|  |
| --- |
| Techforce RPA Studio |
| User Manual |
| Techforce |

Contents

[1. Techforce.ai RPA Studio 5](#_Toc15324950)

[2. Release Notes 6](#_Toc15324951)

[2.1. Feature Comparison 6](#_Toc15324952)

[2.2. Version: 6](#_Toc15324953)

[3. Installation Instructions 7](#_Toc15324954)

[3.1. OS Compatibility 7](#_Toc15324955)

[3.1.1. 7](#_Toc15324956)

[3.2. Minimum Hardware Requirements 8](#_Toc15324957)

[4. RPA Studio & Worker/Runtime 9](#_Toc15324958)

[4.1. How to download Techforce.ai RPA studio 9](#_Toc15324959)

[4.2. Installing RPA Studio 9](#_Toc15324960)

[4.3. Installing Techforce OCR 12](#_Toc15324961)

[5. Enterprise Manager Console 16](#_Toc15324962)

[5.1. Installing Orchestrator (Windows Os) 16](#_Toc15324963)

[5.1.1. Installing Orchestrator in Producer Machine 16](#_Toc15324964)

[5.1.2. Start the servers 24](#_Toc15324965)

[5.1.3. Configure Database 24](#_Toc15324966)

[6. Receiver Installation 25](#_Toc15324967)

[6.1. Load Balancer Requirements 25](#_Toc15324968)

[6.2. Deployment Options (Single Thread or Multithread) 25](#_Toc15324969)

[7. Architecture Overview & Deployment Models 26](#_Toc15324970)

[7.1. Enterprise Model 26](#_Toc15324971)

[7.2. Hybrid Model 26](#_Toc15324972)

[7.3. SaaS/PaaS Model 27](#_Toc15324973)

[8. Design a Bot – Best Practices & Guidelines 28](#_Toc15324974)

[8.1. 28](#_Toc15324975)

[9. Enterprise Manager Console – Management Guide 29](#_Toc15324976)

[9.1. Process Management 29](#_Toc15324977)

[9.1.1. Adding the Business Process 29](#_Toc15324978)

[9.1.2. Executing a Business Process: 29](#_Toc15324979)

[9.2. Worker Management 30](#_Toc15324980)

[9.3. Worker List 32](#_Toc15324981)

[9.4. Schedules 32](#_Toc15324982)

[9.4.1. Trigger 33](#_Toc15324983)

[9.4.2. Execution Target 36](#_Toc15324984)

[9.4.3. Actions 37](#_Toc15324985)

[9.5. Account Management 37](#_Toc15324986)

[9.5.1. Organization: 38](#_Toc15324987)

[9.5.2. User Management: 38](#_Toc15324988)

[9.5.3. Secret Vault: 39](#_Toc15324989)

[9.5.4. Credentials 41](#_Toc15324990)

[10. Developer Guide 43](#_Toc15324991)

[10.1. Studio Guide 43](#_Toc15324992)

[10.1.1. Menu Bars 43](#_Toc15324993)

[10.2. Play 48](#_Toc15324994)

[10.3. Action Library Panel 51](#_Toc15324995)

[10.3.1. Stage Category 52](#_Toc15324996)

[10.3.2. Desktop Automation: 52](#_Toc15324997)

[10.3.3. Web Automation 53](#_Toc15324998)

[10.3.4. General 53](#_Toc15324999)

[10.3.5. Programming 54](#_Toc15325000)

[10.3.6. Mail Integration 55](#_Toc15325001)

[10.3.7. Cognitive 55](#_Toc15325002)

[10.4. Developer Panel 56](#_Toc15325003)

[10.5. Properties Panel 56](#_Toc15325004)

[10.5.1. Variable Panel 57](#_Toc15325005)

[10.5.2. Logs Panel 57](#_Toc15325006)

[10.6. Actions Guide 58](#_Toc15325007)

[10.6.1. Stage 58](#_Toc15325008)

[10.6.2. Desktop Automation 58](#_Toc15325009)

[10.6.3. Web Automation 60](#_Toc15325010)

[10.6.4. General 62](#_Toc15325011)

[10.6.5. Programming 80](#_Toc15325012)

[10.6.6. Mail Integration 81](#_Toc15325013)

[10.6.7. Cognitive 84](#_Toc15325014)

[10.7. Build A Basic Bot 85](#_Toc15325015)

[11. Security Architecture 90](#_Toc15325016)

[12. Troubleshooting Guide 91](#_Toc15325017)

[13. Analytics 92](#_Toc15325018)

1. Techforce.ai RPA Studio

User Manual

Author: Srinivas Kandimalla

Creation Date: July 25, 2019

Last Updated: July 25, 2019

Document Ref: <Document Reference Number>

Version:  DRAFT 1A

1. **Title, Subject, Last Updated Date, Reference Number**, **and** **Version** are marked by a Word Bookmark so that they can be easily reproduced in the header and footer of documents. When you change any of these values, be careful not to accidentally delete the bookmark. **You can make bookmarks visible by selecting Tools->Options…View and checking the Bookmarks option in the Show region.**

Approvals

|  |  |
| --- | --- |
| <Approver 1> | Hari Krishna Gudladona |
| <Approver 2> | Rambabu Datla |

1. Release Notes
   1. Feature Comparison
   2. Version:
2. Installation Instructions
   1. OS Compatibility

**Bot Designer:**

|  |  |  |  |
| --- | --- | --- | --- |
| **On-Premise** | **GCP** | **MS Azure** | **AWS** |
| Windows 10 / Windows Server 2012 R2 / Windows Server 2016 | Windows Server 2012 R2 Data Center Core / Windows Server 2012 R2 Datacenter / Windows Server 2016 Data Center / Windows Server 2016 Datacenter Core | Windows Server 2012 R2 Data Center / Windows 10 Pro Ver 1809, 1803 / Windows Server 2016 Data Center | Microsoft Windows Server 2016 Base/ Microsoft Windows Server 2012 R2 Base |

**Enterprise Manager Console:**

|  |  |  |  |
| --- | --- | --- | --- |
| **On Premise** | **GCP** | **MS Azure** | **AWS** |
| Windows 10 / Windows Server 2012 R2 / Windows Server 2016 / Ubuntu 16.04 LTS / Red Hat Linux 7 | Windows Server 2012 R2 Data Center Core / Windows Server 2012 R2 Data Center / Windows Server 2016 Datacenter / Windows Server 2016 Data Center Core / Ubuntu 16.04 LTS / Red Hat Linux 7 | Windows Server 2012 R2 Data Center / Windows 10 Pro Ver 1809, 1803 / Windows Server 2016 Data Center / / Ubuntu 16.04 LTS / Red Hat Linux 7.6 | Microsoft Windows Server 2016 Base/ Microsoft Windows Server 2012 R2 Base / Red Hat Enterprise Linux 8 |

**Worker:**

|  |  |  |  |
| --- | --- | --- | --- |
| **On-Premise** | **GCP** | **MS Azure** | **AWS** |
| Windows 10 / Windows Server 2012 R2 / Windows Server 2016 | Windows Server 2012 R2 Data Center Core / Windows Server 2012 R2 Data Center / Windows Server 2016 Data Center / Windows Server 2016 Data Center Core | Windows Server 2012 R2 Data Center / Windows 10 Pro Ver 1809, 1803 / Windows Server 2016 Data Center | Microsoft Windows Server 2016 Base/ Microsoft Windows Server 2012 R2 Base |

* 1. Minimum Hardware Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Hardware | Bot Designer | Worker | Enterprise Manager Console | Microsoft SQL Server | PostgreSQL Server |
| **Processor** | Intel Core i5 2.6 GHz | Intel Core i5  2.6 GHz | 8 core - 3.0 GHz Intel Xeon Platinum processor (Turbo Boost to 3.5 GHz) | 4 core Intel Xeon Processor | 2 core Intel Xeon Processor |
| **RAM** | 8 GB | 8 GB | 16 GB | 8 GB | 4 GB |
| **Storage** | 10 GB | 10 GB | 50 GB | 50 GB | 10 GB |
| **Network** | 1 GbE | 1 GbE | 10 GbE | 1 GbE | 1 GbE |
| **Other** | Microsoft .NET Framework 4.6 (Windows 8.1 and Windows Server 2012 R2: 4.6.1) | Microsoft .NET Framework 4.6 (Windows 8.1 and Windows Server 2012 R2: 4.6.1) | Microsoft .NET Framework 4.6 (Windows 8.1 and Windows Server 2012 R2: 4.6.1) | Microsoft SQL Server 2012 or later | PostgreSQL Server |

1. RPA Studio & Worker/Runtime
   1. How to download Techforce.ai RPA studio

To download Techforce.ai RPA studio, click on the below URL. The file size would be around 330 MB.

<https://github.com/digitamizers/techforce-ide/releases/download/v1.0.0-alpha.39/techforceide-install-v1.0.0-alpha.39.exe>

* 1. Installing RPA Studio

RPA studio requires **945 MB** hard disk space to install.

After downloading the RPA studio, move the downloaded studio.exe file to the desired folder where it needs to be installed.

Right-click on the installer and select “Run as administrator” so that RPA studio will have “Admin” privileges.

The following image is an example of how to run as administrator.

A screenshot of a social media post

Description automatically generated

Windows will ask for permission stating “*Do you want to allow the unknown publisher to install the application”.* Click on YES, and the installation process will start.

Accept the license agreement of Techforce.ai to proceed for the further installation process.

A screenshot of a social media post

Description automatically generated

The installation process will show the folder selection screen.

A screenshot of a cell phone

Description automatically generated

Click on BROWSE and select a folder which does not contain a space in its name like shown in the image.

After selecting the folder, click on “INSTALL” which will start installing RPA studio.

|  |  |
| --- | --- |
| TIP ☞ | On the folder selection screen, make sure to select a folder, which **does not contain any space in the folder name.** If the selected folder contains any space in its name, then the RPA scripts built using the studio cannot be executed. |

A screenshot of a cell phone

Description automatically generated

The installation will take a bit time, and after installing the Studio, click on CLOSE button to proceed for further process.

A screenshot of a cell phone

Description automatically generated

RPA studio installer will prompt to download a dependency file. Install the dependency file.

A screenshot of a cell phone

Description automatically generated

Click on the “OK” button as shown in the above image, to install Techforce OCR, which is a dependency to RPA studio.

* 1. Installing Techforce OCR

Techforce OCR needs 164MB of hard disk space for installation. In the first screen of Techforce OCR installation, select preferred language, and click OK, which will start the installation of Techforce OCR.

A screenshot of a cell phone

Description automatically generated

In the next screen of OCR installation, click NEXT to continue the installation process.

Accept the license agreement of Techforce OCR to continue the installation process.

A screenshot of a social media post

Description automatically generated

After accepting the agreement, specify/select who should have the user level access to OCR Select appropriately from. “Install for anyone using this computer or Install just for me”.

A screenshot of a social media post

Description automatically generated

Select any one of the options and click on NEXT button. In the next screen choose the components that are to be installed or else just click on NEXT to continue the installation process.

A screenshot of a social media post

Description automatically generated

After selecting the components, the installation process will show the folder path selection. Keep the default storage space or else specify the installation folder.

A screenshot of a social media post

Description automatically generated

Select the options for start menu action and then click on Install to.

A screenshot of a cell phone

Description automatically generated

After installation, click on NEXT button and then click on FINISH button to complete the RPA Studio installation.

1. Enterprise Manager Console
   1. Installing Orchestrator (Windows Os)

Installation documents will be provided by TechForce in zip files.

databaseScripts, producer, receiver are zip files.

**Step1:** In PRODUCER machine, unzip ‘databaseScript’ zip file.

**Step2:** Open DBScripts file from above unzipped.

**Step3:** Copy Scripts from either ‘MSSQL’ or ‘MYSQL’ as per the requirement/DB and Execute/RUN these scripts in respective DB. (Paste in Query Editor & Execute/RUN it).

**Step4:** Copy data in files TF\_USER\_LOGIN & TF\_USER\_TOKEN (unzipped from ‘databaseScript’) and paste in respective tables in DB.

**Step5:** Now, UNZIP ‘producer’ file, in PRODUCER machine.

* + 1. Installing Orchestrator in Producer Machine

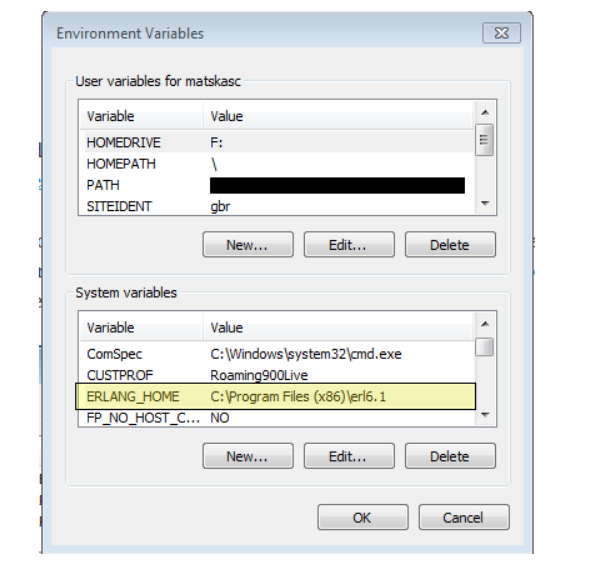
To install Orchestrator successfully in local machine, the following software should be installed first.

1. Erlang
2. Rabbit MQ
3. Hashi corp vault
4. Node JS
5. Yarn.
   * + 1. Erlang:

To install Erlang go to the URL<https://www.erlang.org/downloads> (or the same file is unzipped from ‘producer’ file provided by TechForce)

Click on “[OTP 21.2 Windows 64-bit Binary File](http://erlang.org/download/otp_win64_21.2.exe)“which will download the installer. Navigate to the downloaded path and install Erlang in the machine.

Ensure that the appropriate environment variable (ERLANG\_HOME) has been created during the installation. If, for any reason, the environment variable is missing, create it manually as per the image below:

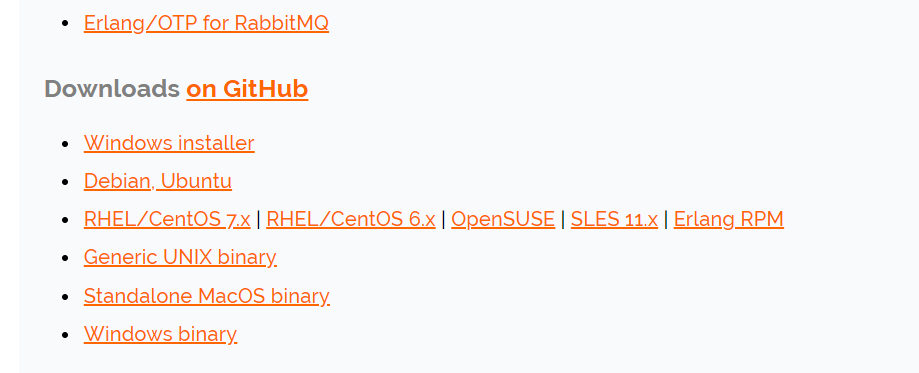


* + - 1. Rabbit MQ:

To install Rabbit MQ go to the URL

<https://www.rabbitmq.com/download.html> (or the same file is unzipped from ‘producer’ file provided by TechForce)

After opening the URL click on “Windows Installer”, which will start downloading Rabbit MQ.



Now open the downloaded folder and double click on Rabbit MQ installer. In the pop-up a screen that shows **“Do u want to allow this app from an unknown publisher”** click **“YES”.**

Click “NEXT” at “choose components” step.

At “Choose Installation Location” browse a specified location to install Rabbit MQ or Click “INSTALL”.

Open an elevated command line (Run as Administrator)

Navigate to the sbin directory of the RabbitMQ Server installation directory. Copy the path and paste it in the “PATH” variable in “SYSTEM VARIABLES” under “ENVIRONMENT VARIABLES”.

Open the command prompt from “C:\Program Files (x86) \RabbitMQ Server\rabbitmq\_server-3.3.4\sbin” path and run the below command to start/run Rabbit MQ plugin.

Run the following command to enable the plugin **rabbitmq-plugins.bat enable rabbitmq\_management**

1. Navigate to C:\Users\<currentLoggedInUser>\AppData\Roaming\RabbitMQ
2. Create a new file rabbitmq.config, and paste the following data:

[{rabbit,

[{heartbeat, 0}]

}].

1. Open Advanced.config from the same folder. Remove the data and paste the following data and save it.

[{rabbit,

[{heartbeat, 0}]

}].

1. Open an elevated command prompt (run as administrator type) and paste the following commands:
2. rabbitmq-service.bat stop
3. rabbitmq-service.bat remove
4. rabbitmq-service.bat install
5. Start the rabbitMQ service using the following command:

rabbitmq-service.bat start

* + - 1. HashiCorp Vault

To download HashiCorp Vault, go to the URL

<https://www.vaultproject.io/downloads.html> (or the same file is unzipped from ‘producer’ file provided by TechForce))

Above URL will download a .zip file. Extract and navigate to the extracted folder and copy the folder path and set that at “PATH” in “ENVIRONMENT VARIABLES”. Below is the folder path.



To Navigate to environment variables, try the below option.

Right-click on “My computer/This PC”, click “Properties”, Click “Advanced system settings”, Click “Environment Variables”.

Now paste previously copied “HashiCorp Vault” path in “PATH” variable under “System variables”.

* + - 1. Starting Hashi Corp Vault:

1. Create a folder anywhere to store the secret data
2. Copy the folder path
3. Create a file config.hcl
4. Paste the below content in the config.hcl file

disable\_cache = true

disable\_mlock = true

ui = true

listener "tcp" {

address = "127.0.0.1:8200"

tls\_disable = 1

}

storage "file" {

path = "E:\\Vault\\FileSystem"

}

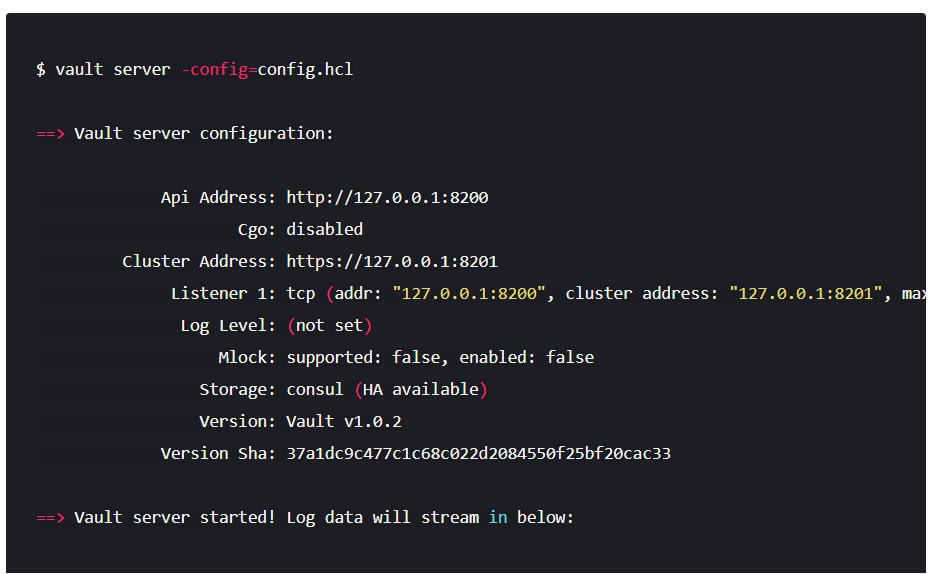
1. Replace the yellow highlighted part with the path of folder created in step one
2. Set two system environment variables

VAULT\_ADDR = <http://127.0.0.1:8200>

VAULT\_API\_ADDR = <http://127.0.0.1:8200>

* + - 1. Starting the vault server

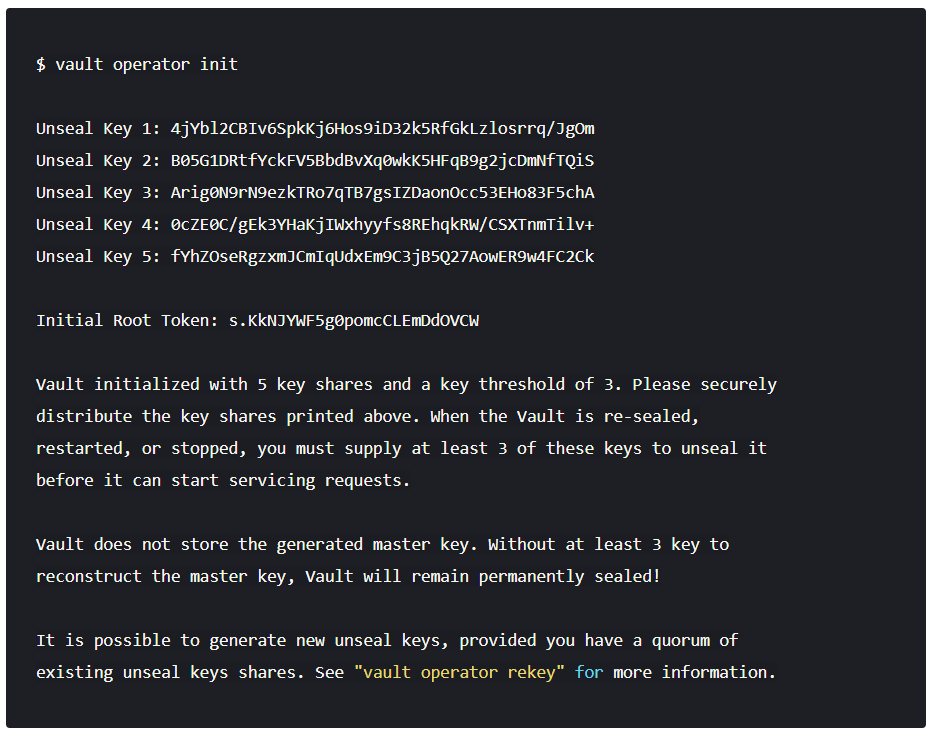
With the configuration in place, starting the server is simple, as shown below. Modify the -config flag to point to the proper path where the configuration was saved above.



* + - 1. Initializing the vault

Initialization is the process configuring the Vault.

During initialization, the encryption keys are generated, unseal keys are created, and the initial root token is setup. To initialize Vault, use vault operator init in a new command prompt



Initialization outputs two incredibly important pieces of information - the **unseal keys** and the **initial root token**. This is the only time ever that all of this data is known by Vault, and also the only time that the unseal keys should ever be so close together.

Save the above 5 unseal keys and initial root token in a file for future use.

**Unseal Vault:** Every initialized Vault server starts in the sealed state. From the configuration, Vault can access the physical storage, but it can't read any of it because it doesn't know how to decrypt it. The process of teaching Vault how to decrypt the data is known as unsealing the Vault.

Unsealing has to happen every time the Vault starts. It can be done via the API and via the command line. To unseal the Vault, it is a must to have the threshold number of unseal keys. In the output above, notice that the "key threshold" is 3. This means that to unseal the Vault, 3 keys are required out of the 5 keys that were generated.

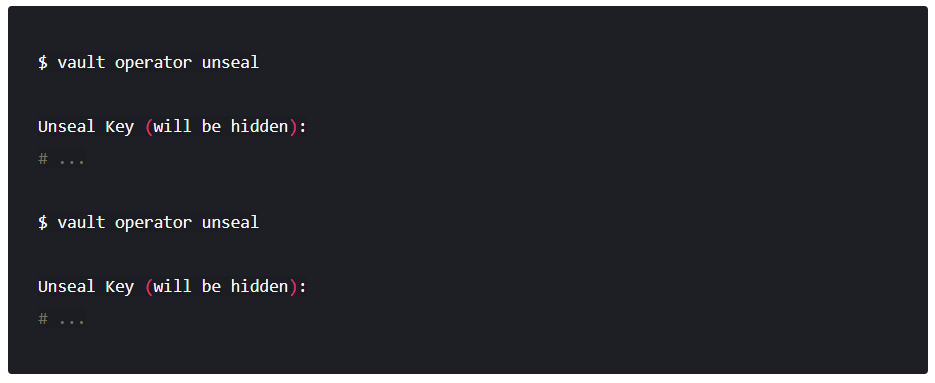
Begin unsealing the Vault:



After pasting a valid key and confirming, the Vault is still sealed, but progress is made. Vault knows it has 1 key out of 3. Due to the nature of the algorithm, Vault doesn't know if it has the correct key until the threshold is reached.

Also, notice that the unseal process is stateful. One can go to another computer, use vault operator unseal, and as long as it is pointing to the same server, that other computer can continue the unsealing process. This is extremely important to the design of the unseal process: multiple people with multiple keys are required to unseal the Vault. The Vault can be unsealed from multiple computers and the keys should never be together. A single malicious operator does not have enough keys to be malicious.

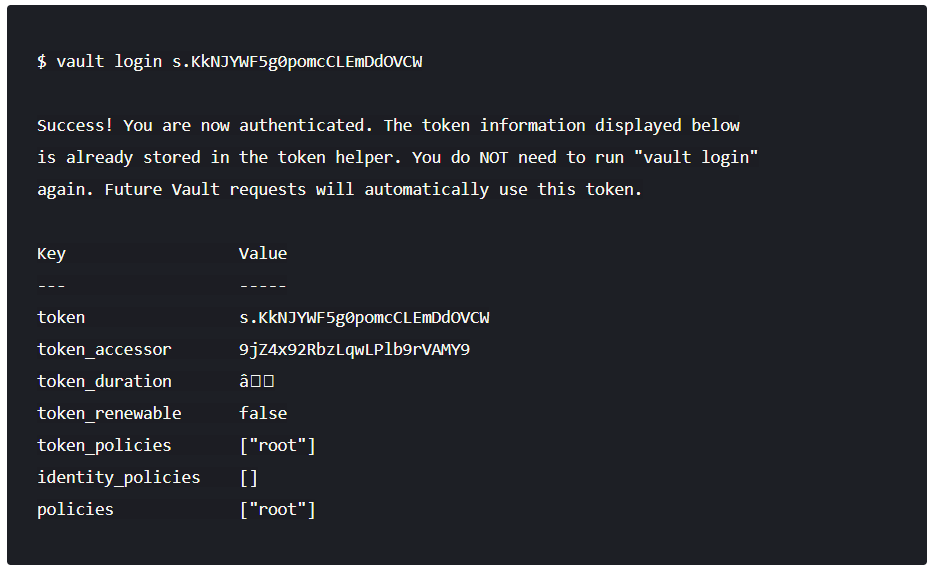
Continue with vault operator unseal to complete unsealing the Vault. To unseal the vault, use three different keys. Repeating the same key will not work. If the keys are used correctly, the following output will display:



When the value for Sealed changes to false, the Vault is unsealed:



Finally, authenticate as the initial root token (it was included in the output with the unseal keys):



Navigate to ENVIRONMENT VARIABLES. Following are the steps to navigate to environment variables:

Right-click on “My computer/This PC”

Click “Properties”, and click “Advanced system settings”

Click “Environment Variables”

Create a new variable as “**VAULT\_DEV\_ROOT\_TOKEN\_ID**” and value as the initial root token value.

* + - 1. Node js:

To download Node.js, go to the URL:

<https://nodejs.org/en/download> (or can be unzipped from ‘producer’ file provided by TechForce)

After completion of downloading, install Node. js.

Now download “YARN” to the machine. To download YARN type below mentioned command in command prompt.

“npm install -g yarn”

* + - 1. Open firewall port 5672

Manually permit a program to access the Internet by opening a firewall port if it is known what port it uses and the protocol to make it work.

1. Navigate to Control Panel, System and Security and Windows Firewall.
2. Select Advanced settings and highlight Inbound Rules in the left pane.
3. Right-click Inbound Rules and select New Rule.
4. Add the port that needs to be opened and click NEXT.
5. Add the protocol (TCP or UDP) and the port number-5672into the next window and click NEXT.
6. Select “Allow the connection” in the next window and click NEXT.
7. Select the network type and click NEXT.
8. Name the rule something meaningful and click FINISH.
   * 1. Start the servers
9. Set a system variable **REACT\_APP\_IPADDRESS**, **local iP** (public IP for VM) as its value.
10. Open three different command prompts. Navigate to the Orchestrator\_API, RPA\_authentication, and RPAWebapp (unzipped content from producer.zip file) paths from the three terminals.
11. Type the following commands in each terminal

* npm install
* npm start
  + 1. Configure Database

1. Open browser
2. Open <http://localhost:4050> or http://<localIP>:4050 or http://<publicIP>:4050 (for VM)
3. Get redirected to a DB credentials Page.
4. Select the Database (Dialect) - MySQL or MS SQL
5. Fill all the fields and submit
6. Get redirected login Page.
7. Login to the account using the credentials given below

* Username: techforce
* Password: Welcome

1. Receiver Installation

Following are the steps for Receiver installation:

1. Install Node.js and Python 3.6 or above
2. Unzip the receiver.zip file.
3. Copy the path of Techforce/src/flows (unzipped receiver file) and set it to a system variable TECHFORCE\_HOME
4. Copy the path of Techforce-windows/src (unzipped receiver file) and add it to the PATH variable under system variables
5. Open a command prompt, navigate to Orchestrator\_receiver (unzipped receiver file) and run the below commands

* npm install
* node index.js

1. Open another command prompt, navigate to Orchestrator\_python\_api (unzipped receiver file) and run the below commands

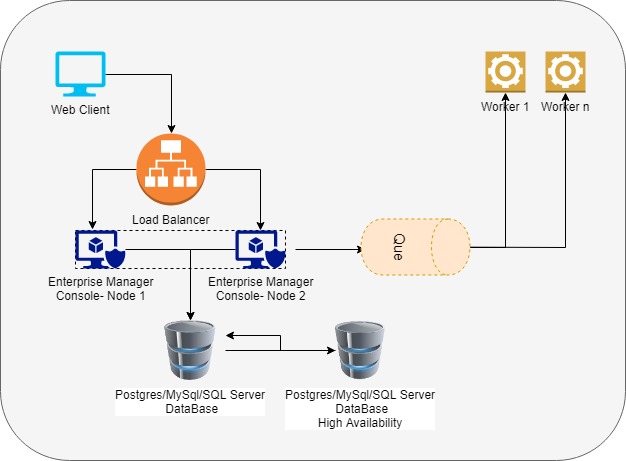
* pip install –r requirements.txt
* python manage.py runserver 0.0.0.0:8000
  1. Load Balancer Requirements

(Content needs to be provided by Srini Kandimalla)

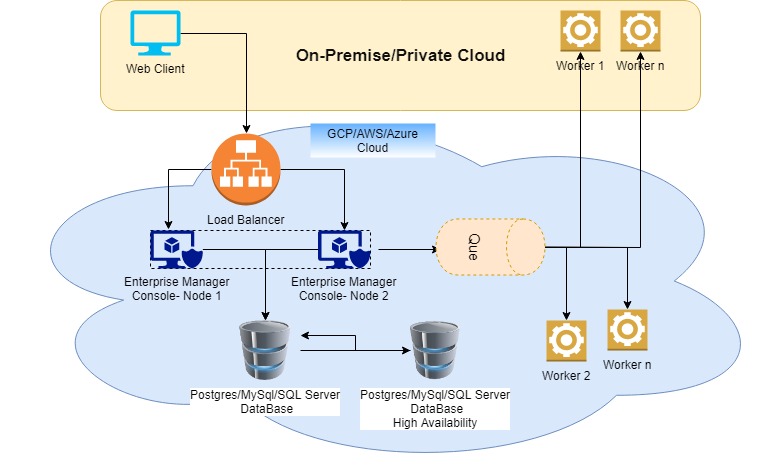
* 1. Deployment Options (Single Thread or Multithread)

(Content needs to be provided by Srini Kandimalla)

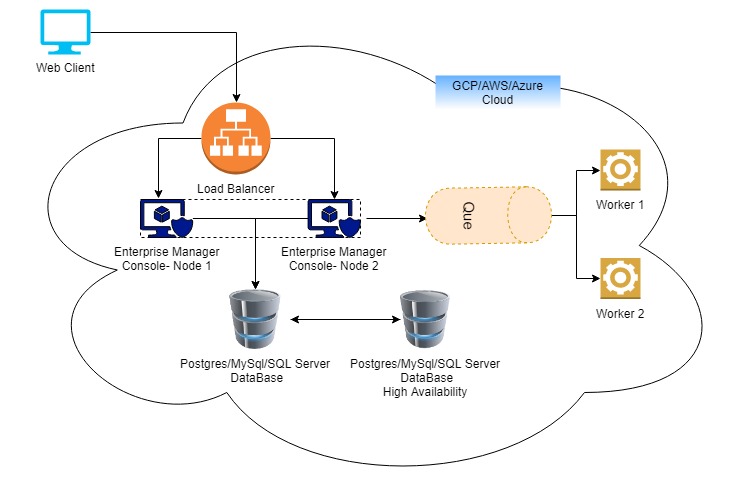
1. Architecture Overview & Deployment Models
   1. Enterprise Model



* 1. Hybrid Model



* 1. SaaS/PaaS Model



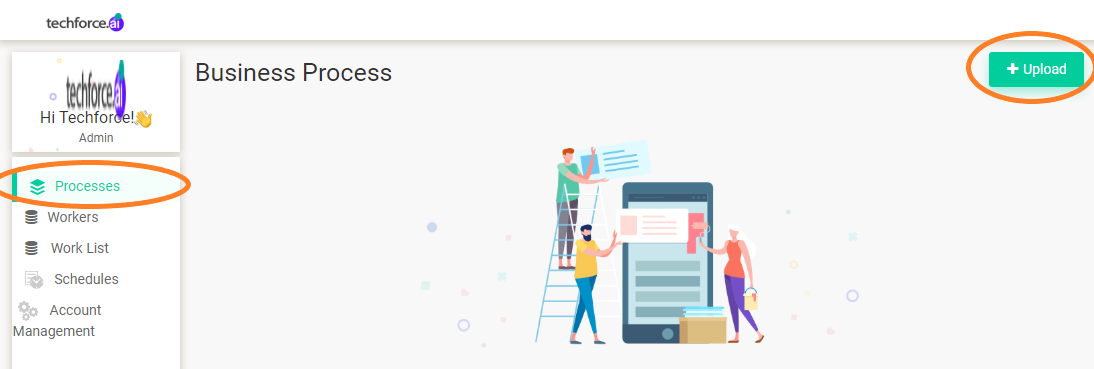
1. Design a Bot – Best Practices & Guidelines

(Content needs to be provided by Srini Kandimalla)



1. Enterprise Manager Console – Management Guide
   1. Process Management
      1. Adding the Business Process

After successful login get redirected to the Business Process screen. Here upload the RPA script files as a Business Process. After uploading, execute the flow.



* + 1. Executing a Business Process:

In the business process page, upload the RPA flow and execute the flow by clicking on the PLAY button from the UI. After clicking on the play button, the below image will appear.

A screenshot of a cell phone

Description automatically generated

Select the Execution target Type like Reserved. If the execution type is selected as reserved, then specify the Worker.

Note: Upload file should be a .Zip file

Note: In on-premise the processes are called “Business Process”.

* 1. Worker Management

The main functionality of a worker is to execute the flows/business processes that are uploaded under business process page. So, after adding the worker while executing the business process, select the worker that was created in worker screen. Below is the image for creating a worker.

A screenshot of a cell phone

Description automatically generated

Click the “Add “option, which will redirect to the Worker environment creation page. Here select the “Create worker”, which will create a new worker.

A screenshot of a cell phone

Description automatically generated

After creating a worker the details of the worker like name, host and so on, can be seen and the worker RDP details will be sent via email.

A screenshot of a social media post

Description automatically generated

* + - 1. Worker Categories

Worker category is the type of worker that is going to be registered. Currently, only “Reserved” type can be created as a worker.

* + - 1. Reserved

The Reserved category defines that, if a worker is created as “Reserved” then flows/business processes can be executed only to that machine.

* 1. Worker List

Worker list/Queues is the functionality where the status of currently executing bots/workers status and scheduled workers execution status can be seen.

**A screenshot of a cell phone

Description automatically generated**

Note: On-premise the Worker List is called as “Queues”

* 1. Schedules

The main functionality of scheduling is, to schedule/execute RPA scripts whenever required. After uploading a business process/RPA flow, schedule those flows. To schedule a flow first create a “Schedule”.

**A screenshot of a cell phone

Description automatically generated**

To create a “Schedule”, navigate to “Schedules” page, and click on “ADD” button to start adding a new schedule.

Give a name to the schedule. Choose the “Business Process” that needs to be scheduled, and select the time zone in which it needs to be run.

* + 1. Trigger

Here specify the time intervals to execute the flows/business processes.

* + - 1. Minutes

If the trigger type is selected as minutes, then the business process will execute at specified intervals measured in minutes. For example, if a business process is scheduled to execute for every 5 minutes then the flow will continuously execute after every 5 minutes until that scheduled flow is stopped/destroyed

**A screenshot of a cell phone

Description automatically generated**

* + - 1. Hourly

If the trigger type is selected as Hours, then the business process will run at specified intervals measured in minutes., For example, if a business process is scheduled to execute for every 3 hours and 30 minutes, then the flow will continuously execute after every 3 hours and 30 minutes until that scheduled flow is stopped/destroyed.

**A screenshot of a cell phone

Description automatically generated**

* + - 1. Daily

If the trigger type is selected as Daily, then the business process will execute every day at the specified time For example, if a business process is scheduled to execute every day at 05:30 a.m. then the flow will continuously execute every day at 05:30 a.m. until that scheduled flow is stopped/destroyed.

A screenshot of a cell phone

Description automatically generated

* + - 1. Weekly

If the trigger type is selected as Weekly, then select the weekday i.e. Mon, Tue, and so on. Then the business process will execute every Week at the time specified. For example, if a business process is scheduled to execute on every Monday at morning 05:30 a.m. then the flow will continuously execute every Monday at 05:30 a.m. until the scheduled flow is stopped/destroyed.

A screenshot of a cell phone

Description automatically generated

* + - 1. Monthly

If the trigger type is selected as monthly, then select the month i.e. Jan, Feb, and so on. Then the business process will execute every Month at the time specified. For example, if the business process is scheduled to execute on every Month at morning 05:30 a.m., then the flow will continuously execute every Month at 05:30 a.m. until the scheduled flow is stopped/destroyed.

A screenshot of a cell phone

Description automatically generated

* + - 1. Advanced:

In Advanced option, one can write a cron expression

**A screenshot of a cell phone

Description automatically generated**

* + 1. Execution Target

In execution target, one can select the Worker.

* + - 1. Reserved Worker

Reserved worker means if a business process is scheduled on a specific machine, then it will be executed only on that machine.

A screenshot of a cell phone

Description automatically generated

* + 1. Actions

In actions, specify when to stop a scheduled bot.

A screenshot of a cell phone

Description automatically generated

* 1. Account Management

In Account Management, the profile details created when the user logged in for the first time can be updated.

A screenshot of a cell phone

Description automatically generated

* + 1. Organization:

Only admin will have access to create an organization. To create an Organization, navigate to Account Management and from the Account Management drop-down, select the Organization tab. In the Organization screen, click on the “ADD” button to create a new organization.

A screenshot of a cell phone

Description automatically generated

* + 1. User Management:

To invite a user to an organization, navigate to the User Management page. Click on INVITE button.

In the invite page screen, enter the email of the person to be invited, and select the Role of that member as Admin or Member for that organization.

A screenshot of a cell phone

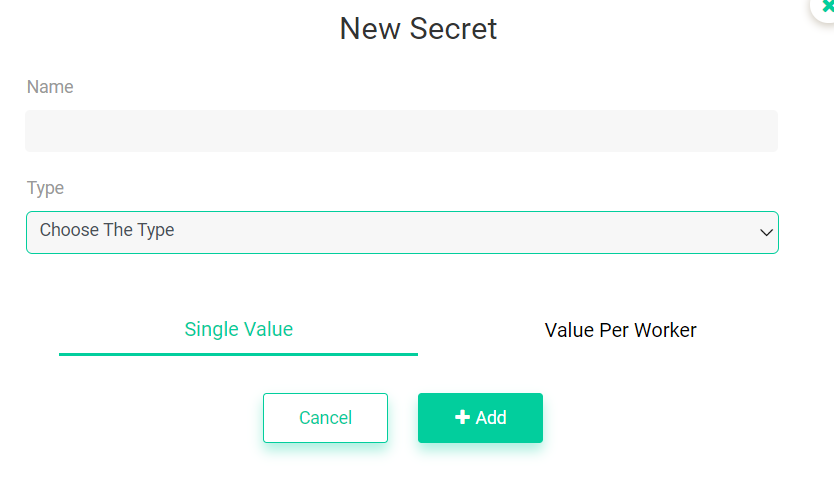
Description automatically generated

* + 1. Secret Vault:

The main functionality of the Secret Vault is to create the values that are not to be shared or shown to others. A secret vault can be created with different types like Integer, Boolean, Text, and Credentials.

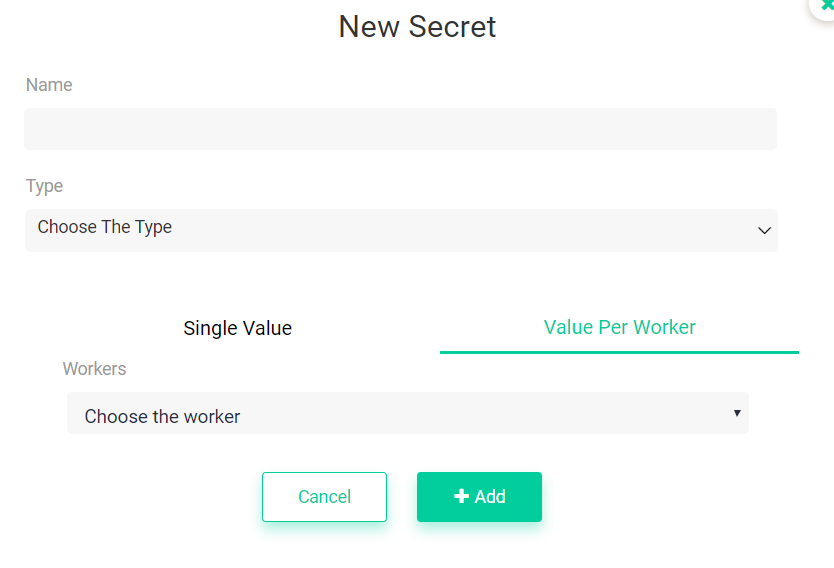
* + - 1. Single value

If a Secret is created as a single value then it can be used for any worker.



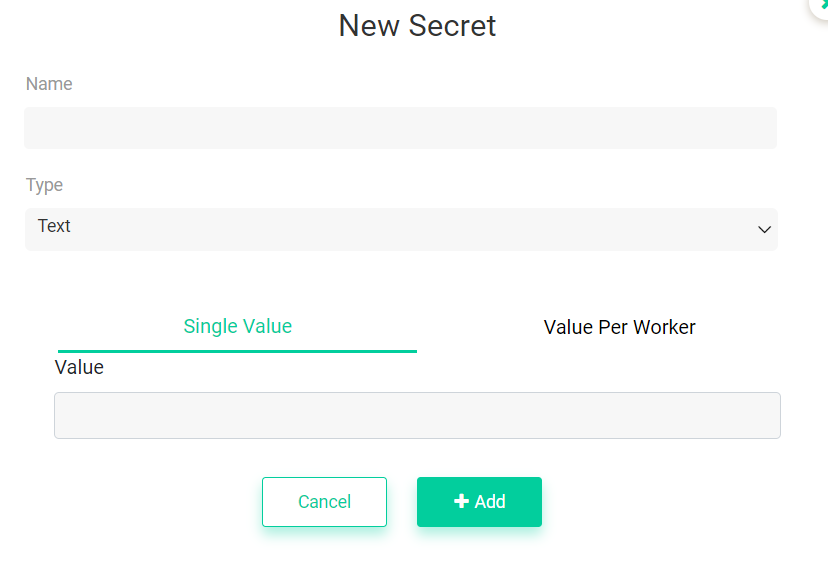
* + - 1. Value per Worker

Value per Worker allows specifying the worker to assign the secret that was created.



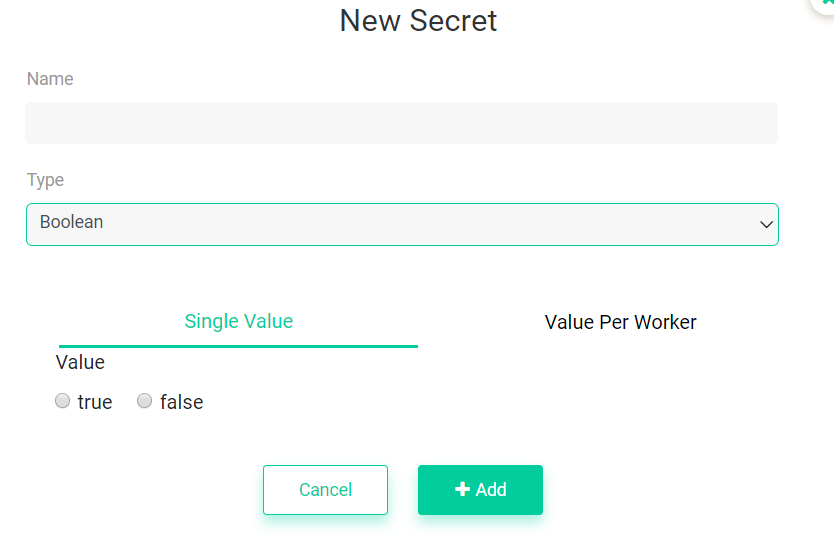
* + - 1. Text:

While creating a new secret select the type of secret. If the text type is selected, then the value will be passed as text to the RPA flow.



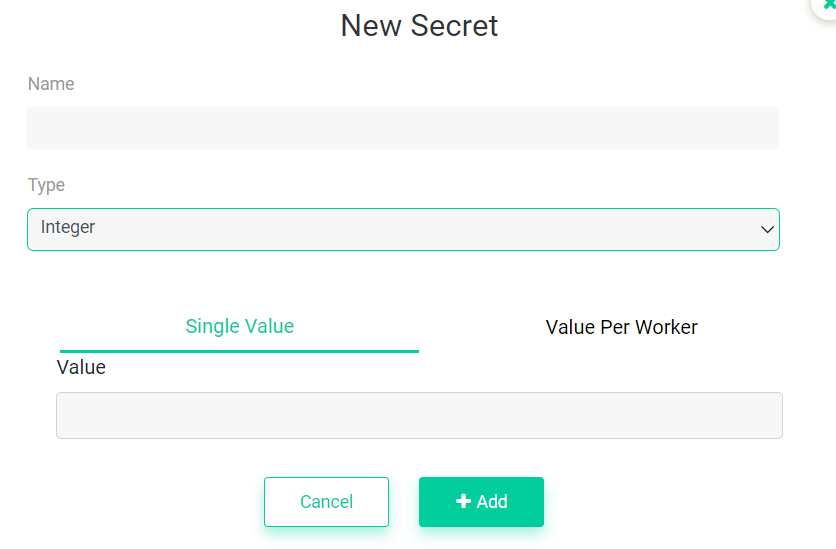
* + - 1. Boolean

While creating a new secret, if the type of secret is selected as Boolean, the value will be passed as Boolean to the RPA flow.



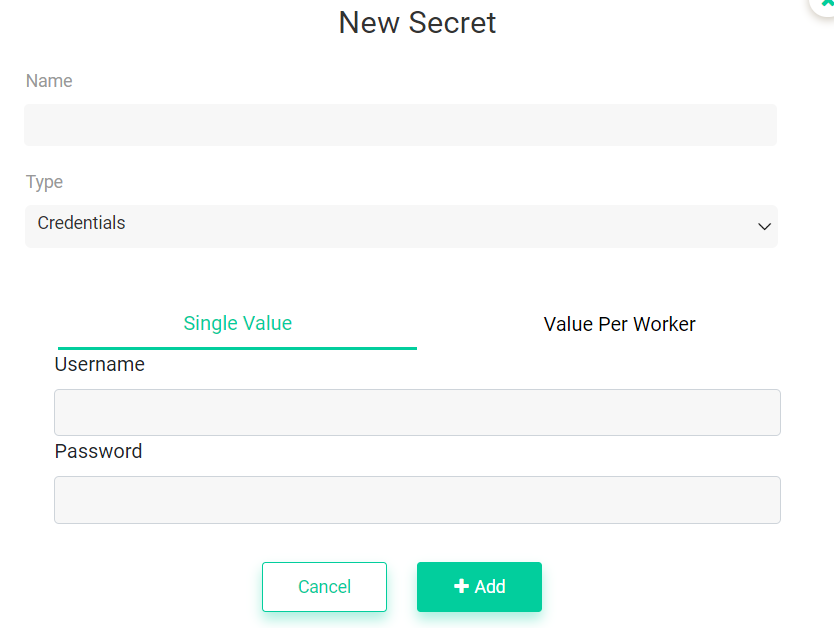
* + - 1. Integer

While creating a new secret, if the type of secret is selected as INTEGER, then the value will be passed as integer to the RPA flow.



* + 1. Credentials

While creating a new secret if the type of secret is selected as CREDENTIALS, then the value will be passed as text to the RPA flow, which will never be shown to any other user.



1. Developer Guide
   1. Studio Guide

Techforce RPA Studio is a complete solution for application integration, and automating third-party applications, administrative IT tasks, and business IT processes. One of the most important notions in Studio is the automation project.

A project is a graphical representation of a business process. It enables automation of rule-based processes, by giving the user full control of the execution order and the relationship between a custom set of steps, also known as activities in Techforce RPA Studio. Each activity consists of a small action, such as clicking a button, reading a file or writing to a log panel.

Techforce RPA Studio contains multiple panels for easier access to specific functionalities.The UI of Techforce RPA Studio contains user-friendly functionalities, which is used to ease any business Process to automate and reduce the manual workforce and rapidly build AI-powered digital employees across the Business & IT workflows.

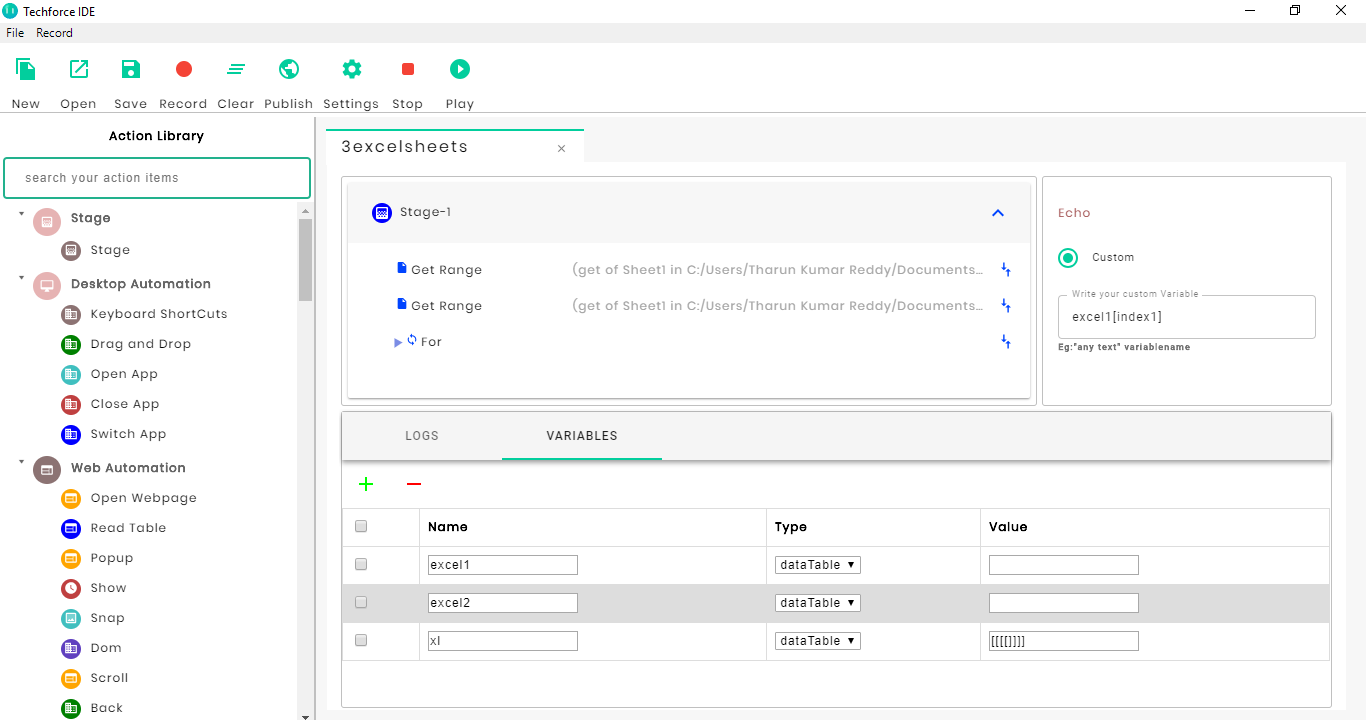


Figure 1: Techforce RPA Studio User Interface

* + 1. Menu Bars

Techforce RPA Studio **Menu bar** contains File Tab and Record Tab. **The menu toolbar** contains the following New, Open, Save, Record, Clear, Publish, Settings, Stop and Play functionalities. Menu bars are very common and easy to find. They help in the efficient use of space.

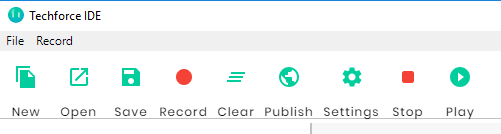


Figure 2: Techforce RPA Studio Menu Panel

* + - 1. File Tab:

The File bar displays Actions and the options in the drop-down menu are Exit, Save and Save As.

* **Exit:** Exit is used to close the Techforce RPA Studio.
* **Save:** It is used to save an RPA task with a name. The shortcut is Ctrl+S.
* **Save As:** It is used to save an already saved document with a new name. The shortcut is Ctrl+Shift+S.

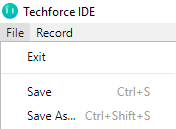


Figure 3: Techforce RPA Studio File Tab

* + - 1. Record Tab

The File bar displays Actions and the options in the drop-down menus are **Web** and **Desktop**.

**Web: Web Automation** with Techforce RPA.

* **Web**: Techforce RPA **Web Automation** uses a built-in recorder that can read and enact **web**-based activities. It identifies **web** elements by their attributes and accurately manipulates them while keeping up with **website** changes.
* **Desktop: Desktop Automation** with Techforce RPA.
* **Techforce RPA** Studio introduces a visual, declarative way of describing how to **automate** a process. Business users can use it in the same way they use a Visio diagram. When working with the presentation layer of other apps, the user simply indicates on the screen what operation needs to perform.

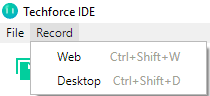


Figure 4: Techforce RPA Studio Record Tab

* + - 1. New

Used to create a new RPA scripting File (BOT) into the respected folder. The user can start a new project from predefined templates.

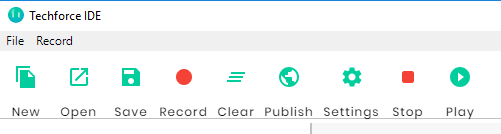


Figure 5: Techforce RPA Studio New

* + - 1. Open

Used to open the existing file. Open a project the user worked on recently. By default, projects are created in

C:\Users\<current\_user>\Documents\Techforce-IDE. The **Open** project dialog searches for project. Json files.

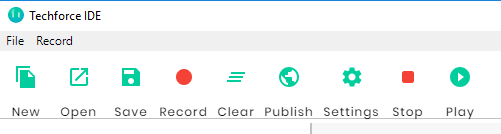


Figure 6: Techforce RPA Studio Open

* + - 1. Save

Used to save the created script or Modified Script. It is used to save an RPA task with a name. Shortcut is Ctrl+S

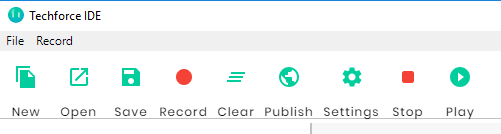


Figure 7: Techforce RPA Studio save

* + - 1. Record

Used to Record the **WEB operations,** simply called a Web-Recorder to automate the browser functionalities.

Techforce RPA **Web Automation** uses a built-in recorder that can read and enact **web**-based activities. It identifies **web** elements by their attributes and accurately manipulates them while keeping up with **website** changes.

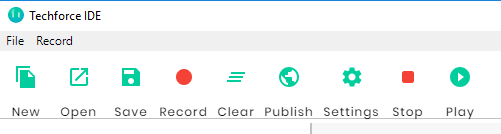


Figure 8: Techforce RPA Studio Record

* + - 1. Clear

Used to close the opened scripts. That means to clear any task (which the user may want to clear or close)

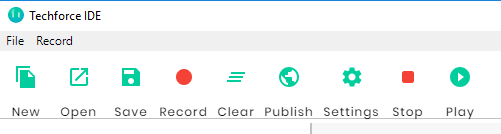


Figure 9: Techforce RPA Studio clear

* + - 1. Publish

Used to publish the developed script using the required user credentials and IP

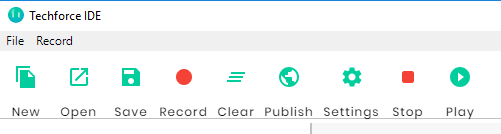


Figure 10: Techforce RPA Studio Publish



Figure 11: Techforce RPA Studio Publish Configuration

* + - 1. Settings

Used to set the global wait time of the flow and basic Chrome dimensions.

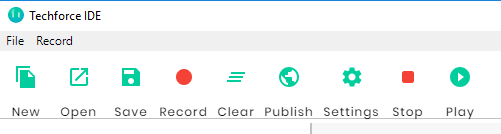


Figure 12: Techforce RPA Studio Settings

Also, it asks Chrome should open or close at the time of exception and at the time of flow completion.

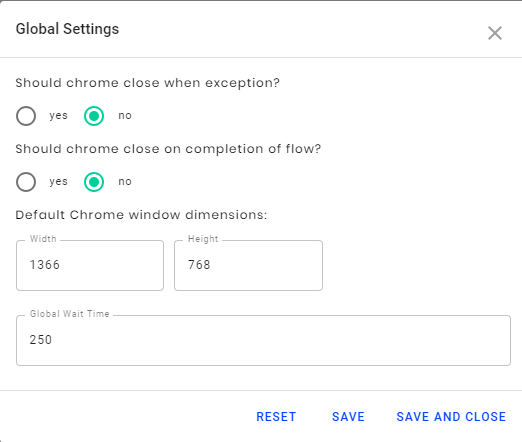


Figure 13: Techforce RPA Studio Global settings

* + - 1. Stop

Used to stop the execution of flow while it is running.

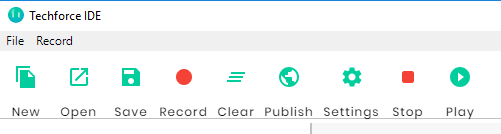


Figure 14: Techforce RPA Studio Stop

* 1. Play

Used to run the flow to start the execution.

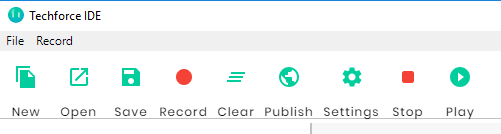


Figure 15: Techforce RPA Studio Play

* + - 1. Play Now

Used to run the flow instantly.

Validate and run the workflow, while using debugging tools to set breakpoints, monitor the execution of activities step by step, and adjust the debugging speed. Open logs to view details regarding execution and any changes made to the project.

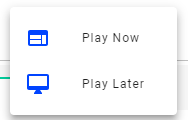


Figure 16: Techforce RPA Studio Play dropdown

* + - 1. Play Later

Used to schedule the execution of the flow and for that, give the name of the flow.

Select the execution options i.e., the execution type like Chrome, Firefox, Headless, Debug, Quiet, Speed, and Baseline.

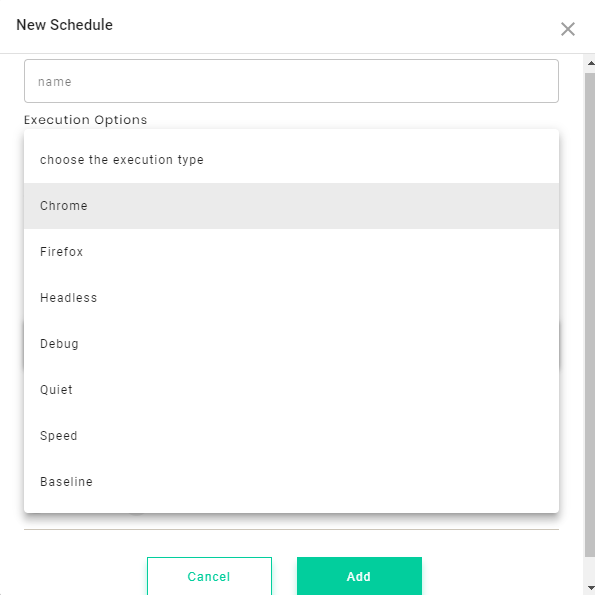


Figure 17: Techforce RPA Studio New schedule

Also, select the time zone of the flow like Pacific, America, Asia, etc., used to trigger the flow at the specific time.

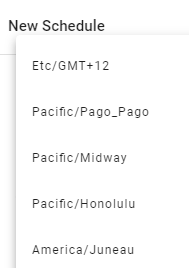


Figure 18: Techforce RPA Studio New Schedule Dropdown

Choose Trigger type as shown in the figure below like Minutes, Hours, Daily, Weekly, Monthly and Advanced to run the flow.



Figure 19: Techforce RPA Studio New Schedule Trigger

Then select the **Actions** type to stop or destroy the flow at the appointment date and time.

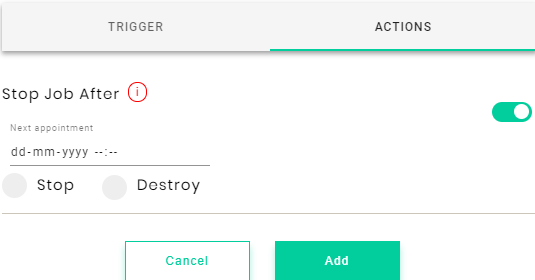


Figure 20: Techforce RPA Studio New Schedule Actions

Then click the **ADD** button to add the created scheduled flow to run or play the developed script.

* 1. Action Library Panel

All the actions in the Action library panel are categorized based on the functionality.

List of action categories in Techforce RPA Studio:

* Stage
* Desktop Automation
* Web Automation
* General
* Programming
* Mail Integration
* Cognitive

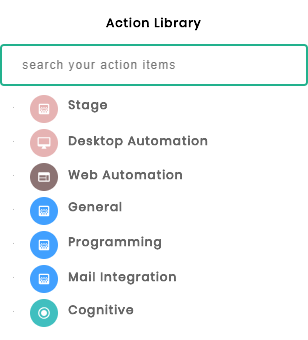


Figure 21: Techforce RPA Studio Action Library

* + 1. Stage Category

In Techforce RPA Studio, Stage Action is used to divide the flow into sub-flows in order to divide the functionalities and reduce the flow confusion to the developers.

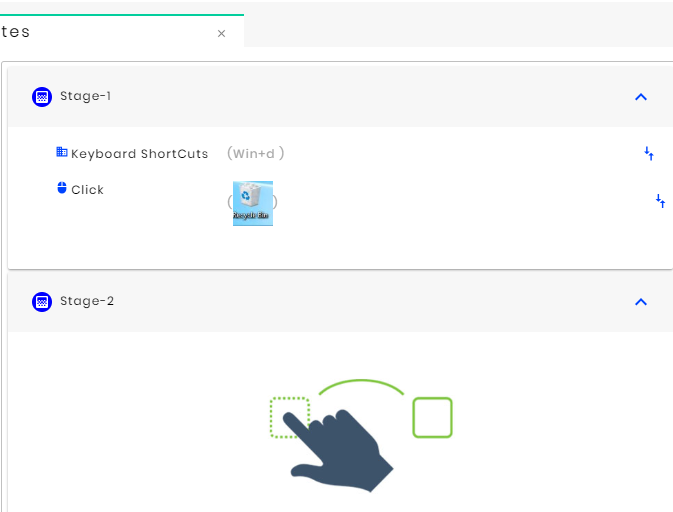


Figure 22: Techforce RPA Studio Stage Action

* + 1. Desktop Automation:

Desktop Automation is used to automate the desktop applications like Notepad, Excel, MS- Word, folders, files, images and screen functionalities using sub-actions like keyboard shortcuts, Drag and Drop, Open App, Close APP and Switch App.

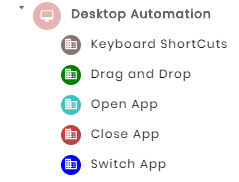


Figure 23: Techforce RPA Studio Desktop Automation

* + 1. Web Automation

Web Automation is used to automate the browser applications like Chrome, Firefox and Internet Explorer, etc., using the Sub-Actions like Open Webpage, Read Table, Popup, Show, Snap, Dom, Scroll and Back.

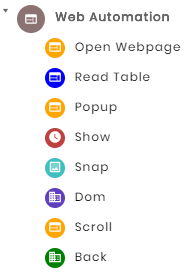
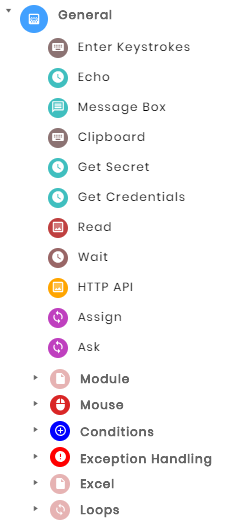


Figure 24: Techforce RPA Studio Web Automation

* + 1. General

General Action is used to automate normal desktop-based and web-based operations using Sub-Actions.

The Sub-Actions like Mouse Clicks, Conditions, Excel, Loops, Jump Statements, OCR, Files and Folders, Secure FTP, CSV, Message Box, Clipboard, Assign, Ask, HTTP API, Wait, Read, Get Credentials, Get Secret, Echo, Enter Keystrokes, Read and Module are used in General Action.



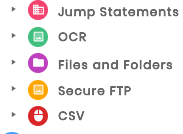


Figure 25: Techforce RPA Studio General Actions

* + 1. Programming

Programming Action is containing String and Technology Sub-Actions. Using String Sub-Action, the user can perform string operations and match case operations. By using Programming Sub-Action, the user can include the programming languages like JavaScript, Python, R and Auto Hotkey scripting, available to automate.

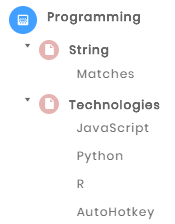


Figure 26: Techforce RPA Studio File Tab

* + 1. Mail Integration

Mail Integration is used to automate mail operations to configure Read Mail, Send Mail and Save Attachments using Sub-Actions: Outlook, IMAP, POP3, OutLook2016 and Save Attachments.

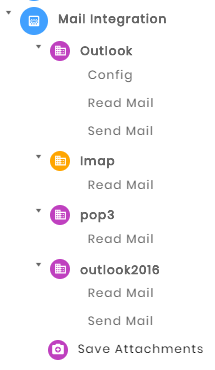


Figure 27: Techforce RPA Studio Mail Integration

* + 1. Cognitive

The Cognitive processes are mainly involved in reading images and there are sub-Actions like Super Resolution, Image Classification and Face Similarity.

Super Resolution: This action item enhances pictures uploaded by the user into higher resolution images.

Image Classification: This action item classifies different objects in a picture uploaded by the user.

Face Similarity: This action item recognizes similarities between two Images.

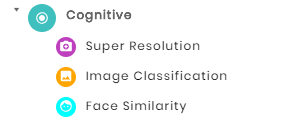


Figure 28: Techforce RPA Studio *Cognitive*

* 1. Developer Panel

Developer panel is used to develop the flow or script to automate any type of operations, either Desktop or Web-based, and using the Record button the developer can also create a flow of web automation.

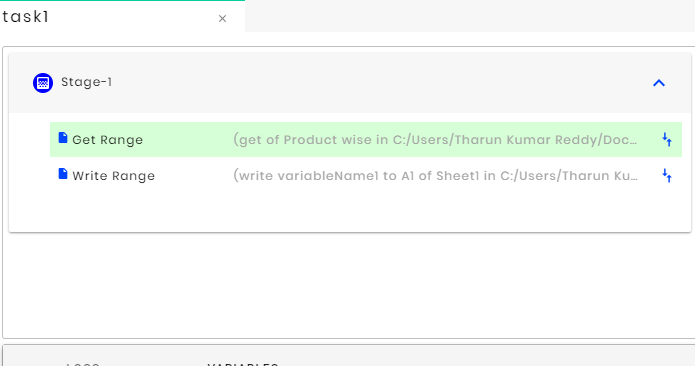


Figure29: Techforce RPA Studio Developer Panel

* 1. Properties Panel

The properties panel is used to specify each Action’s and sub-actions property and helps to automate the bot successfully.

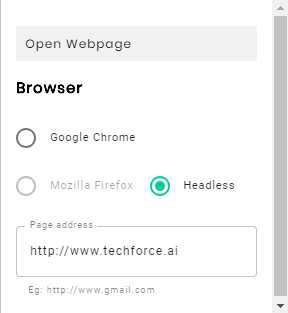


Figure 30: Techforce RPA Studio properties panel

* + 1. Variable Panel

The variable panel is used to specify the names to identify and store the data in different formats like data table, list, and string, etc., According to the usage, variables may be called into the properties panel to automate any script.

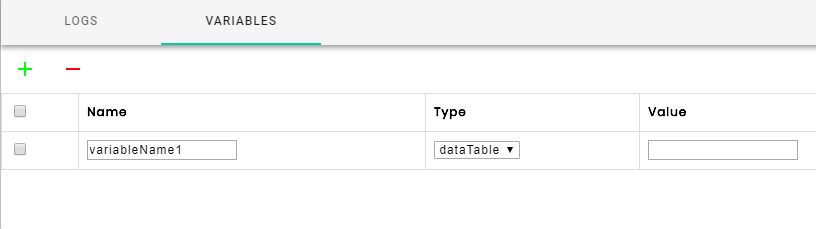


Fig: Techforce RPA Studio File Tab

* + 1. Logs Panel

Logs Panel is used to see the starting and ending of the automation and to find the flow logs and non-executable lines, which can easily get to know the Bugs.

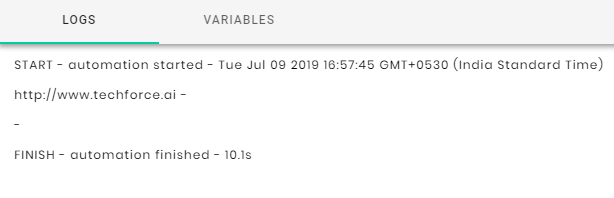


Figure 32: Techforce RPA Studio File Tab

* 1. Actions Guide
     1. Stage

Stage can be considered a group of multiple actions. The actions used in the flow must be included at least in one stage.

* + 1. Desktop Automation

Desktop automation is used to automate desktop applications. In this type of automation, either images or location are used for interacting with the elements. Dom objects of the web page and web-based actions cannot be accessed using desktop automation. Instead, images and locations can be used. While triggering the flow designed using desktop actions, the user should be careful that the UI of the page should be visible on the screen.

The list of actions that come under desktop automation are Keyboard shortcuts · Drag and Drop, Open App, Close App, and Switch App.

* + - 1. Keyboard Shortcuts

This action is used to send keyboard shortcuts to active application in which functions like CTRL, WIN, ALT, and SHIFT have been assorted in the IDE.

* + - 1. Properties
* **Custom keys:** Provides a text box in which the user needs to type the function keys to be passed to the active application.
* **Keys:** Gives out a list of function keys and all the functions available on the keyboard. Click on keys and select the required function to be used on the active application.
  + - 1. Example:

Requirement: Need to send (Control +Copy) and enter key using custom keys and keys.

**Control+ Copy:** Check in the Ctrl box and click on Custom keys and type “c” in the text box **Enter key:** check in the keys and click on drop-down box to find drop-down list of the entire keyboard functions; choose ENTER and click on it.

* + - 1. Drag and Drop

This action is used to drag/change the location of the file/folder from one location to another by using the image selector or location selector.

Properties

From

Input starting point

* **Image Selector:** Where bot enables image capturing mode to select the image of the required file/folder to change.
* Location **selector:** Where bot enables location capturing mode to select the X and Y coordinates of the required file/folder.

To

Output destination

* **Image Selector:** bot allows the user to select the required location in the image from where bot recognizes the file/folder or place in the image format.
* **Location selector:** similarly bot allows the user to select the destination of the selected file/folder by using the X and Y coordinates.

Example

Requirement: Need to change the location .docx file present on the desktop from one location to another by using the image selector in “FROM” and by using location selector in “TO”.

**From:** Check in theimage selector and click on the insert screenshot where the user will give access to select the image of the required .docx file.

**To:** Check in the location selector and click on the find location where bot allows the users to select the required location of destination by using X and Y coordinates.

* + - 1. Open App

Opens Windows applications.

Properties

**File Path:** It is used to open Windows applications by giving the file path of the application with an extension like notepad.exe. Copy the path and paste it in the Open App properties window.

Example

Requirement: Need to open Google by using Open App.

Copy the full path of Google and paste it in the properties window.

* + - 1. Close App

Closes the active applications.

Properties

**File Path:** Similar to the Open App, Close App also works on the same formulae but in order to the file path,just give the header of the existing app.

Example

Requirement: Need to close the Amazon website using Close App.

Copy the header of Amazon.in and paste it in the properties window.

* + - 1. Switch App

Switches between the active applications.

Properties

File Path: Switch app is used to a particular window or app when multiple windows are open. Just write down the header of the required app/window or variable and paste it in the properties window.

Example

Requirement: Need to switch app when two apps/windows are active.

Input: Just write down the heading of the required app/window or variable and paste it in the properties window.

* + 1. Web Automation

Web automation is used to automate browser-based applications. Web actions enable the bot to open a web page, read data from the elements, scroll the webpage, deal with the popup windows, take a screenshot of the page, etc.

* + - 1. Open Webpage

Open a website by passing the URL in the properties window as input with **a** back quote or else by creating the variable for URL in the variables section and pass the variable name in the properties.

Properties:

Google Chrome

**Page address:** Check in the Google box. Pass the URL with back quote symbol (``) example: (`www.amazon.com`) in the page address properties window or else directly pass the variable.

Headless

**Page address:** Used to run a program in the background without any extension. Pass the URL with back quote symbol (``) example: (`www.flipkart.com`) in the page address properties window or else directly pass the variable.

Example

Requirement: Need to open webpage www.irctc.co.in by creating a variable.

**Input:** create a variable in the variables and select the string for variable type and place the [www.irctc.co.in](http://www.irctc.co.in) in value.

Pass that variable into the page address property window.

* + - 1. Show

Show action is used to display the text value in the logs.

Properties

Element: Pass the string or else create a variable in the variables and pass the variable name in back quote symbol (``). Results will be displayed by the bot in the system logs.

Example

Requirement: Show hello world in the logs.

Input: Pass the variable/string in the property window.

NOTE: variables should be passed with back quote (``) like `apple`.

* + - 1. Snap

Used to take a screenshot of a whole page or can be customized to get a specific image by passing the XPath of the required HTML of the window

Properties

Page

* **Image Name:** Provide the image name in the property window and the image will be saved by the given name.

Custom

* **XPath:** Copy and paste the required image HTML path in the XPath properties window or pass the variable.
* **Image Name:** Provide an image name in the property window and the image will be saved by the given name.

**Note**: By default, images are saved in .png format with a default name (snap1 snap2, etc.). To save the image with a given name, then the user needs to mention it with the format also i.e., .png or else each time bot overwrites the existing image and saves it in the same field.

Example: image.png

Example

Requirement: Capture the whole page of google.com.

Page

**Input:** Click on the page and pass the image name in the properties window as the captured image will be placed under the given file name.

Custom

**Input:** Click on the page and paste the XPath of the variable to be captured and pass the image name as string/variable in the properties window.

* + - 1. DOM

When a web page is loaded, the browser creates a Document Object Model (DOM) of the page.

To access HTML elements with JavaScript, find the elements first.

There are a couple of ways to do this:

Changing HTML Elements:

|  |  |
| --- | --- |
| **Method** | **Description** |
| document.getElementById (id) | Find an element by element id |
| document.getElementsByTagName (name) | Find elements by tag name |
| document.getElementsByClassName (name) | Find elements by class name |

* + 1. General

The General Activities pack contains all the basic activities used for creating automation projects. These activities enable the robots to:

* Getting the input to robots
* Printing the output in the console
* Giving an alert message to the user
* Getting credentials to communicate with the orchestrator
* Read data from web
* Assigning the values to variables
  + - 1. General
         1. Enter Keystrokes

Send keystrokes to a UI Element. Enter by using XPath and by selecting Image.

Properties

To Web Element

* **XPath:** The XPath of the corresponding web element.
* **Text:** The text to be written in the specified UI element.

To Image

* **Insert Screenshot:** Capture the corresponding image, which needs to enter the text.
* **Text:** The text to be written in the specified UI element.

Note:

If we give variable name instead of direct text in the text field, the variable name has to be placed between the left quote (` `) command.

* + - * 1. Echo

Prints a string or the value of a string variable to the Output Panel. Strings have to be placed between quotation marks.

Properties

Custom

* **Text/Variable:** The content to be written to the Output panel. This field only accepts Stings and String variables. Strings have to be placed between quotation marks.
  + - * 1. Message Box

Displays a message box with a given text with the button options. User can also set the time to present the message box.

Properties

* **Message**: The text to be displayed in the message box. User has to enter own text in this box or can pass the variable
* **Time out**: The time to display the message box. Directly pass the variable.

Note

Pass the variables between the left quote commands. Ex.(` `)

* + - * 1. Get Secret

Gets a specified secret value from the secret vault in the Orchestrator by using the Asset Name and return a secret value, which can be Text, Boolean, or Integer.

Properties

Configuration

* **User Name:** User Name of the orchestrator account.
* **Password:** Password of the orchestrator account.
* **IP:** The orchestrator IP.

After configuration, the user can access the secret vault in Orchestrator by using the given asset name and can store the values in a variable.

* **Asset Name:** The asset name, created in the orchestrator to contain the secret value.
* **Variable:** The variable used to store returned value from a secret vault.

Note

Once the user was configured with the orchestrator, it will not ask to configure again. Directly the activity shows asset name and variable field

* + - * 1. Get Credentials

Gets a specified credential by using the provided Asset Name and returns a user name and a secure password.

Properties

Configuration

* **User Name:** User Name of the orchestrator account.
* **Password:** Password of the orchestrator account.
* **IP:** The orchestrator IP which the user needs to access.

After configuration, the secret vault is accessible in Orchestrator by using given Asset Name, user name, and password.

* **Asset Name:** The asset name, which was created in the orchestrator to contain a secret value.
* **User Name:** Type the variable name here. The username, which is stored in the Orchestrator secret vault will come to this variable automatically.
* **Password:** Type the variable name here. The password, which is stored in the Orchestrator secret vault will come to this variable automatically.
  + - * 1. Read

Read the data from the web by using the XPath value of that data. It is only used to read values from the web.

Properties

* **HTML Element**: The XPath of that data to be read from the web
* **Variable:** The variable used to contain the read value from the web.
  + - * 1. Wait

Waits for a specified amount of time before continuing the workflow.

Properties

**Time in sec:** The amount of time (in seconds) the following activity to be delayed.

* + - * 1. HTTP API

Enables the user to perform HTTP operations to a specified web API. When first dragging this activity to the Designer panel, the HTTP Request Wizard window is displayed, which provides an easier way of building requests and previewing server responses.

Properties

* **URL:** URL to hit the corresponding API.
* **Get:** One of the methods to hit API. This will get a response from the corresponding API. Here only headers can be added.
* **Post:** One of the methods to hit API. This will send the *json* object and get a response from the corresponding API. Here add body(payload) along with the header.
* **Put:** This request will create a new resource or replaces a representation of the target resource with request payload.
* **Delete**: This request use to delete the specified resource.
  + - * 1. Assign

Allocates any values to a variable. It can be used to increment the value of a variable in a loop. For example, it sums up the values of two or more variables and assigns the result to a different variable.

Properties

* **Custom/Variable**: The name of the variable to be assigned a value.
* **Custom/Variable (second field)**: The value to be assigned to the variable. It may be a string value or can pass variable.
  + - 1. Module
         1. Add External Flow
* Drag the ’Add External Flow’ into the flow.
* This action will help to add sub-flows into the main flow. For this give the exact path to call the sub-flow into the main flow**.** 
  + - 1. Mouse
         1. Click

This Click Action is used to point the particular element or image.

Specify the type of mouse click (single, double, up, down) used when simulating the click event and the mouse button (left, right, middle) used for the click action. By default, the left mouse button is selected.

Properties

Click Action has three different types of selectors.

HTML Selector

* HTML selector is nothing but the element X-Path.

Image Selector

* Image selector is nothing but the Image as an indicator.

Location Selector

* Location selector is nothing but the location of the element, which is X and Y axes.
  + - * 1. Right Click

Drag the Right Click Action into the flow.

Properties

The Right Click Action has three different types of Selectors.

HTML Selector

* HTML selector is nothing but the element X-Path.

Image Selector

* Image selector is nothing but the Image as an indicator.

Location Selector

* Location selector is nothing but the location of the element which is X and Y axes.
  + - * 1. Double Click

Drag Double click action into the flow.

Specifies the kind of click action and it is used to open direct applications like SAP & Outlook etc.

Properties

Double Click Action has three different types of Selectors.

HTML Selector

* HTML selector is nothing but the element XPath.

Image Selector

* Image selector is nothing but the Image as an indicator.

Location Selector

* Location selector is nothing but the location of the element which is X and Y axes.
  + - * 1. Hover

Drag the Hover action into the flow. It can place the mouse cursor at a particular place wherever selected.

Properties

HOVER Action has three different types of Selectors.

HTML Selector

* HTML selector is nothing but the element XPath.

Image Selector

* Image selector is nothing but the Image as an indicator.

Location Selector

* Location selector is nothing but the location of the element which is X and Y axis.
  + - * 1. Select

Drag the SELECT option into the flow.

The SELECT option has only XPath of the selected value and option value (here option will be a variable also).

**Select Variable:** Create one list variableand select that variable from the drop-down. Here only list variable is allowed to select from the drop-down.

* + - 1. Conditions
         1. If, IfElse, ElseIf

Provides the ability to route workflow execution to different step groups depending on conditions provided.

The If Else action compares values of two variables, or of a variable and a specific value and returns a Boolean result (True or False). If the condition result is true, ‘Then’ block will execute. If the condition result is false, ‘Else’ block will execute.

Properties

Custom

Checks the condition. If the condition is met, a block of code will execute. Otherwise, *Else* block will execute.

* **Variable:** Variable name that needs to be compared with the value. The variable field also has a dropdown menu which contains url(), title(), text() to operate on a webpage.
* **Operator Dropdown:** Contains operators such as *contains*, *not contains, equal to, not* *equal to, greater than, less than,* etc.
* **Value:** Value to compare with the variable. Value can be a string or a variable. If the value is a string, it should be included within double quotes (“ ”).

Present

* **HTML Selector:** Checks whether the given XPath exists on a webpage. If exists, a block of code will execute. Otherwise, *Else* block will execute.
* **Image selector:** Checks whether the given image exists on the current window. If exists, a block of code will execute. Otherwise, *Else* block will execute.

Visible

* **HTML Selector:** Checks whether the given XPath is visible on a webpage. If visible, a block of code will execute. Otherwise, *Else* block will execute.
* **Image selector:** Checks whether the given image is visible on the current window. If visible, a block of code will execute. Otherwise, *Else* block will execute.

Count

Checks the count of the XPath based on the condition. If the condition is met, a block of code will execute. Otherwise, *Else* block will execute.

* + - 1. Exception Handling
         1. Try Catch

Drag the Try Catch block into the flow.

Whatever be the actions involved in the flow, add each and every step or action in the Try block. After running the flow, if any exceptions are found, those exceptions will be handled bythe Catch block.

* + - 1. Excel

**Note**: If passing the Excel path by using a variable, then the variable type should be a string. If it is backward slash, use double backward slashes (Example: C:\\test\\excel123.xlsx), and if it is forward slash, then use single slash (Example: C:/test/excel123.xlsx) for every Excel Action. Then pass that variable within the tilde symbol (Example: `VaribleName`)

* + - * 1. Open Spreadsheet

Opens an Excel workbook in the background mode and provides scope for Excel Activities. When the execution of this activity ends, the specified workbook and the Excel application are closed if a Workbook Application variable is provided in the Output. This activity can only be used if the Microsoft Excel application is installed on the machine.

Properties

* **Select an Excel File:** Use the full path of the Excel spreadsheet. Only String variables and Strings are supported. To pass the Excel workbook path through variable, mention that variable name within tilde symbol in that field (Example: `VaribleName`), otherwise click on File symbol and select that file.

**Note:** If passing the Excel path by using a variable then the variable type should be a string. If it is backward slash use double backward slashes (Example: C:\\test\\excel123.xlsx), and if it is forward slash, then use single slash (Example: C:/test/excel123.xlsx) for every Excel Action. Then pass that variable within the tilde symbol (Example: `VaribleName1`).

|  |  |
| --- | --- |
| Note ☞ | If passing the Excel path by using a variable then the variable type should be a string. If it is backward slash use double backward slashes (Example: C:\\test\\excel123.xlsx), and if it is forward slash, then use single slash (Example: C:/test/excel123.xlsx) for every Excel Action. Then pass that variable within the tilde symbol (Example: `VaribleName1`). |

* + - * 1. Get SheetNames

Returns a list of all the sheet names in a workbook as String variables, ordered by their index (index value starts from zero).

Properties

* **Enter Full Path:** Need to give the full path of the Excel spreadsheet. This option is shown only while taking the Get Sheet Names action directly. If dragged into the open spreadsheet activity, it will not ask the path, and no need to mention the path.
* **Select Variable:** Create one list variableand select that variable from the drop-down. Here only list variable is allowed to select from the drop-down. The result value will be stored in the list variable.
  + - * 1. Get Column

Reads the values from a column beginning with the cell specified in the Starting Cell property field, and stores them in a list variable.

Properties

* **Enter Full Path:** Need to give the full path of the Excel spreadsheet. This option is shown only while taking the Get Column action directly. If dragged into the open spreadsheet activity, it will not ask the path, and no need to mention the path.
* **Enter Sheet Name: Enter** the name of the sheet containing the column that needs to be read. Only String variables and strings are supported (Example: Sheet1).
* **Enter Cell Address:** The cell from which to start extracting the column data. Mention Column name A or B.
* **Select Variable:** Need to create the variable as a list. Stores the information from the specified spreadsheet column in a variable. Only list variables are supported.
  + - * 1. Delete Column

Deletes a table column from a spreadsheet-based on its name.

Properties

* **Enter Full path: Enter** the full path of the Excel spreadsheet to be used. This option is shown only while taking the Delete Column action directly. If dragged into the open spreadsheet activity, it will not ask the path, and no need to mention the path.
* **Enter Sheet Name:** Enterthe name of the sheet containing the column that needs to be deleted. Only String variables and strings are supported (Example: Sheet2).
* **Delete Column:** The exact number of the column to be deleted. Only String variables and number values are supported.
  + - * 1. Get Row

Reads the values from a row beginning with the cell specified in the Starting Cell field and stores it in a list variable.

Properties

* **Enter Full path:** Enter the full path of the Excel spreadsheet being used. This option is shown only when taking the Delete Column action directly. If dragged into the open spreadsheet activity, it will not ask the path, and no need to mention the path.
* **Enter Sheet Name:** Enter the name of the sheet containing the column to be deleted. Only String variables and strings are supported (Example: Sheet1).
* **Get Row: Enter** the exact row number to be read. Only String variables and number values are supported.
* **Select Variable:** Need to create the variable as a list. Stores the information from the specified spreadsheet row in a variable. Only list variables are supported.
  + - * 1. Delete Row

Remove a specified row at a certain position.

Properties

* **Enter Full path:** Enter the full path of the Excel spreadsheet. This option is shown only when taking the Delete Row action directly. If dragged into the open spreadsheet activity, it will not ask the path, and no need to mention the path.
* **Enter Sheet Name:** Enter the name of the sheet containing the Row that needs to be deleted. Only String variables and strings are supported (Example: Sheet1).
* **Delete Row:** Enter the exact number of the row to be deleted. Only String variables and number values are supported.
  + - * 1. Get Cell

Reads the values from a row beginning with the cell specified in the Starting Cell field and stores it in a string variable.

Properties

* **Enter Full Path:** Enter the full path of the Excel spreadsheet. This option is shown only when taking the Get Cell action directly. If dragged into the open spreadsheet activity, it will not ask the path, and no need to mention the path.
* **Enter Sheet Name:** Enter the name of the sheet containing the Cell to be read. Only String variables and strings are supported (Example: Sheet1).
* **Enter Cell Address:** The exact Cell address that needs to be read. Only String variables and number values are supported.
* **Select variable:** Create the variable as string Stores the information from the specified spreadsheet Cell in a variable. Only string variables are supported.
  + - * 1. Write Cell

Writes a value into a specified spreadsheet cell. If the sheet does not exist, a new one is created with the Sheet Name value. If a value exists, it is overwritten. Changes are immediately saved.

Properties

* **Enter Full Path:** Enter the full path of the Excel spreadsheet being used. This option is shown only when taking the Write Cell action directly. If dragged into the open spreadsheet activity, it will not ask the path, and no need to mention the path.
* **Enter Sheet Name: Enter** the name of the sheet containing the Cell to be written. Only String variables and strings are supported (Example: Sheet1).
* **Enter Cell Address:** Enter the exact Cell address to be written. Only String variables and number values are supported.
* **Enter Cell Value:** Enter the cell value to be written. It allows only String and String variable. If passing the value through variable, mention the variable within the tilde symbol (Example: `VaribleName`).
  + - * 1. Get Range

Reads the value of an Excel range and stores it in a DataTable variable. If the range isn't specified, the whole spreadsheet is read. If a particular range is specified, it reads the values from that specific range only.

Properties

* **Enter Full Path: Enter** the full path of the Excel spreadsheet. This option is shown only when taking the Get Range action directly. If dragged into the open spreadsheet activity It will not ask the path, and no need to mention the path.
* **Enter Sheet Name:** Enter the name of the sheet containing the Cell that needs to be read. Only String variables and strings are supported (Example: Sheet1).
* **From Range:** Specify the range of cells to be read. If this value is not specified, the whole spreadsheet is read. Only String variables and strings are supported to mention that range (Example: B5: G15).
* **Select variable:** Create the variable as DataTable Stores the information from the specified spreadsheet range values in a variable. Only DataTable variables are supported select.
  + - * 1. Write Range

Writes the data from a DataTable variable in a spreadsheet starting with the cell indicated in the **Starting Cell** field. If the starting cell is not specified, the data is written starting from the A1 cell. If the sheet does not exist, a new sheet is created with the value specified in the **Sheet Name**property. All cells within the specified range are overwritten. Changes are immediately saved.

Properties

* **Enter Full Path:** Enter the full path of the Excel spreadsheet. This option is shown only when taking the Write Range action directly. If dragged into the open spreadsheet activity, it will not ask the path, and no need to mention the path.
* **Enter Sheet Name:** Enter the name of the sheet containing the range that needs to be written. Only String variables (Example: `VaribleName`) and strings are supported (Example: Sheet1).
* **Write Range: Specify** the cell from which to start writing the data. Only string variables and strings are supported (Example: A1 or B1).
* **Select variable:** Select the variablethe data that needs to be written to the specified range, as a DataTable variable. Only DataTable variables are supported.
  + - * 1. Append Range

Adds the information stored in a DataTable variable to the end of a specified Excel spreadsheet. If the sheet does not exist, a new one is created with the name indicated in the Sheet Name field.

Properties

* **Enter Full Path:** Enter the full path of the Excel spreadsheet. This option is shown only when taking the Append Range action directly. If dragged into the open spreadsheet activity, it will not ask the path, and no need to mention the path.
* **Enter Sheet Name:** Enter the name of the sheet in which the range that needs to be appended. Only String variables (Example: `VaribleName`) and strings are supported (Example: Sheet1).
* **Select variable:** Select the variablethe data that needs to be appended to the specified range, as a DataTable variable. Only DataTable variables are supported.
  + - * 1. Delete Range

Deletes a specified range in an Excel workbook.

Properties

* **Enter Full Path:** Enter the full path of the Excel spreadsheet. This option is shown only when taking the Delete Range action directly. If dragged into the open spreadsheet activity, it will not ask the path, and no need to mention the path.
* **Enter Sheet Name:** Enter the name of the sheet containing the range that needs to be deleted. Only String variables (Example: `VaribleName`) and strings are supported (Example: Sheet1).
* **From Range:** Specify the range of cells to be deleted. If this value is not specified, the whole spreadsheet is read. Only String variables and strings are supported to mention that range (Example: B5: G15).
  + - * 1. Get Cell Color

Extracts the background Color of a cell and saves it as a Color variable. It can be used within the **Open Spread Sheet** Action and outside the Open Spread Sheet Action.

Properties

* **Enter Full Path:** The full path of the Excel spreadsheet that you want to use. This option is shows only you are taking the Get Cell Color action directly. If we dragged into the open spreadsheet activity It will not ask the path, then no need to mention the path.
* **Enter Sheet Name:** The name of the sheet in which the cell color name that you want to read. Only String variables (Eg: `VaribleName`) and strings are supported (Eg: Sheet1).
* **Enter Cell Address:** The exact Cell address that you want to get the color. Only String variables and stings are supported (Eg: B1).
* **Select Variable:** Create and select the variable as string Stores the name of the color from the specified spreadsheet Cell in a variable. Only string variables are supported.
  + - * 1. Set Range Color

Changes the color of a specified cell or cell range using a Color variable. Can only be used within the Open Spread Sheet action and outside the Open Spread Sheet Action.

Properties

* **Enter Full Path:** The full path of the Excel spreadsheet that you want to use. This option is shows only you are taking the Set Range Color action directly. If we dragged into the open spreadsheet activity It will not ask the path, then no need to mention the path.
* **Enter Sheet Name:** The name of the sheet in which the cell color name that you want to read. Only String variables (Eg: `VaribleName`) and strings are supported (Eg: Sheet1).
* **Enter Range Address:** The range address that you want to set the color. Only String variables and stings are supported (Eg: A1:B5 or A3).
* **Color:** Select thecolor form drop down which you want to set the range color. (Eg: green or black etc)
  + - * 1. Get Cell Formula

Extracts the formula used in the specified Excel cell.

Properties

* **Enter Full Path:** The full path of the Excel spreadsheet that you want to use. This option is shows only you are taking the Get Cell Formula action directly. If we dragged into the open spreadsheet activity It will not ask the path, then no need to mention the path.
* **Enter Sheet Name:** The name of the sheet in which the cell formula that you want to read. Only String variables (Eg: `VaribleName`) and strings are supported (Eg: Sheet1).
* **Enter Cell Address:** The exact Cell address that you want to get the cell formula. Only String variables and stings are supported (Eg: B1).
* **Select Variable:** Create and select the variable as string Stores the formula from the specified spreadsheet Cell in a variable. Only string variables are supported.
  + - * 1. 9.2.4.6.16 Create Pivot Table

(Cotent to be provided by Srinivas Kandimalla)

* + - * 1. Convert XLS to XLSX

Converts the Excel workbook from XLS to XLSX format.

Properties

**Source File Path:** Givethe full path of the Workbook (.xls file) which you want to convert from XLS to XLSX format.

**Enter File Name:** File nameisoptional. If file name is not mentioned, then the Workbook is converted from xls to xlsx format with same in the same location. If File name is mentioned in that file, the new file was created in xlsx format with that filename. If in conversion already file is existed with same name in that location with xlsx format, then it will overwrite with same name. If passing the file name through variable need to mention that variable within tilt symbol (Eg: `varibleName’).

* + - 1. Loops
         1. For

Loops are an efficient way to complete work via multiple number of iterations until the targeted result is achieved.

Properties

* **Element:** The variable that the user wants to iterate.
* **Initial value:** The index value of the element to start with. In general, the index starts with zero.
* **Range:** The index value of the element to end with. To loop indefinitely, use range value as infinity. This loops 1024 times as infinity variable is preset to 1024
  + - 1. Jump Statements
         1. Break

Drag the statements action into the flow.

Use this statement only for a loop. If the iteration needs to stop after a certain number of executions, then use this Break statement**.**

* + - * 1. Continue

Drag the statements action into the flow.

Use this Continue action in a loop to execute a number of iterations continuously.

* + - 1. OCR
         1. Techforce OCR (Optical Character Recognition)

This action is used to extract the data from the inputs provided. The extracted data will be stored in a variable for further use.

Properties

* **Indicate on-screen using location:** Click on **Select Location** button. It will open the new window to select the location. Select the location and press Enter. Select any string variable from the drop-down to store the result of the OCR.
* **Indicate on Screen:** Click on **Insert Screenshot** button. It will open the new window to select the area. Select the area and press Enter. Select any string variable from the drop-down to store the result of the OCR.
* **Select Image:** Click on **Insert Image** button. It will open the new window to select the image. Select the image from the path. Select any string variable from the drop-down to store the result of the OCR.
* **Select PDF:** Select the required input PDF file by clicking on the path button. Select an option from All/ Single/ Multiple to extract data from all pages/ from a single page/ from multiple pages.

**Range of PDF:** The user can select the range from the PDF document to extract the data. *Ex: 2-5.OCR action will be performed on the pages from 2 to 5.*

Select any string variable from the dropdown to store the result of the OCR.

* + - 1. Files and Folders
         1. Create Files /Folders

Creates a file/folder in the specified location.

Properties

File

* **File Path:** Give the full path of the file to be created with extension. The path can be given by assigning it to a variable using double slashes. For example, path = C:\\Desktop\\*filename.Extension*. The variable should be included within `` while passing to file path.

Folder

* **Folder Path:** Give the full path of the folder to be created. The path can be given by assigning it to a variable using double slashes. For example: path = C:\\Desktop\\*foldername.* The variable should be included within `` while passing to file path.
* **Overwrite:** The folder will be overwritten if there is any folder existing with the same name in the given path.
* **Skip:** The creation of the folder will be skipped if there is any folder existing with the same name in the given path.
  + - * 1. Write To File

Appends/Overwrites the specified string or the data assigned to a variable to the specified file.

Properties

* **File Path:** Give the full path of the file. The path can be given by assigning it to a variable using double slashes. For example: path = C:\\Desktop\\*filename.Extension*. The variable should be included within `` while passing to file path.
* **Content to Write:** Give the string/variable to be written to the file specified. If it is a string, it can be given directly. If it is a variable, it should be included in ``.
* **Overwrites:** The file data will be overwritten with the string/variable data if there is any data existing in the file.
* **Append:** The file data will be appended with the string/variable data if there is any data existing in the file.
  + - * 1. Copy File/Folder

Copy/Move a file from one location to another location as specified.

Properties

File

* **Source Path:** Give the full path of the file to be copied/moved including the extension. The path can be given by assigning it to a variable using double slashes. For example: path = C:\\Desktop\\*filename.Extension*. The variable should be included within `` while passing to file path.
* **Destination Path:** Give the full path of the location where the file needs to be copied/moved including the extension. The path can be given by assigning it to a variable using double slashes. For example: path = C:\\Desktop\\*filename.Extension*. The variable should be included within `` while passing to file path.
* **Copy:** Makes a copy of the file specified from the source path to the destination path.
* **Move:** Moves the file specified in the source path to the destination path.
* **Overwrite:** Overwrites the file while moving/copying the file specified if there is a file existing with the same name in the destination.
* **Skip:** Skips the copying/moving the file specified if there is a file existing with the same name in the destination.

Folder

* **Source Path:** Give the full path of the folder to be copied/moved. The path can be given by assigning it to a variable using double slashes. For example: path = C:\\Desktop\\*foldername.* The variable should be included within `` while passing to file path.
* **Destination Path:** Give the full path of the location where the folder needs to be copied/moved. The path can be given by assigning it to a variable using double slashes. For example: path = C:\\Desktop\\*foldername.* The variable should be included within `` while passing to file path.
* **Copy:** Makes a copy of the folder specified in the source path to the destination path.
* **Move:** Moves the folder specified in the source path to the destination path.
* **Overwrite:** Overwrites the folder while moving/copying the folder specified if there is a folder exists with the same name in the destination.
* **Skip:** Skips the copying/moving the folder specified if there is a folder existing with the same name in the destination.
  + - * 1. Get Folder Contents

Lists the contents of the folder based on the filter criteria.

Properties

* **Folder Path:** The full path of the folder from where the contents to be retrieved. The path should be given using the forward-slash (/). Example: C:/Desktop/*FolderName*. If it is given by variable, it should be included in ``. for ex: `*variable name*`.
* **Variable:** Variable name to list the contents of the folder. The result will be stored in a list.

Which Contents to List

* **Folders:** Lists the folders in the folder specified in the **Folder Path.**
* **Files:** Lists the files in the folder specified in the **Folder Path.**
* **Filter:** Filters the files based on the given extension like *.png, .xls, .xlsx* etc.

Filter Criteria

* **Date Modified:** Filters the contents of the folder based on the date modified selected in the From Date and To Date fields.
  + - * 1. Delete File/Folder

Deletes the file/folder specified.

Properties

File

* **File Path:** The full path of the file to be deleted including the extension. The path can be given by assigning it to a variable using double slashes. Example: path = C:\\Desktop\\*filename.Extension*. The variable should be included within `` while passing to file path.

Folder

* **Folder Path:** The full path of the folder to be deleted. The path should be given using the forward-slash (/). Example: C:/Desktop/*FolderName*. If it is given by variable, it should be included in ``. For example: `*variable name*`.
  + - * 1. Read File

Reads all the content from the file specified and stores it into a variable specified.

Properties

* **File Path:** The full path of the file to be read including the extension. The path can be given by assigning it to a variable using double slashes. Example: path = C:\\Desktop\\*filename.Extension*. The variable should be included within `` while passing to file path.
* **Variable:** Variable name to store the contents of the file specified.
  + - 1. Secure FTP
         1. Connect

Enables USER to connect to a secured server via entering credentials

Properties

Host - User needs to provide the IP address of the host.

Username -Need to provide the Username for the host to connect.

Password – Need to provide the password for the host to connect.

Port – Need to give the port address

* + - * 1. Get File

User can GET a single file, upon specifying a remote & local path including the file name.

Properties

Remote Path – Need to give the server file path, which is the target path

Local Path – Need to give the destination Path

* + - * 1. Put File

User can PUT a single file, upon specifying a remote & local path including the file name

Properties

Remote Path **--** Need to give target path which is from the local server

Local Path **–** Need to give the destination path

* + - * 1. Get File

User can GET a single file, upon specifying a remote & local path including the file name.

Properties

Remote Path – Need to give the server file path which is target path

Local Path – Need to give the destination path

* + - * 1. Put Files

User can PUT a single file, upon specifying a remote & local path including the file name

Properties

Remote Path --Need to give target path, which is from the local server

Local Path–Need to give the destination path

* + - * 1. Delete

User can delete a file in a specified path or folder.

Properties

Remote Path – Delete files from the given path

* + - * 1. Delete Files

User can delete the file(s) in a specified path or folder.

Properties

Remote Path – Delete files from the given path

* + - 1. CSV
         1. CSV Header

This action is used to send the headers to a specific csv file.

Properties

* **CSV File Name:** The full path of the CSV file.
* **Header Values:** Provide the header values to be passed to the CSV file. The user can give multiple values by pressing Enter.
  + - * 1. CSV Row

This action is used to read the row values in a variable provided.

Properties

* **CSV FileName:** Use **t**he full path of the CSV file.
* **Variable:** Provide any list variable to store the read data from the file.
  + 1. Programming
       1. String
          1. Matches

The Matches action helps the user to write regular expressions whenever needed.

Properties

* **Select Variable:** Assign the input data to a string variable in the variables. Select that variable from the drop-down.
* **Regular expression Pattern:** Give the regular expression to be performed in the field. *Ex: [b-e]. This expression matches the characters in the input, which are in the range b to e.* The user can give regular expression using a variable also, by assigning the expression to a variable in the variable section. The variable should be passed using the ``codes. *Example: `variableName`.*
* **Result variable:** The result of the Matches action will be stored in a *list* variable provided.
  + - 1. Technologies

Techforce RPA supports three programming languages. The supported languages are JavaScript, Python, R, and AutoHotkey. The user can add simple programming code snippets using this action.

* + - * 1. JavaScript

The user can add Java script code snippet using this action. It will support all ES5 methods and functions.

* + - * 1. Python

The user can add Python code snippet using this action.

* + - * 1. R

The user can add R code snippet using this action.

* + - * 1. AutoHotkey

The user can add AutoHotkey code snippet using this action.

* + 1. Mail Integration

The Mail Integration Activity contains all the mail activities used for creating automation projects. These activities enable robots to:

* Configuring bot with corresponding Outlook account
* Sending email from Outlook account
* Receiving email from corresponding Gmail/Outlook email
* Saving an attachment with the email
  + - 1. Outlook
         1. Config

Configures the bot with corresponding Outlook account.

Properties

* **Sign In:** Used to sign in the account.
* **Sign Out:** Used to sign out the account.

**Note:** The send and read email activities can be used only after finishing the configuration. User should drag the send/read mail actions on to the config tab.

* + - * 1. Read mail

Reads email from the configured Outlook account.

Properties

Inbox

* **From Specific Mail:** Reads emails from a specific email account. The user can give one or more email IDs here. To separate mail IDs, use a semicolon.
* **All:** Reads emails from all folders.
* **Unread:** Reads only unread emails from Inbox.
* **Top:** Filter Particular number of mails, which are present on the top.
* **Select Variable:** Store emails in the variable() based on the filter.

Sent Items

**Top:** Filter a particular number of emails, which are present on the top.

**Select Variable:** Stores emails in the variable() based on the filter.

**Note:** Here array variable are used to store emails.

To access the subject of the mail, use variableName[index].Subject.   
To access the body of the mail, use variableName[index].Body.   
To access the from the address of the mail, use variableName[index].From. To access the to address of the mail, use variableName[index].Received.

* + - * 1. Send mail

Send email from the configured Outlook account.

Properties

* **To:** The receiver mail id. The user can give one or more email IDs. To separate mail IDs, use semicolon
* **Subject:** The subject of the email which we need to send.
* **Body:** The body of the corresponding email.

**Note:** We can send variable instead of direct text in to, subject, body

* + - 1. Imap

IMAP (Internet Message Access Protocol) is a standard email protocol that stores email messages on a mail server but allows the end-user to view and manipulate the messages as though they were stored locally on the end user's computing device(s). This allows users to organize messages into folders, have multiple client applications know which messages have been read, flag messages for urgency or follow-up and save draft messages on the server

* + - * 1. Read Mail

Reads email from the corresponding GMAIL account.

Properties

Login

* **Email:** The mail ID, which needs to be accessed to read emails. The user can also use the variable.
* **Password:** Password of the corresponding mail, which needs to be read. The user can also use the variable.

Connection

* **Host:** The host for the IMAP connection.
* **Port:** The port for the IMAP connection.

Options

* **Un Read:** Use to Read Only Unread messages from Inbox.

Select Mail

* **Today:** Reads only today’s (System Date) emails from the corresponding account.
* **Yesterday:** Reads only yesterday’s emails.
* **Start of Week:** Reads emails from the start day of the week.
* **Specific Date:** Reads emails from a specific date.
* **Select Variable:** Stores emails, which are read from the corresponding Gmail account.

**Note:** Here array variable are used to store emails.

To access the subject of the mail, use variableName[index].Subject.   
To access the body of the mail, use variableName[index].Body.  
To access the from the address of the mail, use variableName[index].From. To access the to address of the mail, use variableName[index].Received.

* + - 1. Pop3

POP3 (Post Office Protocol 3) is the most recent version of a standard protocol for receiving e-mail. POP3 is a client/server protocol in which e-mail is received and held for the user by the Internet server. Periodically, the user (or user’s client e-mail receiver) checks the mail-box on the server and downloads any mail, probably using POP3.

* + - * 1. Read Mail

Read emails from corresponding GMAIL account.

Properties

Login

* **Email:** The mail ID required to access to read emails. The user can give one or more email ids. To separate email IDs, use a semi-colon. The user can also use a variable here.
* **Password:** Password of the corresponding mail account, which needs to be read. The user can also use a variable here.

Connection

* **Host:** The host for the POP3 connection.
* **Port:** The port for the POP3 connection.

Select Mail

* **Top:** Filter articular number of emails, which are present on the top.
* **Select Variable:** Store emails, which are read from the corresponding Gmail account.

**Note:** Here array variable is used to store emails. To access the subject of the mail, use variableName[index].Subject. To access the body of the mail, variableName[index].Body, To access the from the address of the mail, variableName[index].From, To access the to address of the mail, variableName[index].Received.

* + - 1. Outlook2016

This is also for the same process mentioned above in the Outlook section. But here the user should have Outlook application in the local system.

* + - * 1. Read mail

Reads emails from configured outlook account.

Properties

Inbox

* **From Mail:** Gmail account to filter emails.
* **Unread:** Filters only unread emails from Inbox.
* **Top:** Filters Particular no of emails which is present on the top.
* **Select Variable:** Stores the emails collected from the corresponding email account.

Sent Items

* **Top:** Filter a particular number of emails, which are present on the top.
* **Select Variable:** Stores the emails collected from the corresponding email account.

**Note:** Here array variable is used to store emails. To access the subject of the mail, use variableName[index].Subject. To access the body of the mail, use variableName[index].Body, To access the from the address of the mail, use variableName[index].From, To access the to address of the mail, use variableName[index].Received.

* + - * 1. Send mail

Send email from configured Outlook account.

Properties

* **To:** The receiver’s mail id
* **Subject:** The subject of the mail r to be sent.
* **Body:** The body of the corresponding mail.
* **Attachments:** The path of the corresponding attachment to upload on Gmail.
  + - 1. Save Attachments

Saves attachments in the local path, which is downloaded from corresponding mail account.

Properties

* **File Path:** The corresponding folder path to save the attachment
* **Content to Write:** Variable, which is used to get an attachment from an email

Note: If there is no attachment in an email, it will ignore automatically

* + 1. Cognitive

The Cognitive Activities pack contains all the cognitive activities used for creating automation projects. These activities enable the robots to:

Increase the resolution of the Picture.

Classify the things mentioned in the picture.

Compare the image to find matches.

* + - 1. Super Resolution

Increases the resolution of the selected image.

Properties

**Current Selected File:** Selects the file for which the resolution needs to be increased.

* + - 1. Image Classification

Classifies the things in the selected image.

Properties

**Current Selected File:** Select file in which things need to be classified.

* + - 1. Face Similarity

Compare two different faces to check whether they are same or not.

Properties

**Current Selected File:** Select first face to compare.

**Current Selected File:** Select the second face to compare.

**Note:** The Result will be displayed in the output panel.

* 1. Build A Basic Bot

Use case: Get the Price of iPhone from Amazon

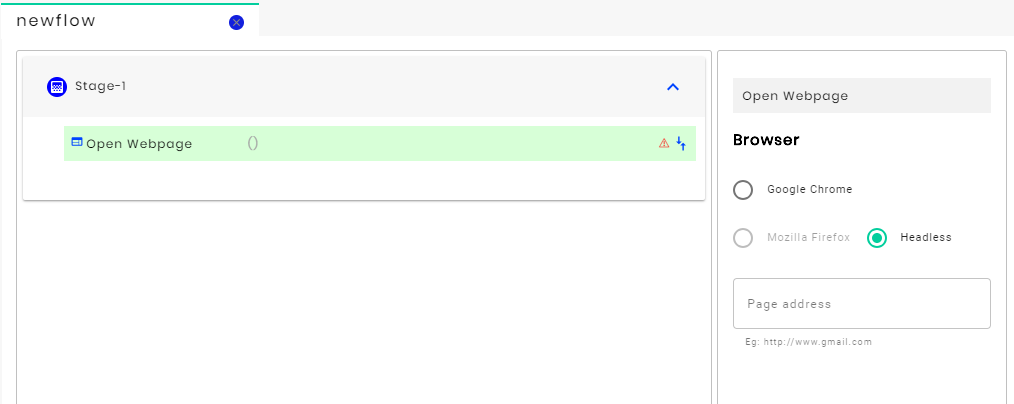
**Launch the studio**

Step 1:

Create a new flow by clicking on the . A new flow will be displayed.

Step 2:

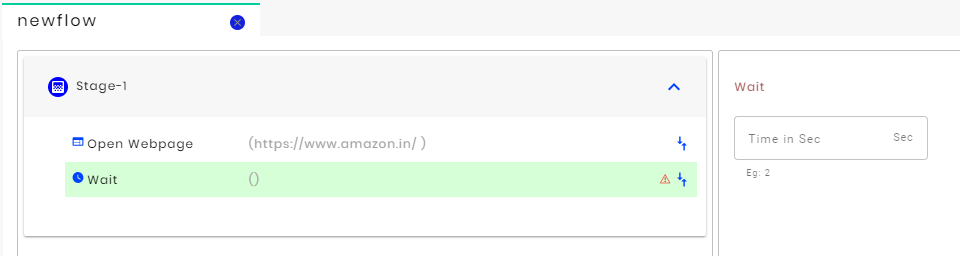
Click on the and drag it into the flow, where a new property window will pop up:



Select Google Chrome and pass the URL of the website in the “Page Address”: https://www.amazon.in/

Step 3:

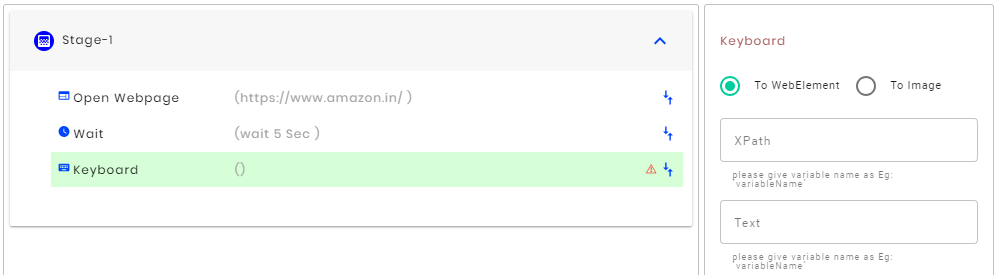
Now, drag and drop the “wait” action  into the flow.



Pass “5 sec” in the Time in sec. Here 5 seconds wait time is given to load the webpage.

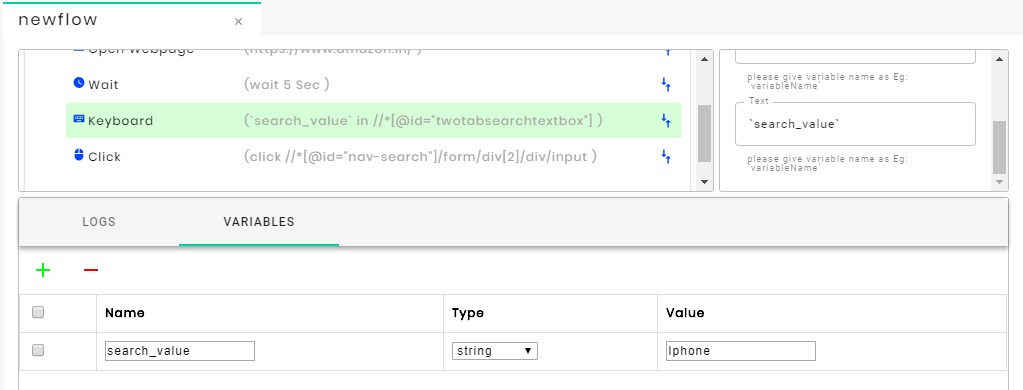
Step 4:

Drag and drop the “Enter Keystrokes”  action into the flow.



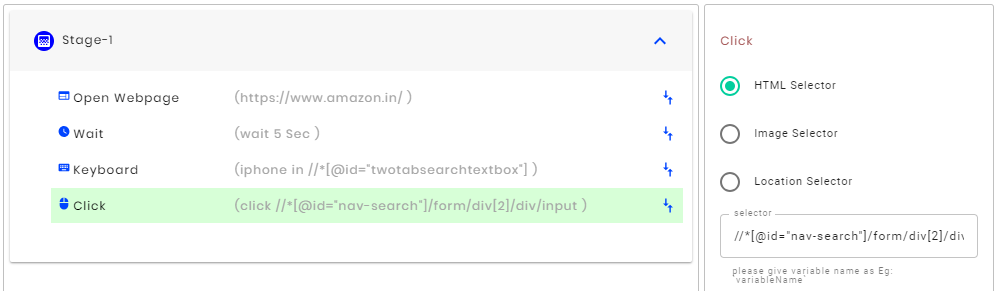
Capture the XPath of the HTML that needs to be written in the webpage and pass that XPath in the XPath.

Create a variable in the variable window and set iPhone as the value pass that variable into the Text.



Step 5:

Drag and drop the click action  into the flow



Select the HTML selector and pass the XPath of the search button in the Selector

Step 6:

Drag and drop the *wait* stage and give 5 seconds like in “step 3”

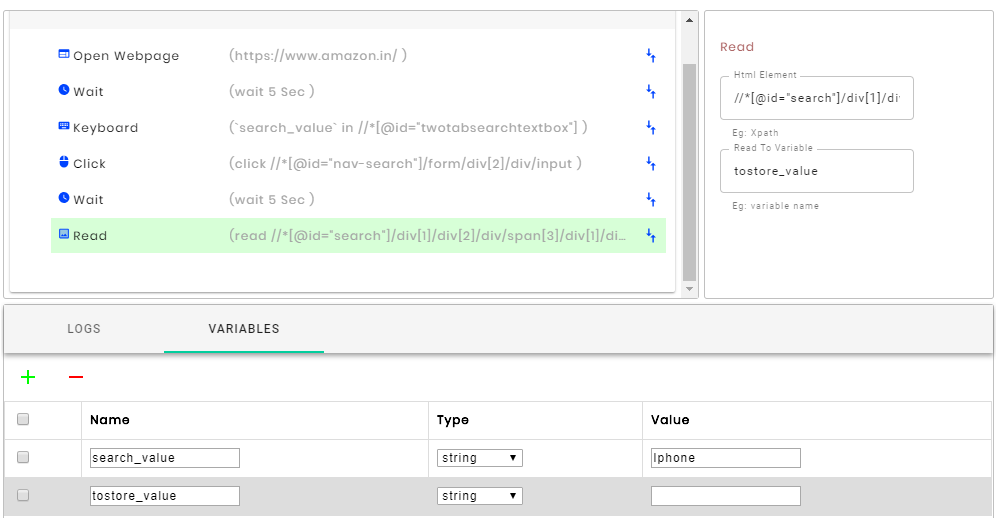


Step 7:

Drag and drop the *read* action  into the flow



Pass the XPath of which you want to read into the HTML element and create a variable in the variable window and pass the variable name in the Read to variable.

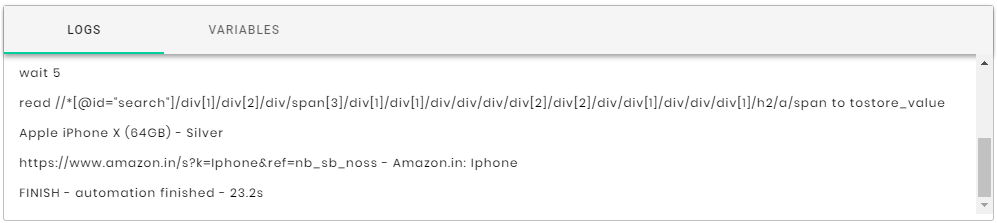


**Step 8:**

Drag and drop the *echo* action  into the flow 

Pass the variable name created to store the read value from the webpage in the Write your custom Variable.

Echo will show the results in the logs.



Step 9:

Now save the flow by clicking  and run the flow by clicking the  button in the menu bar.

THUS, THE FLOW HAS BEEN COMPLETED.

1. Security Architecture
2. Troubleshooting Guide
3. Analytics