IBM Cloud Pak for Business Automation Demos and Labs 2021

Chatbot Development and Configuration

V 1.2

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.

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.

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 develop a chatbot script using IBM RPA Studio, and then shows how to run it externally on a SaaS tenant through a chat mapping configuration. As part of this exercise, we will also show how to create and train a knowledge base that is then used in a chatbot script for national language understanding.

2.1 Pre-requisites

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

All the pre-requisites have been pre-installed/configured in this lab template. The information below is just for information purposes.

IBM Products:

- IBM Robotic Process Automation Studio v20.12.5.

Custom Solutions/Code:

- Client Onboarding Toolkit which contains the predefined business object definition and service flow to start the RPA bot matching the information required by the two backend systems below.
- 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

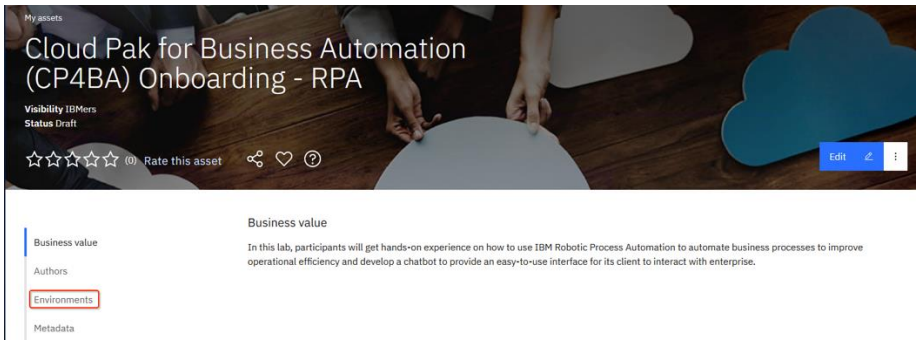
3 Accessing the Environment

If you have already reserved lab environment from IBM Asset Repo and registered your RPA account, please go to Chapter 4 directly.

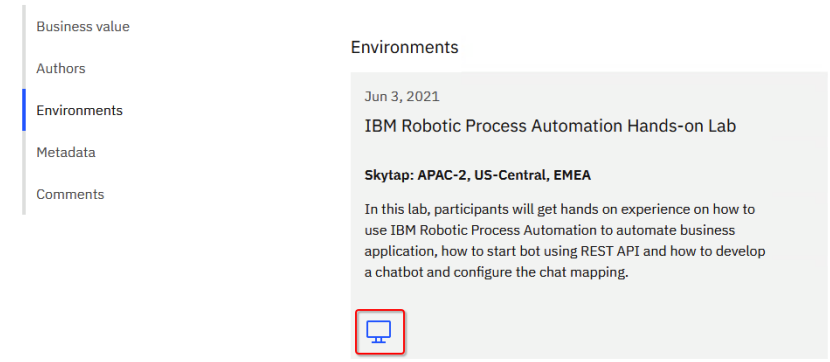
3.1 Reserve Environment

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

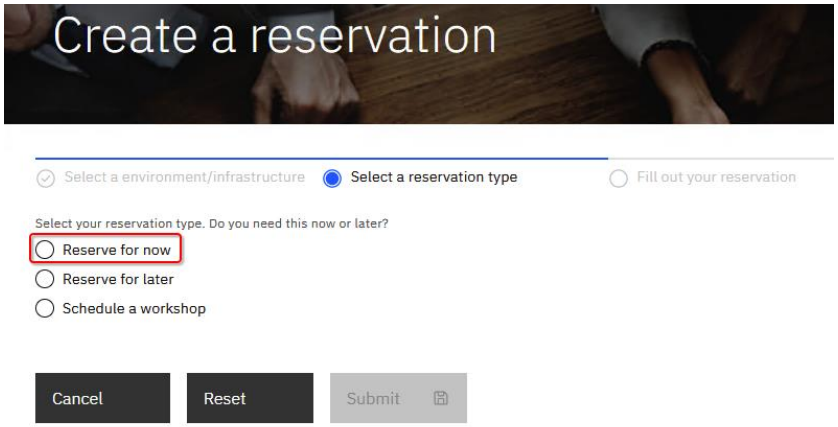
1. Click here to open IBM Asset Repo Reservation portal. You need to use your IBMID to login to the portal.



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



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



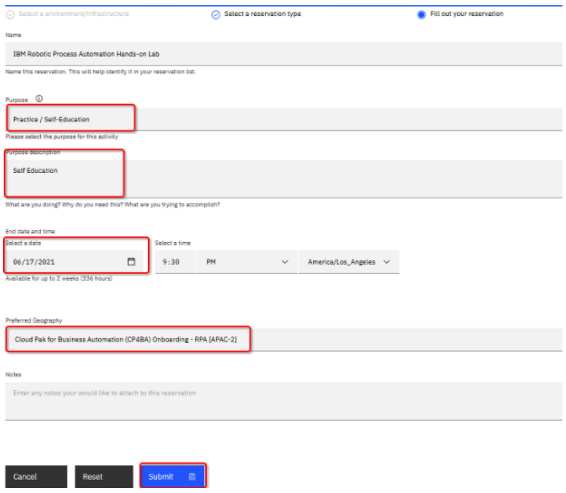
4. On the reservation page, make the appropriate selections as below. Once done, click **Submit**.

   Purpose: Select Practice/Self-Education.

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

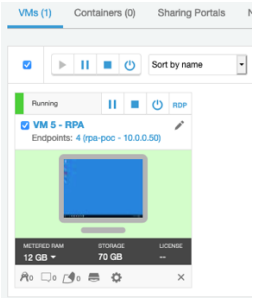
   End date and time: Select the end date and time that the environment will be deleted.

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



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 Access Information**). It also contains a URL to access the IBM RPA Rest Service remotely (**Application Service Information)**. This will be used in the Workflow and RPA integration scenario exercise.

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.

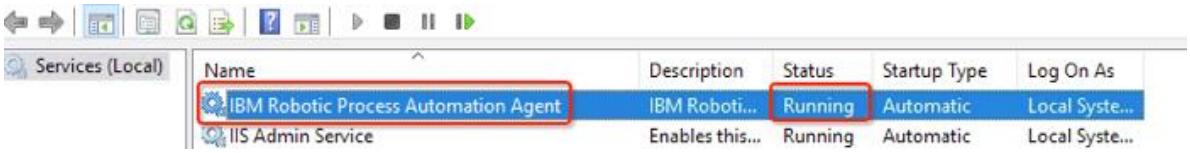


3.2 Activate RPA license

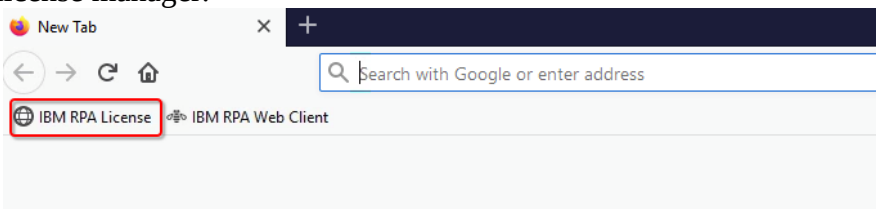
Before you can start and log into IBM RPA Studio, you need to re-activate the RPA license every time the RPA agent machine (or in our case the VM 5) is restarted. This is caused by special lab infrastructure setup and configuration and only required for this lab.

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

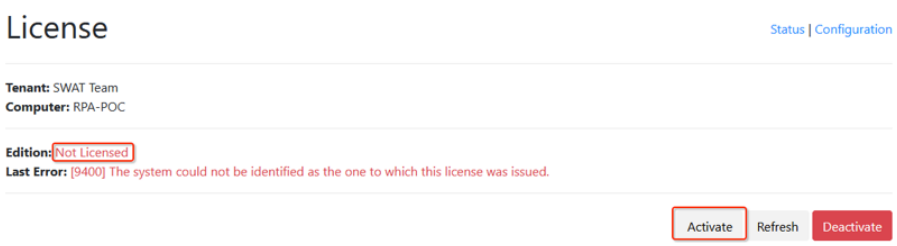
3. Check and ensure that the **IBM Robotic Process Automation Agent** service is in running status through Windows Service Manager.



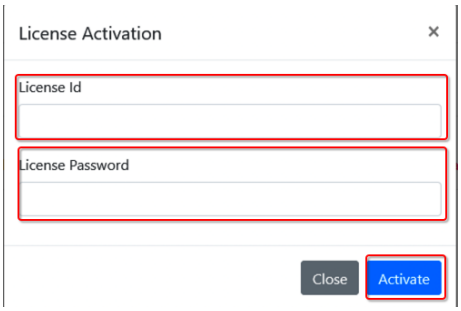
4. Start Firefox, click **IBM RPA license** from the bookmark toolbar to open IBM RPA license manager.



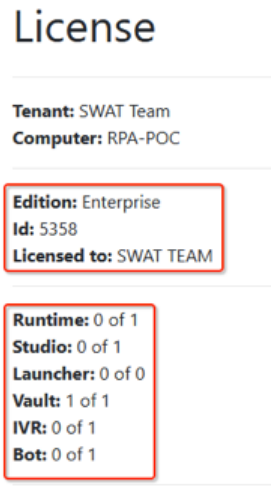
5. You will see the message **Not Licensed**. Click **Activate** button to open the License Activation window.



6. Enter the License ID and License Password and click the **Activate** button. You can get the License ID and License Password from here.



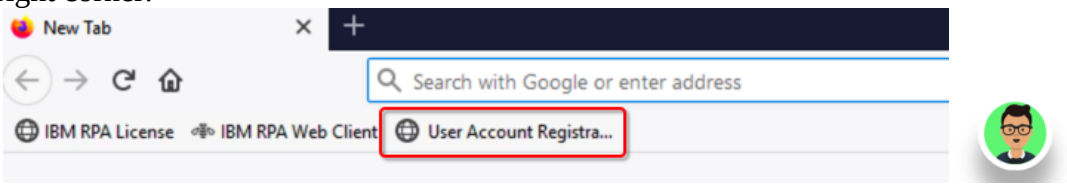
When the license is activated, you should be able to see the number of licenses available for each component.



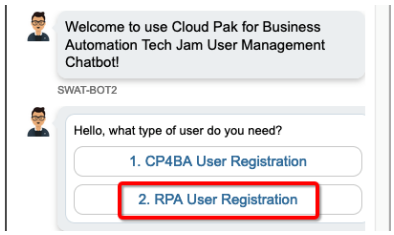
3.3 Register your user account

This lab requires two types of user accounts. One is for IBM RPA to grant you access to the IBM RPA tenant and Studio to develop, test and publish bot scripts. This is required for exercise 1 and exercise 2. The second user account grants you access to IBM Cloud Pak for Business Automation components running on Red Hat OpenShift Kubernetes Service hosting on IBM Cloud. This is required for the Workflow to RPA integration scenario in exercise 4. Please follow below steps to register your RPA account if you don't have access to the IBM RPA tenant yet.

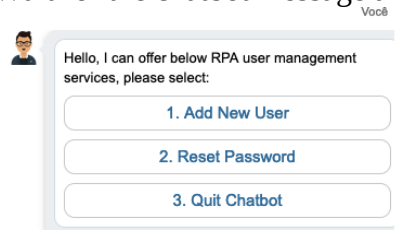
1. Start Firefox, click **User Account Registration** from the bookmark toolbar to open the user account management chatbot. Click the floating robot button at the bottom-right corner.



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

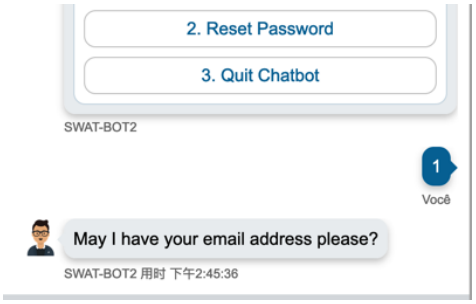


3. Wait for the chatbot message and select “Add New User”.

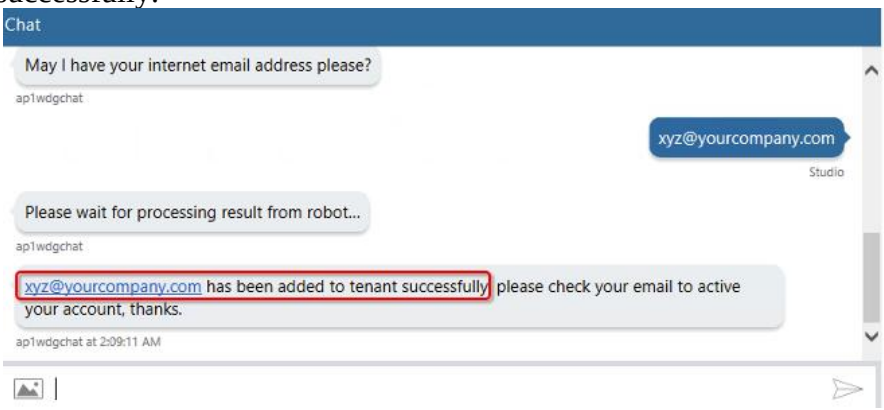


4. Wait for message from the chatbot requesting to enter your email address.

   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.

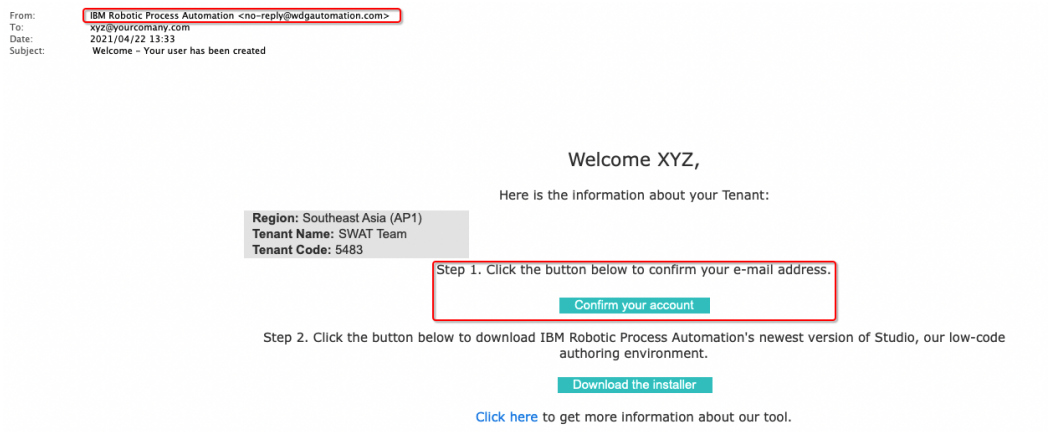


5. Enter your email address. Then wait for the chatbot to register your account. This may take a minute or two. Once the chatbot finishes registering your account, it will prompt a message to indicate that your account has been added to the RPA tenant successfully.

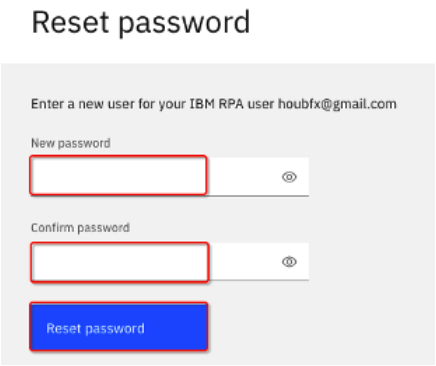


6. Check your mailbox. You should receive an email from **IBM Robotic Process Automation** as below. Click **Confirm your account** from **Step 1.**

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.

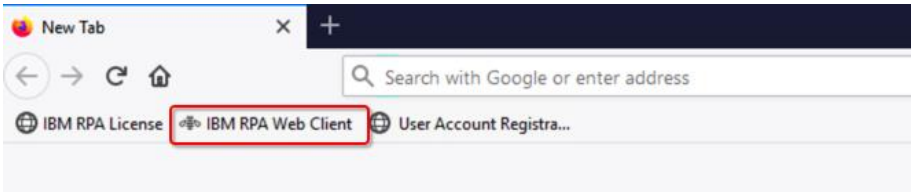


7. Enter new password for your account. Once done, click **Reset password** which will activate your account with the password you set here.

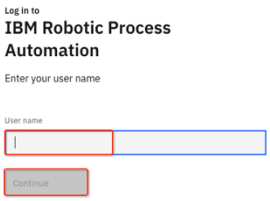


Please follow below steps to check if your account has been registered and activated successfully:

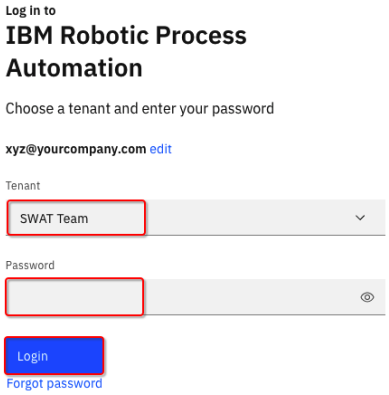
8. Start Firefox, click **IBM RPA Web Client** from the bookmark toolbar.



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



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



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

This exercise will be performed on the RPA VM, **VM 5 – RPA** and takes about 1.5 hours.

One of the key advantages of IBM RPA is the embedded Chatbot support. It provides rich commands for end-users to quickly develop a chatbot and provide an easy-to-user interface for enterprises to interact with its customers, partners or employees. Combining the chatbot and RPA commands, it can create chatbots through multiple channels that provide engaging client interactions. Embedded knowledge base support through machine learning technologies can make your chatbot even smarter and fulfill all kinds of different business requirements. A typical process to develop a chatbot includes three steps:

1. Create and train a knowledge base.
2. Develop a chatbot script using various RPA commands using IBM RPA Studio.
3. Configure a chat mapping within a tenant.

As part of the end-to-end client onboarding solution, we will develop a chatbot to enable a client to easily manage and update its basic client information when there are any changes. The client onboarding solution contains lots of information including primary contact, address, financial details and the number of employees the client company has. To illustrate the typical process on how to create and configure a chatbot, we will focus on creating a chatbot to only update the number of employees covering below aspects:

1. How to use the Machine Learning Model Builder to create a knowledge base.
2. How to use the Knowledge Base Training tool to train and publish the knowledge base.
3. How to develop a chatbot script using IBM RPA Studio.
4. How to configure a chat mapping for a tenant.

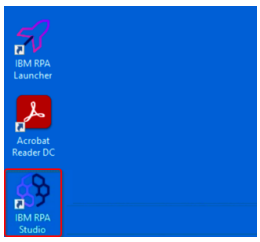
Due to a limitation for the number of bot integration services available in a tenant, you are not able to actually perform the chat mapping configuration by yourselves in this lab. Instead, we will provide instructions on how to do that to help you understand the typical procedure.

4.1.1 Create Knowledge Base

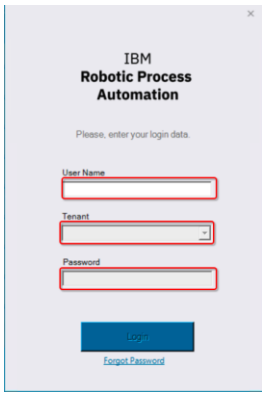
To create a knowledge base, a knowledge base file is needed which must be an Excel XLSX file. The knowledge base file name is not relevant, but its inner structure must have three spreadsheets – **KB**, **Words** and **Synonym.** Please refer to the IBM Robotic Process Automation Documentation to understand the details of the knowledge base file.

Perform the following steps to create a knowledge base:

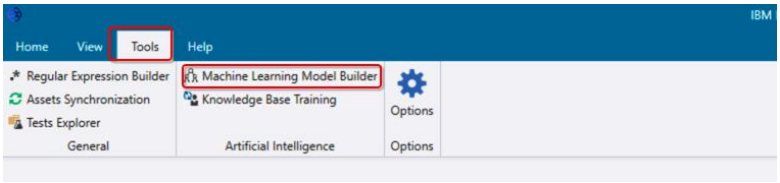
1. Start the IBM RPA Studio from windows desktop.



2. Log into the IBM RPA Studio using your IBM RPA account. Make sure to select the **SWAT Team** tenant.



3. Click **Tools → Machine Learning Model Builder** from the IBM RPA Studio toolbar.



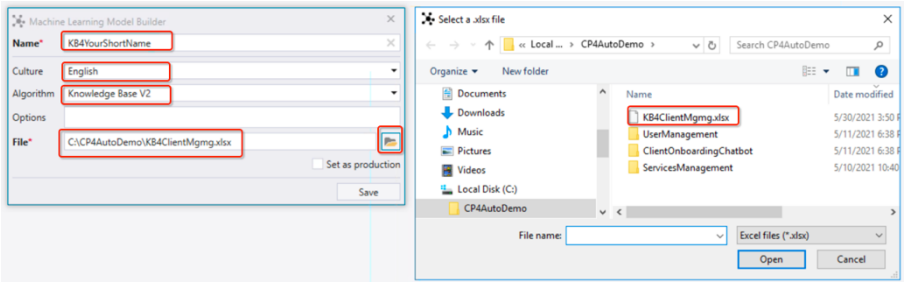
4. Configure the knowledge base model as below. Once done, click **Save**.

   Name: To avoid naming conflicts when publishing the model to the tenant, please make sure to use a unique name, for example – KB4**YourShortName**.

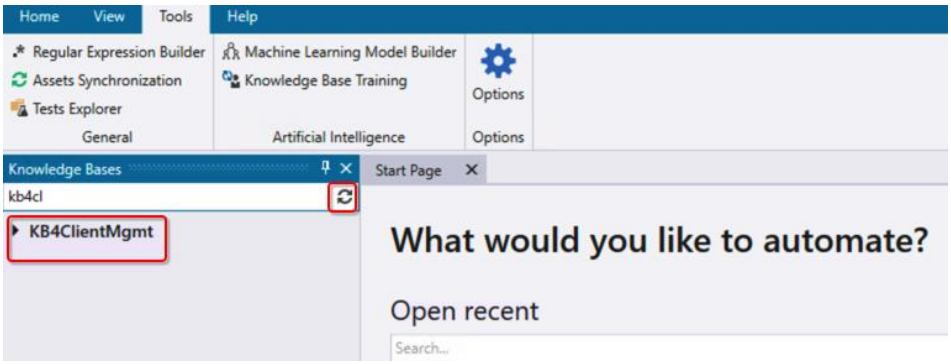
Culture: Select **English**.

Algorithm: Select **Knowledge Base V2**.

File: Click the icon on the right, browse to **C:\CP4BADemo** and select the provided knowledge base model file **KB4ClientMgmt.xlsx**.



5. The knowledge base model will be created and listed in knowledge base list on the left panel. In case you can't see your knowledge base model, click **Tools→Knowledge Base Training**, then click the icon to refresh the list.

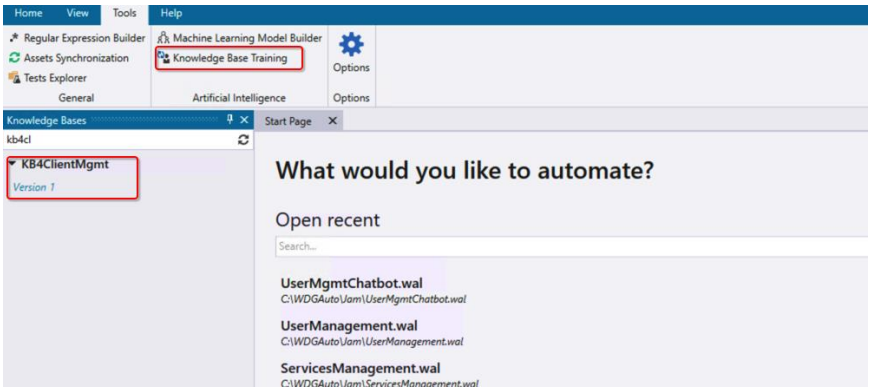


4.1.2 Train Knowledge Base

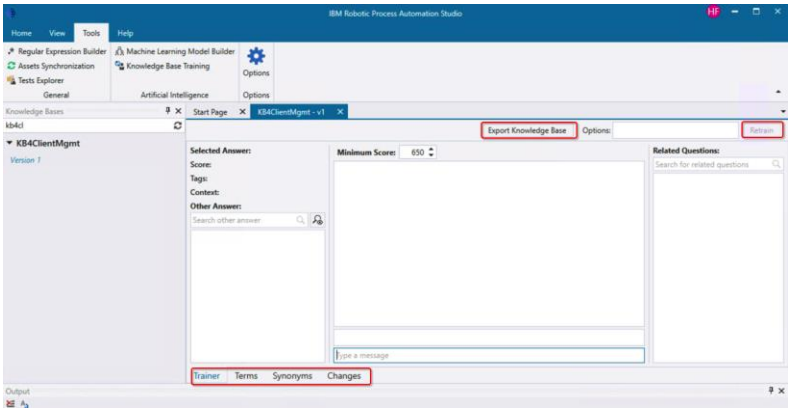
The Knowledge Base Training tool is an interface used to train Knowledge Base models, modify words, synonyms and review changes in the knowledge base file spreadsheet and publish it to a tenant. Please read the documentation to understand how to train a knowledge base model.

1. Click on **Knowledge Base Training**, find the knowledge base model you created above, and click the **version 1** to open the Knowledge Base Training interface.

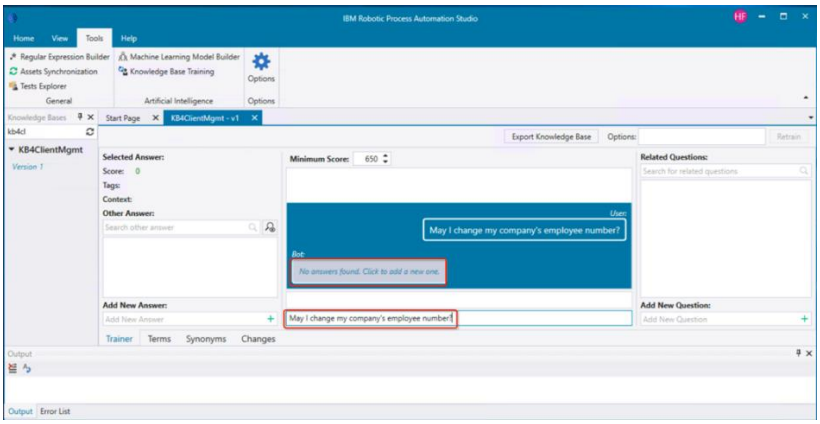
   Please note it will create a new version every time you publish a new knowledge base model to a tenant. If you see multiple versions of knowledge base model listed, select the latest version (recommended) or any version to train the knowledge base model.



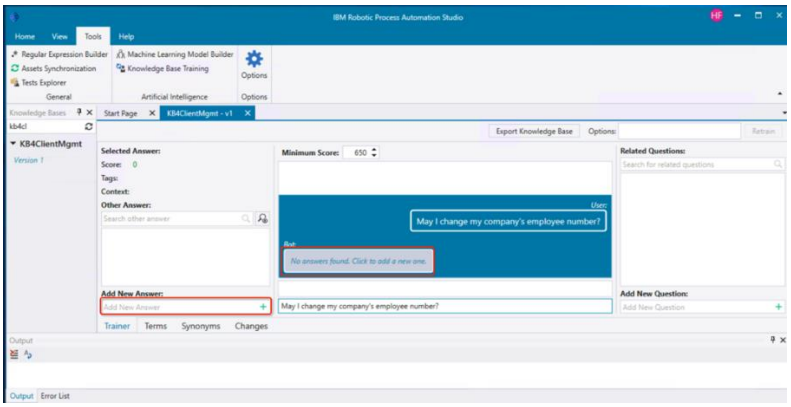
2. Familiar yourself with Knowledge Base Training tool. As mentioned above, the Knowledge Base Training tool provides an interface to train the knowledge base model, change Terms and Synonyms and publish the knowledge base to a tenant. You can click the corresponding tab to take a look at each interface. You can also use the tool to export the knowledge base model to an XLSX file.



3. Type in "**May I change my company's employee number**" and press **Enter**. It will try to get the best answer from the knowledge base model. Since the initial knowledge base model is empty, it doesn't have any answer. The bot will say "**No answers found, click to add a new one.**".



4. You can click the bot message or click the + icon below to **Add New Answer** to add a new answer. **Click** the + icon.

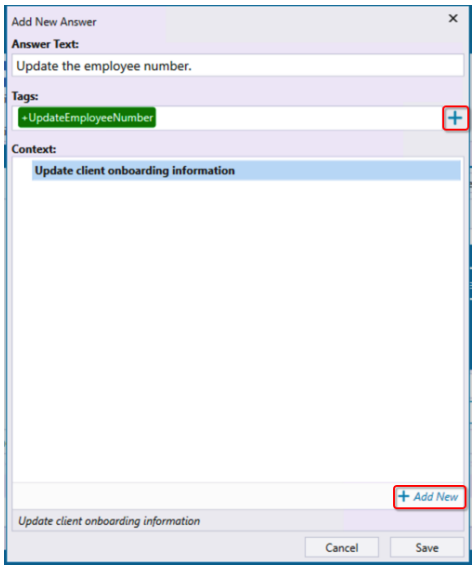


5. Configure the **Add New Answer** as below. Once done, click **Save**.

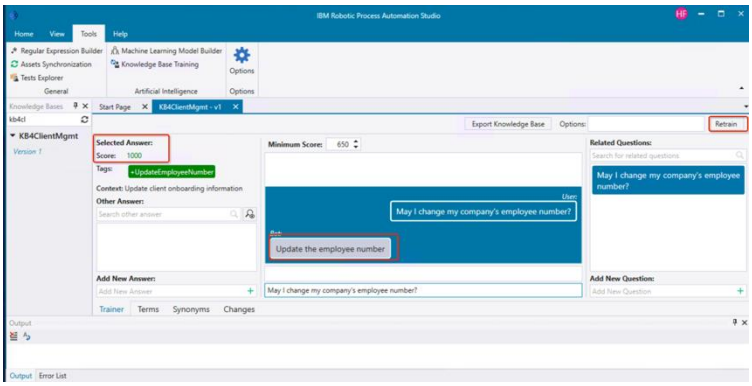
   Answer Text: Enter **Update the employee number**.

   Tags: Tags are text triggers that guide the conversation intent between bot and user. The "+" or the "-" sign at the beginning of a tag increments or decrements a tag, respectively. **Click** the "+" icon, enter **+UpdateEmployeeNumber**.

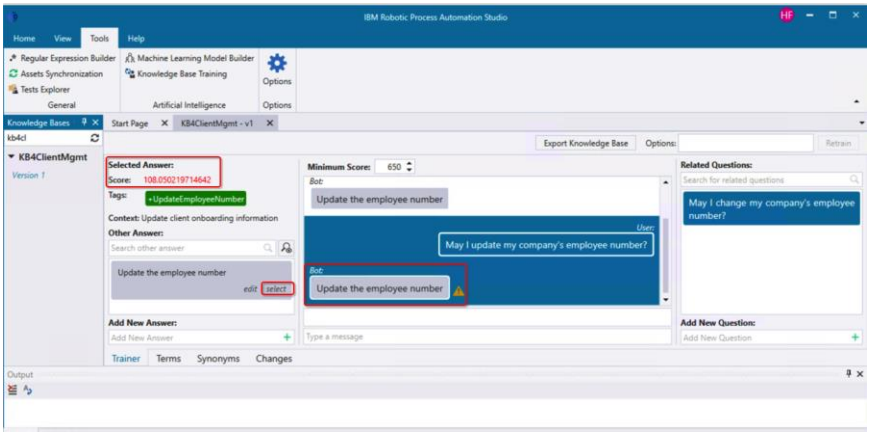
   Context: Click the **Add New** button and enter **Update client onboarding information**.



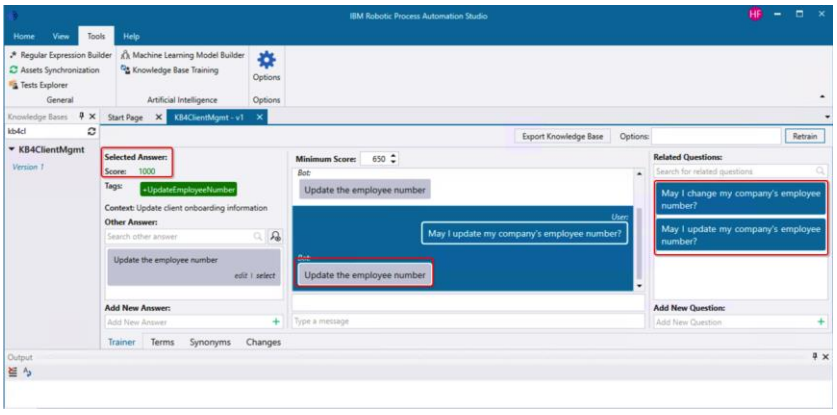
6. The bot will automatically pick up the answer and change from "*No answer found, click to add new one*" to the answer added above. In the **Selected Answer** panel, it will also list the score, which is between 0 and 1000. 1000 means that it is completely certain that it found the correct answer. Click the **Retrain** button in the top-right corner to train the knowledge base model with the question and answer.



7. Enter another question “May I update my company’s employee number?”. The bot will get the same answer, but the score is only 108 with the Yellow triangle besides to answer.

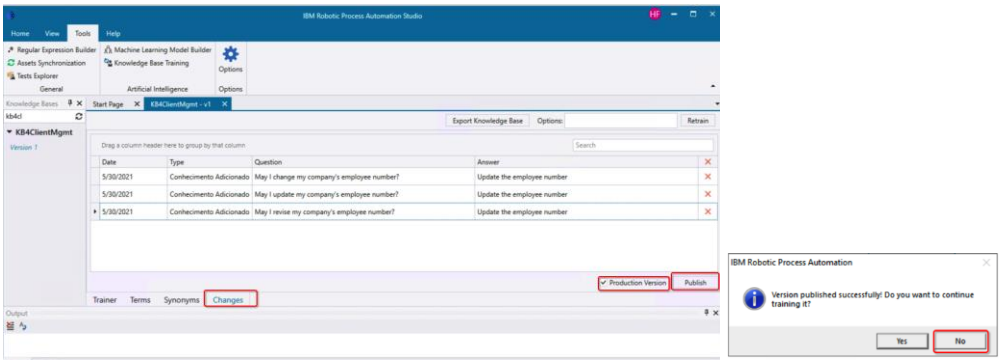


8. To increase score, click **select** from **Selected Answer** panel, then click **Retrain**. It will change the core to 1000 which means the correct answer.



You can continue training the knowledge base model to make it more comprehensive and answer more questions. Instead of continuing to train the knowledge base model with more question which is not the focus of this lab, we will publish the trained knowledge base and use it in chatbot.

9. Switch to the **Changes** tab and **check** the **Production version** checkbox. **Click Publish** to publish the knowledge base model to tenant and select **No** in following pop-up window.



4.1.3 Develop Chatbot Script

You have created and trained a knowledge base model. You can read the documention to learn how to use the knowledge base in a chat. Any automation can use a knowledge base model. However, the most common applications for knowledge base models are:

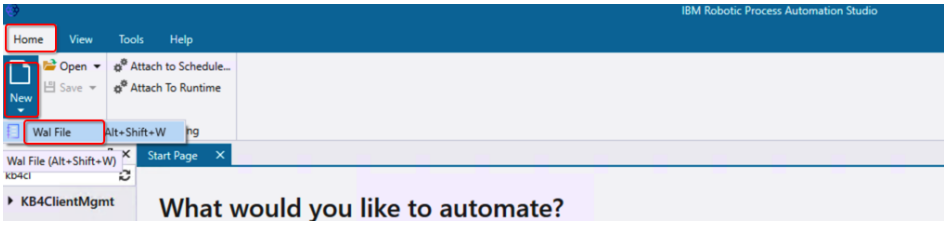
- Decision making based on uncontrolled text input.
- Non-linear interaction with user by text input.

IBM RPA provides two commands **Answer question** (answerQuestion) and **Bot ask and answer** (botAnswerQuestion) to handle questions using a knowledge base model. They will return the best answer along with tags, score, and context. These two commands also return alternative answers. The bot then must use the returned response in its favor during the automation, either for decision making by flow control or by returning the best answer to the user.

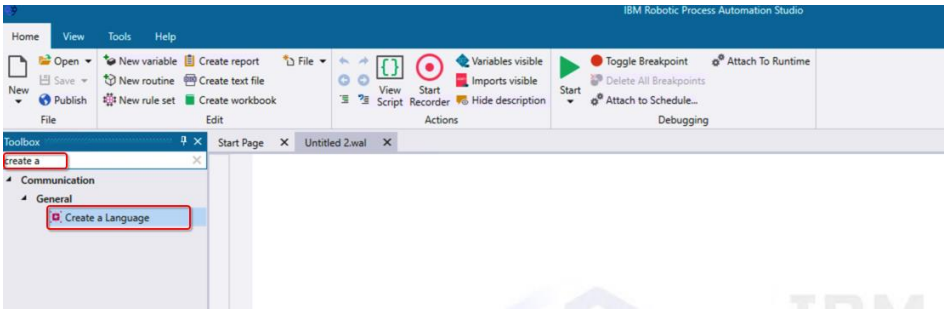
Besides those two commands, IBM RPA also provides a rich set of communication commands for general chatbot development. You can read the documentation to get familiar with those commands.

In this exercise, we will use communication commands to develop a simple chatbot to update the number of employees. An **UpdateEmployeeNumber** script has already been developed and published to the tenant. It will start the Client Management System Java application, query the client information based on the client name, and then update the number of employees. Both the client name and the number of employees are defined as Script Input Parameter.

1. Click **Home** to switch to the IBM RPA Studio script development perspective.
2. Click **New** and select **Wal File**.



3. Enter **create a** in the command search toolbox to filter out the **Create a Language** command. Double click it.



4. Configure the **Create a Language** command as below. Once done, click **Save.**

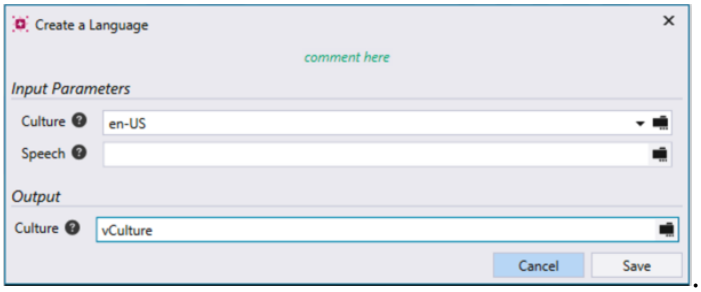
Input Parameters:

Culture: Select **en-US**.

Speech: Leave blank as this option applies to IVR (Interactive Voice Response) only.

Output: Enter **vCulture**.

Once you click **Save**, a variable with this name and type Language will be created.



5. Find and double-click the **Connect to Chatbot or IVR** command from the communication command category. Configure it as below. Once done, click **Save**.

Input Parameters:

Type of communication: Select **Chat**.

Language: Click the icon and select variable **vCulture** defined above.

Timed Out: Leave it blank.

This parameter is to specify a routine to be executed if there is no response from the user within the time specified in the **Timeout** parameter.

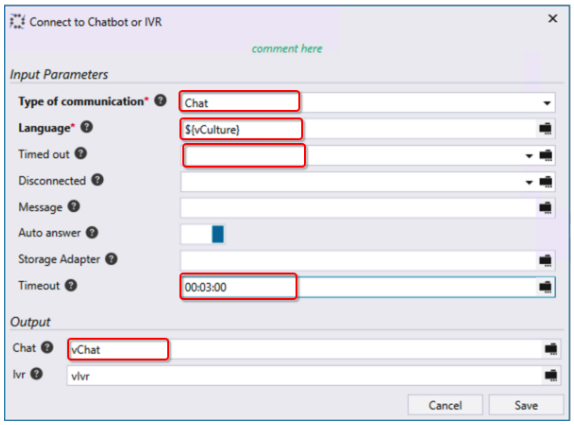
Timeout: Enter **00:03:00** to specify that the chat will be timed out if there is no response with 3 minutes.

Please note if **Timeout** is not specified, the execution uses the context timeout defined by the **Set Timeout** command. If this command is not used either, the default timeout is 5 seconds.

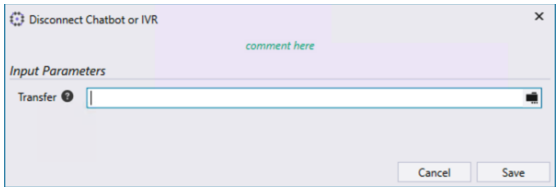
Output:

Chat: Enter **vChat**.

Ivr: Leave it blank or enter **vIVR.** IVR is not covered in this lab.



6. Once you click **Save**, it will automatically add a **Disconnect Chatbot or IVR** command. Leave the **Transfer** field blank and click **Save**.



Next is to use communication commands to develop chatbot scripts to interact with user. You can use the **Bot Ask** command to get input first, then use **Answer Question** to get the answer. Or you can just use **Bot Ask and Answer** command to get input and answer together. We will use the latter one in this lab.

7. First to use the **Bot Ask** command to request the client's company name. Drag and drop it to before the **Disconnect Chatbot or IVR** command. Configure it as below. Once done, click **Save**.

   Input Parameters:

   Language: Click the icon and select variable **vCulture**.

   Text: This is the question asking for input from user. Enter **Hello, may I have your company name please?**.

Output:

Value: Returns the company name, enter **vClientName**.

Success: Returns a Boolean value indicating if this command execution was successful or not, enter **vSuccess**.

8. Find the **Bot Ask and Answer** command. Drag and drop it to before the **Disconnect Chatbot or IVR** command. Configure it as below. Once done, click **Save**.

   Input Parameters:

   Language: Click the icon and select the variable **vCulture**.

   Text: This is the question to ask to the user. Enter a generic question like **Hello, what can I help you?**

   Sub-rountine: Leave it blank.
   This specifies the subroutine that runs if there is no response from the user within the duration specified in the Timeout parameter.

   Knowledge Base: Enter the knowledge base model name you created, trained and published to the tenant. Please make sure the right name.

   This is the knowledge base model to be used for questions and answers.

   Version: Enter **2** or leave it blank.

   This is the knowledge base model version to be used by bot. If the version number is not specified, the bot will use the production version by default, or will use the version you specified here.

   Minimum score: Enter 850.

This is to specify the minimum acceptable score indicating whether the answer is correct in relation to the question when trying to get an answer from knowledge base model.

Timeout: Enter 00:05:00.

Output:

Utterance: Enter **vUtterance** to hold the first text entered by the user.

Time Out: Enter **vTimeout** to hold if the execution of this command is timeout or not.

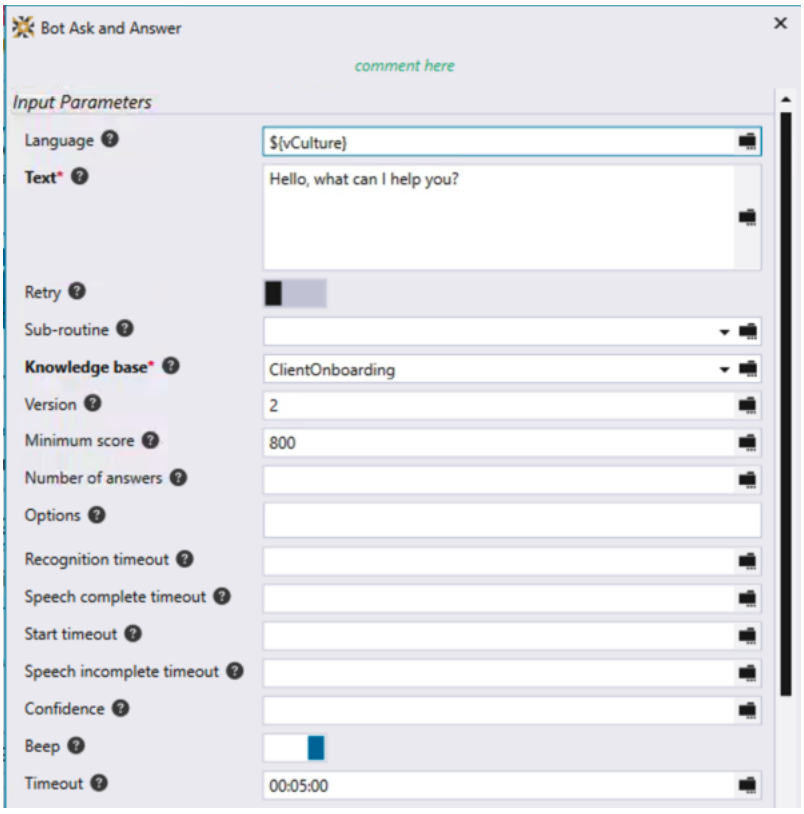
Answer: Enter **vAnswer** to hold the best answer that gives the highest score to the question asked.

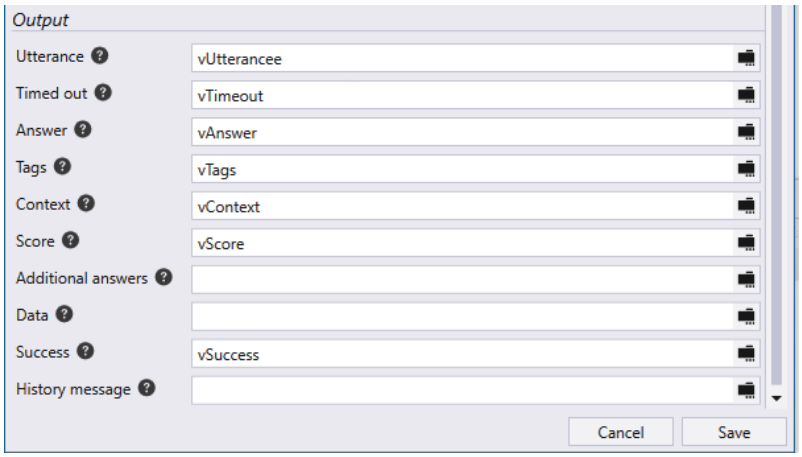
Tags: Enter **vTags** to hold the conversation question context tags.

Context: Enter **vContext** to hold the conversation context.

Score: Enter **vScore** to hold the score for the best answer.

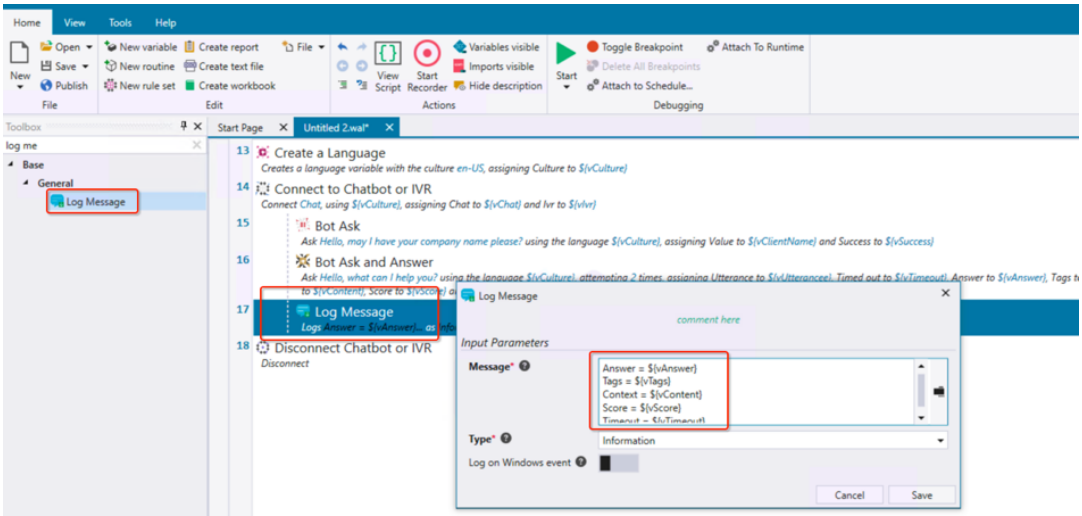
Success: Enter **vSuccess** to hold "True" if the script executed successfully or "False" if not.





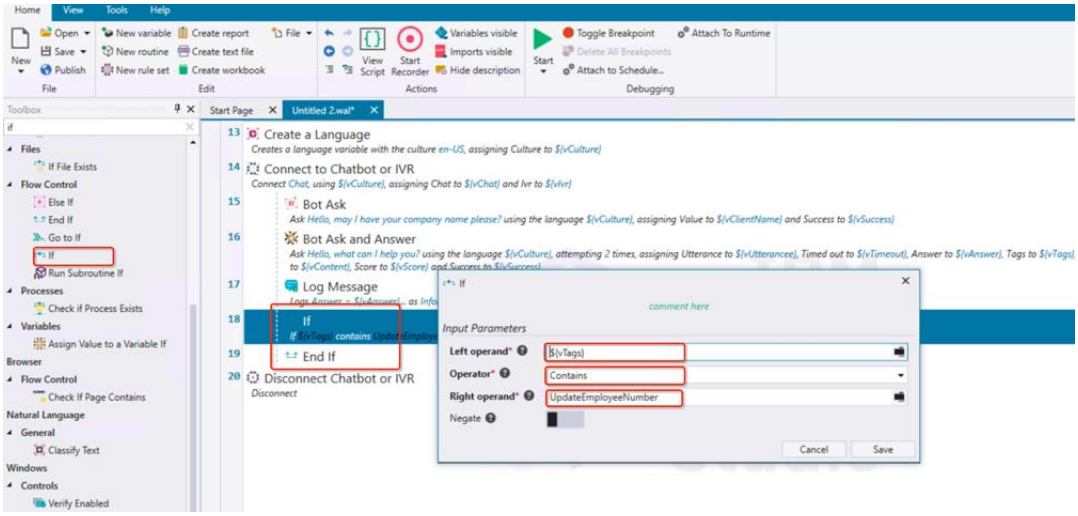
After you click **Save**, your script should be similar as the one below. The **Bot Ask and Answer** command will get the best answer to the question. Next, we will add flow control to update the number of employees.

9. Add a **Log Message** command. Put it before the **Disconnect Chatbot or IVR** command, configure it as below to log he answer, tags, context, score and whether chatbot is timeout or not.

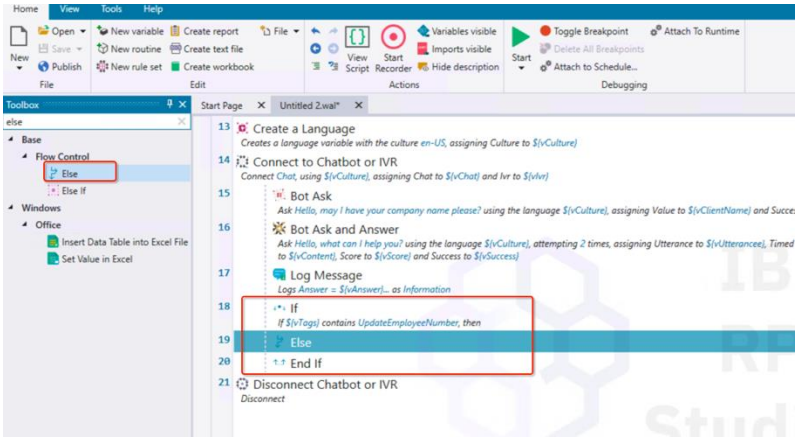


10. Add an **If** command before the **Disconnect Chatbot or IVR** command. Configure it as below which will add flow control if the answer tags contain UpdateEmployeeNumber. Once done, click **Save**.

Note: In a real project, you will likely need to add more conditional decision logic to control the flow.



11. Add an **Else** command in the middle of the **If/End If** command**.**



12. If the user expects to update the number of employees, the chatbot needs to request the number of employees first. Add a **Bot Ask Number** command and put it in the middle of the **If/Else** command. Configure it as below. Once done, click **Save**.

    Input Parameters:

    Language: Select the variable **vCulture** using the picker.

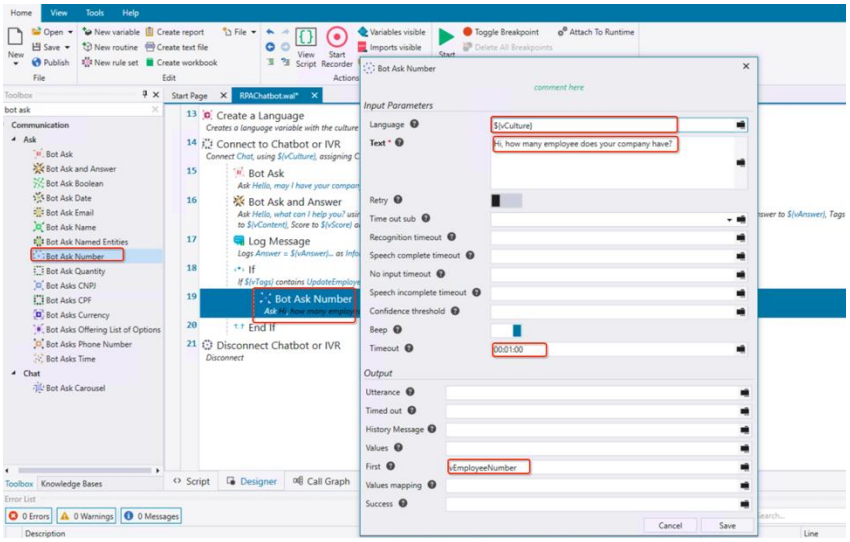
    Text: Enter **Hi, how many employees does your company have?**

    Timeout: Set it to **00:03:00**.

    Output:

    This command has many output parameters. To simplify the exercise, we will only care about one of them.

    First: Enter **vEmployeeNumber** to hold the first number recognized by the bot.

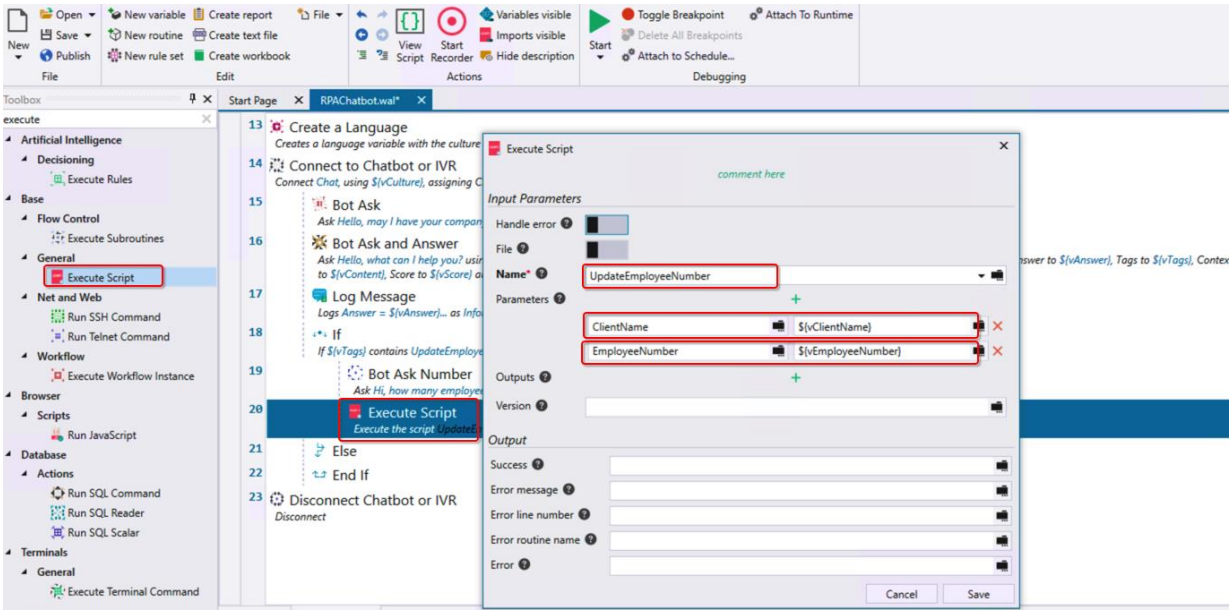


13. Next is to call the **UpdateEmployeeNumber** script to update the employee number. Drag and drop an **Execute Script** command after the **Bot Ask Number** command. Configure it as below. Once done, click **Save**.

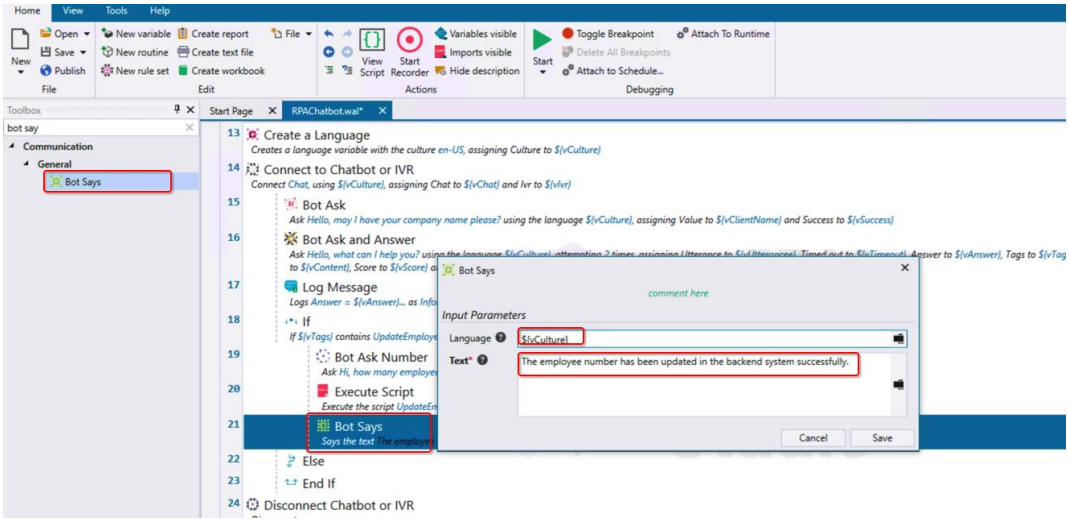
    Input Parameters:

    Name: Select **UpdateEmployeeNumber**.

    Parameters: Map **ClientName** and **EmployeeNumber**.

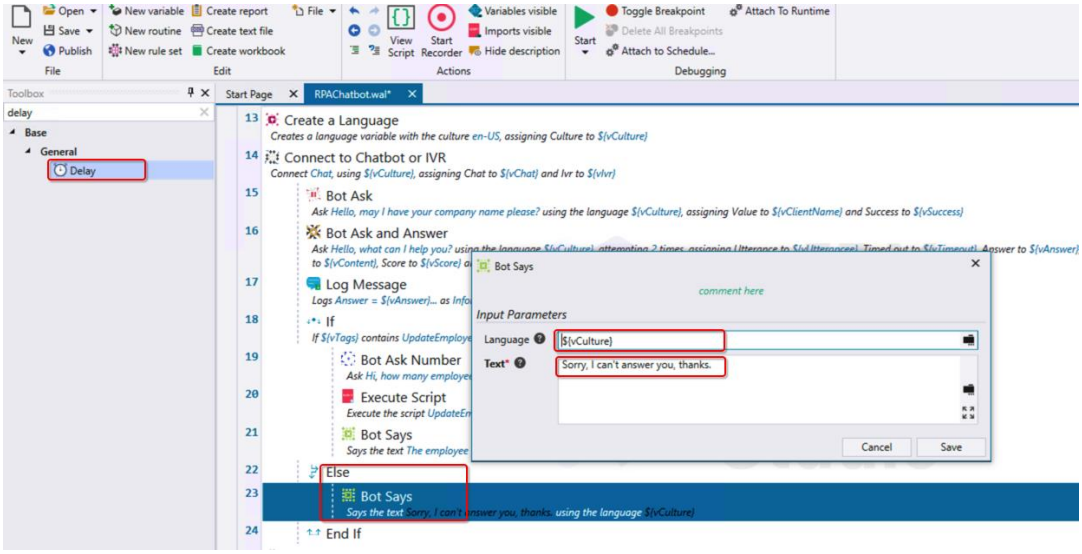


14. Drag and drop the **Bot Says** command after the **Execute Script** command. Configure it as below. For the **Text** parameter, enter something like **The employee number has been updated in the backend system successfully.** Once done, click **Save**.

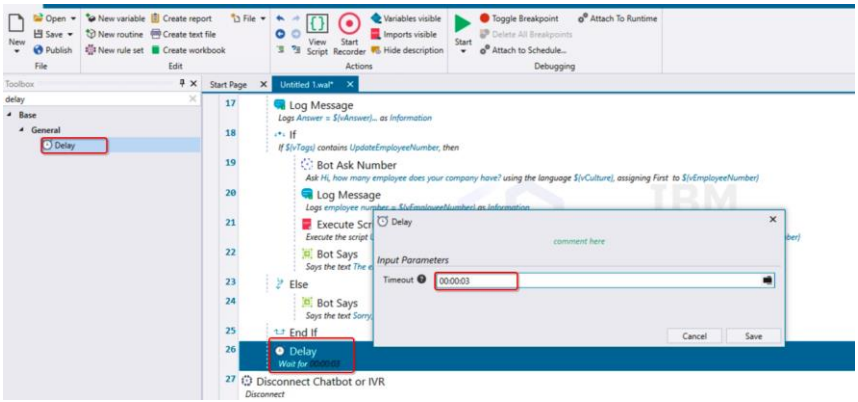


In real project, If the bot can't answer a user's question, the bot needs to have more sophisticated logic to handle the situation. In this lab, it will simply print a message and close the chatbot. You can use other IBM RPA communication commands to further enhance the logic.

15. Drag and drop another **Bot Says** command between the **Else/End If** commands. Configure the command as below. For Language select the variable **vCulture**. For Text, enter a message, for example **Sorry, I can't answer you, thanks**. Once done, click **Save**.

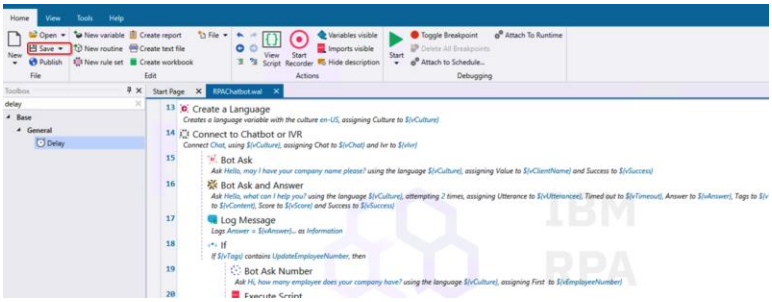


16. Before closing the chatbot, you may want to add a delay command. This will give you a chance to look at the chatbot message. Drag and drop a **Delay** command before the **Disconnect Chatbot or IVR** command. Set the Timeout to **00:00:03**. Once done, click **Save**.



You have successfully created a simple chatbot script. Save your script as **C:\CP4AutoDemo\Chatbot4You.wal** by clicking **Save** in the Studio menu toolbar.

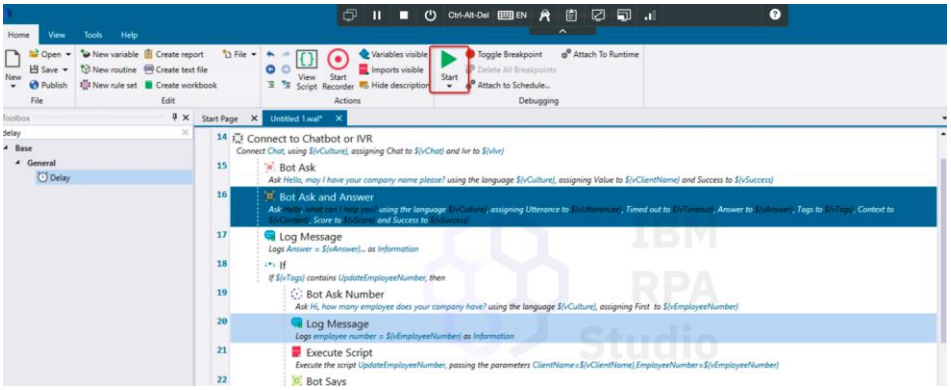
Note: Please make sure to use a unique name e.g. including your initials to avoid naming conflicts when publishing your script to the shared tenant.



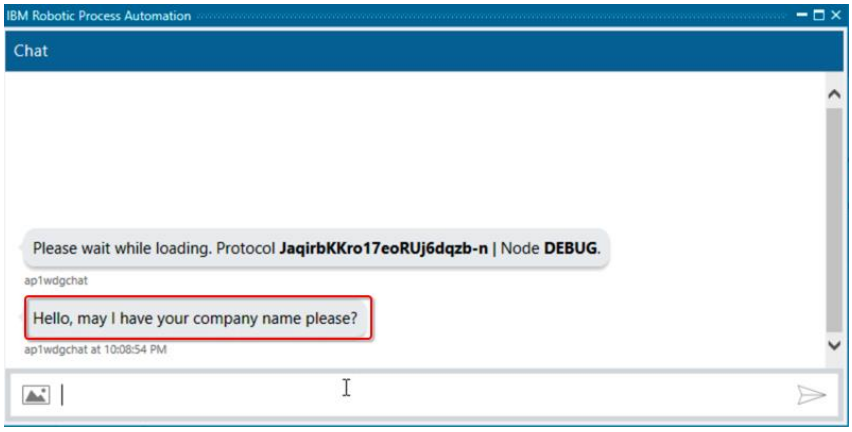
4.1.4 Validation Instructions

Typically, when you develop and debug a chatbot script, you can simply run it from IBM RPA Studio. For a production environment, additional configuration is needed in the tenant to enable execution of chatbots without using Studio. This allows you to embed a chatbot in any application. In this exercise, you will follow the instruction below to validate your chatbot scrips from RPA Studio. It is similar to run and debug any other script.

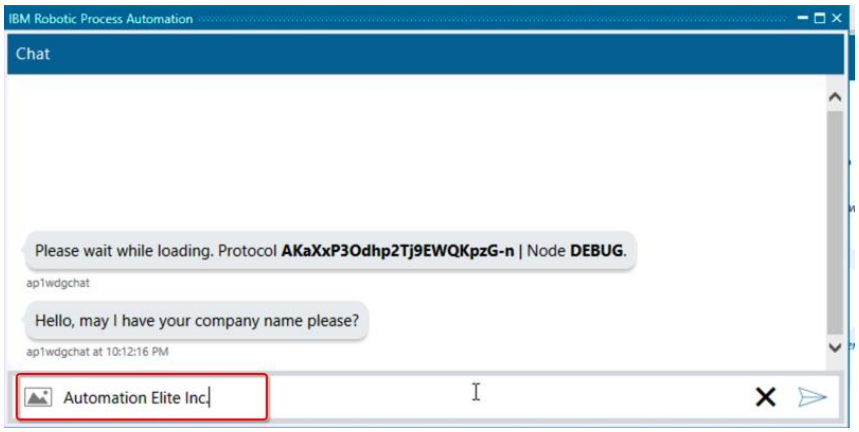
1. Click the **Start** icon in the Studio menu toolbar.



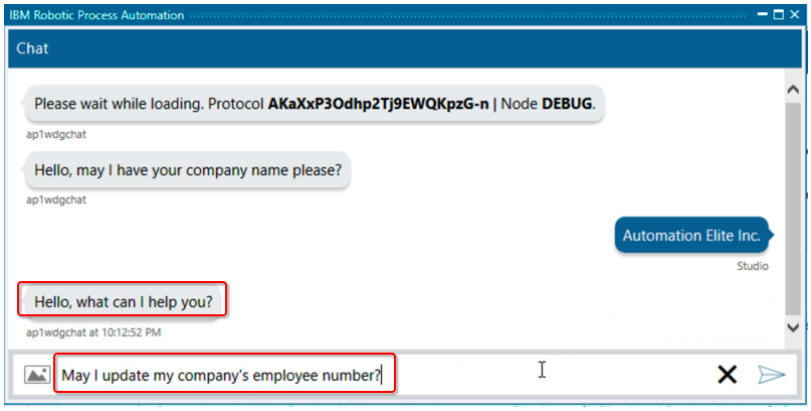
2. The chatbot will start. It may take a few seconds since it needs connect to some services running on the SWAT team tenant. Once connected, it will prompt a message to request you enter your company name.



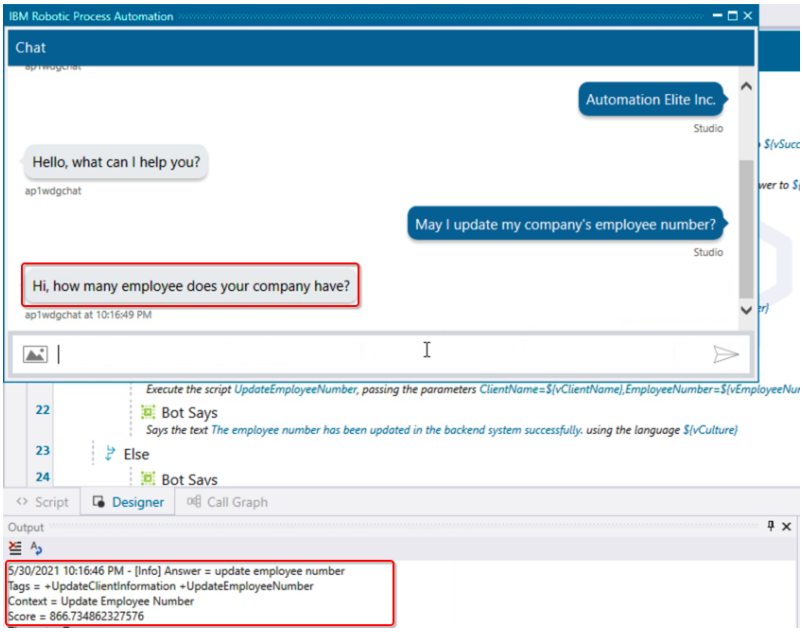
3. Enter **Automation Elite Inc.** in the input text and press enter to send the message.



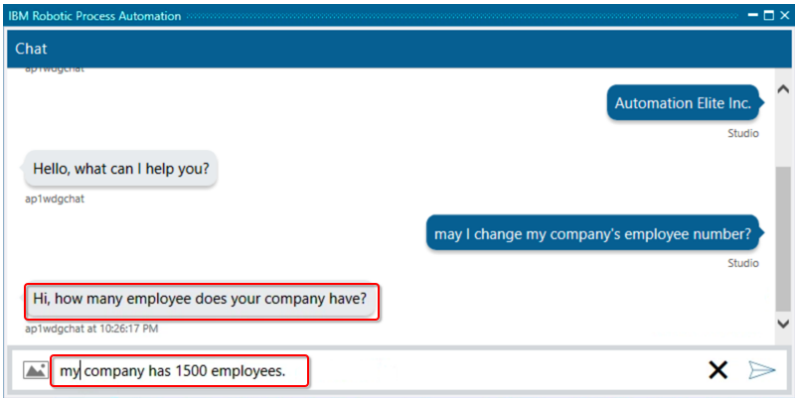
4. The bot will then prompt another message. Enter the following question **May I change my company's employee number?** into the textbox and press **Enter** to send the message.



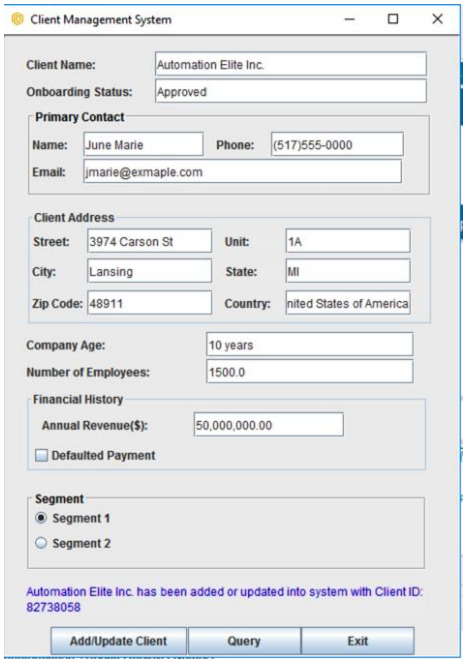
5. The bot will try to get the best answer from the knowledge base model against your question. You can look at the log message in the Studio log window to check the answer, tags and context returned from the knowledge base.



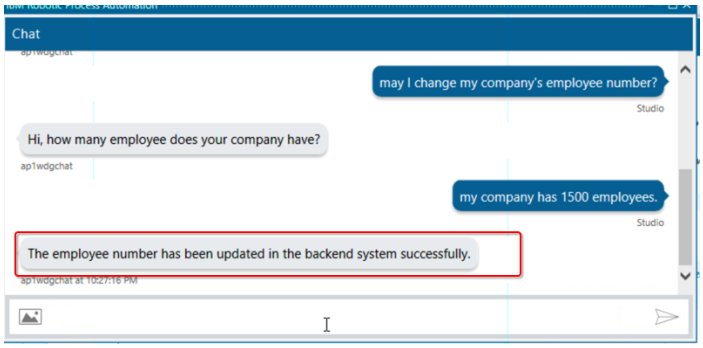
6. Once you see the chatbot message "Hi, how many employees does your company have?", enter something like **My company has 1500 employees**. Once done, press **Enter** to send the message.



7. The bot will extract the employee number from your input first, then start the Client Management System application, and query and update the employee number.



8. Once the employee number has been updated in backend system, the bot will prompt another message indicating the update and then close the chatbot session.



4.1.5 Configure Chat Mapping

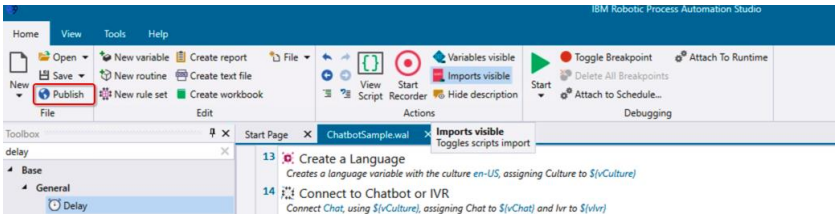
A chatbot script can't be simply scheduled to run from a tenant as a normal bot script. To execute chatbots externally without using IBM RPA Studio, it is required to configure the chat mapping in a tenant. Due to a limitation for the number of bot integration services, you are not able to perform the chat mapping yourselves. Instead, we will provide a guide to work through the process to help you understand the general chat mapping procedures.

Please don't make any change to the chat mapping in the SWAT Team tenant or create new chat mapping.

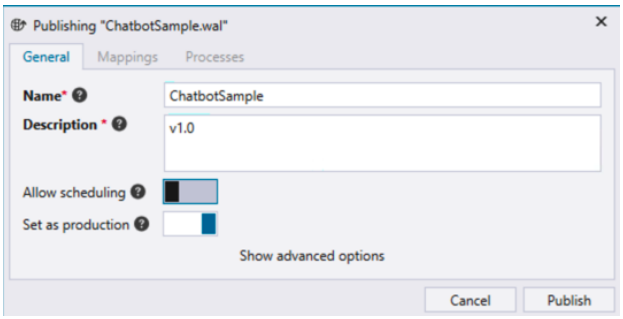
Both available bot integration services have been configured and used for other important business objectives, any change will break them.

Perform the following steps (**without Saving at the end!**):

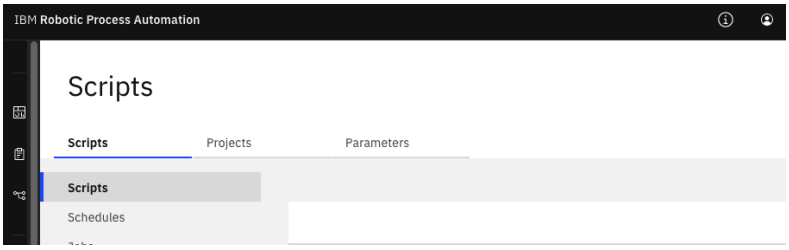
1. Publish your bot script to the tenant by clicking **Publish** in the IBM RPA Studio.



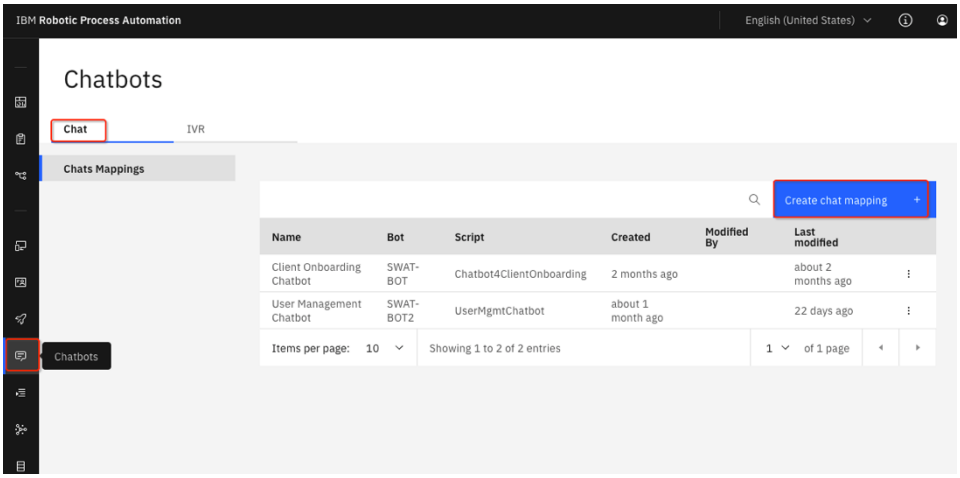
2. Configure the **Publishing** window as below. Give a name to the chatbot script, make sure to use a unique name to avoid naming conflicts and **check Set as production**. Once done, click **Publish** to publish script to the tenant.



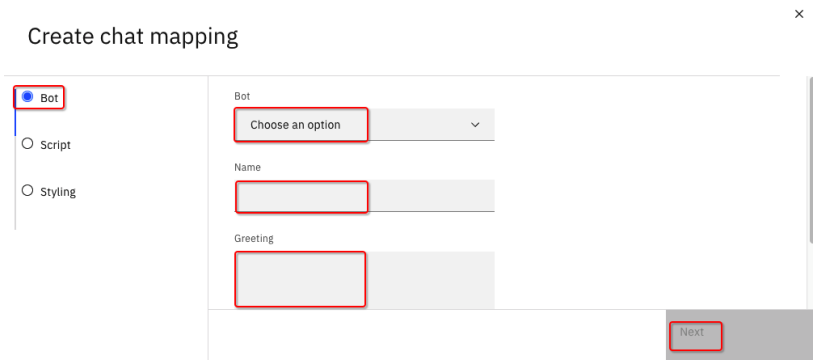
3. Open the web client using Firefox, log in with your RPA account.



4. Click **Chatbot** icon in the left panel, then click **Create Chat mapping**.



5. Follow below steps to configure **New Chat mapping**.
    1. Select appropriate bot integration service, input a chat mapping name and greeting message. Once done, click **Next**.

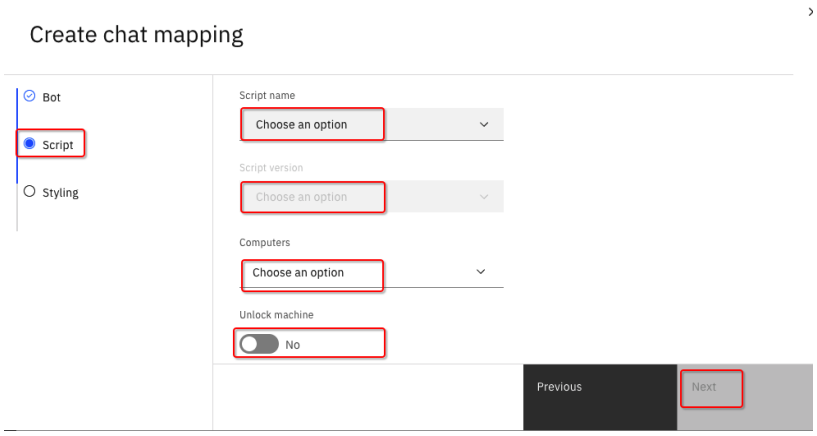


Bot: The bot integration service needs to be configured in the IBM RPA SaaS tenant. This can only be configured by the IBM RPA support team. It is recommended to open a support ticket with your tenant information to request help from support team.

Name: Chat mapping name. You can use any name, suggestion is to give a meaningful name.

Greeting: Greeting message when the chatbot initially starts up.

2. Select the chatbot script you published to tenant and its version, then select the computers to execute the chatbot. Once done, click **Next**.



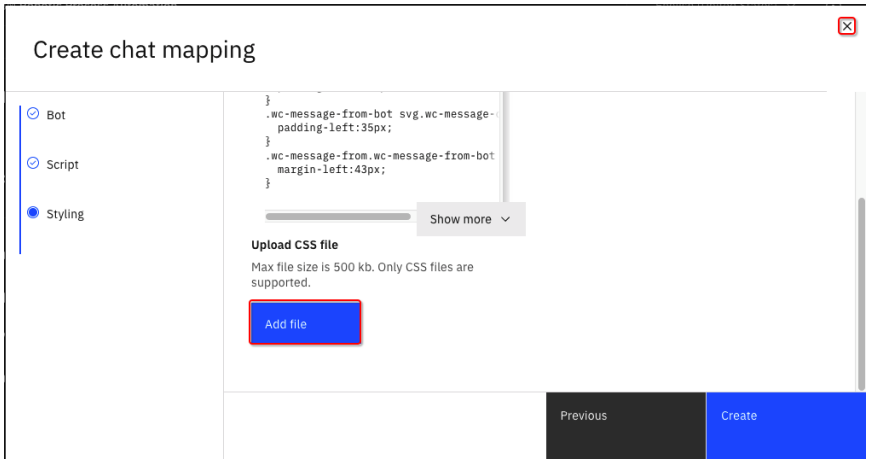
Script Name: Chatbot script name.

Script Version: Chatbot script version.

Computers: A chatbot essentially is still an RPA bot. It needs a machine with the IBM RPA agent installed to execute the chatbot. Specify which machine or group will be used to execute the chatbot.

Unlock machine: Specify whether to unlock the machine to execute the chatbot. If you turn on the option, it is required to configure the RPA agent machine's user credentials for RPA to use to unlock the machine.

3. Click **Add file** button and upload a CSS file. **As already mentioned, don't save your Chat mapping.** Click **X** icon on the top-right window to exit Chat Mapping configuration without saving.



CSS file: The CSS file is used to control chatbot dialog window's user interface. A sample avatar file is uploaded to the RPA folder. You can take it as a sample to customize your chatbot dialog window interface.

After you have configured the chat mapping, you can call your chatbot with an URL. The format of URL is

https://**TenantRegion**bot.wdgautomation.com/integration/**Bot**

TenantRegion: This is the region name of your tenant. You can get it from your web client URL. For example, if your web client URL is https://**ap1**app.wdgautomation.com/, your tenant region is **ap1**.

Bot: This is the bot integration service name that the IBM RPA SaaS support team created for you.

Summary

In this exercise, you have learned:

- The knowledge base model base file format.
- How to use Machine Learning Model Builder to create a knowledge base.
- How to use Knowledge Base Training tool to train and publish a knowledge base.
- How to develop and test a chatbot using the knowledge base model.
- How to develop chatbot scripts using the communication commands.

Congratulations, you have successfully completed IBM RPA Lab!!!