UI Path

**Installation:**

<https://www.uipath.com/> 🡪 Try UiPath Free 🡪 Sign in with Google 🡪 Make a note of the project key highlighted in the URL 🡪 Download UiPath Studio

Graphical user interface, text, application, email, Teams

Description automatically generated Graphical user interface, application, website

Description automatically generated

Skip the sign in part and click “More Options” 🡪 Standalone Options 🡪 Community offline

A picture containing graphical user interface

Description automatically generated Graphical user interface, application

Description automatically generated

**Home screen of UiPath Studio:**

Graphical user interface, application

Description automatically generated

Process 🡪 Project / Workspace which stores all files

Dependencies 🡪Libraries to support Automation

Xaml 🡪 Design automation flow

Project.json 🡪 Project Information and Metadata to publish project

Activities 🡪 UiPath Core packages for design workflow

Properties 🡪 Properties list for selected Activity

Outline 🡪 Output console to log workflow results

Ribbon Menu 🡪 UiPath Tool features

**Recording:**

Click Recording (Basic Option) and you will see a pop up

**Open Application** 🡪 open any window application

Type Into 🡪 type text into the opened application (alphabets and numeric)

Hot keys 🡪 when you want to perform activity using Ctrl, shift, alt, win keys

Click 🡪 when you want to perform any click operation

Open Browser 🡪 Open web browser

Get Text 🡪 Get text from the browser

Messagebox 🡪 to show a text message in the message box

**Data Scrapping:**

Get Text 🡪 Grab text from specific location.

Get OCR Text 🡪 Grab text from images

Get Full Text 🡪 use Screen scrapping = Full Text (it captured all the lines irrespective of the application(notepad, word) size)

Get Visible Text 🡪 use Screen scrapping = Native (assuming out of 10 lines in notepad, only 5 are visible since the notepad is not maximized)

**How to create a workflow:**

Right click on the project 🡪 Add 🡪 Flowchart 🡪 Give a Name 🡪 Click Create

Activities tab 🡪 Sequence 🡪 Open Browser 🡪 Give the URL in quotes 🡪 select the browser under properties for Open Browser activity 🡪 Get Text (Activity under UI automation / control) 🡪 Click the text in the browser which you want to grab (before that click the “indicate on the screen”) 🡪 the grabbed text will get stored in the output property of “get text” activity” 🡪 to fetch that text we need to create a local variable and assign that value to the output field 🡪 crate a variable under “variables” tab below flowchart named “webText” and the type is string” 🡪 type the string name into the output field of get text property 🡪 now to see the output in a msgbox 🡪 Messagebox(from activity tab) 🡪 pass the string variable name in the textbox.

**Instead of message box, we can use write line activity also. But this time the output will be printed at the console area or the same can be called as output in UiPath terminology**

**Note**:

1. To grab text from a browser, first we need to give UiPath knowledge to the browser. In order to do that, we need to go to UiPath 🡪 Tools 🡪 Uipath Extensions 🡪 Click Chrome and Install. Similarly, we need to do for other browsers as well if required.
2. Also, we need to navigate to “Ui Explorer” on the top of the screen, and then in the new window 🡪 Ui Frameworks 🡪 UI Automation

**Screen scrapping:**

The icon available in the top of the window. It is used to capture all the text in an application. Automatically sequence will get created for this method and no need to manually create it. Here the first step is not open application but attach application since we are using an opened notepad.

**Note**: The same can be obtained using “Get Full text” activity

**How to Build Robot which suggests to start business for the given country or not:**

**Condition** 🡪 If country’s GDP > 10 trillion display not a good place to start business else display good place to start business

Create a new workflow under the process 🡪 Add a sequence to get the input (country name) from customer using “**Input Dialogue**” 🡪 Give the input dialogue a header and then the question text which should display on screen in the given text boxes respectively 🡪 Create a variable and assign it to the output field in “Input Dialogue” to take it to the next process 🡪 **“Open Browser”** activity 🡪 set the UI Automation in the Ribbon to “Ui Automation” 🡪 Browser name to chrome 🡪 type the URL to access 🡪 “**Type Into**” activity to pass the value captured on “Input Dialog” activity under “output” field with the variable name “Country” 🡪 pass the value as Country + “ GDP” 🡪 **“HotKey”** activity and just select “enter” in the drop down 🡪 **“get text”** activity to capture the text 🡪 it will capture all the text including the text next to the GDP value for e.g., 20.19 Trillion USD” 🡪 but we are interested only in the number 🡪 also we need to set the regular expression else it will keep look for “20.10 Trillion USD” 🡪 to set the regular expression go to hamburger menu on the Get text activity and click “Edit Selector” 🡪 under the “Edit Selector” section set the name from “19 Trillion USD” to “\* Trillion USD” (this can be achieved by using the “Repair” option on the same screen and recapture the element”) 🡪 use “Get OCR Text” activity to capture only the number which is in string format and store it in a GDPActualNumber 🡪 Now we need to add a decision based on the output 🡪 **“Flow** **Decision” activity can’t be added inside a sequence or along with a sequence 🡪** add a **“Flow Decision”**activity and add a condition under property tab 🡪 we need to give the condition as “GDPActualNumber>10” 🡪 it won’t work due to 2 conditions 🡪 1. Change the scope of the variable to the workflow level instead of “DO” sequence level and 2. Convert the variable type form String into double by rewriting the condition to **“convert.ToDouble(GDPActualValue)>10”** 🡪 now based on the result set the true and false node with message as desired. For e.g., to “True” node connect a “message box” activity with a message “Good place to start business”

**Read and write from Excel:**

1. Read Cell 🡪 Use the activity and mention the path in quotes, sheet name and cell name like A1, B4 etc. Assign a variable in the output field under property tab (using Ctrl + K). Use a “**Message Box / writeline**” activity to print the output
2. Write Cell 🡪 Similar to read cell, use the activity named “Write Cell” and simply pass the “Cell Name”, “Sheet Name” and the value to be written in the cell within quotes.
3. Read Range with Specific Index
4. Read Range on Full sheet
5. Iterating over Data Table using for Each Row

To work with excel we need to first use “**Excel application scope**” activity in the sequence similarly how we use “**Open Browser**” to open a browser etc., Once it is dragged into the sequence container all actions need to be performed must be made inside the “excel application scope” container which is again inside the sequence.

**Read Range with Specific Index:**

Use “**Read Range**” activity and specify the sheet name and the range of cells (let us say A2 to C3). The output of this activity will be a data table and not a string. First create a variable and assign to data table. Now create one more activity named “**Output Data Table**” which takes the output of “**Read Range**” data table variable as input. Create a new variable and assign it to the output field under the property of **“Output Data Table”** activity. Now if we use the “**write Line**” activity with the variable name created in “Output Data Table” then the values will get printed in the output area.

**Read Ranges with full sheet:**

This is very simple where we just remove the cell range mentioned in the “Read Range” activity.

**How to iterate the data tables and get the value:**

**Precondition** 🡪 Read the entire excel and have the result stored in Data table using “Read Range” activity

Get the data into a variable by creating a new variable and assign it to the data table (as discussed above). Now create a new activity named **“For each Row”,** mention the data table variable in the text box. Use “Write Line” activity and say row(A1).toString which will print the output of Column A1 alone instead of entire sheet content.

**How to read a data from an excel and do a math based on the given input and update the value of the cells:**

For the given student id, we need to apply discount of $300 and update the status to “Concession applied” and if there is not match then update the status as “NA”. Take the below as an example table

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Fees** | **Status** | **Name** |
| 123 | 500 |  | Ragavendran |
| 456 | 500 |  | Sharanya |
| 789 | 500 |  | Niharika |

1. We need to get the id from the end user. Hence use **“Input Dialog”** activity and name the header and statement for the end user. Create a string variable “**inputData**” and assign it to the output field under the property tab
2. Create a **“Excel application scope”** to deal with the excel. Give the path of the excel.
3. Create a **“Read Range”** activity and mention the sheet name but not the cell name, so that it will read all the data and stores in a data table. Create and assign a variable (named rrd) to the data table of type data type and set the scope to sequence level.
4. Create **“For Each”** activity and pass the data variable “**rrd**”.
5. To check whether it works fine till this line, use **“Write line”** activity and say row(“ID”).toString
6. Create **“If”** activity which will have “a condition”, “then” block and “else” block with it. Set the condition as row(“ID”).toString.equals(**inputData**)
7. **Inside “Then” block**, have the following things
8. Create “**Assign**” activity which will reset the value of an existing cell based on the given rules.
9. Write the following code in assign activity where it has 2 text boxes one is for the variable name and other to set the formula
10. Set the variable name text box to “updatedFees” (this need to be created as int variable) and formula text box to Cint(row(“Fees”))-300
11. Now to assign the updated fees based on the id, we need to find a logic because if we simply use write cell, then we need to hard code the cell value but to make it as not hard-coded, let us make use of the “Index” field available under property tab of “For Each” activity
12. Create a variable named “Counter” of type integer and set it to the field “Index” under “For Each” activity property tab
13. Index start from “0” and gets incremented every time a loop is completed. Let us say, we need to run 4 iterations and the counter for the first iteration is 0 and the last is “3”
14. Let us say I want the fees for the first iteration to be updated and the column is “B2” where “B” is the column name and “2” is the row number. So, for this instance the counter value will be “0” since it is the first iteration. Hence to get B2, I need to set the counter value = counter + 2
15. **Create one more “Assign” activity and set the counter = counter+2 and place it before if condition so that is applicable to both “Then” and “Else” block**
16. Since we have find the logic to write a general expression for cell name in “Write Cell” activity, we can set the sheet name and cell value =”B”+counter.toString and set the value = updatedFees.toString
17. Create one more “Write Cell” activity to set the status of the corresponding cell to “Concession Applied” by setting the sheet name and column name = “C”+Counter.toString and value = “Concession Applied”
18. **For Else block**,
19. using “Write Cell” activity, set the sheet name as “Sheet 1” and column name = “C”+counter.toString and value = “NA”

Now to print the final message to the end user, create a “message box” activity under “Then” block itself with the message “The concession is applied for” + row(Name).toString + “ and the updated amount is “ + updatedFees.toString.

**Note**: We can’t use row(“Fees”).toString since if we do it will still display the old value because during the initial step we read all the value and store them into data table but we made the changes in excel. So, when print row(“Fees”), it will print the value from the data table and not from the excel.

=====================\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*==========================\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*============

**Demo on Config Files and Dictionary Variables:**

Preconditions:

* Have a excel file with list of project info such as url, username, password
* Create a folder in project explorer and place the excel there

Create a flow chart type

Add 3 sequencies

**In first sequence:**

* Get an “Assign” activity and create a string variable which holds the path of the excel such as Path = “Data\Config.xlsx”
* Get an “Assign” activity and create a String variable which holds the sheet name of the above excel such as sheet=” Config”

**In 2nd sequence**

* Get a “Read Range” activity and in place of path put the variable “path” and in place of sheet name put the variable “sheet” and leave the range
* Assign the variable datatype to the output field in properties tab named “ConfigDT”
* Get a “For each row in DT” activity and in the body have an Assign activity
* Create a Dictionary variable which helps us to hold the data in key, value pair like HashMap in Java
* Create a variable named “Out\_config” under variable tab and set the type of variable to System.collections.Generic.Dictionary and set the key = String and value = object
* To use the dictionary variable, first step is to initialize it.
* To initialize the dictionary variable, use an “Assign” activity at the start of the sequence with variable name = config\_out and value = new Dictionary (Of String, Object)
* Now get back to body of “For each row” activity and in “assign” activity, set out\_config(row(“Name”).ToString) to row(“value”)
* Since our dictionary is of type <Sting, Object>, we have just converted key alone to string and not the value
* After for each, add a message box activity to display the dictionary variable using “out\_config(“url\_prod”).ToString

**In 3rd Sequence:**

* Get a open browser activity and in url instead of hardcoding use the dictionary variable and say out\_config(“url”).toString
* Get a “Kill Process” activity and set the Processname = “iexplore” for internet explorer if open browser is launching using IE

Now instead of having 2 different variables to store the different sheet name, let us use a single variable which is of type Sting[] and to create and declare the value

* Create a variable ConfigSheets of type String of Array
* In the first sequence, have a new assign activity and set the value to {“Sheet1”,”Sheet2”} for the variable ConfigSheets
* In the 2nd sequence, change the “read range” activity by adding a “for each” (under workflow/control) activity just before the read range
* For each = item and in = configsheets and body = pull other activity into it
* Change the sheet name to item.tostring in read range activity
* In the 3rd sequence, inside open browser activity, add a “Delay” activity and set the duration field under properties to “TimeSpan.FromSecond(cint(Out\_config(“Delay”)))
* Add “Log Message” activity and set the log level as well as message Out\_config(“LogMessage”).ToString

Dictionary can not store null or empty key. To check it, we can use the function String.IsNullOrWhiteSpace(StringValue).

To implement the check, add an if activity in body of sequence 2 just before adding the values to the out\_config dictionary variable

If condition =Not String.IsNullOrWhiteSpace(row(“Name”).ToString.Trim)

Add the assign activity to then box in If activity