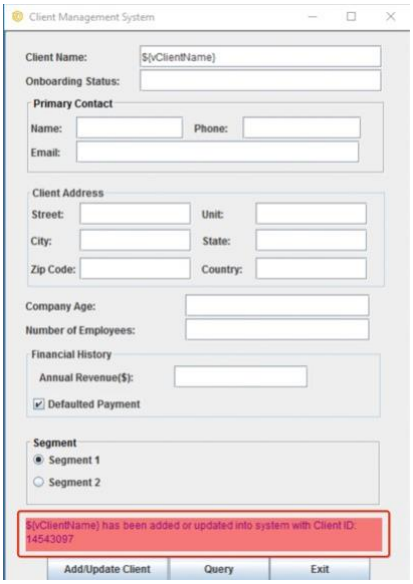
17. Drag and drop the **Click segment 1** radio button command (the first of the two) between the **If/Else** command and the **Click Segment 2** radio button command between the **Else/End If** command. Once done, your scripts should look similar as the one below.



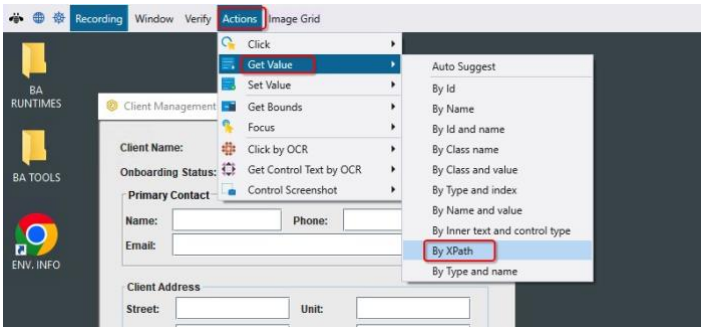
18. Switch back to the Recorder window by minimizing the Studio window again.
19. Press and hold the **left CTRL key**, and move the mouse to the **Add/Update Client** button. Once the button is captured, release the CTRL key, select **Actions ➔ Click ➔ By XPath** from the Recorder toolbar menu. Keep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



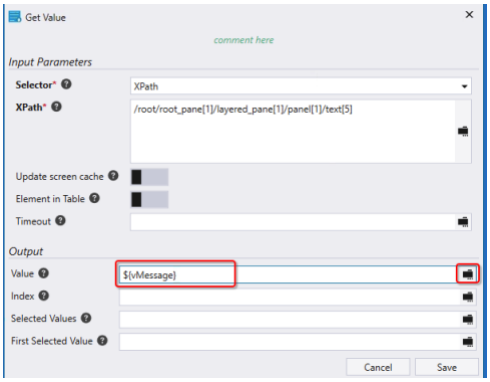
21. Select **Actions ➔ Get Value ➔ By XPath** from the Recorder toolbar menu.



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

    For **Input parameters**, leave all default settings.

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



23. Press and hold the **left CTRL key**, and 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.

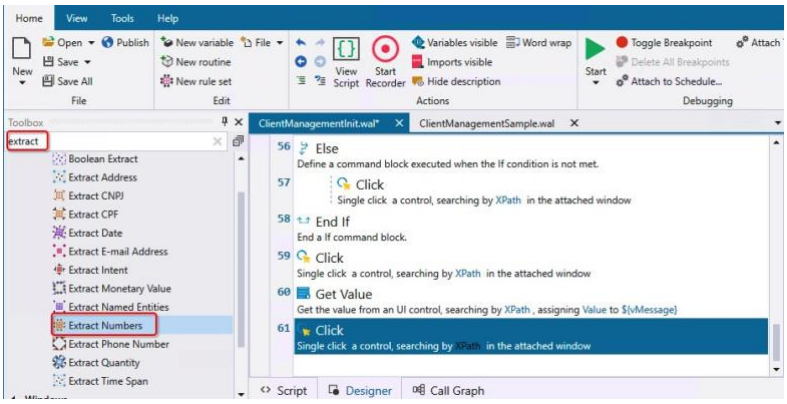
    Keep all default settings in the Click command configuration window and click **Save**.

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

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

25. 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 with the web application.

    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:

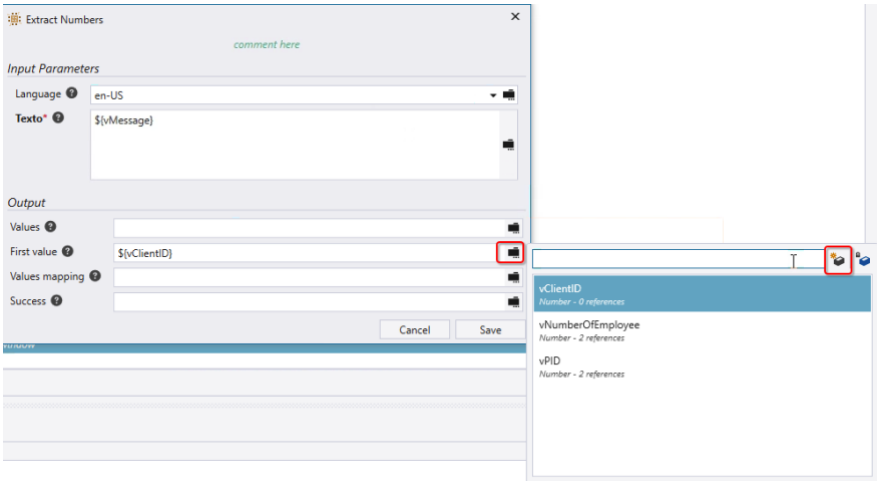
    Language: Select en-US

    Texto: Click the icon and select the variable vMessage

    Output:

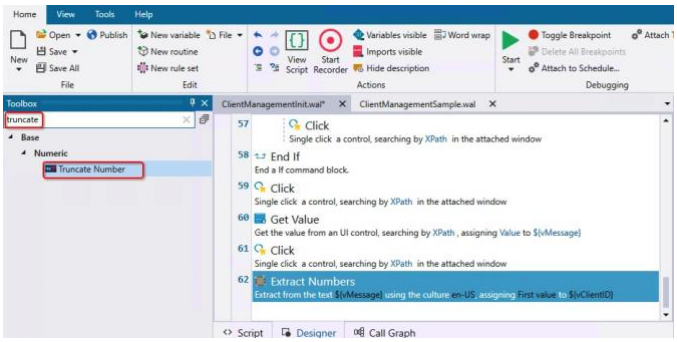
    Values: The command will retrieve all numbers as a list if the source string contains multiple numbers. In this lab, it only has one number; leave it blank.

    First Value: Click the icon on the right, and select **vClientID** variable to assign the first number found to this variable.

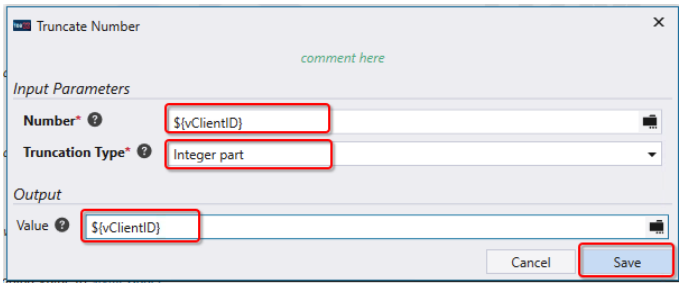


27. The clientID extracted above contains decimals which are not required and need to be truncated.

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

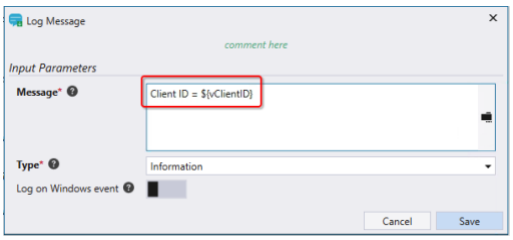


28. Configure the **Truncate Number** command as below. Once done, click **Save**.

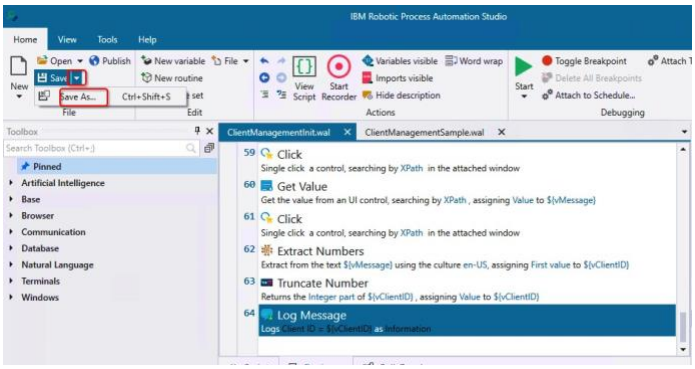


29. Add a **Log Message** command and configure it as below. Once done, click **Save**.

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



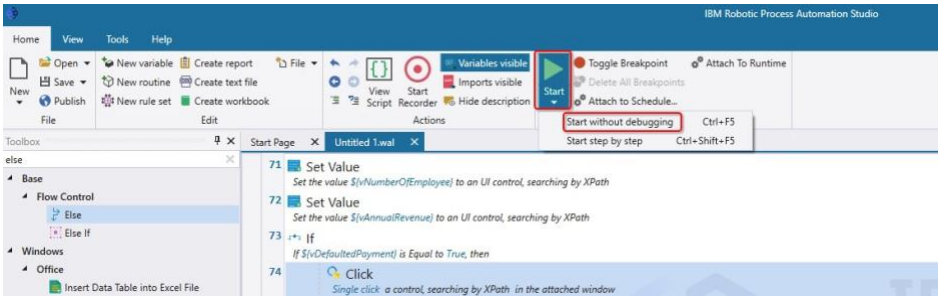
30. Now you have automated the Client Management System Java application. Click the triangle icon beside **Save** and select **Save As** to save your Script to the **C:\CP4AutoDemo\Lab 1 – Application Automation** folder, please use a different name to avoid overwriting init script.



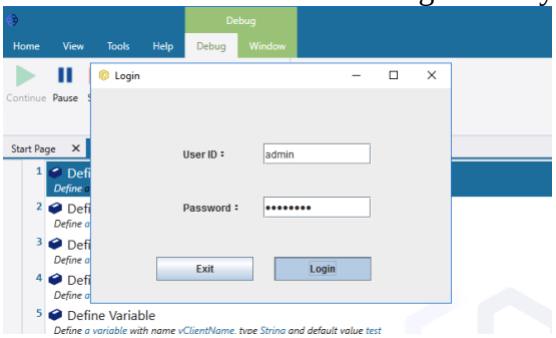
4.1.2 Verification Instructions

IBM RPA Studio provides two types of 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.

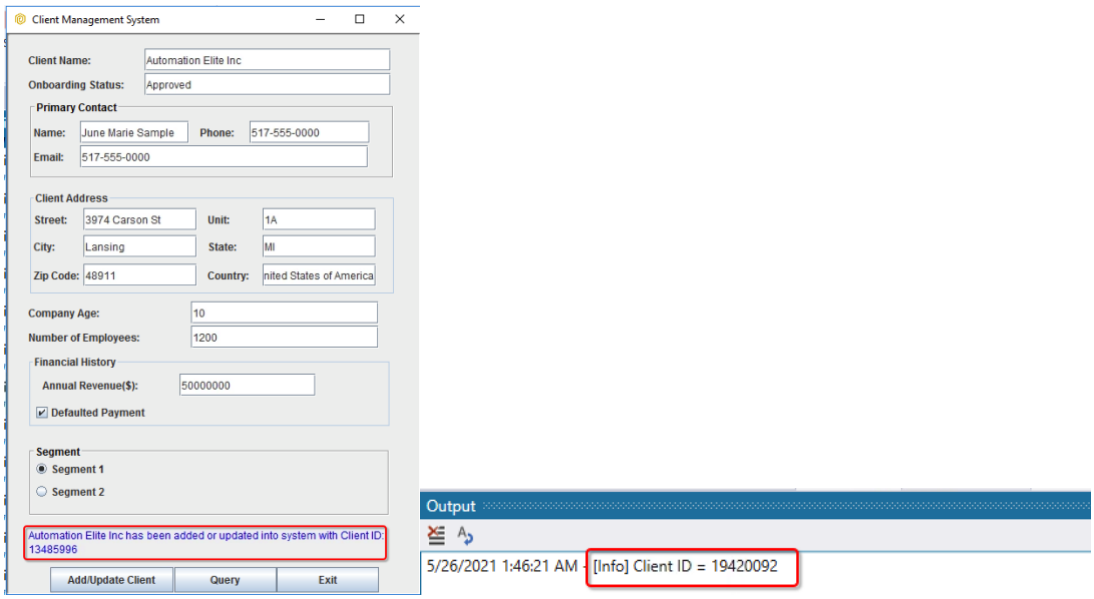
1. Click the **Start** icon and select **Start without debugging** to execute the bot script.



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.



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



Summary

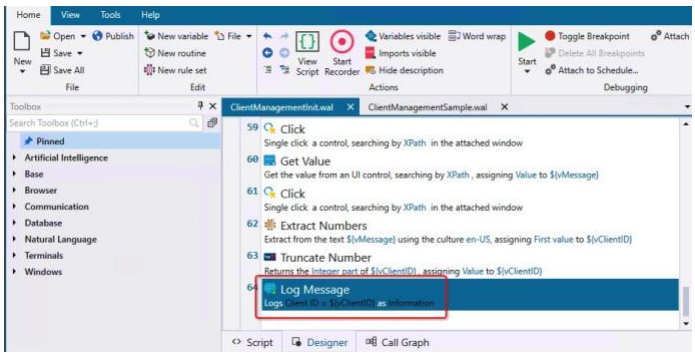
In this exercise, you have learned:

1. How to use IBM RPA Studio to develop and test an automation script.
2. How to use the IBM RPA Recorder to automate a Java Application by starting it, getting/setting UI control values, and clicking UI controls.
3. How to use various other IBM RPA commands to automate your application by, for example controlling the execution flow.
4. 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 take about 45 minutes to complete.

Please open the bot script created from exercise 1 in IBM RPA Studio if not yet and go to the end of the script, please note the line number may be different. You will continue with the web application automation from there.



4.2.1 Develop Bot Script

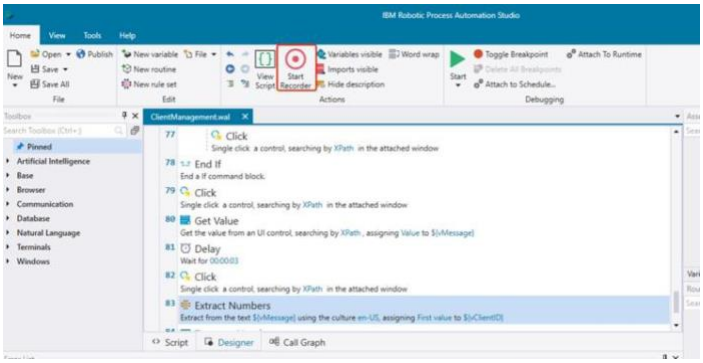
The typical process to automate a web application is to start a web browser first, then navigate to the corresponding web page, capture the controls from the HTML document and take appropriate actions.

The IBM RPA Web recorder is a tool in IBM RPA studio that can record user actions in the browser and automatically generate the equivalent commands in your script. After recording, you can review the commands to remove redundancy, tweak the selectors, etc.

The web recorder is similar to the Java recorder you used in the first part of the lab. First you place your mouse cursor over an element and press **Ctrl** key to capture the element. Once an element is captured, it will be highlighted in **yellow**, and then you can use the Recorder to add the command. In this lab we will use this approach to automate the sample web application.

To use the web recorder, it requires the installation of the IBM RPA web recorder extension in Google Chrome, which has been already installed in the lab environment. Please refer to Web Recorder for more details.

1. Click **Start Recorder** from IBM RPA Studio to start the Recorder.



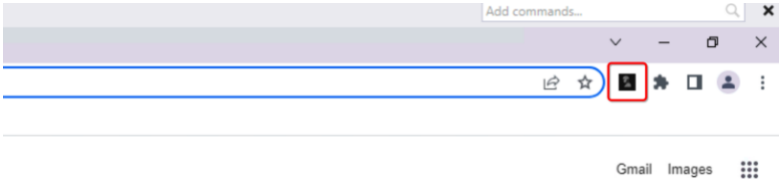
2. Click the **Activate Web Recorde**r icon in top-left corner in recorder window and select **Chrome**.

Important: If you are using Google Chrome, **refresh the page after enabling the recorder** to update the elements. If you are using Firefox, this is not necessary, follow the steps as normal.

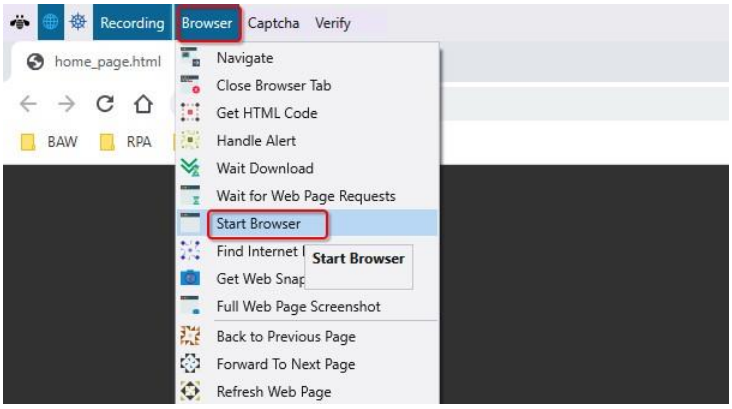
3. Manually start **Google Chrome** by clicking the icon from the Windows system toolbar.

4. Click the **web recorder** icon in top-right corner of your Chrome browser to activate it. Please wait and make sure the icon turns from grey ( ) to blue ( ).

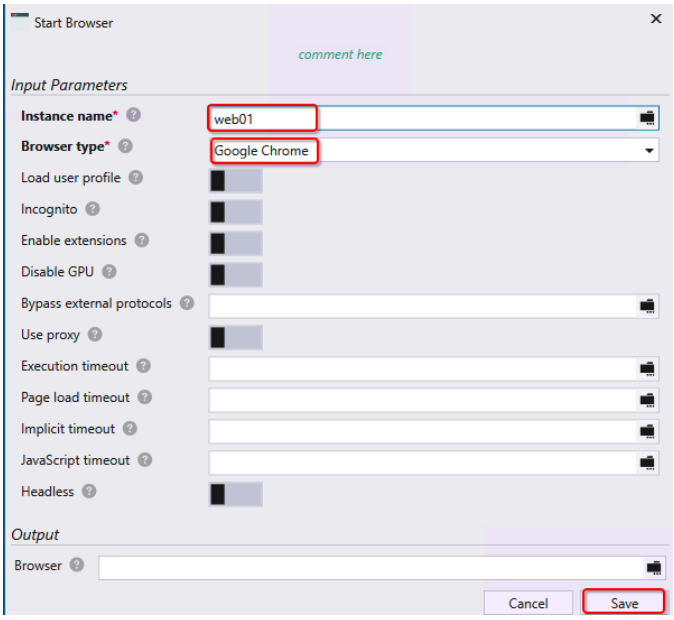
   Please note that after the web recorder is activated, it will record all the actions you perform in the browser. To avoid recording too many redundant commands, please don't click any controls in the browser moving forward.



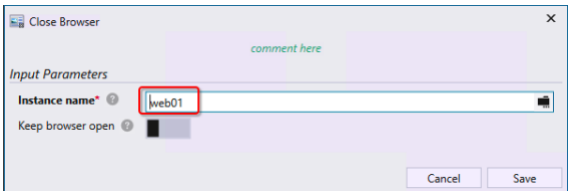
5. Click **Browser** from the top menu bar and select **Start Browser**.



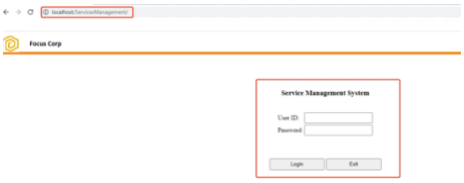
6. Configure the **Start Browser** command as below. As **Instance name**, enter **web01**. For **Browser Type**, select **Google Chrome**. Once done, click **Save**.


Start Browser
comment here
Input Parameters
Instance name*
web01
Browser type*
Google Chrome
Load user profile
Incognito
Enable extensions
Disable GPU
Bypass external protocols
Use proxy
Execution timeout
Page load timeout
Implicit timeout
JavaScript timeout
Headless
Output
Browser
Cancel
Save

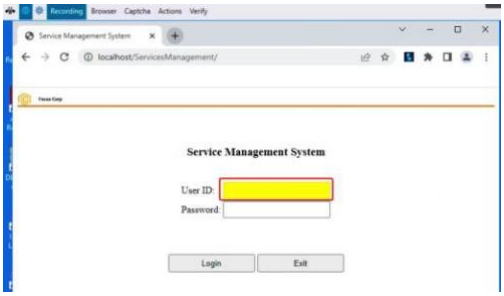
Once you click the **Save** button, RPA Studio will automatically add a **Close Browser** command. As **Instance name**, please enter the same name you entered above – **web01**. Once done, click **Save**.
Please note that this command will be added just after the **Start Browser** command.



8. Manually enter http://127.0.0.1/ServicesManagement in the Google Chrome address bar to open the Service Management System solution login page.

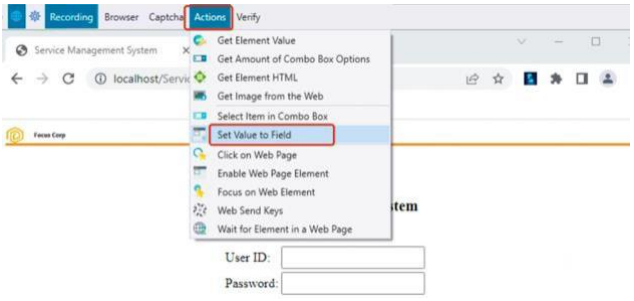


9. Automate the **Login page** using the steps below:
    1. **Press and hold the left CTRL** key, **move the mouse to the User ID textbox**, and wait for a few seconds. The User ID textbox will be captured and marked with **yellow** color, as shown below. **Once** the User ID textbox is **captured, release the CTRL key**.



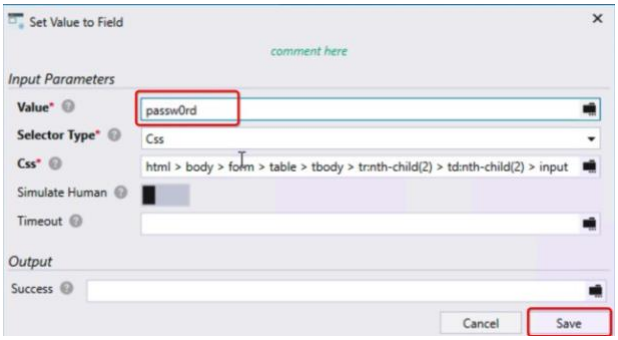
    2. Select **Actions ➔ Set Value to Field** from the Recorder toolbar menu.

Note: In case you notice that the yellow highlighting goes away when you click through the recorder menu, this is the expected behavior.

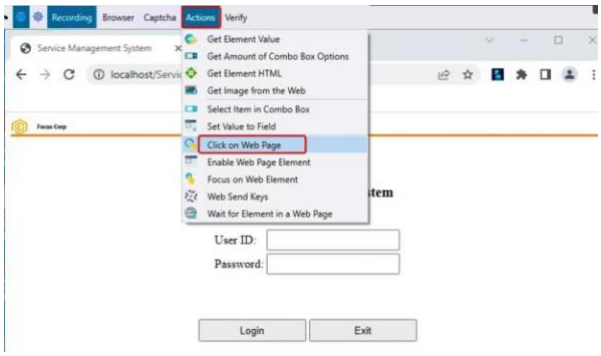


3. Configure the **Set Value to Field** command as below. Please enter **admin**, which is the only user that can log in to the Service Management System application. Once done, click **Save**.

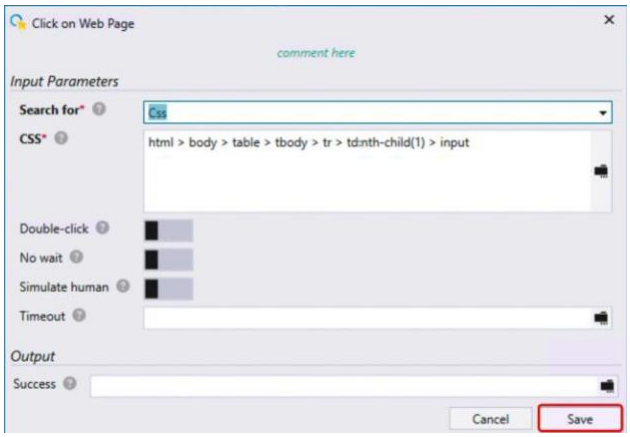
4. Follow the same steps to automate the Password field. Enter **passw0rd** (make sure to use zero as part of the password), which is the only valid password that can be used to login.



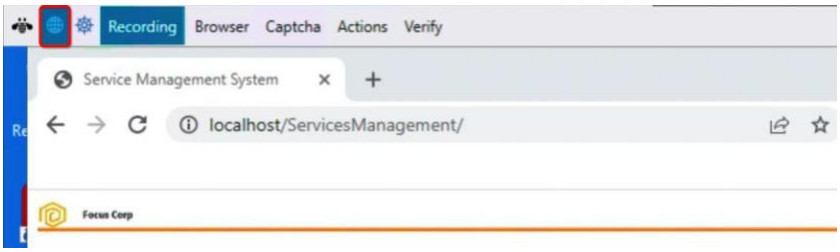
5. **Press and hold the left CTRL** key, **move the mouse to the Login button**, and wait for a few seconds. The Login button will be captured and marked as a **yellow** color. **Once** the **Login button** is **captured, release the CTRL key**, and select **Actions ➔ Click on Web Page** from Recorder toolbar menu.



6. You don't need to change anything for the **Click** command as shown below, just click **Save** which will complete the Login page automation.

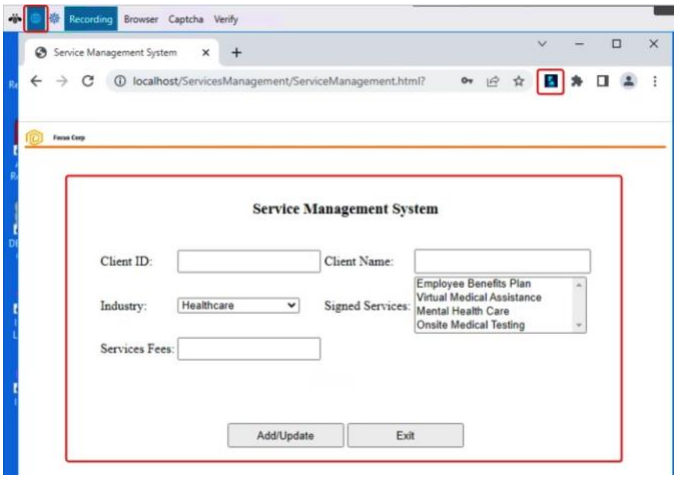


10. **To avoid redundant recording commands, please deactivate web recorder by clicking the** icon in the top-left corner of the recorder window.

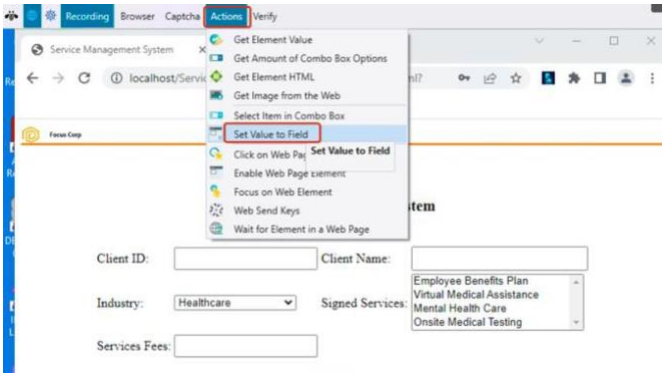


11. Manually enter **admin** in the user ID textbox and "p**assw0rd"** in the password field. Then click the **Login** button to log into the Services Management System as below. Once done, please **re-activate** the web recorder and make sure the web recorder in Google Chrome is also activated by checking the icon as shown below.

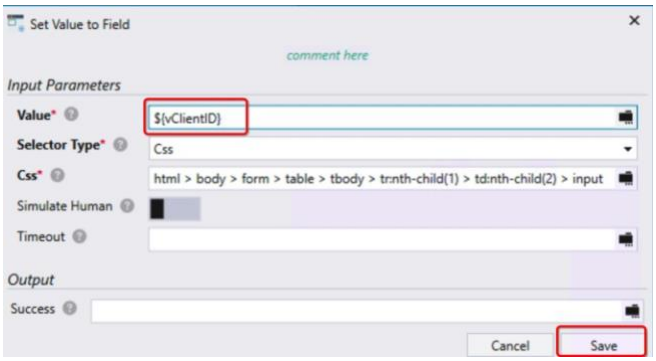
Important: If you are using Google Chrome, **refresh the page after enabling the recorder** to update the elements. If you are using Firefox, this is not necessary, follow the steps as normal.



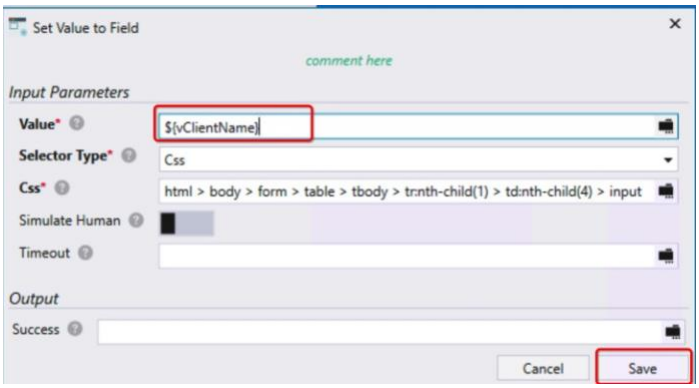
12. **Press and hold the left CTRL** key, **move the mouse to the Client ID** text field, and wait for a few seconds. The client ID text field will be captured and marked with **yellow** color. Once it is captured, release the CTRL ke**y**, and select **Actions ➔ Set Value to Field** from the recorder toolbar menu.



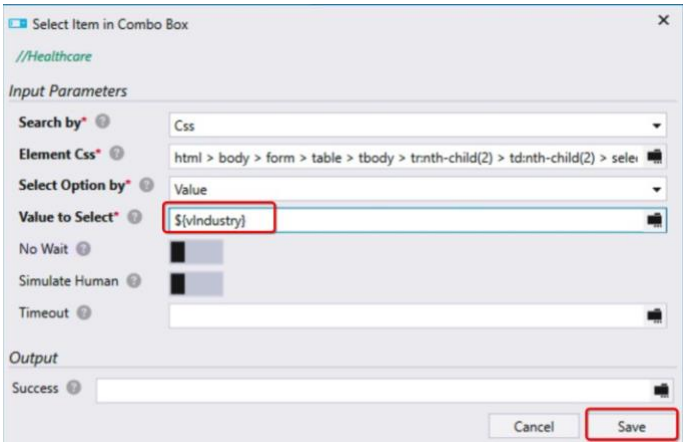
13. Configure the **Set Value to Field** command as below. For Input Parameters, click the icon and select the variable **vClientID**. Once done, click **Save**.



14. Follow the same steps to automate the **Client Name** field and configure the **Set Value to Field** command as below. Once done, click **Save**.

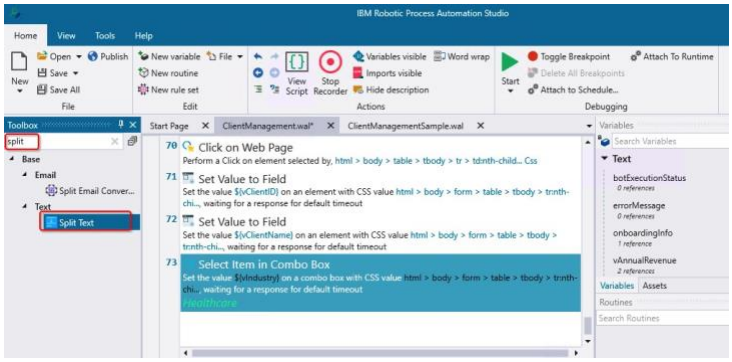


15. Follow the same approach to capture the **Industry** combo box, and then select **Actions ➔ Select Item in Combo Box**. Configure the command as below and select the variable **vIndustry** as value to select. Once done, click **Save**.



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

    1. Switch to IBM Robotic Process Automation Studio.
    2. Find and double-click the **Split Text** command.



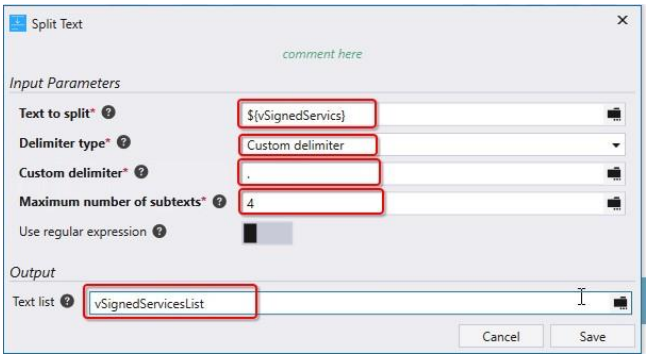
    3. Configure the Split Text command as below. Once done, click **Save**.

       Input Parameters:

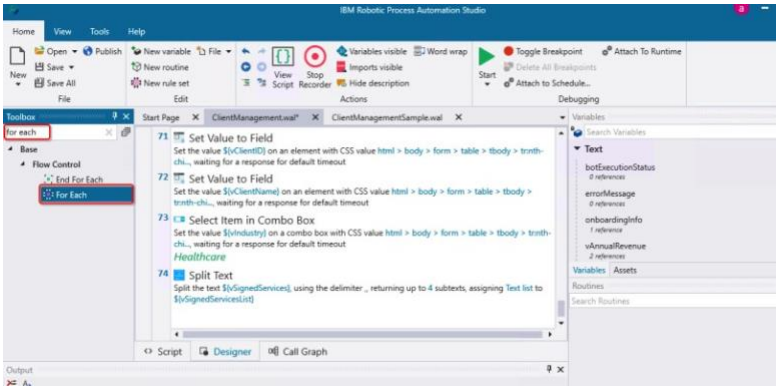
       Text to split: Select the variable **vSignedServices**
       Delimiter type: Select **Custom delimiter**
       Custom delimiter: Enter comma **(,)**
       Limit of subtexts: Enter 4 since there are up to 4 services for each industry that can be selected in the client onboarding application

       Output:

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



4. Find and double click the **For Each** command.

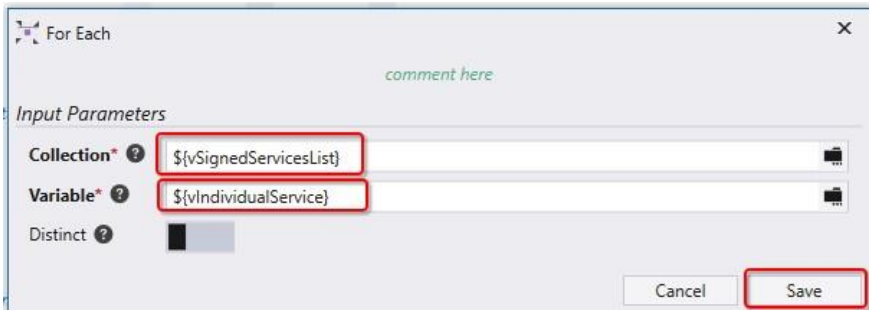


5. Configure the **For Each** command as below. Once done, click **Save**.

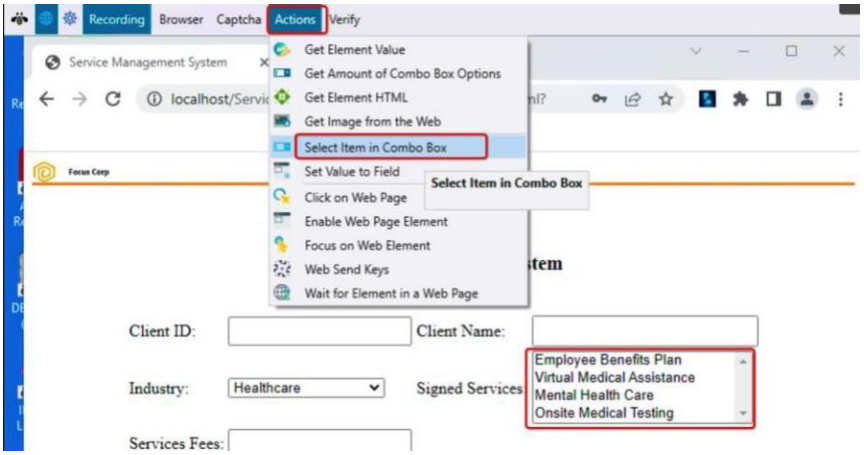
Input Parameters:

Collection: Select the variable **vSignedServicesList**.

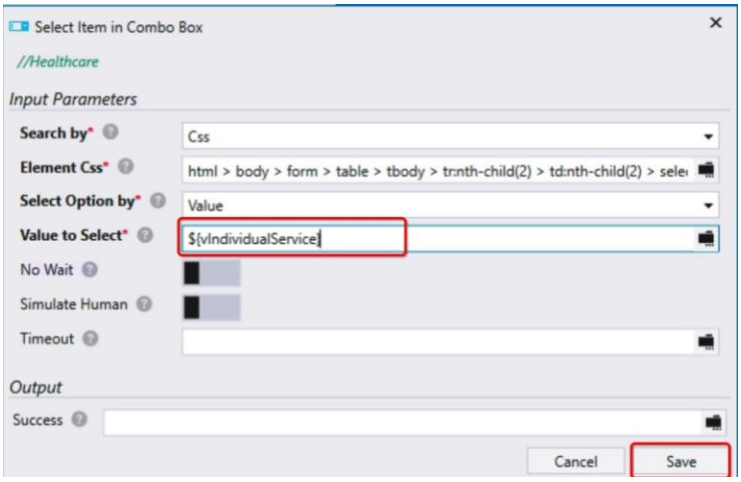
Variable: Select the variable **vIndividualService**.



6. Switch to the **web recorder** window, capture the **Singed Services** combo box, and select **Actions ➔ Select Item in Combo Box**.

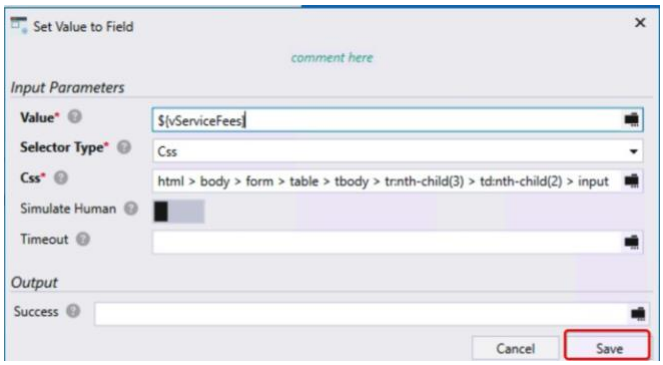


7. Configure the **Select Item in Combo Box** command as below. For **Value to Select**, select the variable **vIndividualService**. Once done, click **Save**.

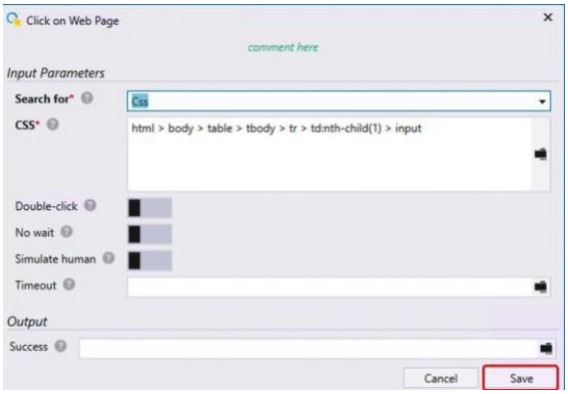


8. Switch back to the IBM RPA Studio. Find the last **Select Item in Combo Box** command and drag and drop it into between the **For Each** and **End For Each** command.

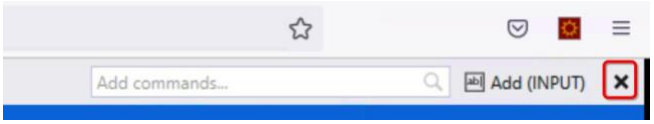
17. Automate the **Services Fees** field and Configure the **Set Value to Field** command as below. Once done, click **Save**.



18. Automate clicking the **Add/Update** button, select **Actions ➔ Click on Web Page**. You don't need to change anything in the **Click on Web Page** command configuration panel, just click **Save**.



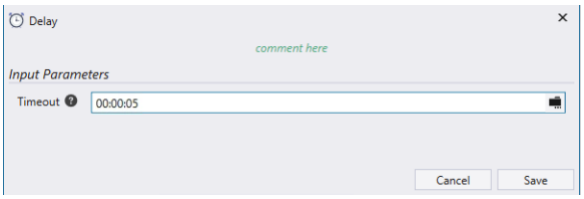
19. You have automated all fields in the Service Management system. You can close **Recorder** by clicking **X** icon in the top-right corner in the recorder window and return to IBM RPA Studio. Please also close Google Chrome.



20. Find the **Close Browser** command, and move it to the last line of the script.

*All commands that interact with the web page must be between the **Start browser** and **close browser** commands.*

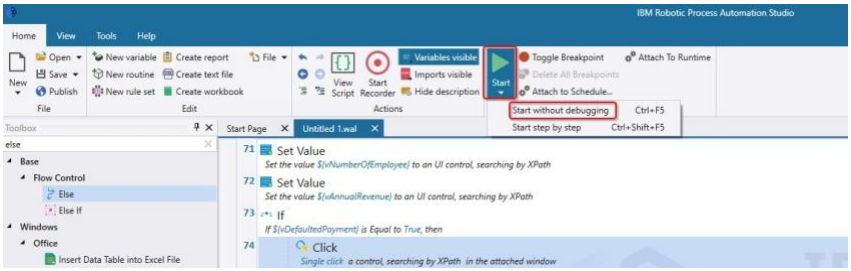
21. Since the bot executes very fast, for demo purposes, you can intentionally add a **Delay** command to check the execution result before closing the browser. Below setting will delay the execution for 5 seconds. Once done, click the **Save** button from the Studio toolbar to save your script.



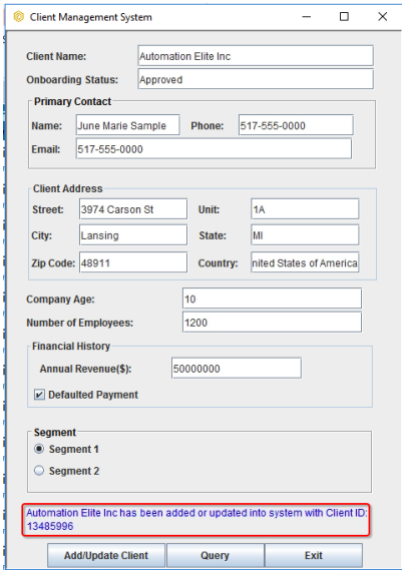
You have successfully automated a web application using IBM RPA. The bot first starts Google Chrome and logs into Services Management System. After that, it adds the client information and information about the signed services before saving these changes. Finally, it closes the browser.

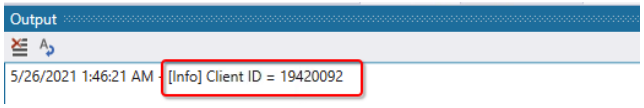
4.2.2 Verification Instructions

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

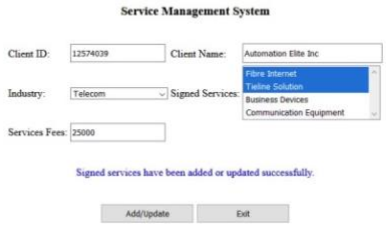


2. The bot script will be executed. It will launch the Client Management System Java application and add the client information first. Once done, it will retrieve the client ID from the result message.





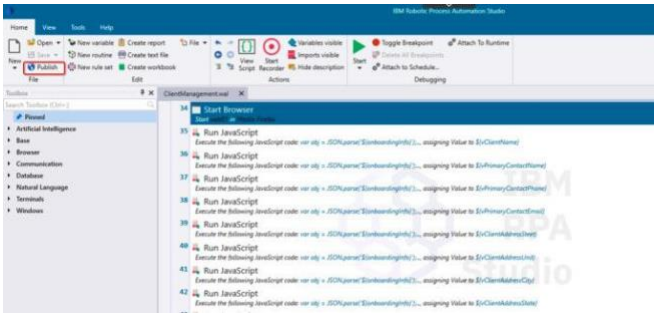
3. The bot will then start Google Chrome and login into the Services Management System web application. It will add the client information and the signed services information. Once done, it will close the browser window.



4.2.3 Publish Script to RPA Server

You have validated your script, let's publish it to the RPA server now.

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



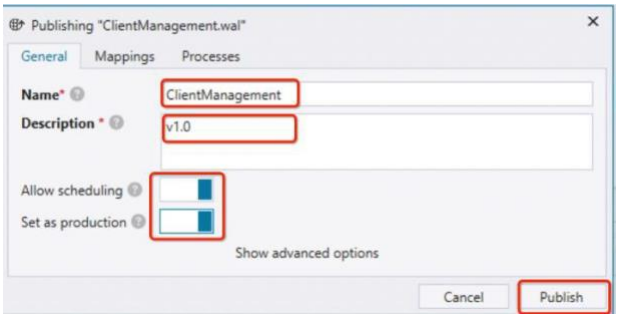
2. In the Publishing script window, enter appropriate values as below. Once done, click **Publish**.

   Name: it will automatically populate with your script's 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 the same script can exist on the tenant, but only one can be the production version. If you execute the script without specifying the version, IBM RPA will use the production version by default. Click **Set as production** to set it as a production version.



Your script should have been published to the tenant successfully, this script will be used in Workflow and RPA integration lab later.

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 the tenant.

Congratulations, you have successfully completed this lab!!!