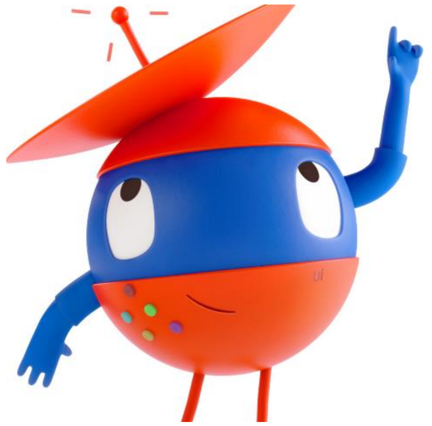
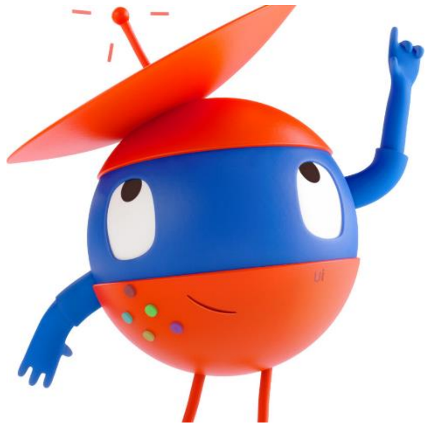
Introduction to Robotics Process Automation (RPA)

Subject Code: CSE552





Department of Computer Science and Engineering

October-December 2021

UNIT IV

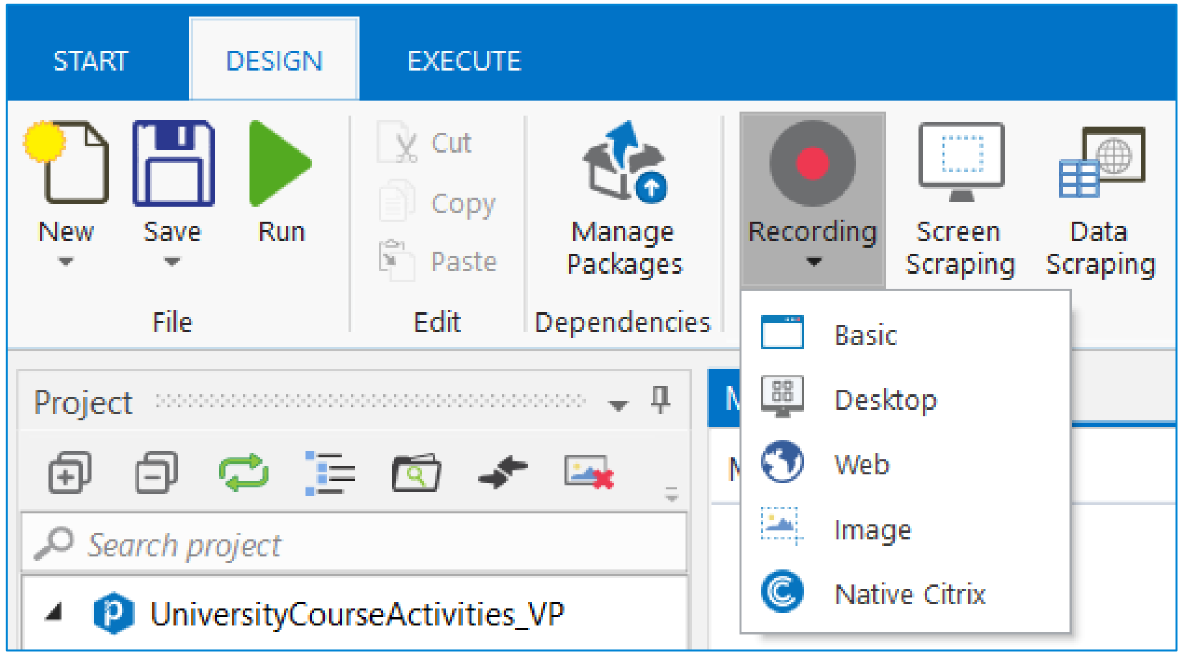
AUTOMATION CONCEPTS AND TECHNIQUES:

- Desktop and Web Recording,
- Extraction and its techniques-
- Screen scraping, Data scraping and PDF Extraction.
- Automation Techniques- Workbook and Excel automation(read/write).

EMAIL AUTOMATION:

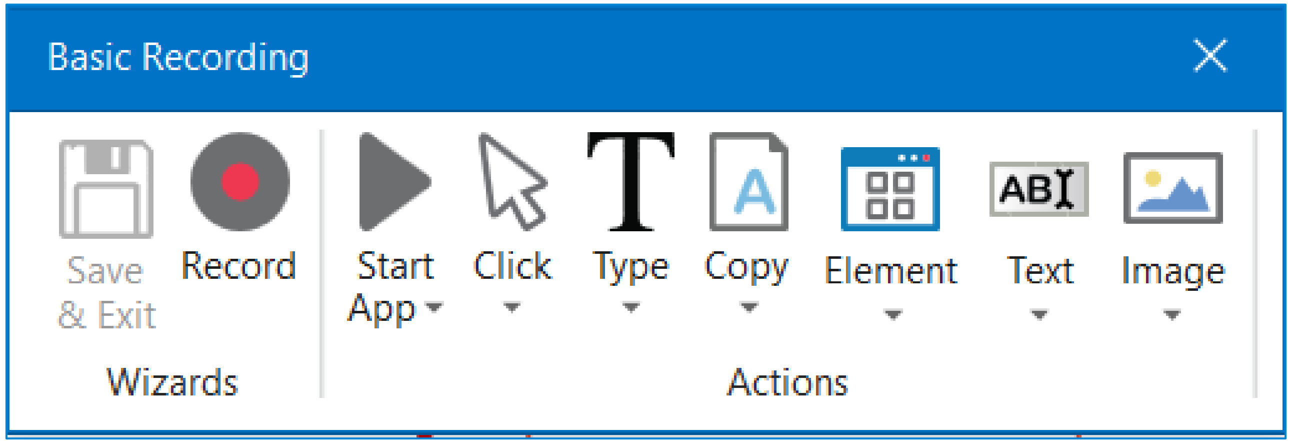
- Incoming Email automation –
- Sending Email automation

Recording



- The recording tool can be accessed from the 'Design' tab in UiPath Studio.
- While recording, all user interface elements are highlighted, allowing easy identification of buttons, fields, menus, or elements the user interacts with.
- Once the recording ends, a sequence will be created, containing the activities performed by the user.

Components of Recording Wizard



- **Save & Exit:** This option is enabled only when the recorder is on; it closes the recorder and reverts to the Designer window, displaying the automatically generated activities

- **Record:** Starts automatic recording mode, in which **multiple input activities are generated**. The user will see a blue screen over the area where an action is captured and yellow outlines for selectors.

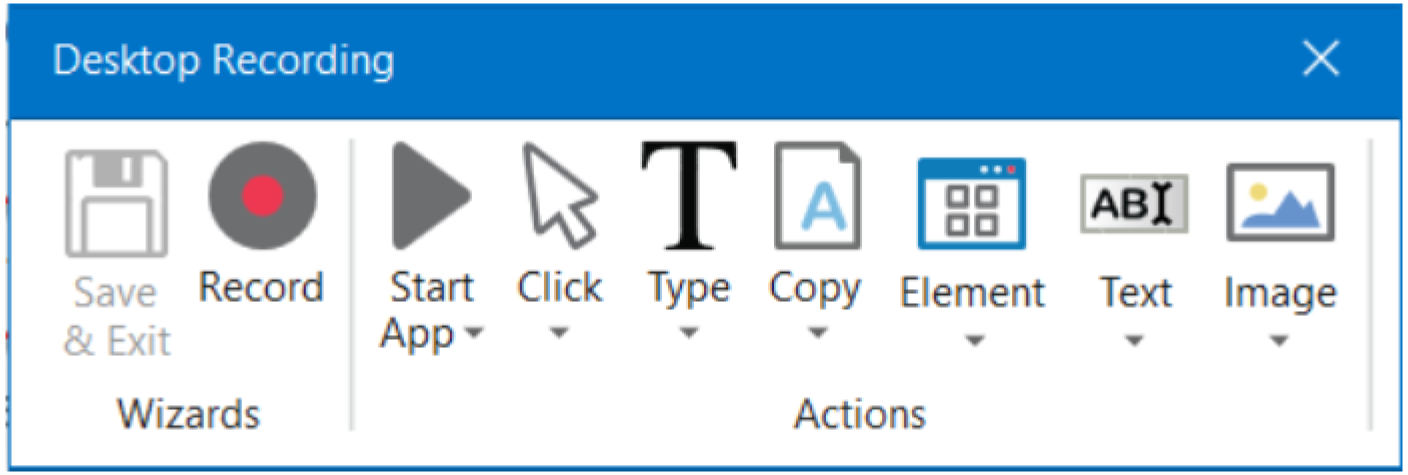
- **The options in the orange frame:** Manual Recording Actions – **starting apps, clicking, and so on.** Actions that users can select to generate single activities during their recording process.

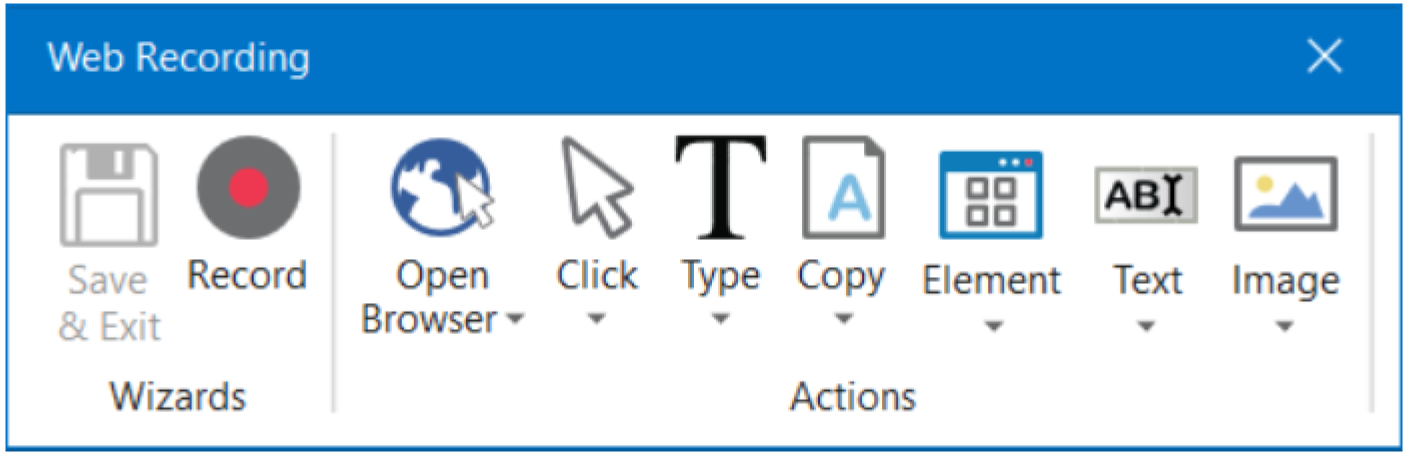
Desktop and Web Recording

❖ Recording is an important part of UiPath Studio, that can help you save a lot of time when automating your business processes. This functionality enables you to easily **capture a user's actions on the screen and translates them into sequences.**

•**Desktop** – Suitable for all types of desktop apps and multiple actions. It is faster than the Basic recorder, and generates a container (with the selector of the top-level window) in which activities are enclosed, and partial selectors for each activity.

•**Web** – Designed for recording in web apps and browsers, generates containers and uses the Simulate Type/Click input method by default

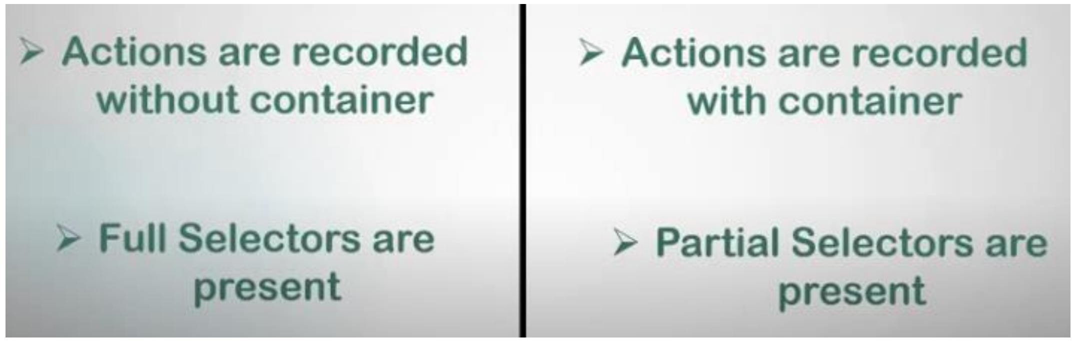




- **if its a desktop application,** it creates **“Open Application”, “Attach Window” activities** which we use to perform activities related to that particular application window.

    - This shows “Start App” and “Close App” activities But nothing related to opening a browser or closing a browser.

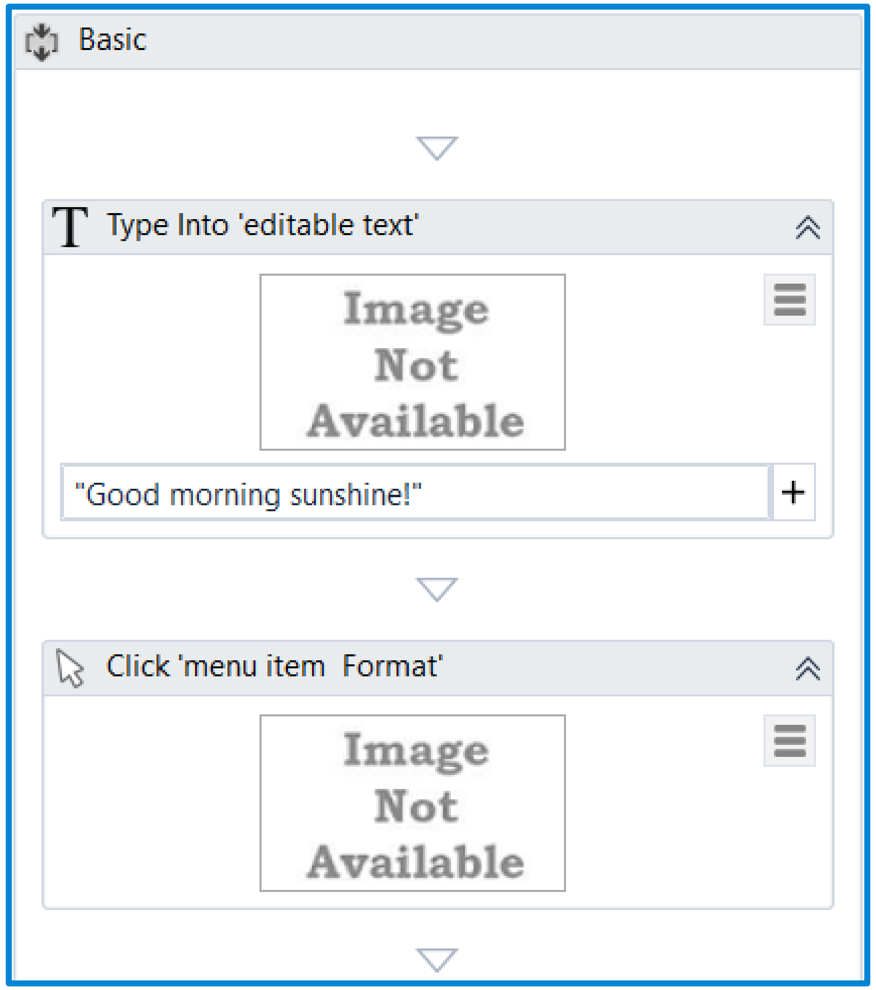
- **If its a web application,** open application and attach window is not what it generates. For web applications, it generates two different types of activities like **“Open Browser” and “Attach Browser” activity**. These activities are used when we are performing an web application loading in a browser and when we are performing different activities in a particular browser page.



➤ Actions are recorded without container	➤ Actions are recorded with container
➤ Full Selectors are present	➤ Partial Selectors are present

Basic vs Desktop Recording - Containers

Basic Recording



Basic

Type Into 'editable text'

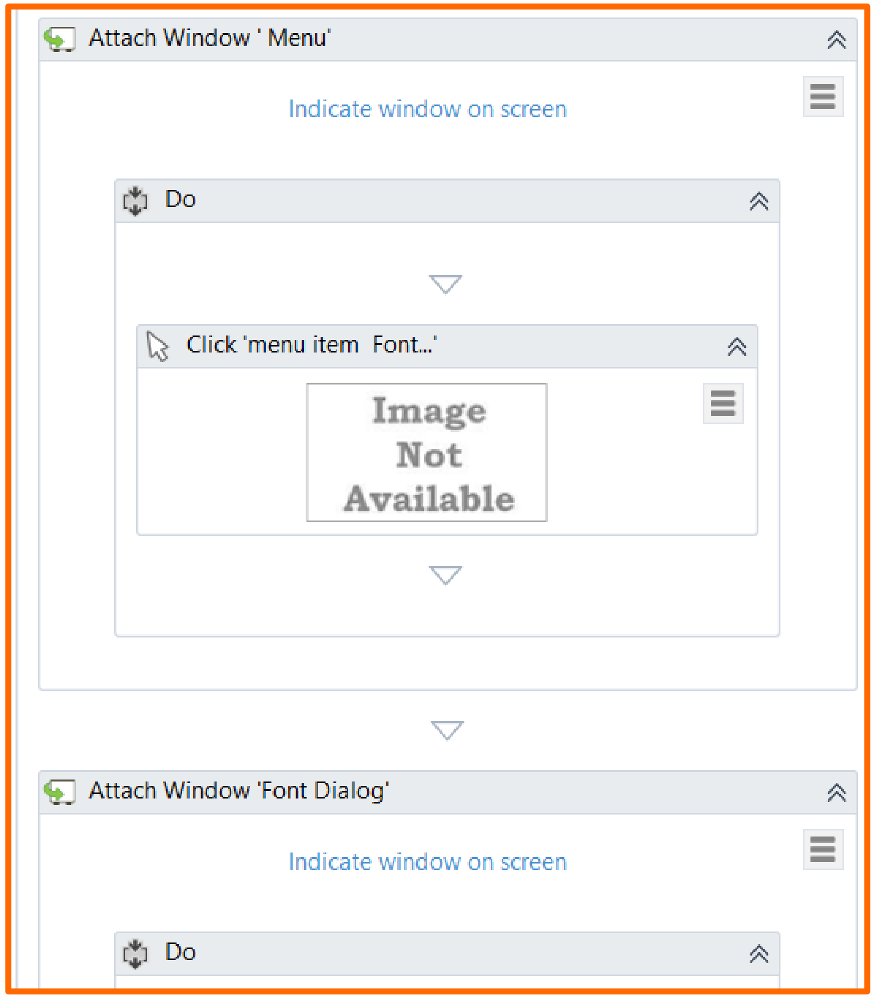
Image Not Available

"Good morning sunshine!"

Click 'menu item Format'

Image Not Available

Desktop Recording



Attach Window ' Menu'

Indicate window on screen

Do

Click 'menu item Font...'

Image Not Available

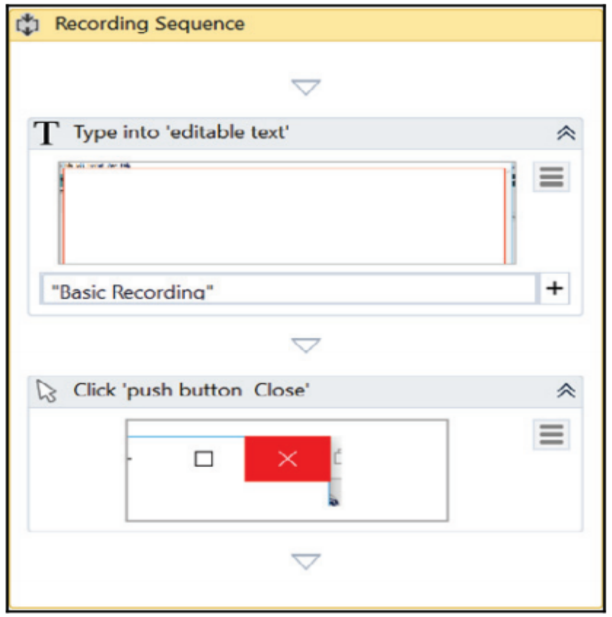
Attach Window 'Font Dialog'

Indicate window on screen

Do

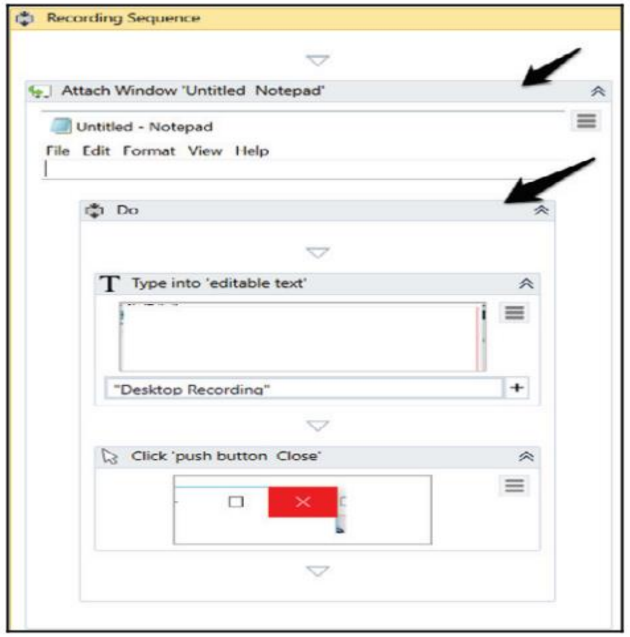
Basic Recording: This is used to record the actions of applications that have a single window

- Basic Recording uses a **full Selector.**
- It works better for applications performing **a single action.**
- It is not suitable for applications with multiple windows.
- A Full selector has all the attribute to recognize a control or application.



Desktop recording: This is similar to Basic recording with the added advantage of **working with multiple actions**.

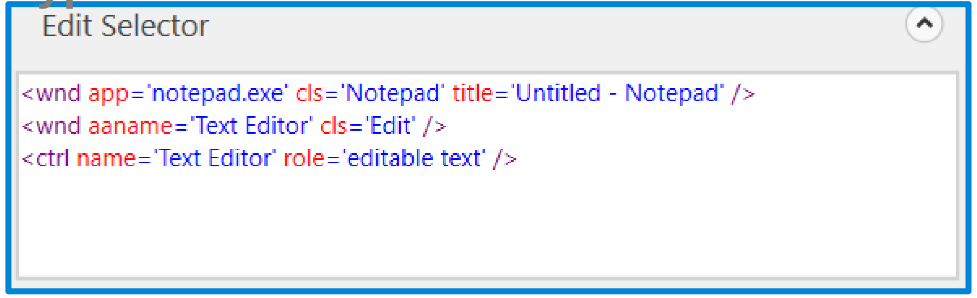
- It is most suitable for automating Desktop applications.
- Desktop recorder generates **Partial selectors.**
- The Partial selectors, have a hierarchical structure. They are split into parent child views for recognizing the UI element properly


Recording Sequence
Attach Window 'Untitled Notepad'
Untitled - Notepad
File Edit Format View Help
Do
Type into 'editable text'
"Desktop Recording"
Click 'push button Close'

Basic vs Desktop Recording

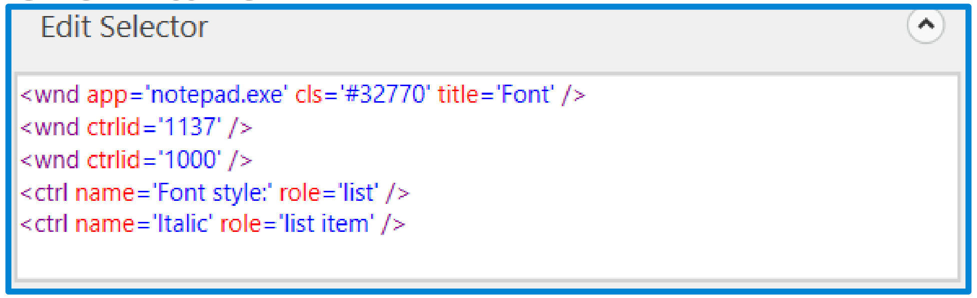
Basic Recording

Type into


Edit Selector

```
<wnd app='notepad.exe' cls='Notepad' title='Untitled - Notepad' />
<wnd aaname='Text Editor' cls='Edit' />
<ctrl name='Text Editor' role='editable text' />
```

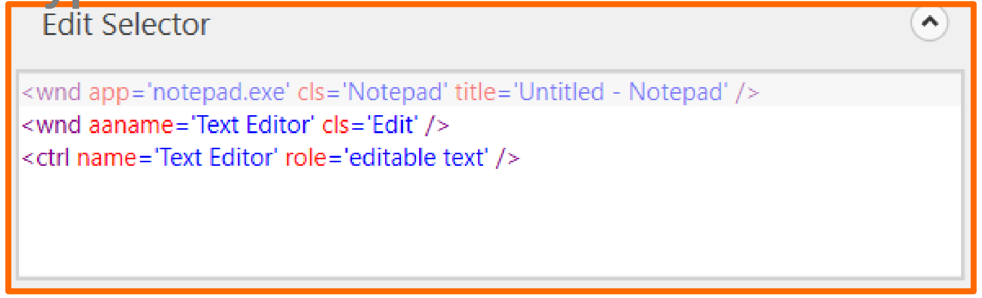
Click ‘Italic’


Edit Selector

```
<wnd app='notepad.exe' cls='#32770' title='Font' />
<wnd ctrlid='1137' />
<wnd ctrlid='1000' />
<ctrl name='Font style:' role='list' />
<ctrl name='Italic' role='list item' />
```

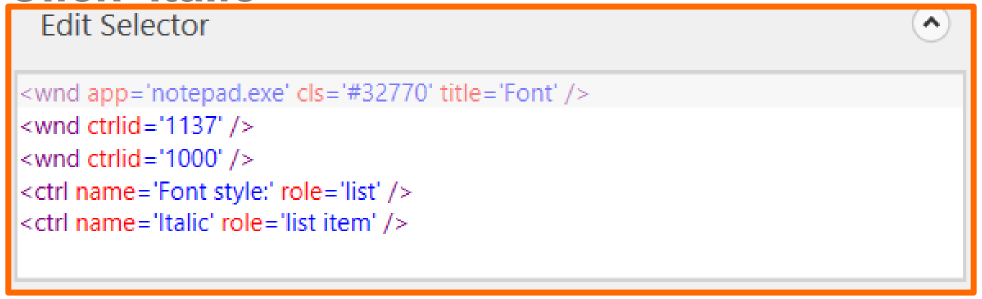
Desktop Recording

Type into


Edit Selector

```
<wnd app='notepad.exe' cls='Notepad' title='Untitled - Notepad' />
<wnd aaname='Text Editor' cls='Edit' />
<ctrl name='Text Editor' role='editable text' />
```

Click ‘Italic’


Edit Selector

```
<wnd app='notepad.exe' cls='#32770' title='Font' />
<wnd ctrlid='1137' />
<wnd ctrlid='1000' />
<ctrl name='Font style:' role='list' />
<ctrl name='Italic' role='list item' />
```

Web Recording

Start app

It is used to open the application. It is similar to the **‘Open Application’ activity** of UiPath studio. To use this, we have to select “Start App” and then click on the application that we want to open.

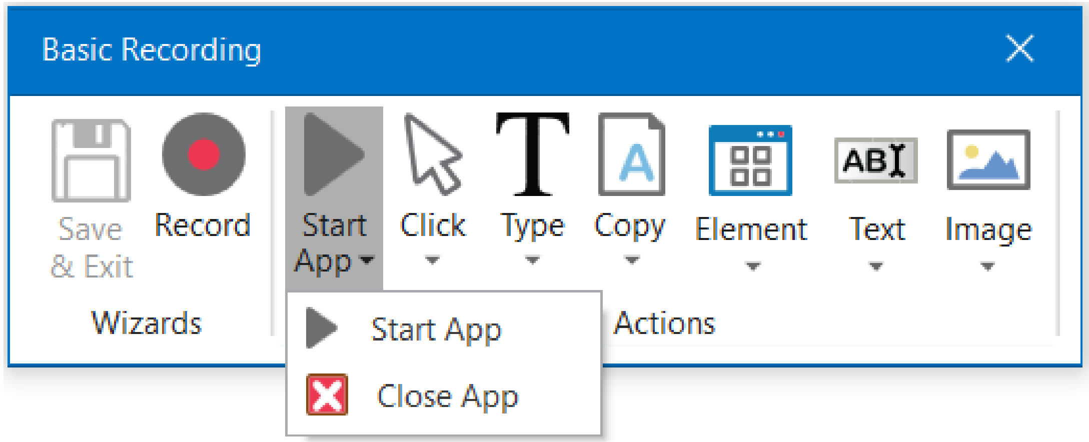
Close app

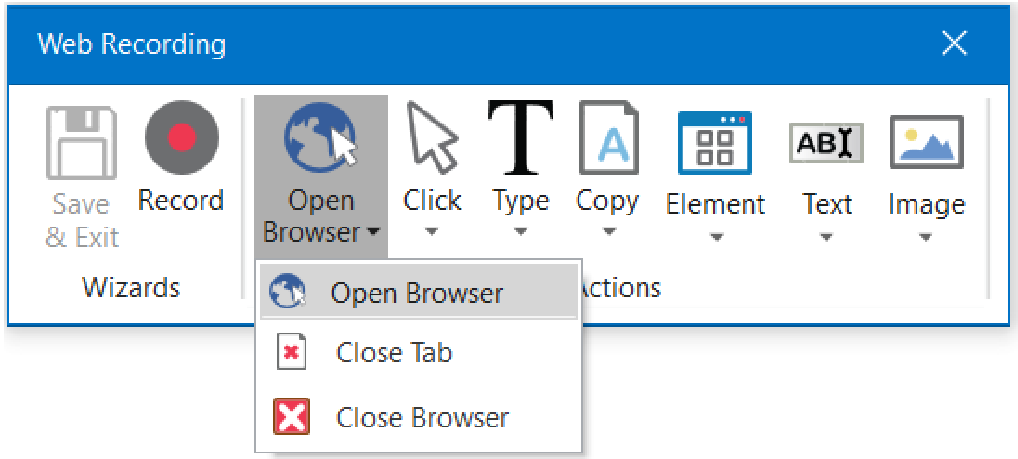
It is used to close the opened application. It is similar to the **‘Close Application’ activity** of UiPath studio. To use this, we have to select “Close App” and then click on the application that we want to close.

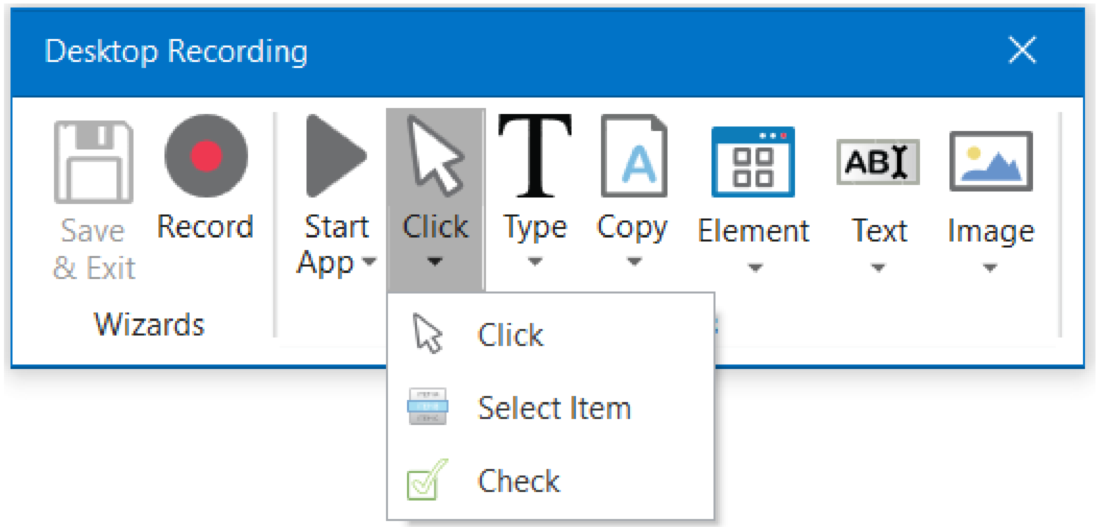
Open Browser: The browser can be chosen (Internet Explorer, Firefox or Chrome), the URL can be specified inside the “***”, or we can store the URL inside the string variable and pass that inside it.

Close Tab: Closes the Browser page that you want to close.

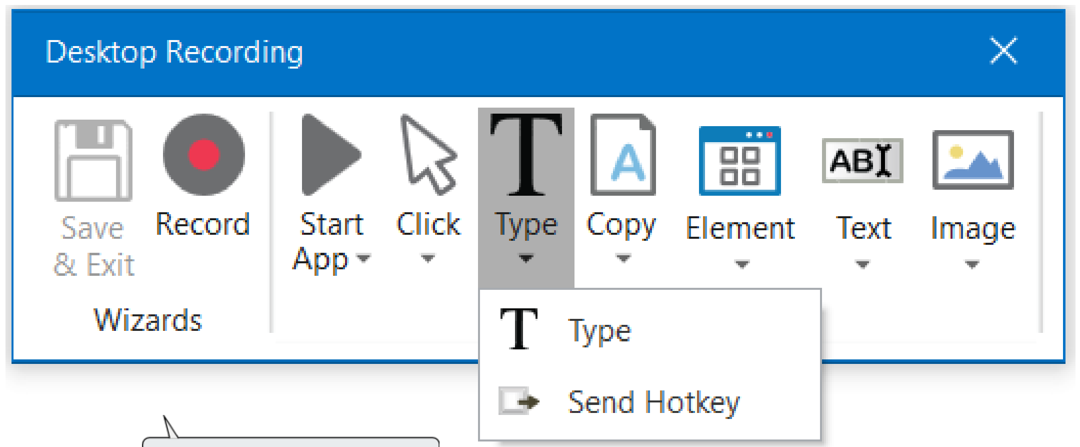
Close Browser: Closes the browser that is opened. You have to select “Close Browser” and then click on the browser that you want to close.

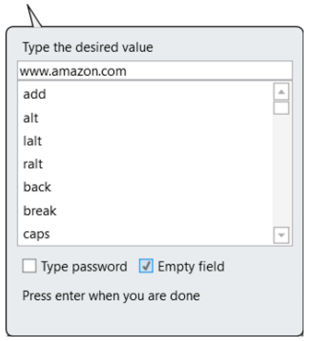






- Click activity is used to click on the option or any button.
- Select Item is used to select an item from a combo box or list box.
- Check is used to check or uncheck the options or toggle.



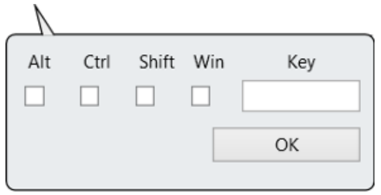

Type the desired value

www.amazon.com

add
alt
lalt
ralt
back
break
caps

☐ Type password ☑ Empty field

Press enter when you are done

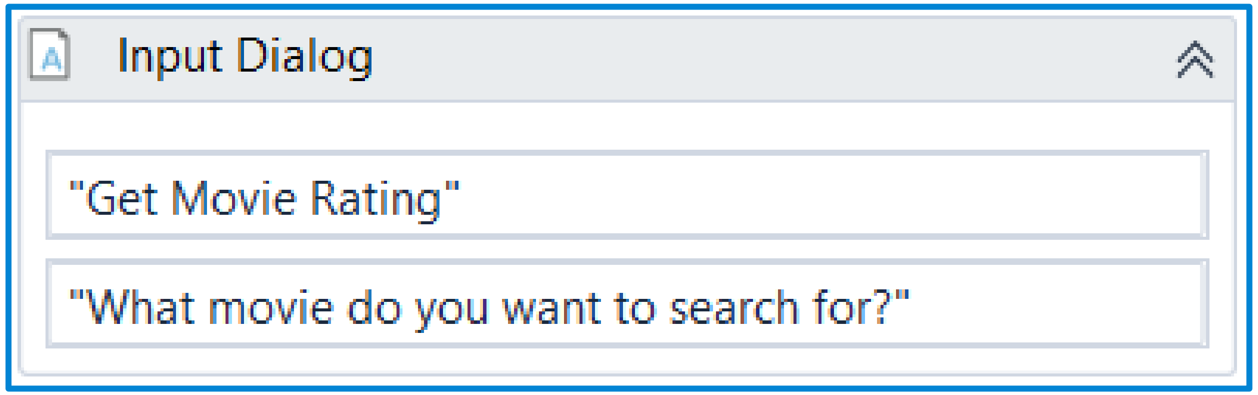


Type is the same activity that is created when the Automatic Recording functionality identifies the text typing action. It is generated when a text is typed into an editable Ui **UI element.**

Send Hotkey is an activity that can be created only using Manual Recording since hotkeys are not captured under Automatic Recording. The action designed is used to define a hotkey (using one or **several modifier keys – Alt/Ctrl/Shift/Win + one key)** and to send it towards the target app. It can be beneficial for actions that are hard to perform using the mouse and keyboard.

Web Recording: IMDb Rating

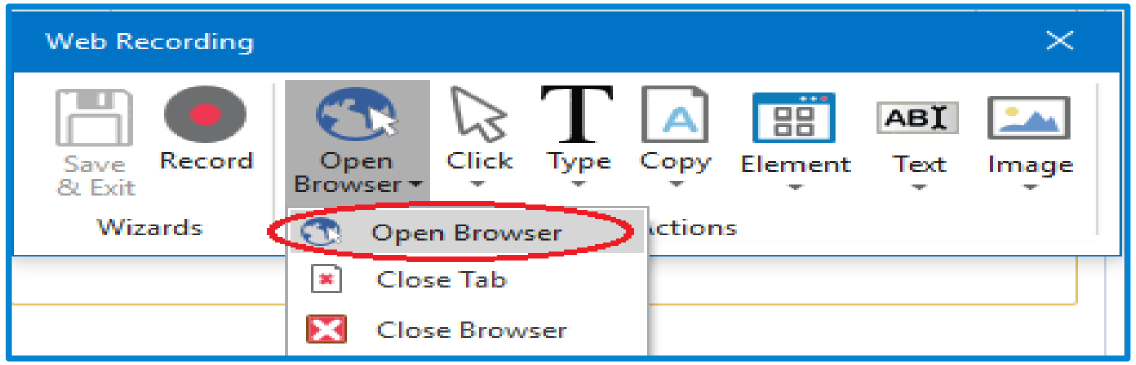
1. Input Dialog: user provides a movie title
2. Open browser: www.imdb.com
3. Type into the navigation bar the movie title
4. Click search button
5. Click on the first title from the results page
6. Get value: movie rating
7. Message box: display movie rating to user



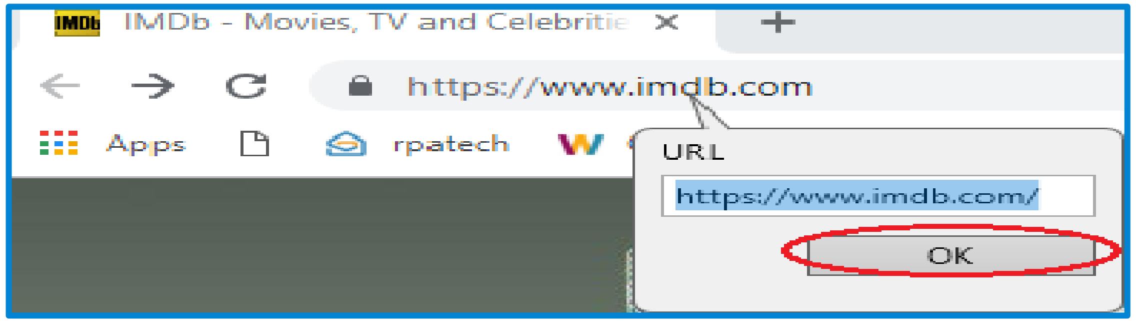
"Get Movie Rating"

"What movie do you want to search for?"

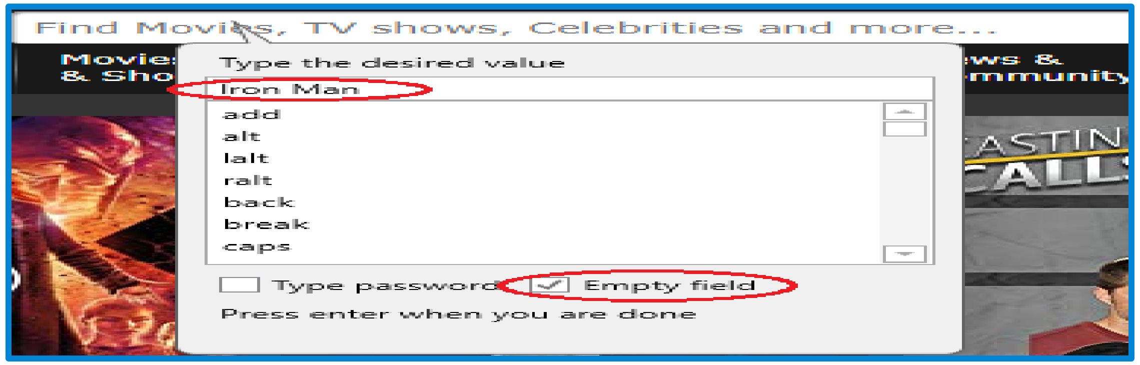
- 1. The first step is to take the movie title from the user. This is done in our case with an Input Dialog.
- The outcome (what the user types in) is stored in a string variable (named 'Movie' in our case).
- This activity is outside the web recording feature, being added manually



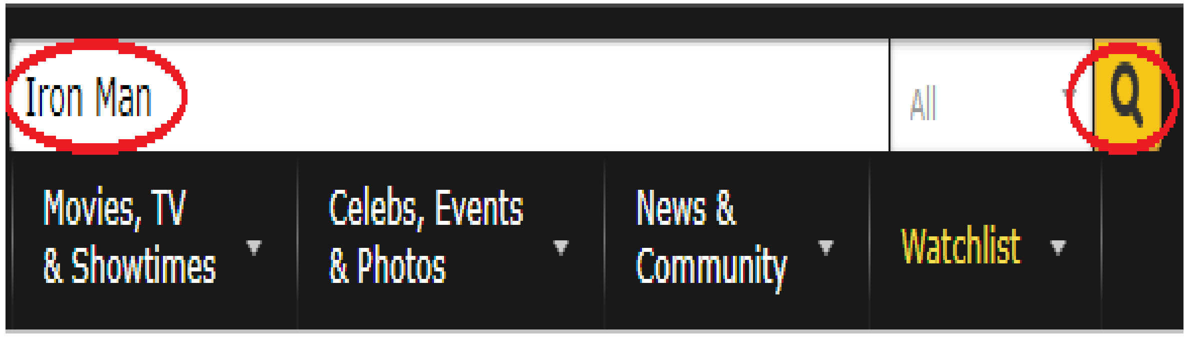
2. The next step is to open a browser and type in the www.imdb.com address. Remember, you can choose the browser. Once the browser is open, the activities to follow are placed inside the container.



3. Indicate the URL and Click OK.

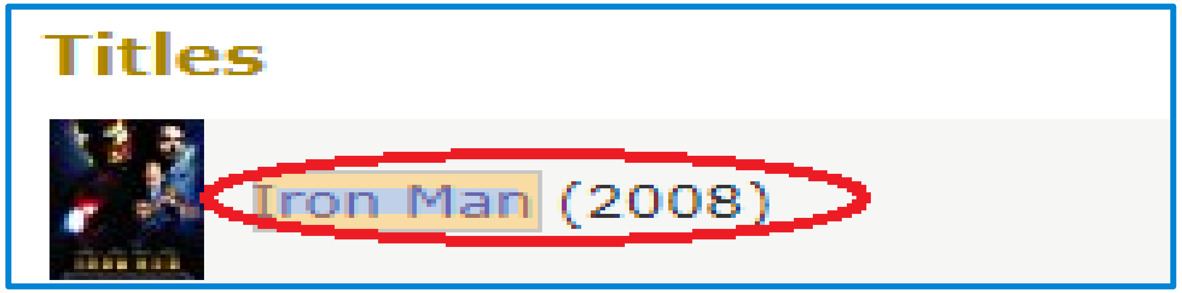


4. Click on the record option.
- Type the movie name “Iron Man”.
- Click on the empty field box.
- Press the enter button.



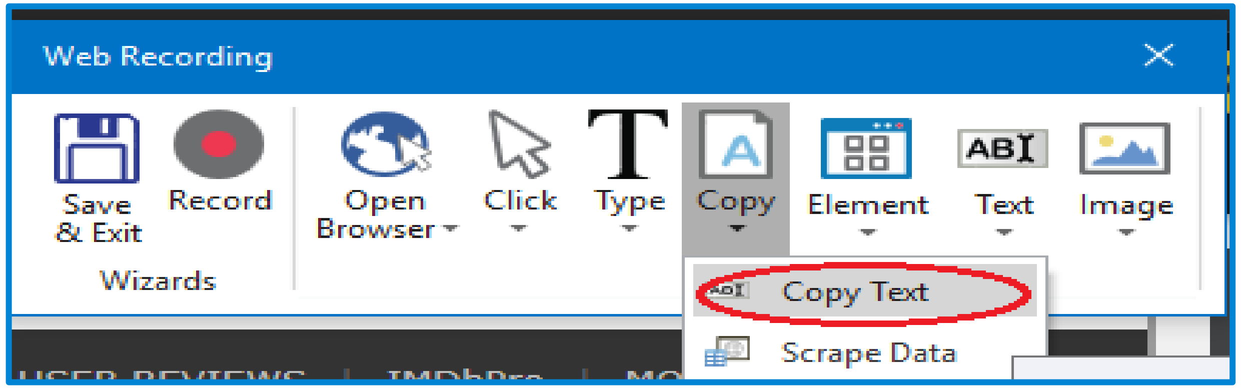
5. Click on the search bar option.

6. Click on the Iron Man movie that is displayed in the result.

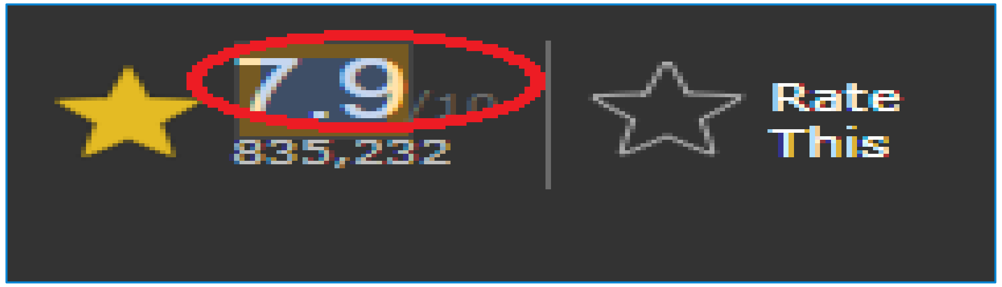


7. Click on Copy.

Select "Copy Text" from Dropdown list.

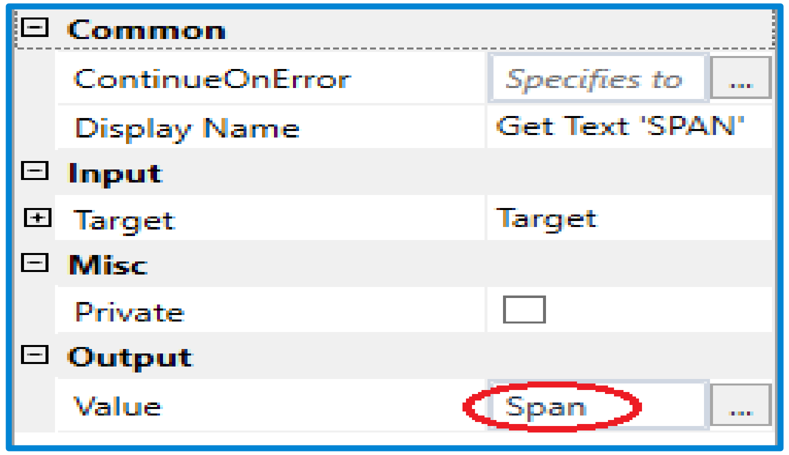


8. Indicate the rating of the movie.



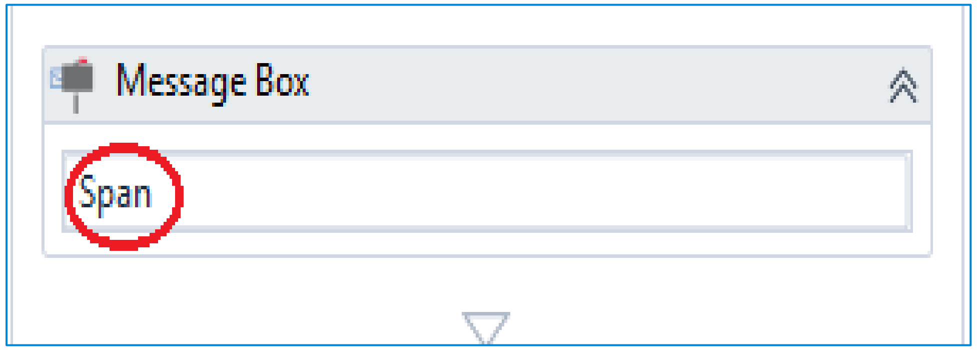
9. Go Back to UiPath Studio.

- Click on the copy text activity.
- Open the Properties of the copy text.
- The output variable is Span.



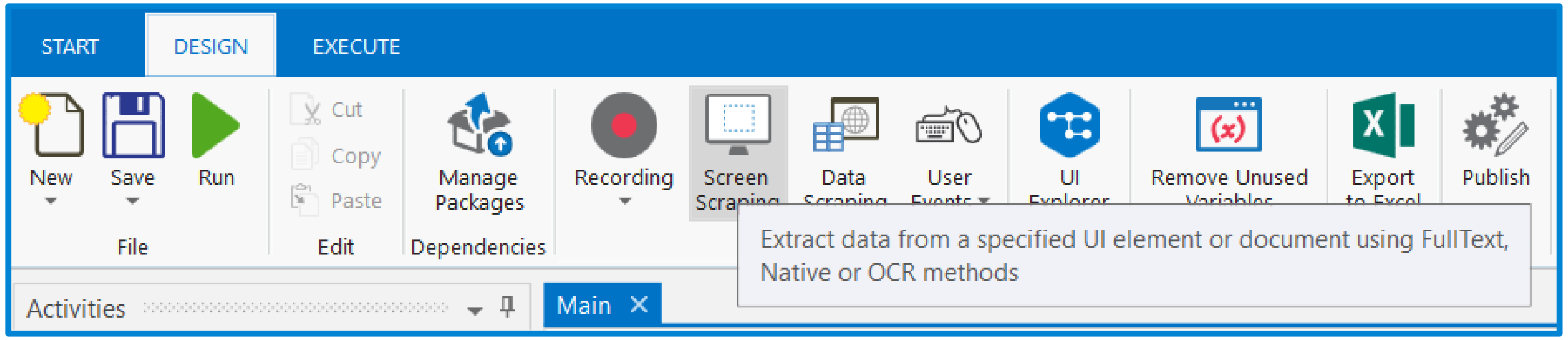
Common	
ContinueOnError	Specifies to
Display Name	Get Text 'SPAN'
Input	
Target	Target
Misc	
Private	☐
Output	
Value	Span

- 10. Drag the message box from the activity panel.
- Pass the variable span inside it.

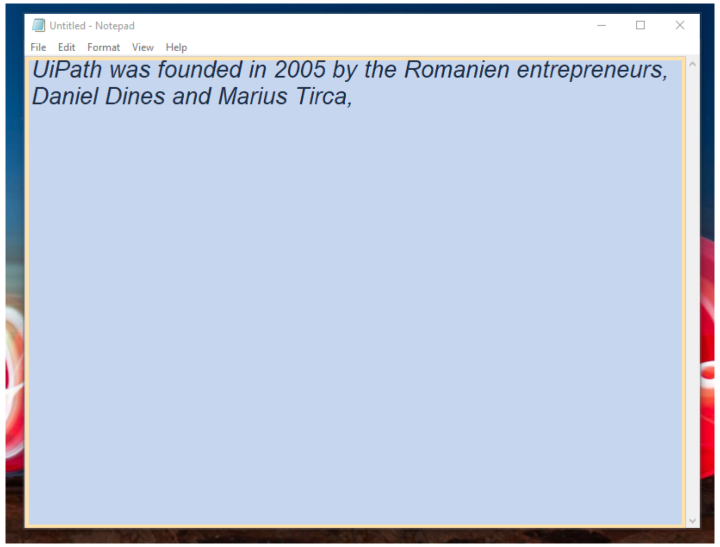


Extraction and its techniques- Screen scraping

The **Screen Scraping Wizard** enables you to point at a UI element and extract text from it, using one of the three output methods. The steps are as follows:

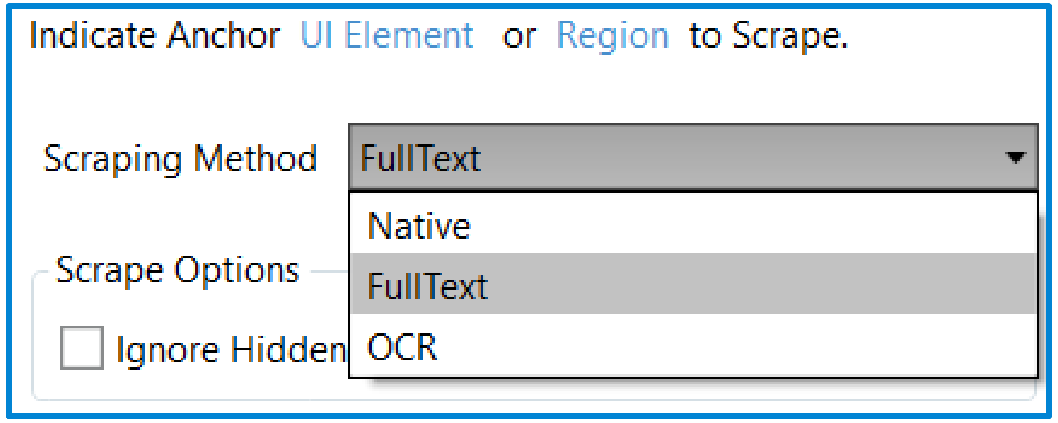


Extract data from a specified UI element or document using FullText, Native or OCR methods



UiPath was founded in 2005 by the Romanien entrepreneurs, Daniel Dines and Marius Tirca,

1. Start the Screen Scraping Wizard from the Studio menu bar
2. Select the UI element in Computer Vision mode
3. Select the screen scraping method from the Options panel



Indicate Anchor UI Element or Region to Scrape.

Scraping Method: FullText

Screen Scraping Screen Scraping is a method of **extracting data from documents, websites, and PDFs.** It is a very powerful method for extracting text. We can extract text using the Screen Scraper wizard. The Screen Scraper wizard has three scraping methods: Full Text ,Native and OCR

Full text:

- The Full text activity is used to extract information from various types of documents and websites.
- It has a 100% accuracy rate.
- It is the fastest method among all three methods.
- It even works in the background.
- It is also capable of extracting hidden text. However, it is not suitable for Citrix environments.

Native: This is similar to the Full text method but has some differences.

- It has a slower speed than the Full text method.
- It has a 100% accuracy rate, like the Full text method.
- It does not work in the background.
- It has an advantage over the Full text method in that it is also capable of extracting the text's position.
- It cannot extract hidden text. It also does not work with a Citrix environment.

OCR: This method is used when the previous two methods fail to extract information.

- It uses the two OCR engines: Microsoft OCR and Google OCR.
- It has also a scale property: you can choose the scale level as per your need.
- Changing the scale property will give the best results:

Types of OCR:

There are two OCRs available in our UiPath Studio:

1. Microsoft OCR
2. Google OCR

However, we are free to import other OCR engines into our project.

Both the Microsoft and Google OCR engines have their own advantages and disadvantages. The advantages of Google OCR include the following:

- Multiple language support can be added in Google OCR
- Suitable for extracting the text from a small area
- It has full support for color inversion
- It can filter only allowed characters

The advantages of Microsoft OCR include the following:

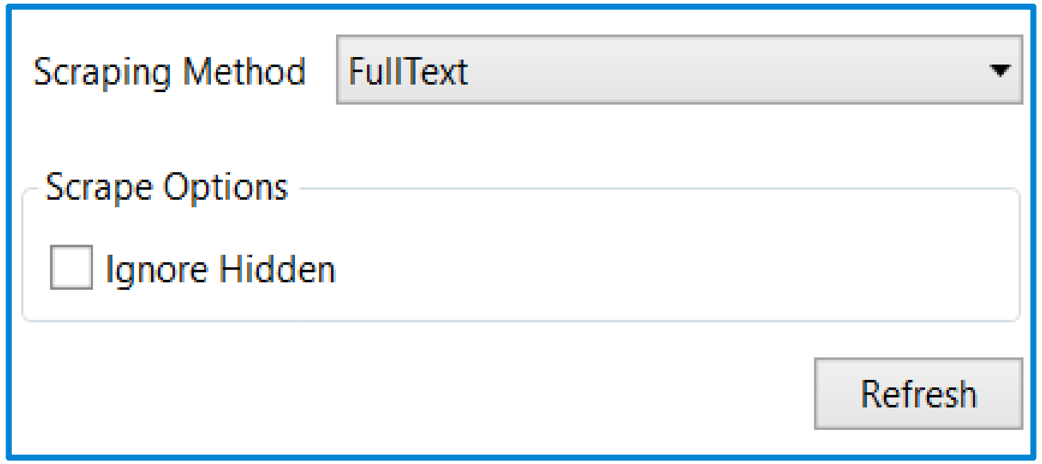
- Multiple languages are supported by default
- It is suitable for extracting text from a large area

OCR is not 100% accurate. It is useful for extracting text that other methods cannot successfully do. It works with all applications, including Citrix.

Microsoft and Google's OCRs are not the optimum for every situation. Sometimes, we have to look for more advanced OCRs to recognize more sophisticated text, such as handwritten documents and so on.

There is another OCR available in UiPath Studio, known as the Abbyy OCR Engine. You can find this OCR engine in the **Activities** panel by searching for `OCR`.

The FullText Output Method


Scraping Method FullText

Scrape Options

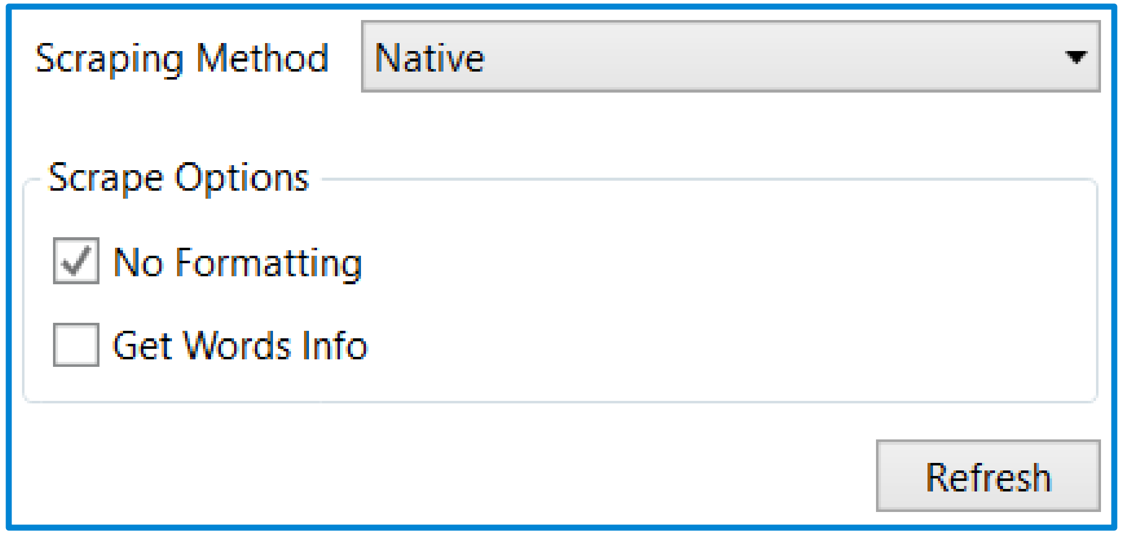
☐ Ignore Hidden

The **FullText output method** captures all the text from a terminal screen. It is the default method in UiPath.

- It is fast, accurate and can work in the background. It captures all the text from the terminal screen, including the hidden text (from options, drop-down lists, and so on).

Ignore Hidden: When this check box is selected, the hidden text from the selected UI Element is not copied.

The Native Output Method



Scraping Method: Native

Scrape Options

[✓] No Formatting

[ ] Get Words Info

The **Native output method** works with applications that are built to render text with Graphics Device Interface (GDI). It can extract the screen coordinates of the text

- It is very helpful when coordinates of each word are needed, and when font and color of the text is needed to be retained accurately.
- On the other hand, it only extracts the visible text, it cannot work in the background and it doesn't support Citrix.

The Native method offers the following options:

- **No Formatting:** When the font and color are not important, this box can be checked, and the wizard extracts only the text, just like the FullText method.
- **Get Words Info:** This option can extract the position of each word and can support several separators. By default, it can process all known characters as separators (comma, space, and so on), but when only certain separators are specified, it can ignore the others. Additionally, the Custom Separators field is displayed, enabling the user to specify the characters used as separators. If the field is empty, all known text separators are used.

The OCR Output Method

The **OCR output method** uses the **Optical Character Recognition technology**, to extract information from virtual environments. It has two default engines: **Google Tesseract and Microsoft** MODI

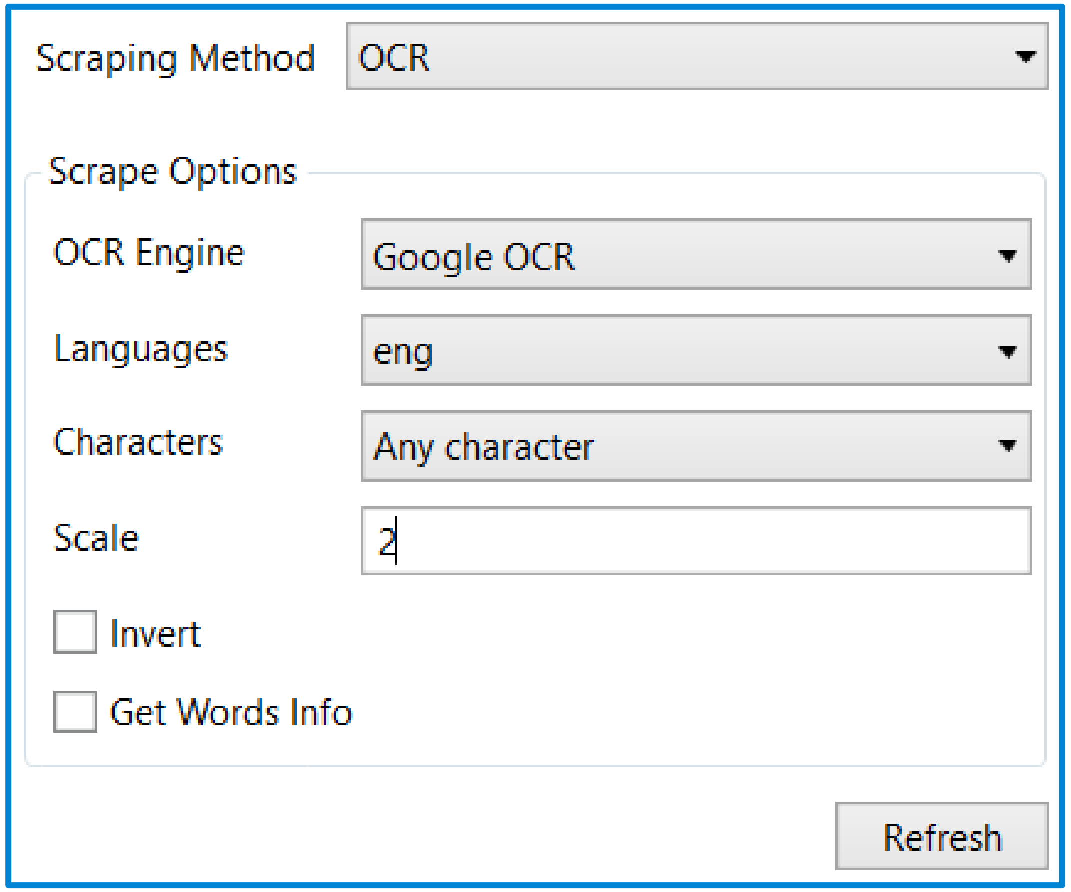
The OCR method has two default engines:

- **Google Tesseract:** It gets better results on smaller size areas and supports color inversion. It has filters that can be used to select only certain categories of characters.
- **Microsoft MODI:** It performs better when it comes to Microsoft fonts and on larger size areas. Moreover, it supports multiple languages.

	Multiple Languages Support	Preferred Area Size	Support for Color Inversion	Filter Allowed Characters	Best with Microsoft Fonts
Google Tesseract	Can be added	Small	YES	YES	NO
Microsoft MODI	Supported by default	Large	NO	NO	YES

Google Tesseract

The Google Tesseract OCR engine is more effective with character recognition on small size areas. It offers multiple customization options:



Scraping Method	OCR

Scrape Options

OCR Engine	Google OCR
Languages	eng
Characters	Any character
Scale	2

- [ ] Invert
- [ ] Get Words Info

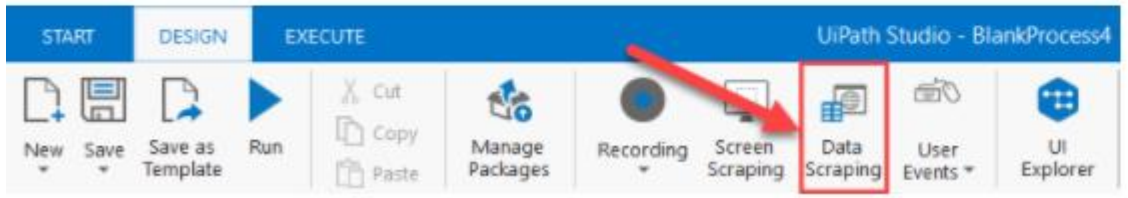
Languages: Enables language change for the scraped text. By default, English is selected.

Characters: Enables the selection of type of characters to be extracted: any character, numbers only, letters, uppercase, lowercase, phone numbers, currency, date and custom.

Screen scraping, Data scraping

Screen scraping methods refer to those activities that enable you to **extract data from a specified UI element** or document, such as a .pdf file. The screen scraping wizard enables you to point at a UI element and extract text from it, using one of the three output methods

Data scraping enables you to **extract structured data** from your browser, application or document to a database, .csv file or even Excel spreadsheet.



There are three different methods to perform screen scraping –

- **Full text –** all visible object on UI object is scrapped
- **Native –** scrapes text, also captures the position, color, font style of the text.
- **OCR –** scrapping in virtual desktops and Citrix applications

Capability Method	Speed	Accuracy	Background Execution	Extract Text Position	Extract Hidden Text	Support for Citrix
FullText	10/10	100%	yes	no	yes	no
Native	8/10	100%	no	yes	no	no
OCR	3/10	98%	no	yes	no	yes

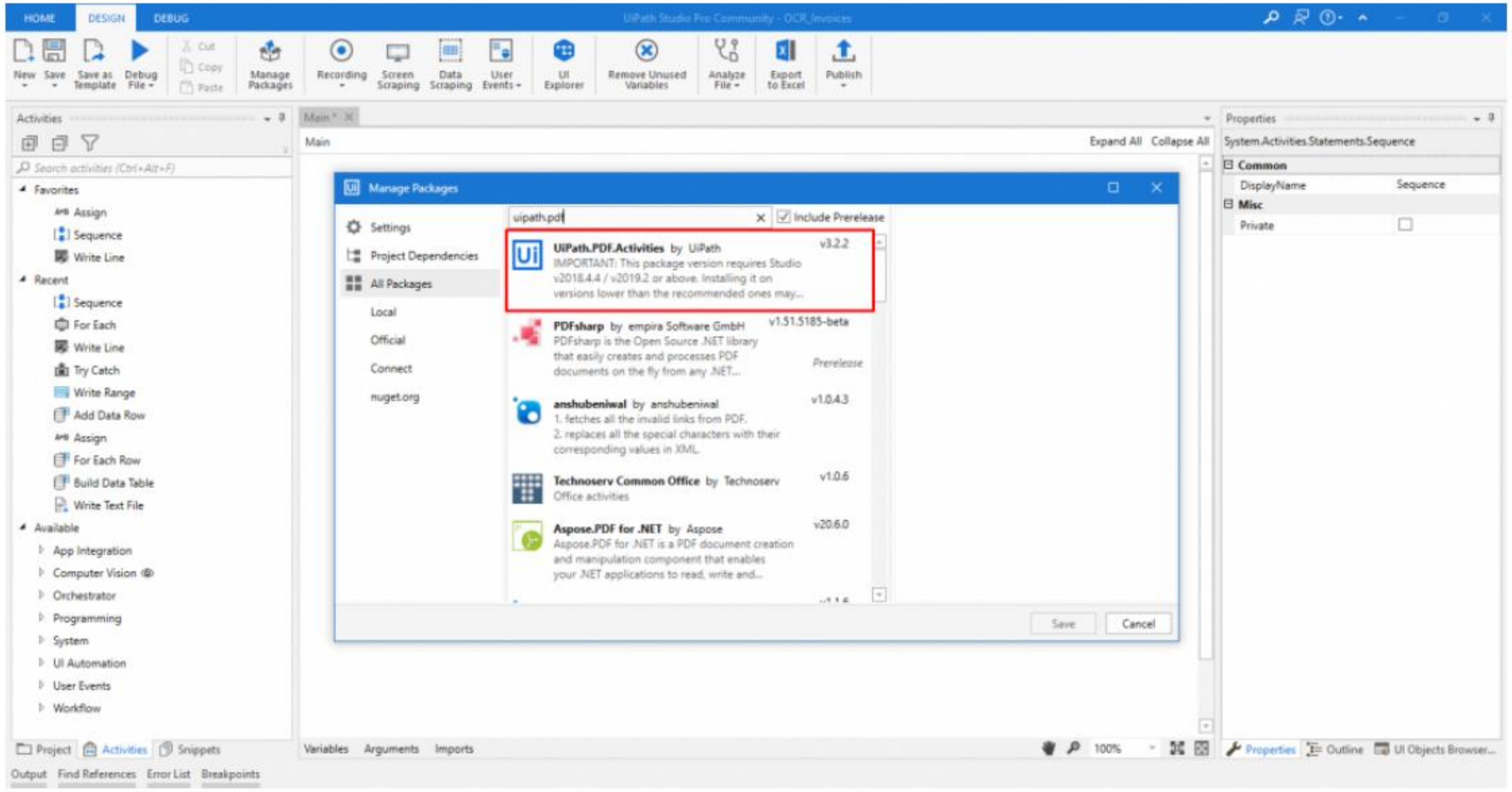
Task: Extract data from Google Contacts and store it in a file.

Follow the below steps to achieve the task:

•Use the Data Scraping tool to extract data.
•Extract the correlated values accordingly.
•Store the data in a CSV file by using the Write CSV activity.

PDF Extraction

In order to read the PDF document, the **UiPath.PDF.Activities** package should be installed (Figure). Hence, in the ribbon, you can find the Manage Packages button where you can search and install the required package. This package needs to be installed every time a new process in UiPath is created.



UiPath.PDF.Activities by UiPath v3.2.2
IMPORTANT: This package version requires Studio v2018.4.4 / v2019.2 or above. Installing it on versions lower than the recommended ones may...

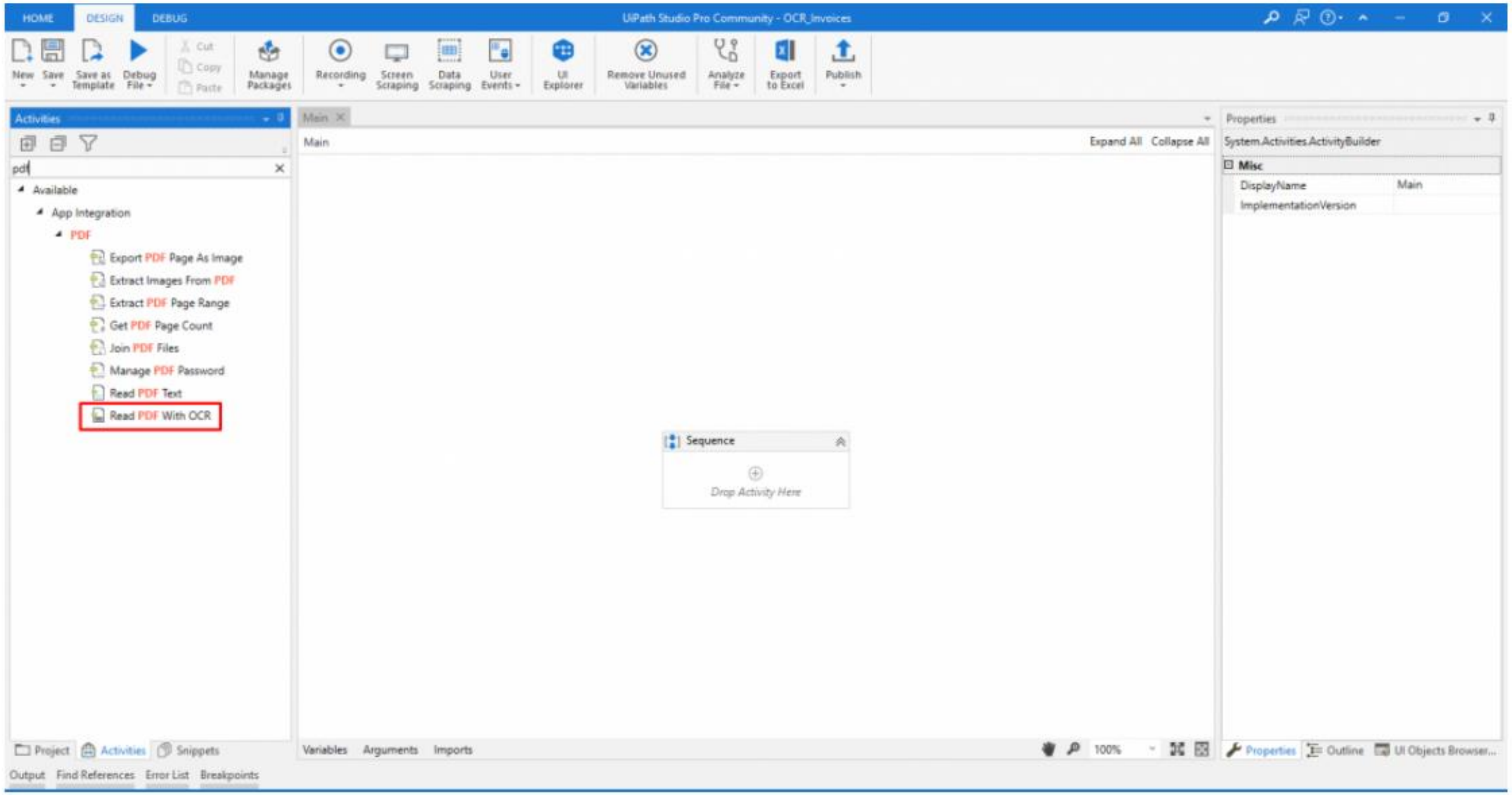
PDFsharp by empira Software GmbH v1.51.5185-beta Prerelease
PDFsharp is the Open Source .NET library that easily creates and processes PDF documents on the fly from any .NET...

anshubeniwal by anshubeniwal v1.0.4.3
1. fetches all the invalid links from PDF.
2. replaces all the special characters with their corresponding values in XML.

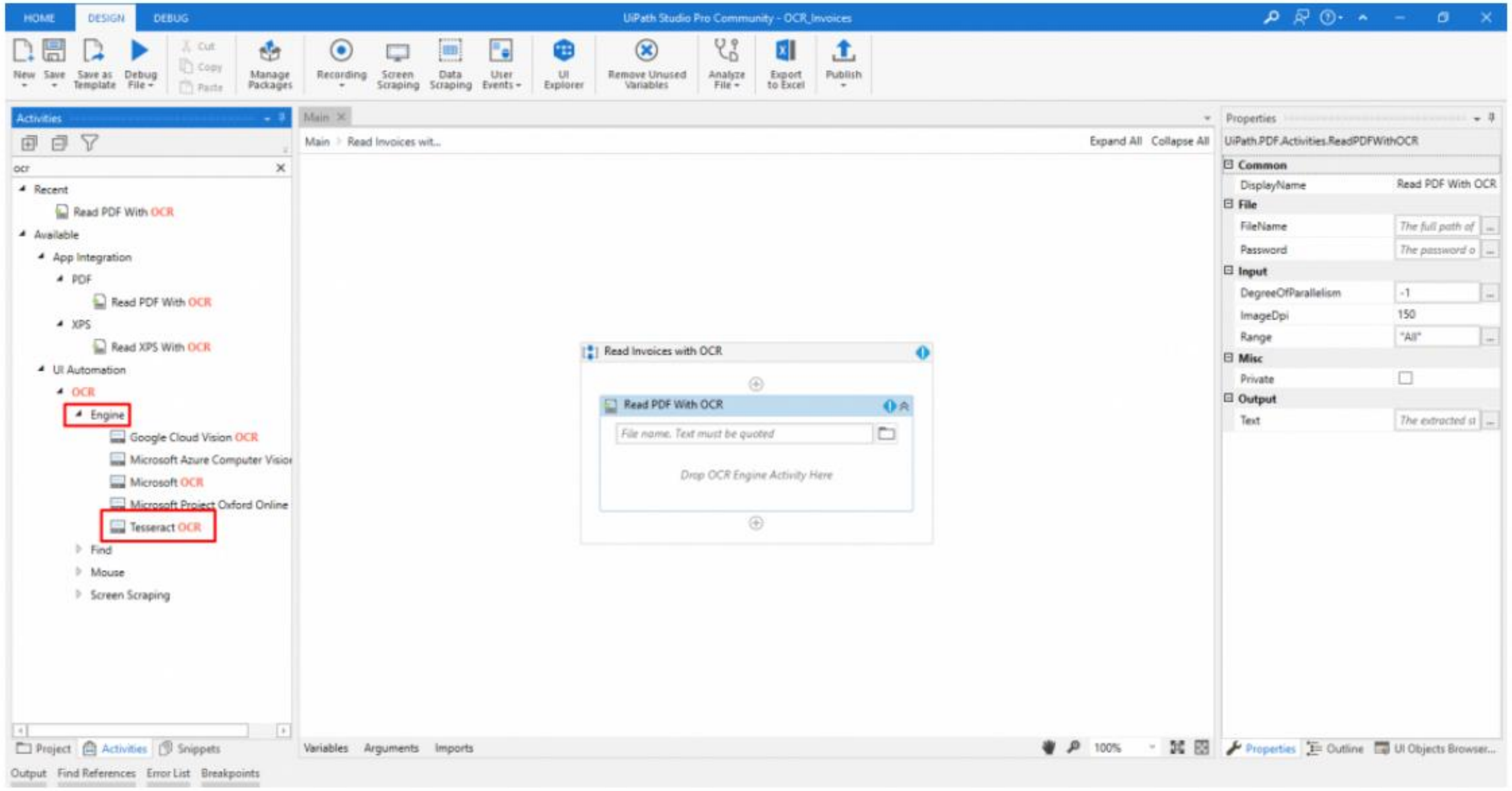
Technoserv Common Office by Technoserv v1.0.6
Office activities

Aspose.PDF for .NET by Aspose v20.6.0
Aspose.PDF for .NET is a PDF document creation and manipulation component that enables your .NET applications to read, write and...

After the installation is over, in the left panel, the activities tab, search for the activity **Read PDF with OCR**, and add it to the sequence (Figure).



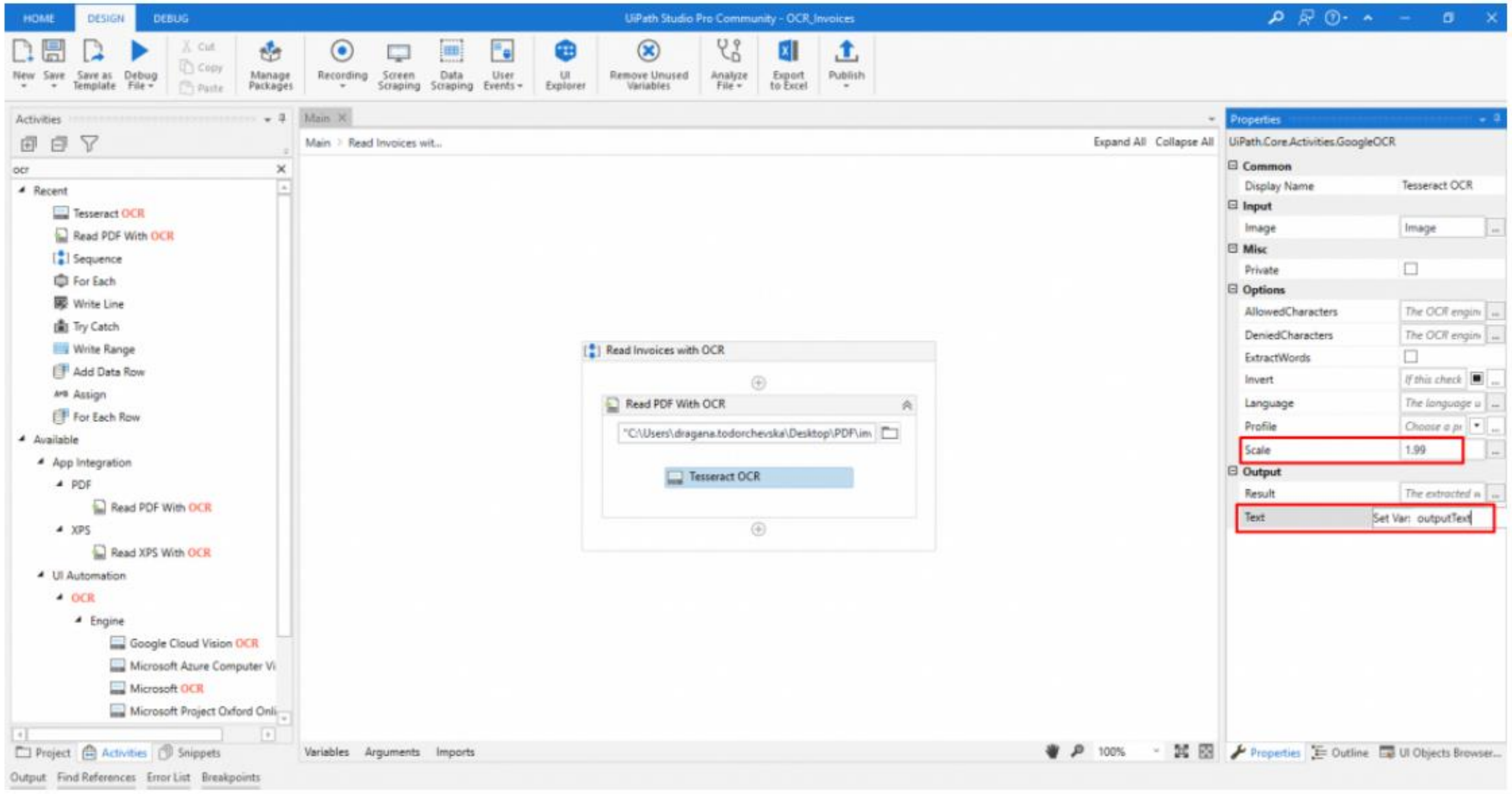
After the read activity is added, the next required fields are the file name and the OCR Engine (Figure). **Click on the folder to browse for the PDF file that you want to extract data from**, and afterward search in the activities panel for the OCR engine. In this process the **Tesseract OCR engine** will be used. Tesseract will return results as plain text, which will be overlaid on the original document.



Properties
UiPath.PDF.Activities.ReadPDFWithOCR

Common	
DisplayName	Read PDF With OCR
File	
FileName	The full path of
Password	The password o
Input	
DegreeOfParallelism	-1
ImageDpi	150
Range	"All"
Misc	
Private	☐
Output	
Text	The extracted st

However, when the Tesseract OCR is clicked, a properties panel appears on the right side where some options can be changed., under the Output section in the Text field, **a new variable should be added**, which will have the text, read from the PDF, saved as a string value (Figure).



Display Name	Tesseract OCR
Image	Image
Private	☐
AllowedCharacters	The OCR engin
DeniedCharacters	The OCR engin
ExtractWords	☐
Invert	If this check
Language	The language u
Profile	Choose a pr
Scale	1.99
Result	The extracted w
Text	Set Var: outputText

Read PDF Text

https://docs.uipath.com/activities/docs/read-pdf-files

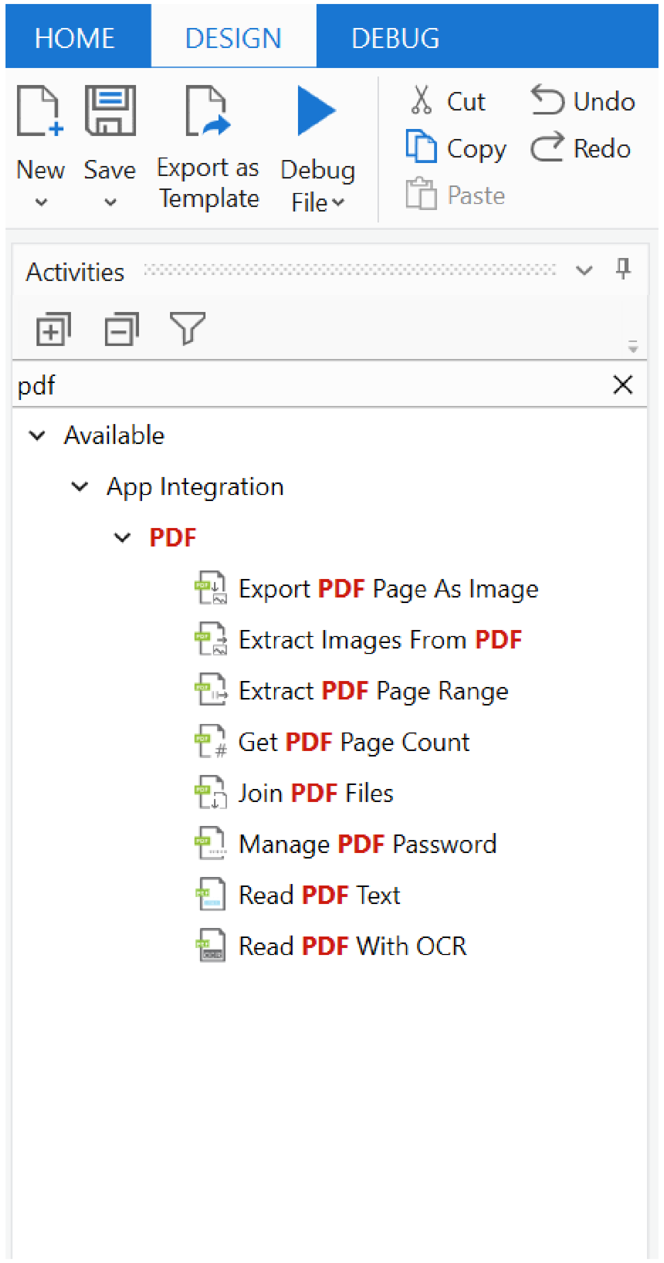
UiPath.PDF.Activities.ReadPDFText

- **Reads all characters from a specified PDF file and stores them in a string variable.**

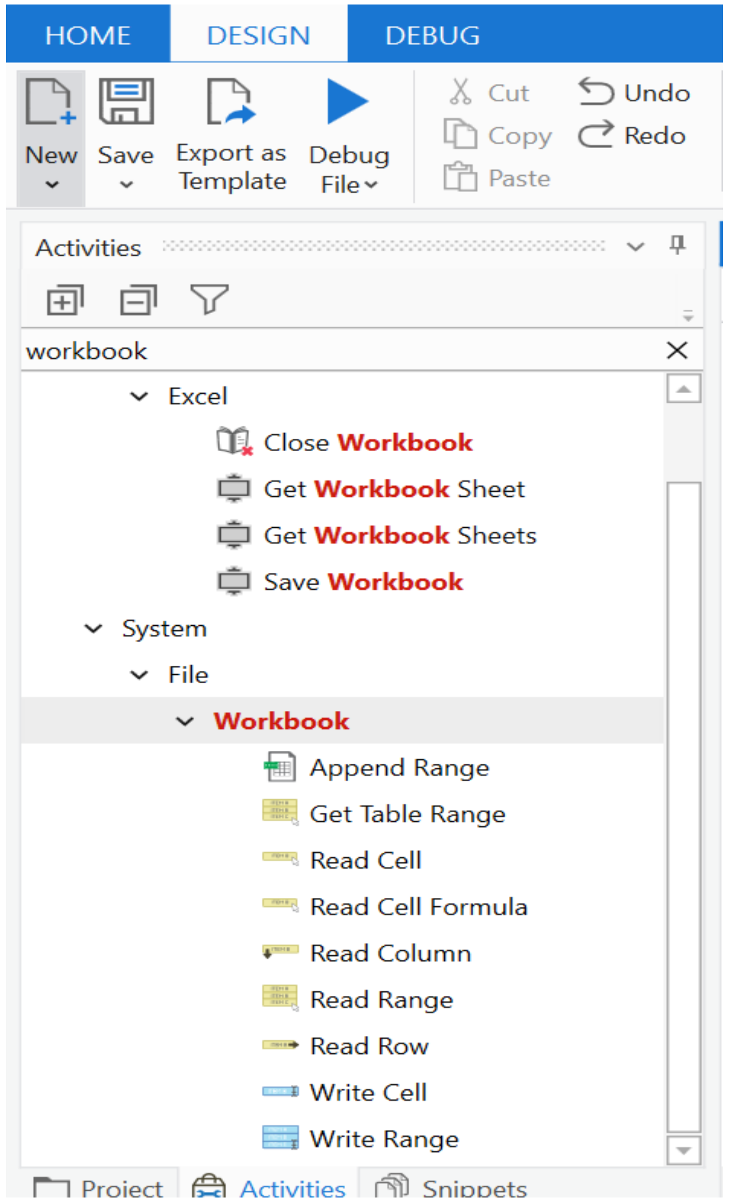
Read PDF With OCR

UiPath.PDF.Activities.ReadPDFWithOCR

- **Reads all characters from a specified PDF file and stores it in a string variable by using OCR technology.**

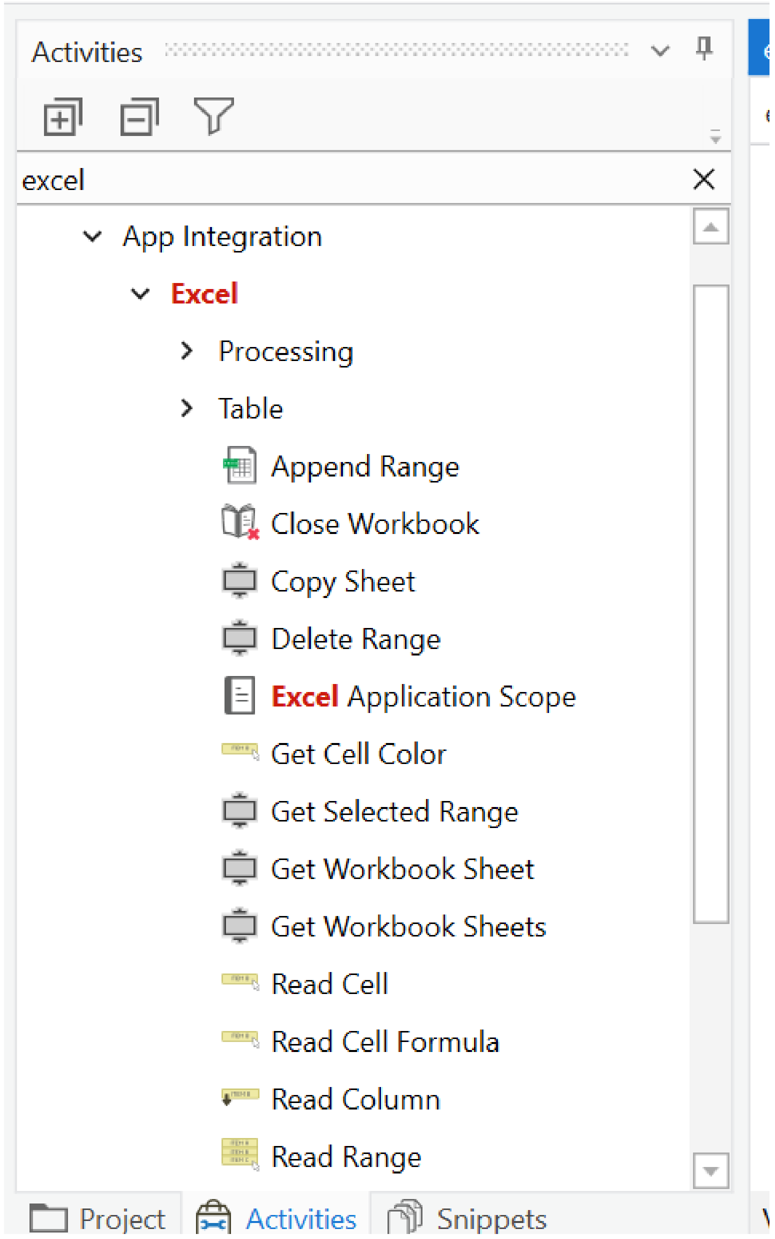


Workbook and Excel automation (read/write).



workbook

- Excel
  - Close Workbook
  - Get Workbook Sheet
  - Get Workbook Sheets
  - Save Workbook
- System
  - File
    - Workbook
      - Append Range
      - Get Table Range
      - Read Cell
      - Read Cell Formula
      - Read Column
      - Read Range
      - Read Row
      - Write Cell
      - Write Range



excel

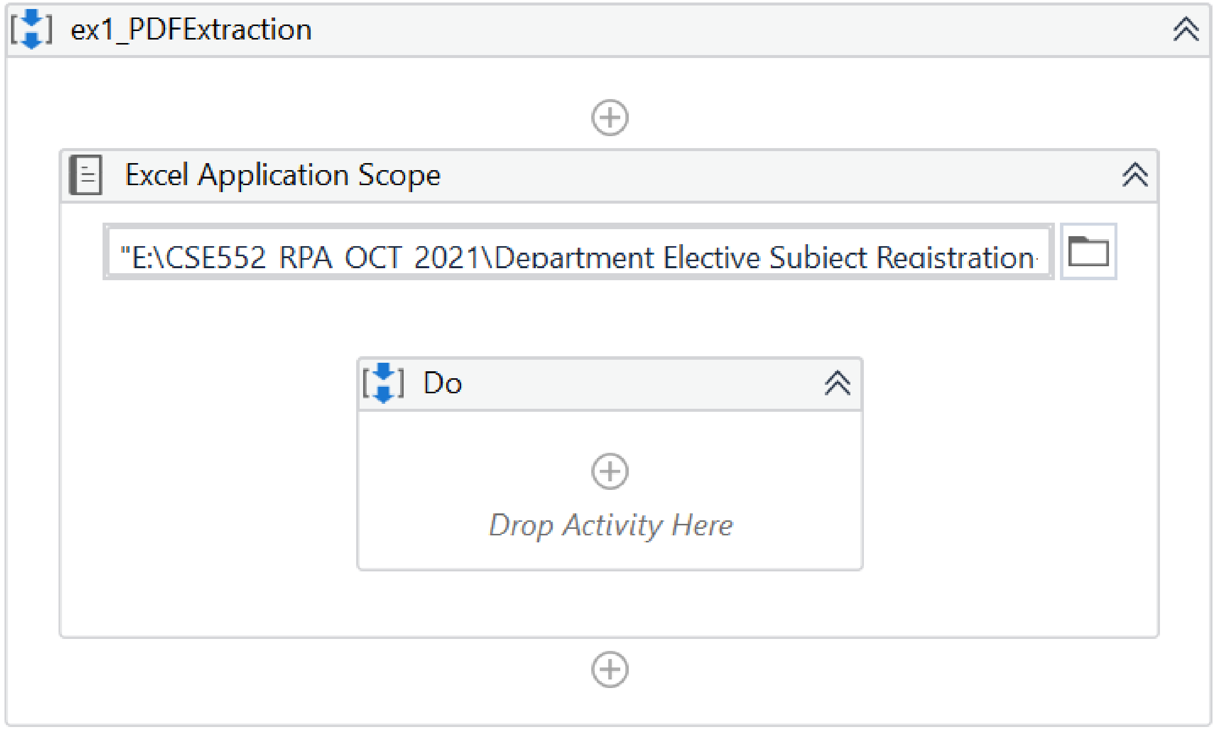
- App Integration
  - Excel
    - Processing
    - Table
    - Append Range
    - Close Workbook
    - Copy Sheet
    - Delete Range
    - Excel Application Scope
    - Get Cell Color
    - Get Selected Range
    - Get Workbook Sheet
    - Get Workbook Sheets
    - Read Cell
    - Read Cell Formula
    - Read Column
    - Read Range

Excel Application Scope

UiPath.Excel.Activities.ExcelApplicationScope

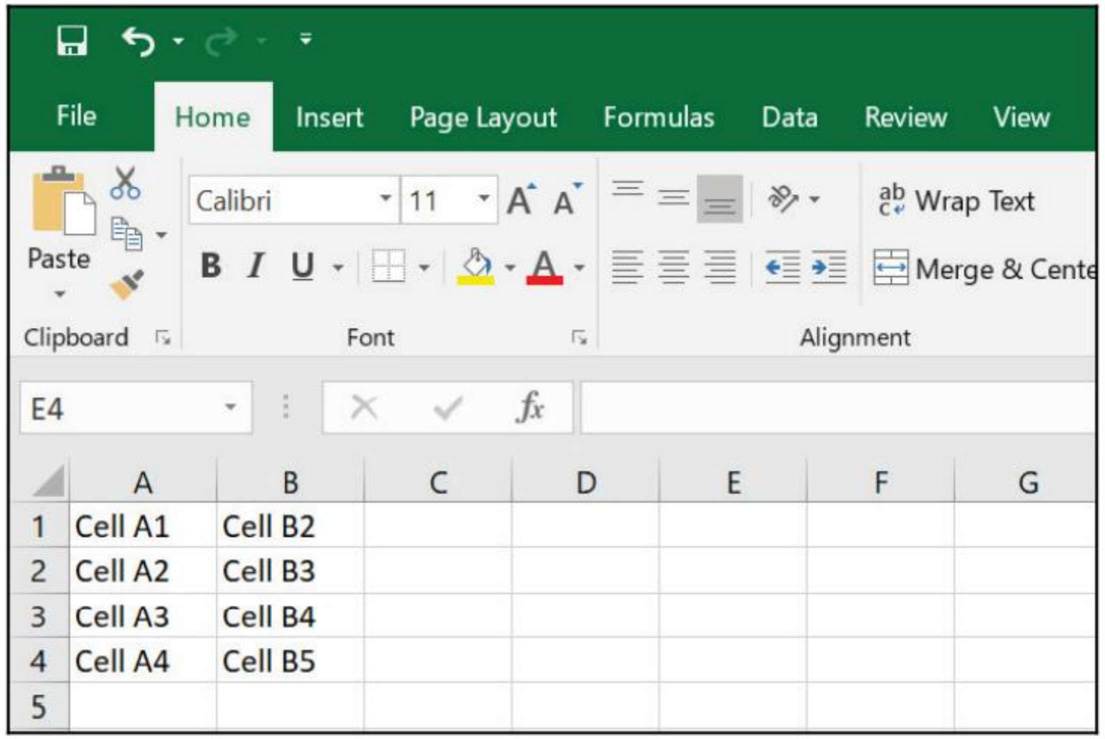
Opens an Excel workbook and provides a scope for Excel Activities.

- When the execution of this activity ends, the specified workbook and the Excel application are closed.
- If a WorkbookApplication variable is provided in the Output > Workbook property field, the spreadsheet is not closed after the activity ends. If the specified file does not exist, a new Excel file is created.
- This activity can only be used if the Microsoft Excel application is installed on your machine.

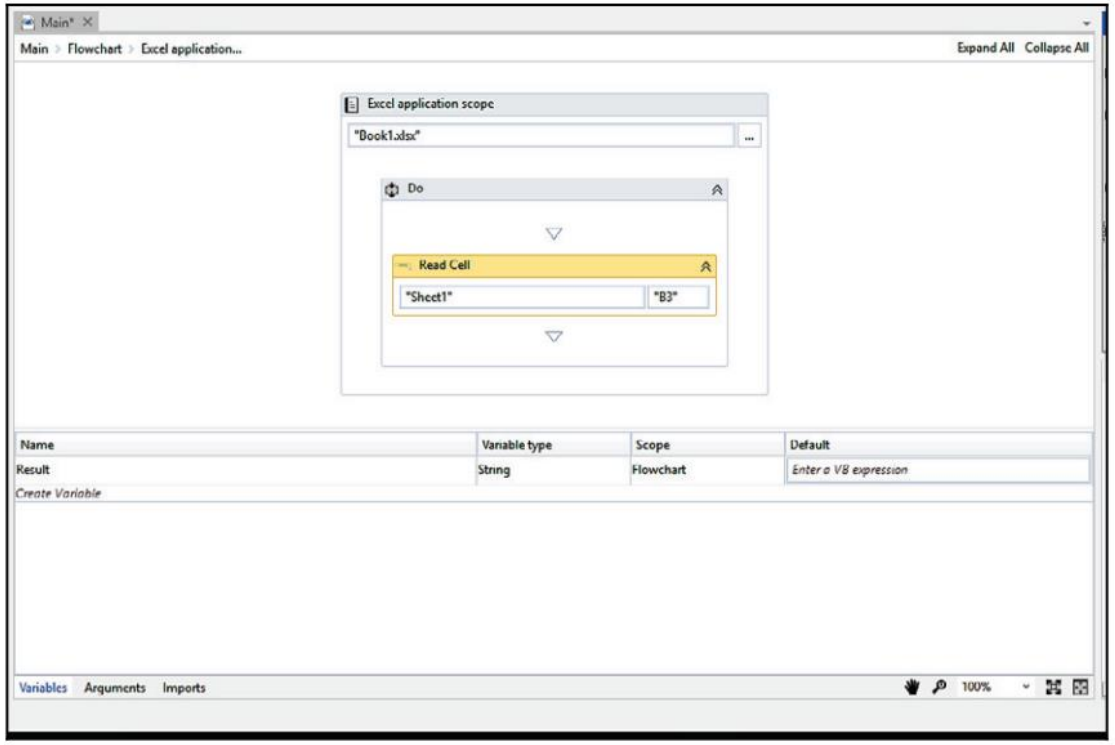

ex1_PDFExtraction
Excel Application Scope
"E:\CSE552_RPA_OCT_2021\Department Elective Subject Registration
Do
Drop Activity Here

File operation with step-by-step example

Read cell : This is used to read the value of a cell from an Excel file. We have a sample Excel file that we will use in this example:



	A	B	C	D	E	F	G
1	Cell A1	Cell B2					
2	Cell A2	Cell B3					
3	Cell A3	Cell B4					
4	Cell A4	Cell B5					
5							



Excel application scope

"Book1.xlsx"

Read Cell

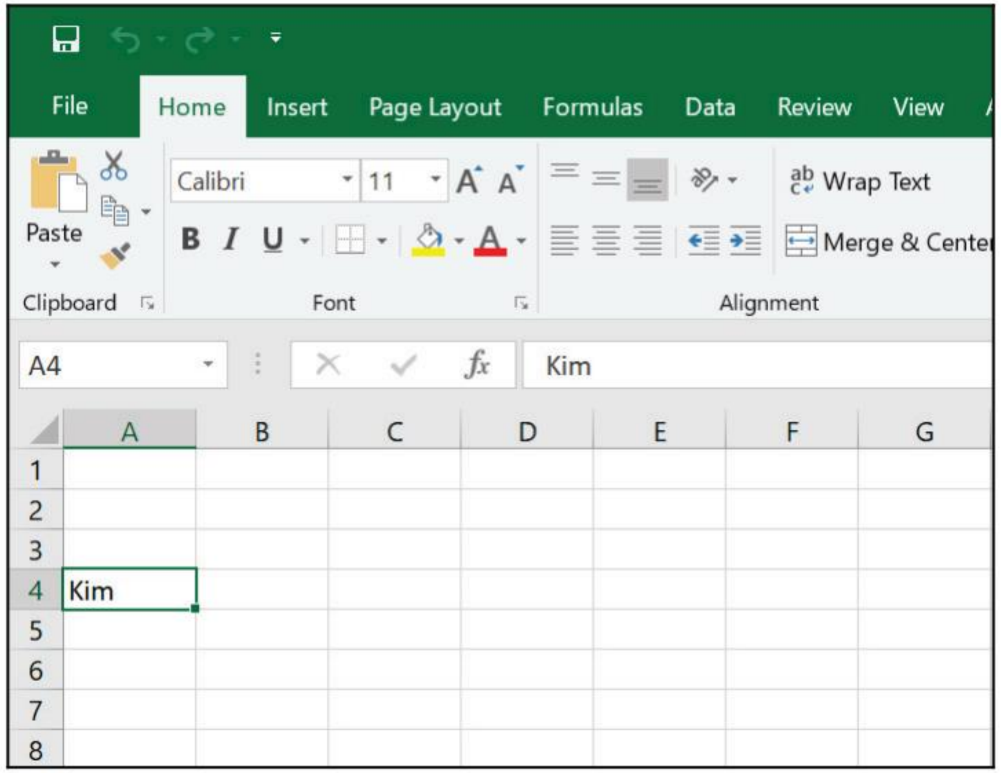
"Sheet1" "B3"

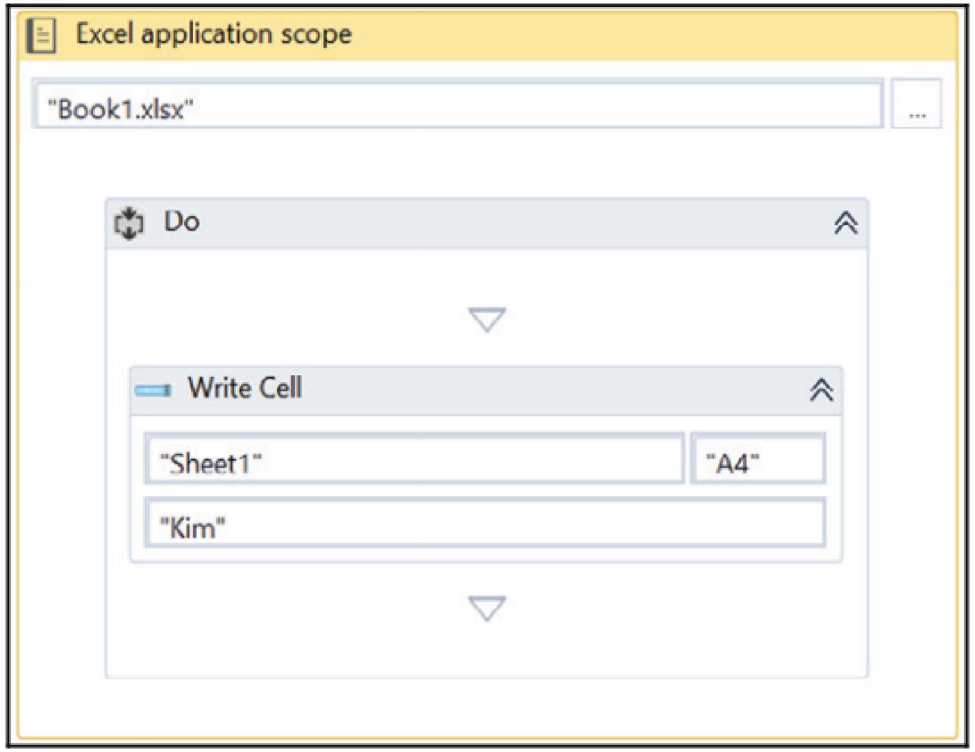
Name	Variable type	Scope	Default
Result	String	Flowchart	Enter a VB expression

Create Variable

Write cell : This activity is used to write a value in a cell of an Excel file:

1. Drag and drop a Flowchart activity on the main Designer panel. Also, drag and drop an Excel application scope inside the Flowchart activity. Connect it to the Start node.
2. Drag and drop a Write Cell activity inside the Excel application scope. Specify the cell value in which we want to write in the Range pr





"Book1.xlsx"

"Sheet1" "A4"

"Kim"

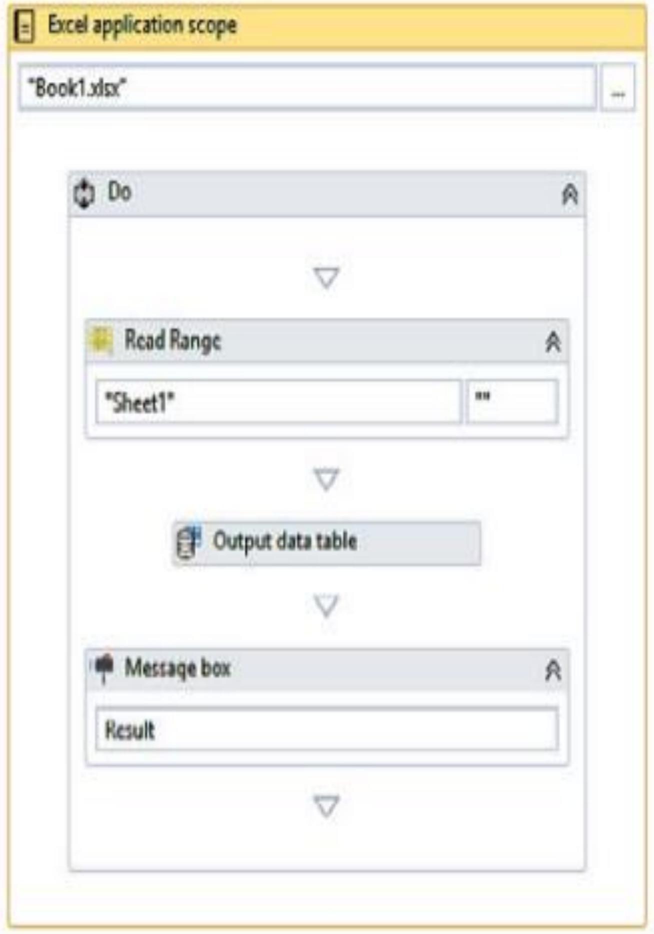
Read range : This is used to read the value up to the specified range. If the range parameter is not specified, it will read the entire Excel file:

1. Drag and drop a Flowchart activity on the main Designer panel. Also, drag and drop an Excel application scope inside the Flowchart activity. Connect it to the Start node.

2. Drag and drop a **Read Range activity** inside the Excel application scope activity. The Read Range activity produces a data table. We have to receive this data table in order to consume it. We need to create a data table variable and specify it in the Output property of the Read Range activity.

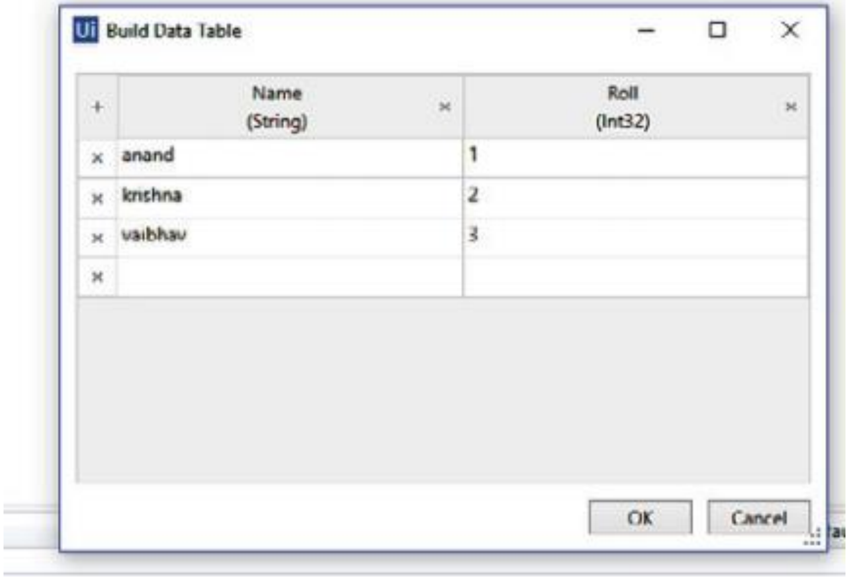
3. Drag and drop an **Output Data Table activity** inside the Excel application scope activity. Now, we have to specify two properties of the Output Data Table activity: Data Table property and text property. The Data Table property of the Output Data Table activity is used to convert the data table into a string format. The text property is used to supply its value in a string format. We have to receive this value in order to consume it. For this, let us create a variable of type string. Give it a meaningful name (in our case, it is Result):

4. Drag and drop a **Message box activity** inside the Excel application scope activity. Also, specify the string variable's name that we created earlier inside the Message box activity:

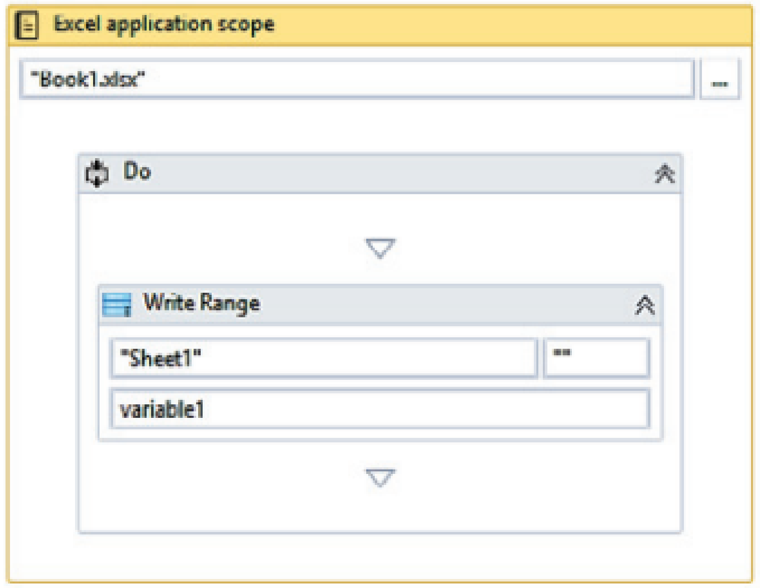


Write range : This is used to write a collection of rows into the Excel sheet. It writes to the Excel file in the form of a data table. Hence, we have to supply a data table:

1. Drag and drop a **Build data table activity** from the Activities panel. Double-click on this activity. A window will pop up. You will notice that two columns have been generated automatically. Delete these two columns. Add your column by clicking on the + icon and specify the column name. You can also select your preferred data type. You are free to add any number of columns:
2. In this project, we are **adding two columns**. The procedure for adding the second column is almost the same. You just have to specify a name and its preferred data type. We have added one more column (Roll) and set the data type to Int32 for the data table. We have also initialized this data table by providing some values in its rows.
3. Drag and drop an **Excel application scope** inside the main Designer panel. You can either specify the Excel sheet path or manually select it. Connect this activity to the Build Data Table activity. Inside the Excel application scope activity, just drag and drop the Write Range activity:
4. Specify the data table variable name that we created earlier and set it as a Data table property inside the Write Range activity. We can also specify the range. In this case, we have assigned it as an empty string:



Name (String)	Roll (Int32)
anand	1
krishna	2
vaibhav	3



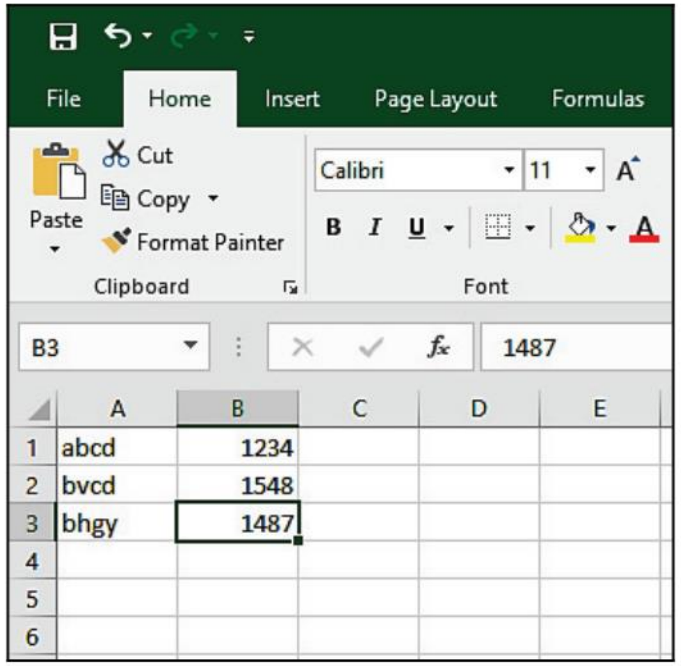
Append range : This is used to add more data into an existing Excel file. The data will be appended to the end.

1. Drag and drop the Flowchart activity on the main Designer window. Also, drag and drop the **Excel application scope** inside the Flowchart activity. Connect it to the Start node.

   - The Append Range activity requires a data table. In this program, we are going to use another sample Excel file, which has some raw data. Then, we will read this Excel file and append the data to another Excel file.

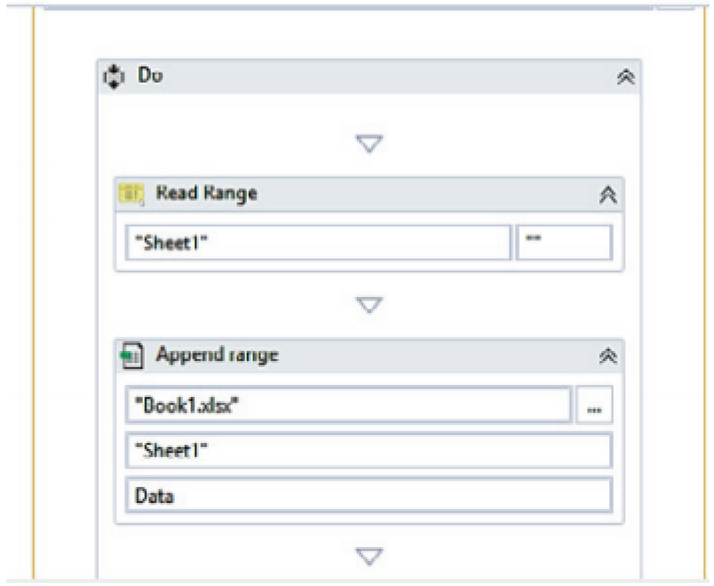
2. Drag and drop the Read Range activity inside the Excel application scope activity. The Read Range activity produces a data table. We have to receive this data table in order to consume it. Create a data table variable and specify it in the Output property of the **Read Range activity:**

3. Drag and drop the Append Range activity inside the Excel application scope activity. Specify the Excel file path in the **Append Range activity** (in which we want to append the data). Also, specify the data table (which is generated by the Read Range activity):



B3 1487

	A	B	C	D	E
1	abcd	1234			
2	bvcd	1548			
3	bhgy	1487			
4					
5					
6					



Read Range: "Sheet1" ""

Append range: "Book1.xlsx" "Sheet1" Data

EMAIL AUTOMATION:

- **Incoming Email automation**
- **Sending Email automation**

Introduction to Email Automation

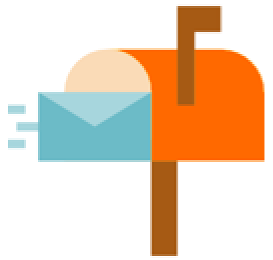
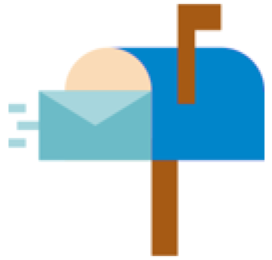
To this day, many business processes are triggered by email or generate email output.

Estimates show that an average person spends between 5 and 11 hours a week checking, reading and responding to emails.

What can we automate?

§ Generating and sending automated messages
§ Retrieving messages and extracting data
§ Managing messages
§ Saving attachments
§ Saving messages

Key Concepts of Email Automation

				
Email Client	**SMTP**	**POP3**	**IMAP**	**Exchange**
⇧ ⇩	⇧	⇩	⇩	⇧ ⇩

Email Client:

- An email client, (MUA), is a desktop or **browser-based application** used to access and manage a user's email.
- For example, Microsoft Outlook is an email client. Email clients are used to send and retrieve emails.

SMTP:

- Simple Mail Transfer Protocol (SMTP) is a basic protocol used **only for sending messages**.

POP3:

- Post Office Protocol (POP3) is an old and almost obsolete **protocol for reading messages**, but most email servers support it.

IMAP:

- Internet Message Access Protocol (IMAP) is only used for receiving messages but offers **features to mark messages as read or move them between folders.**

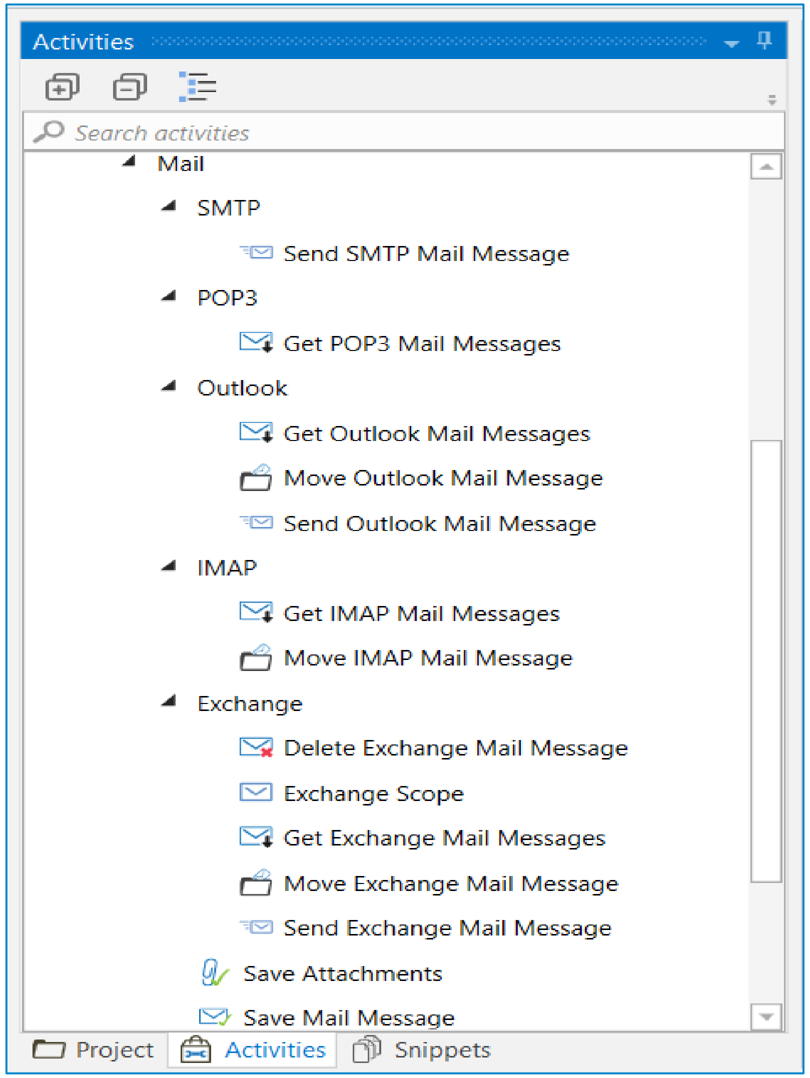
Microsoft Exchange:

- Exchange is Microsoft’s enterprise email solution that UiPath integrates perfectly.

UiPath Studio Activities

UiPath Studio includes activities for **sending, retrieving and organizing messages**. The available categories are:

- **Generating and sending automated messages:**
  - **SMTP, Outlook and Exchange**
- **Retrieving messages and extracting data:**
  - **POP3, Outlook, IMAP and Exchange**
- **Managing Messages:**
  - **Outlook, IMAP, Exchange**
- **Saving attachments and emails:**
  - **Generic Email Activities**


Activities

Search activities

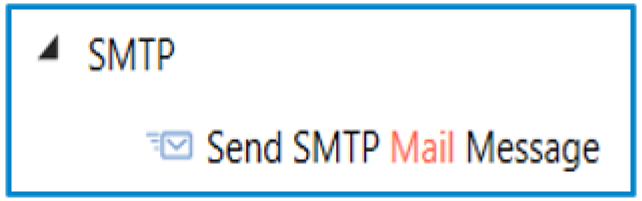
- Mail
  - SMTP
    - Send SMTP Mail Message
  - POP3
    - Get POP3 Mail Messages
  - Outlook
    - Get Outlook Mail Messages
    - Move Outlook Mail Message
    - Send Outlook Mail Message
  - IMAP
    - Get IMAP Mail Messages
    - Move IMAP Mail Message
  - Exchange
    - Delete Exchange Mail Message
    - Exchange Scope
    - Get Exchange Mail Messages
    - Move Exchange Mail Message
    - Send Exchange Mail Message
  - Save Attachments
  - Save Mail Message

Project | Activities | Snippets

Email Protocols

Depending on the scope of automation, we will use different sets of activities and protocols. For using email as input, depending on the client's setup, we have the options to use: **POP3, IMAP, Outlook or Exchange.** For using email as output: **SMTP, Outlook, or Exchange.**

SMTP

Definition	SMTP stands for Simple Mail Transfer Protocol. Mail servers and other mail transfer agents use the SMTP protocol to send mail messages on TCP port 25.
Scope	Email as output. SMTP can be used with Gmail to send emails.
UiPath Studio Activities	

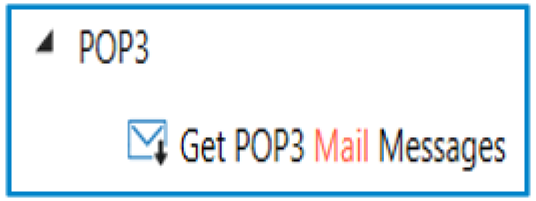
Scope: SMTP is only used to send emails.

UiPath Studio Activities: Depending on how an organization has their email processing set up, there is a chance that you use the SMTP activity to send an email.

Minimum Requirements:

- To
- Subject
- Body
- Port
- Server
- Email
- Password

Email Protocols:POP3

Definition	POP3 stands for Post Office Protocol and is an old and almost obsolete protocol for reading messages but most email servers support it.
Scope	Input via email. POP3 can be also used to retrieve email from Gmail.
UiPath Studio Activities	

Scope: POP3 is only used to retrieve emails.

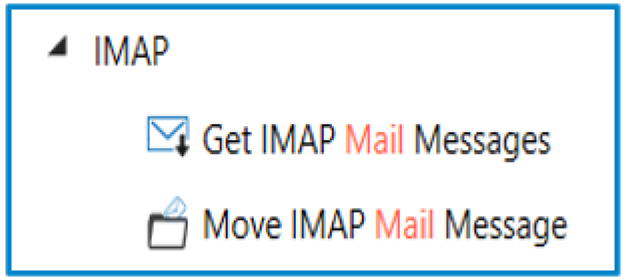
A POP3 server listens on well-known port number 110 for service requests.

UiPath Studio Activities: As it is commonly spread, this activity allows you to easily retrieve emails. The most commonly used variable type to store emails in is List of mail messages.

Minimum Requirements:

- Port
- Server
- Email
- Password
- Top: The number of messages to be retrieved starting from the top of the list.
- Output > Messages: The name of the List of Mail Messages variable used to store the messages.

Email Protocols: IMAP

Definition	IMAP or Internet Message Access Protocol is only used for receiving messages This offers some useful features to mark messages as read or move them between folders.
Scope	Input via email. IMAP can be also used to retrieve email from Gmail.
UiPath Studio Activities	 IMAP Get IMAP Mail Messages Move IMAP Mail Message

Scope: IMAP is only used to retrieve emails.

An IMAP server typically listens on port number 143. IMAP over SSL (IMAPS) is assigned the port number 993.

UiPath Studio Activities: The practical use is to retrieve emails and also the possibility to move the email to different inbox folders.

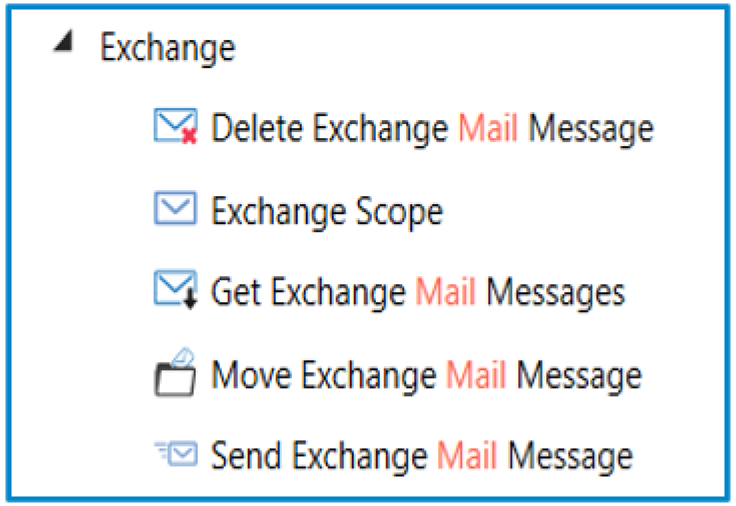
Minimum Requirements:

- Port
- Server
- Email
- Password
- Top: (The number of messages to be retrieved starting from the top of the list)
- Output > Messages: The List of Mail Messages variable used to store the messages.

Email Protocols: Microsoft Exchange

UiPath Studio Activities: Exchange automation requires less setup than SMTP, POP3 and IMAP as it can use the default and auto-discover settings in UiPath Studio.

Method: Create and define an Exchange Scope activity. Include other email activities in the Exchange Scope activity container.

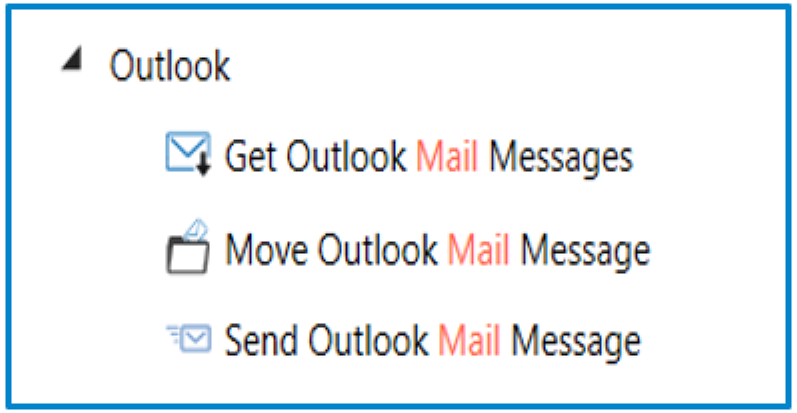
Definition	Exchange is Microsoft's enterprise email solution that UiPath integrates perfectly, which help to sends and received emails.
Scope	Sends and receive emails
UiPath Studio Activities	

Exchange
Delete Exchange Mail Message
Exchange Scope
Get Exchange Mail Messages
Move Exchange Mail Message
Send Exchange Mail Message

Requirements Exchange Scope:

- **Server:** The email server host that is to be used.
- **ExchangeVersion:** Specifies the lowest version of the Exchange server that is used. The options displayed in this field range from the 2007 to the 2013 version. **EmailAutodiscover:** Searches automatically for an Exchange server by using an email address from that server. This works only if the Exchange server has Autodiscover enabled.
- **ExistingExchangeService:** Allows connecting through a pre-existing ExchangeServer object from another Exchange Scope. This field supports only ExchangeServer objects.
- **User:** The username of the Exchange account to be used.
- **Password:** The password of the Exchange account to be used.
- **Domain:** The Active Directory domain to connect to

Email Protocols: Microsoft Outlook

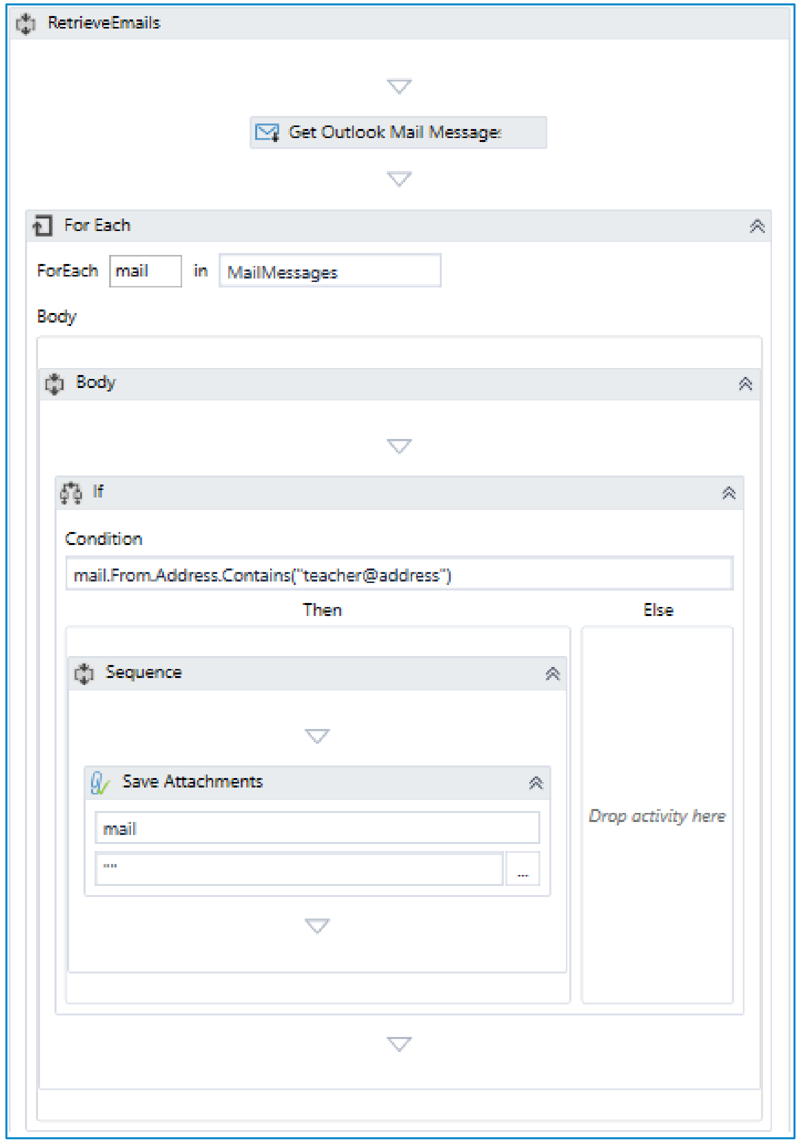
Definition	Microsoft Outlook is a personal information manager from Microsoft. It is mainly used as an email client. This functionality is what we are focusing on in this course.
Scope	Input via email and email as output.
UiPath Studio Activities	

Outlook
Get Outlook Mail Messages
Move Outlook Mail Message
Send Outlook Mail Message

Requirements Get Outlook Mail Messages:

- **Account:** Insert your email address.
- **MailFolder:** Generally you will grab your emails from "Inbox", but you could also place there any folder from your Outlook application.
- **MailMessages:** The easiest way is to right click in the blank space (or Ctrl+K) to create the output variable which is a list of Mail Messages.
- **MarkAsRead:** If you grab some emails that were not read, these will be checked as read in Outlook.
- **OnlyUnreadedMessages:** Check this option to get only the emails that were not read.
- Top: Indicate the number of emails you want to extract.

Retrieve and Save Emails from Outlook



```
mail.From.Address.Contains("teacher@address")
```

Main steps of the solution

1. **Get Outlook Mail Messages -** Retrieve a set amount of emails from the Outlook inbox.
2. **For each -** Type of argument used is System.Net.Mail.MailMessage
3. **If -** The condition here is used to see if one of the emails in the list retrieved is from the teacher or not.
4. Inside the 'If' we have the **Save Attachments** activity, which saves the attachments from the email sent from the teacher. The attachments are saved in the local folder of the project.

Retrieve and Save Emails from Outlook

1. Get Outlook Mail Message

- Saves a set number of emails from Outlook in a variable.
- Input > Mail folder: The email folder from which the messages are to be retrieved. For this example, use “Inbox”.
- Options > Top: The number of messages to be retrieved starting from the top of the list.
- Output > Messages: The variable in which you want to store the retrieved emails. The used variable type is List <Email Messages>

2. For Each:

- Performs an operation for each object stored in the list variable
- Misc > Type Argument: System.Net.Mail.MailMessage
- Misc > Values: Use the output variable from the previous activity.

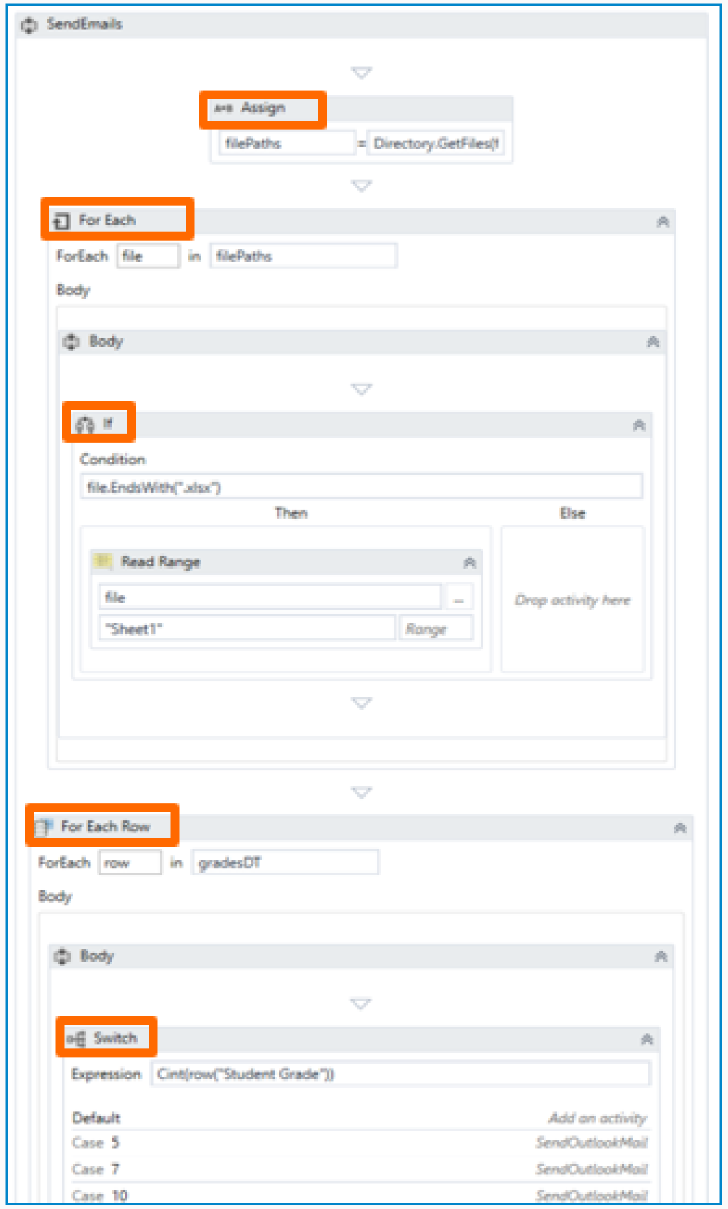
3. If:

- Checks if the email “from” address contains teacher@address. If yes, it saves the attachment.
- Condition > mail.From.Address.Contains(“teacher@address”)
- The condition must be changed a little as it is dependent on a real email address, as a placeholder it's the used the "teacher@address".

4. Save Attachments:

- Saves the attachments that meet the “If” criterion to the defined path.
- Input > Folder Path: The full path of the folder where the attachments are to be saved.
- Message > The MailMessage object whose attachments are to be saved.

Sending Emails from Outlook



SendEmails

Assign

filePaths = Directory.GetFiles(

For Each

ForEach file in filePaths

Body

Body

If

Condition

file.EndsWith(".xlsx")

Then

Else

Read Range

file

"Sheet1" Range

Drop activity here

For Each Row

ForEach row in gradesDT

Body

Body

Switch

Expression CInt(row("Student Grade"))

Default Add an activity

Case 5 SendOutlookMail

Case 7 SendOutlookMail

Case 10 SendOutlookMail

Main steps of the solution

1. **Assign Activity -** This is used to retrieve all the files inside the folder project.
2. **For Each -** Simple string argument.
3. **If -** If the file ends with .xlsx we will read the whole file.
4. **For Each Row -** This will iterate through the rows of the previously read excel file.
5. **Switch -** Depending on the grade the student received, he will also receive a personalized message inside the email.

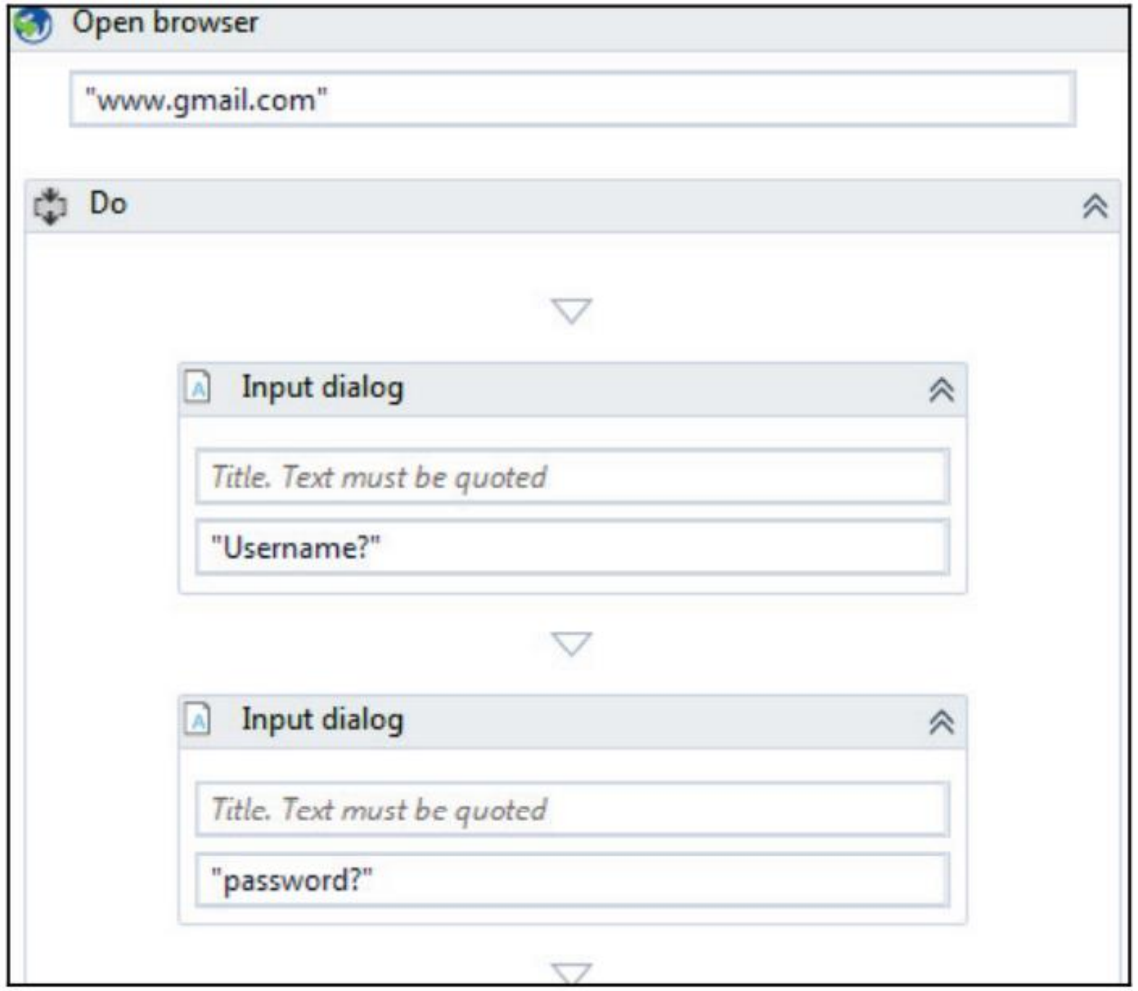
An example of Monitoring Email

1. **Open the browser** and browse to www.gmail.com: To do this, drag and drop the **Open browser activity**. In the required field for the address, enter www.gmail.com .

2. **Getting username and password:** After typing in the address, we have to ask the user for a username and password. For this, we will use the **Input dialog activity** as shown in the following screenshot.

- We have dragged and dropped two Input dialog activities to ask the user for a username and password respectively. Until the user types in each dialog and presses okay, the Robot will not work:

- Once the user types in the username and password, we save these details into **two variables: User and Password.** You can convert their values into a variable by going to the **Input dialog property in the Properties panel.**

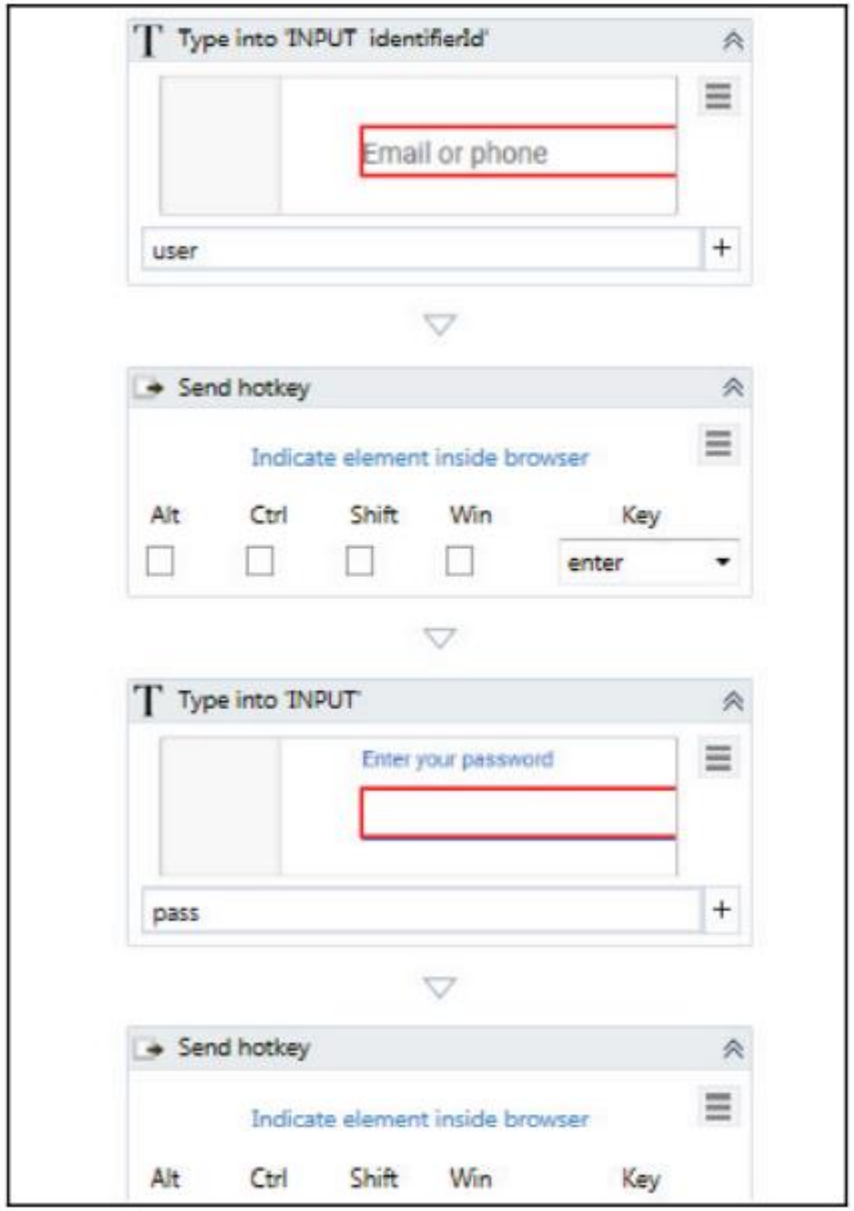

Open browser
"www.gmail.com"
Do
Input dialog
Title. Text must be quoted
"Username?"
Input dialog
Title. Text must be quoted
"password?"

3. **Entering a username and password:** We shall use the Type into activity to enter a username and password by indicating the respective fields for typing in the username and password.

- Once the user enters the username and password, he needs to login which he can either do by clicking on the login button or by pressing the Enter key on the keyboard.

- We will use the **Send hotkey activity** to send the Enter key (as shown in the following screenshot). By doing so, the login button is clicked:

4. **Trigger the send email event with a Hotkey trigger:** Our next step is to trigger the send mail event. Here, pressing the Enter key will be the trigger. On pressing it, the Robot performs the rest of the send email task. For this, we will use the Hotkey trigger activity. We first have to drag and drop the Monitor events activity as trigger activities only work under it:

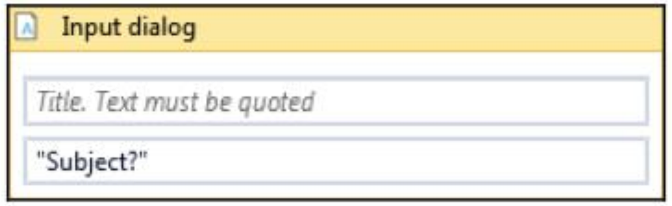

Type into 'INPUT identifierId'
Email or phone
user
Send hotkey
Indicate element inside browser
Alt Ctrl Shift Win Key
enter
Type into 'INPUT'
Enter your password
pass
Send hotkey
Indicate element inside browser
Alt Ctrl Shift Win Key

5. **Ask the user for the email ID of the recipient, the subject of the email, and its body:** Our next step is to ask the user for details. We will use three Input dialogs, one for the email ID, one for the subject, and one for the content.

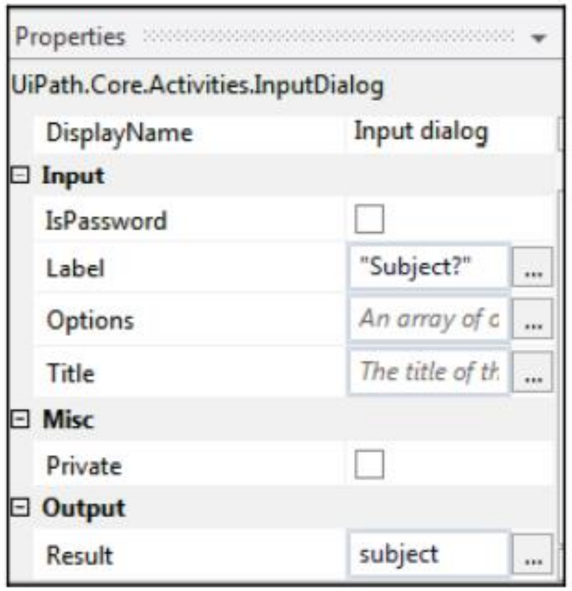
6. Type in the details: Now that we have all the details that are required for sending the mail, our next step will be to type into the required fields for sending the email. We will use the Type Into activity for this step:

7. Click on Send and confirm if successfully sent: Our final step is to click on the Send button so that the mail is sent and the process is completed. In order to click on the Send button, we will use the Click activity and indicate the Send button. Doing so enables the Robot to easily recognize where to click:

In the second **Input dialog**, we will ask the user to input the subject for the email:

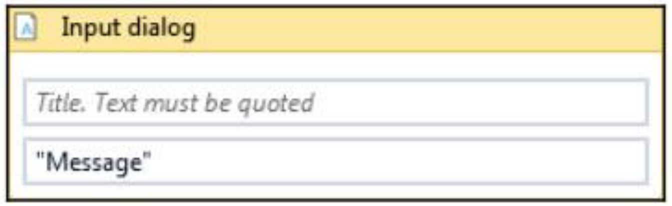

Input dialog
Title. Text must be quoted
"Subject?"

The output, that is, the response entered by the user, is saved as a new variable called `Subject?` as shown in the following screenshot:


Properties
UiPath.Core.Activities.InputDialog

DisplayName	Input dialog
Input	
IsPassword	☐
Label	"Subject?"
Options	An array of c
Title	The title of th
Misc	
Private	☐
Output	
Result	subject

In the third input dialog, the user has to input the message/mail he or she wants to send:


Input dialog
Title. Text must be quoted
"Message"