RPA Design and Development

v4.0

UiPath

Lesson 09 | Ui Automation

Modern Ui Automation – Exam Topics

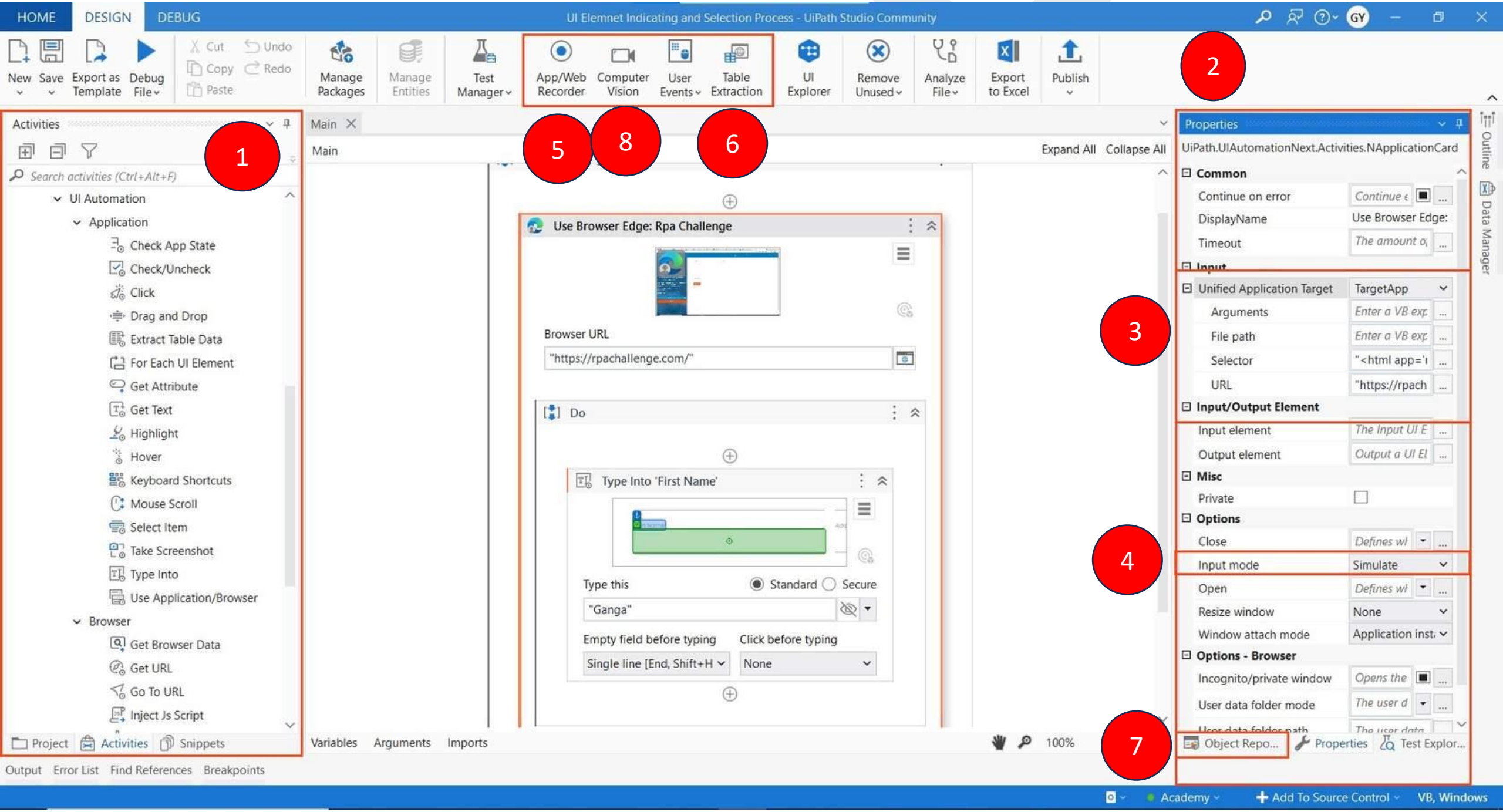
1. Explain how UI Automation works and switch between the modern and classic design experiences
2. Use the Modern Recorder to create UI Automation.
3. Use Modern UI Automation Input Activities and Input Methods.
4. Use Modern UI Automation Output Activities and Output Methods.
5. Use UI Synchronization with activities available in the Modern Design Experience.
6. Evaluate and configure static and dynamic Descriptors.

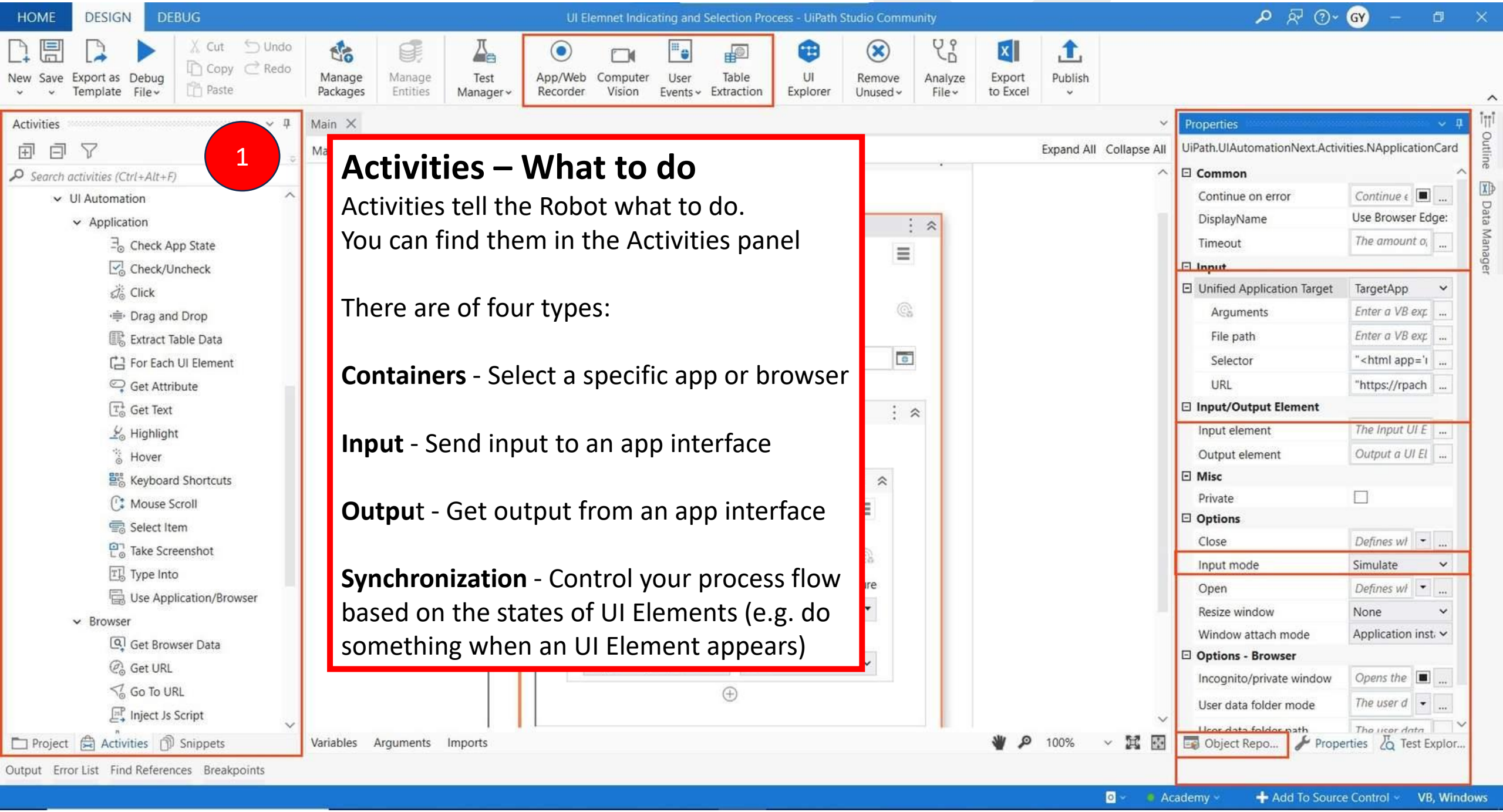
What is UI Automation?

- Automates interactions with GUI elements in software apps.
- Saves time and minimizes errors in the process
- Utilizes robots that mimic human user tasks
- Facilitates automation of complex business processes
- Particularly effective for repetitive tasks.

UI Automation Key concepts

1. UI Automation activities
2. Activity properties
3. Targeting methods
4. Input and output methods
5. Recorders
6. Extraction Wizard
7. The object repository
8. AI Computer Vision


HOME
DESIGN
DEBUG
UI Elemnet Indicating and Selection Process - UiPath Studio Community
New
Save
Export as Template
Debug File
Cut
Copy
Paste
Undo
Redo
Manage Packages
Manage Entities
Test Manager
App/Web Recorder
Computer Vision
User Events
Table Extraction
UI Explorer
Remove Unused
Analyze File
Export to Excel
Publish
2
Activities
1
Search activities (Ctrl+Alt+F)
UI Automation
Application
Check App State
Check/Uncheck
Click
Drag and Drop
Extract Table Data
For Each UI Element
Get Attribute
Get Text
Highlight
Hover
Keyboard Shortcuts
Mouse Scroll
Select Item
Take Screenshot
Type Into
Use Application/Browser
Browser
Get Browser Data
Get URL
Go To URL
Inject Js Script
Project
Activities
Snippets
Main
Main
Expand All
Collapse All
5
8
6
Use Browser Edge: Rpa Challenge
Browser URL
"https://rpachallenge.com/"
Do
Type Into 'First Name'
Type this
Standard
Secure
"Ganga"
Empty field before typing
Click before typing
Single line [End, Shift+H
None
Variables
Arguments
Imports
100%
Properties
UiPath.UIAutomationNext.Activities.NApplicationCard
Common
Continue on error
DisplayName
Use Browser Edge:
Timeout
Input
Unified Application Target
TargetApp
Arguments
File path
Selector
"<html app='
URL
"https://rpach
Input/Output Element
Input element
Output element
Misc
Private
Options
Close
Input mode
Simulate
Open
Resize window
None
Window attach mode
Application inst
Options - Browser
Incognito/private window
User data folder mode
3
4
7
Object Repo...
Properties
Test Explor...
Outline
Data Manager
Output
Error List
Find References
Breakpoints
Academy
Add To Source Control
VB, Windows



Activities – What to do

Activities tell the Robot what to do.
You can find them in the Activities panel

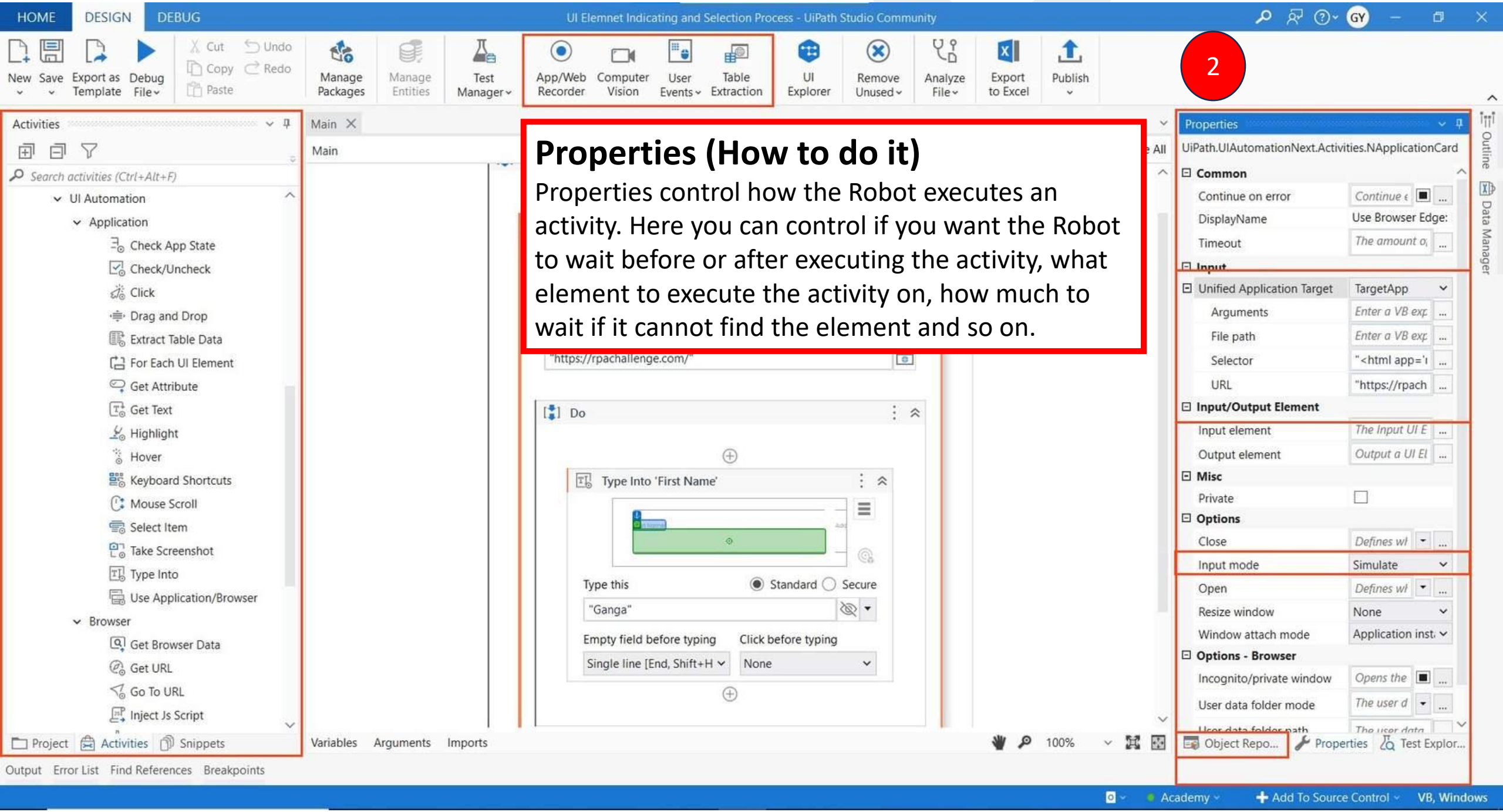
There are of four types:

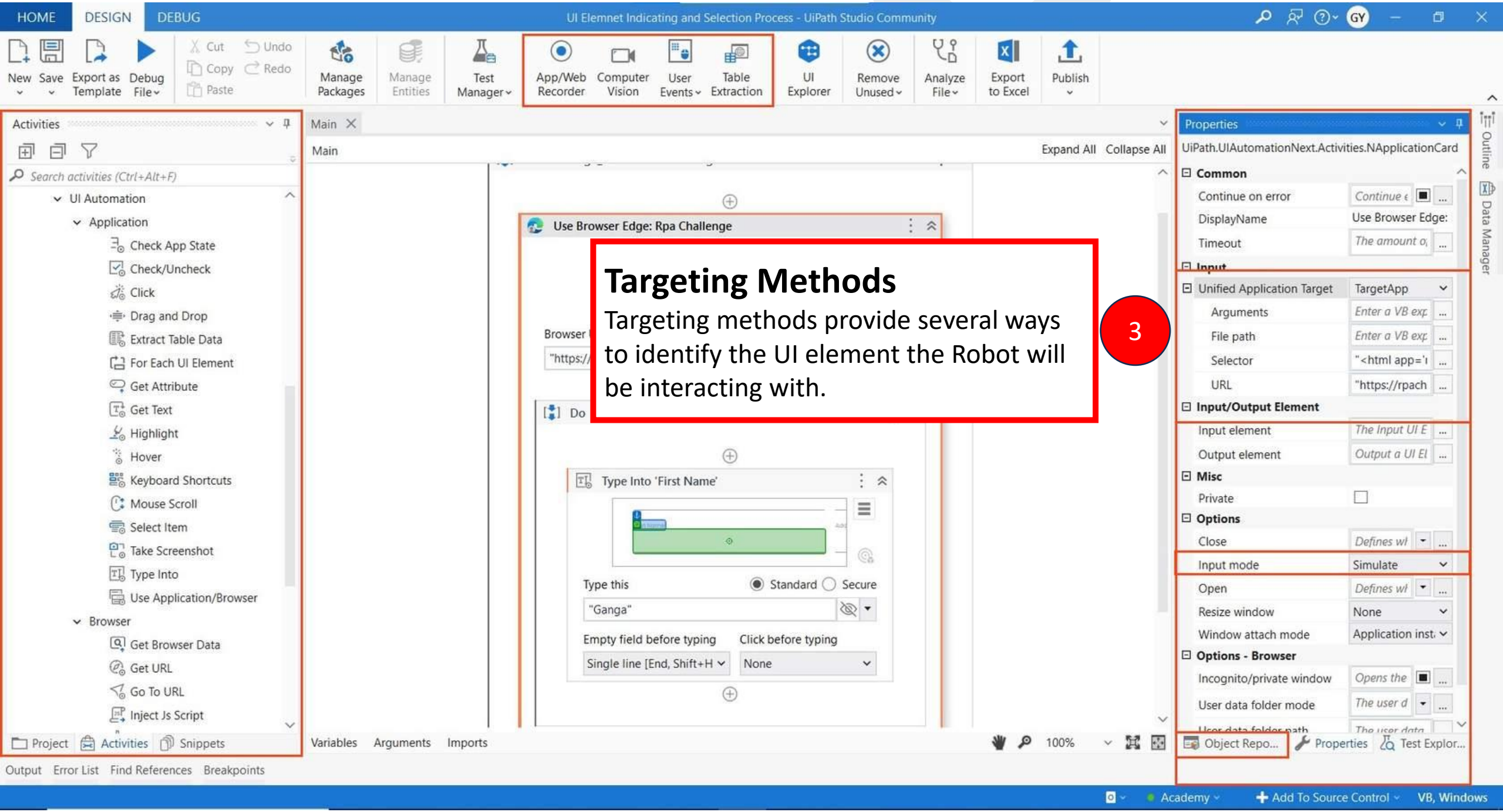
Containers - Select a specific app or browser

Input - Send input to an app interface

Output - Get output from an app interface

Synchronization - Control your process flow based on the states of UI Elements (e.g. do something when an UI Element appears)

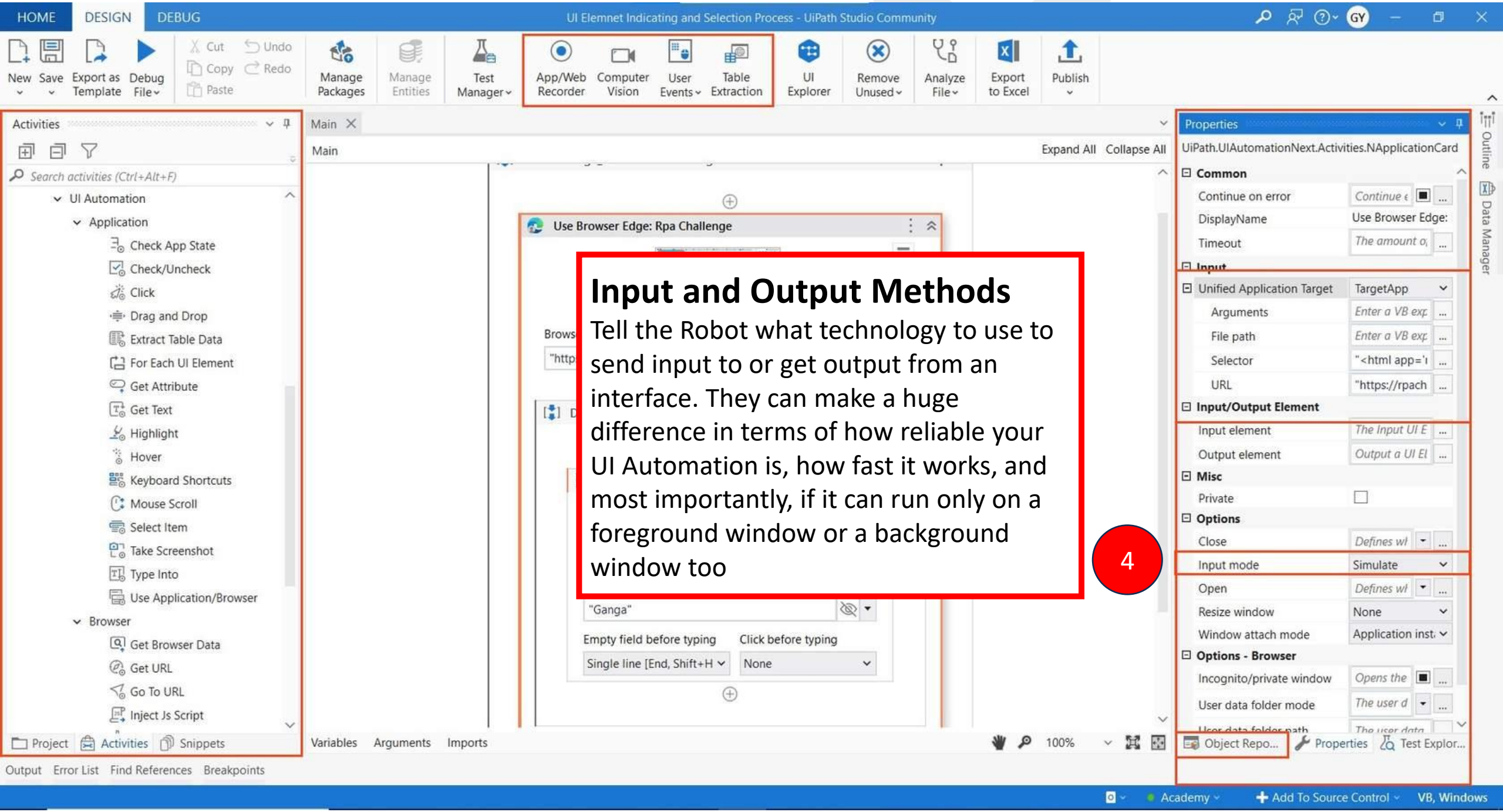

HOME
DESIGN
DEBUG
UI Elemnet Indicating and Selection Process - UiPath Studio Community
New
Save
Export as Template
Debug File
Cut
Copy
Paste
Undo
Redo
Manage Packages
Manage Entities
Test Manager
App/Web Recorder
Computer Vision
User Events
Table Extraction
UI Explorer
Remove Unused
Analyze File
Export to Excel
Publish
2
Activities
Search activities (Ctrl+Alt+F)
UI Automation
Application
Check App State
Check/Uncheck
Click
Drag and Drop
Extract Table Data
For Each UI Element
Get Attribute
Get Text
Highlight
Hover
Keyboard Shortcuts
Mouse Scroll
Select Item
Take Screenshot
Type Into
Use Application/Browser
Browser
Get Browser Data
Get URL
Go To URL
Inject Js Script
Project
Activities
Snippets
Main
Main
Properties
Properties (How to do it)
Properties control how the Robot executes an activity. Here you can control if you want the Robot to wait before or after executing the activity, what element to execute the activity on, how much to wait if it cannot find the element and so on.
"https://rpachallenge.com/"
Do
Type Into 'First Name'
Type this
Standard
Secure
"Ganga"
Empty field before typing
Click before typing
Single line [End, Shift+H
None
UiPath.UIAutomationNext.Activities.NApplicationCard
Common
Continue on error
DisplayName
Use Browser Edge:
Timeout
Input
Unified Application Target
TargetApp
Arguments
File path
Selector
URL
Input/Output Element
Input element
Output element
Misc
Private
Options
Close
Input mode
Simulate
Open
Resize window
None
Window attach mode
Options - Browser
Incognito/private window
User data folder mode
Variables
Arguments
Imports
100%
Object Repo...
Properties
Test Explor...
Output
Error List
Find References
Breakpoints
Academy
Add To Source Control
VB, Windows



Targeting Methods

Targeting methods provide several ways to identify the UI element the Robot will be interacting with.

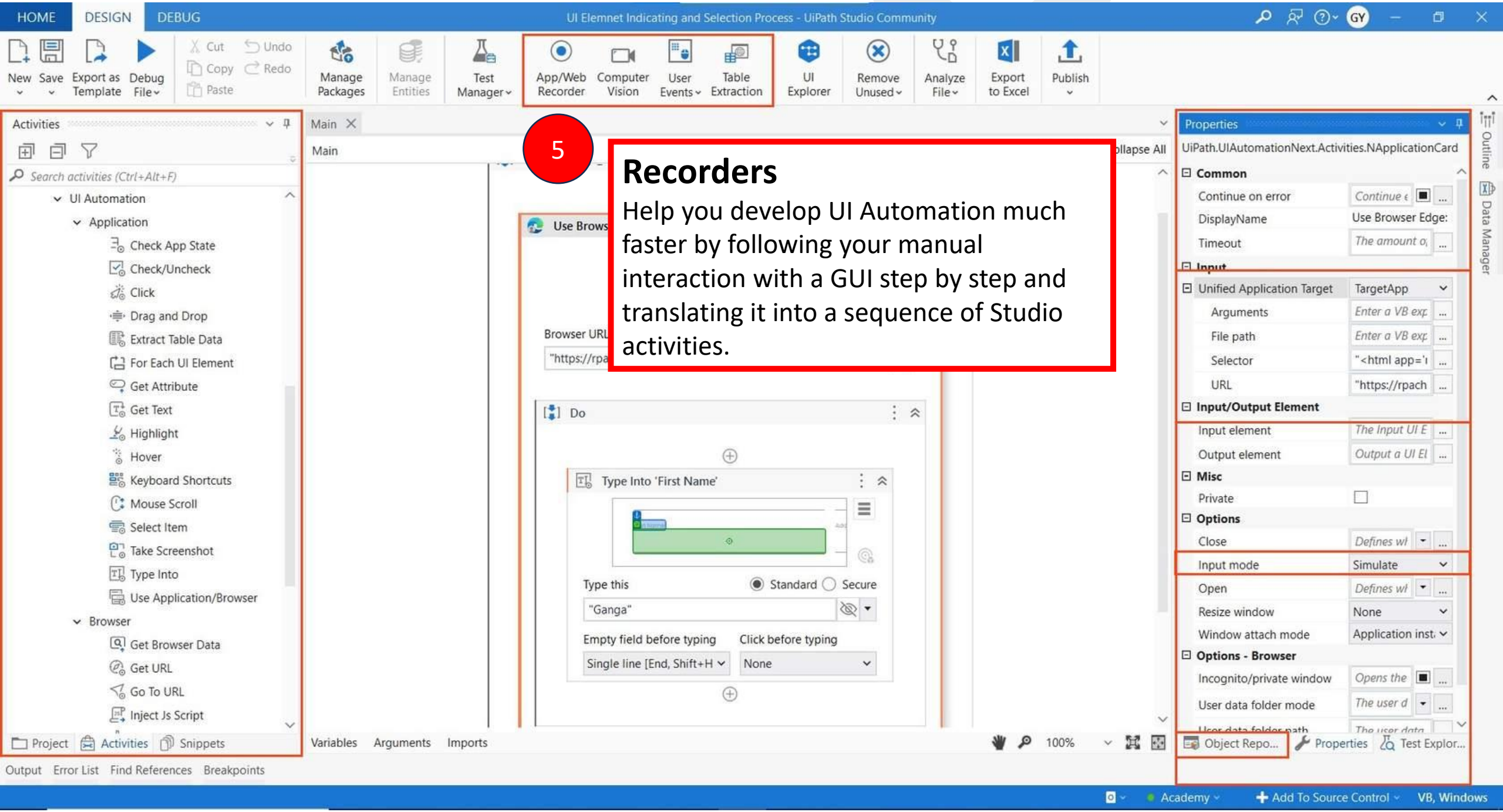
3

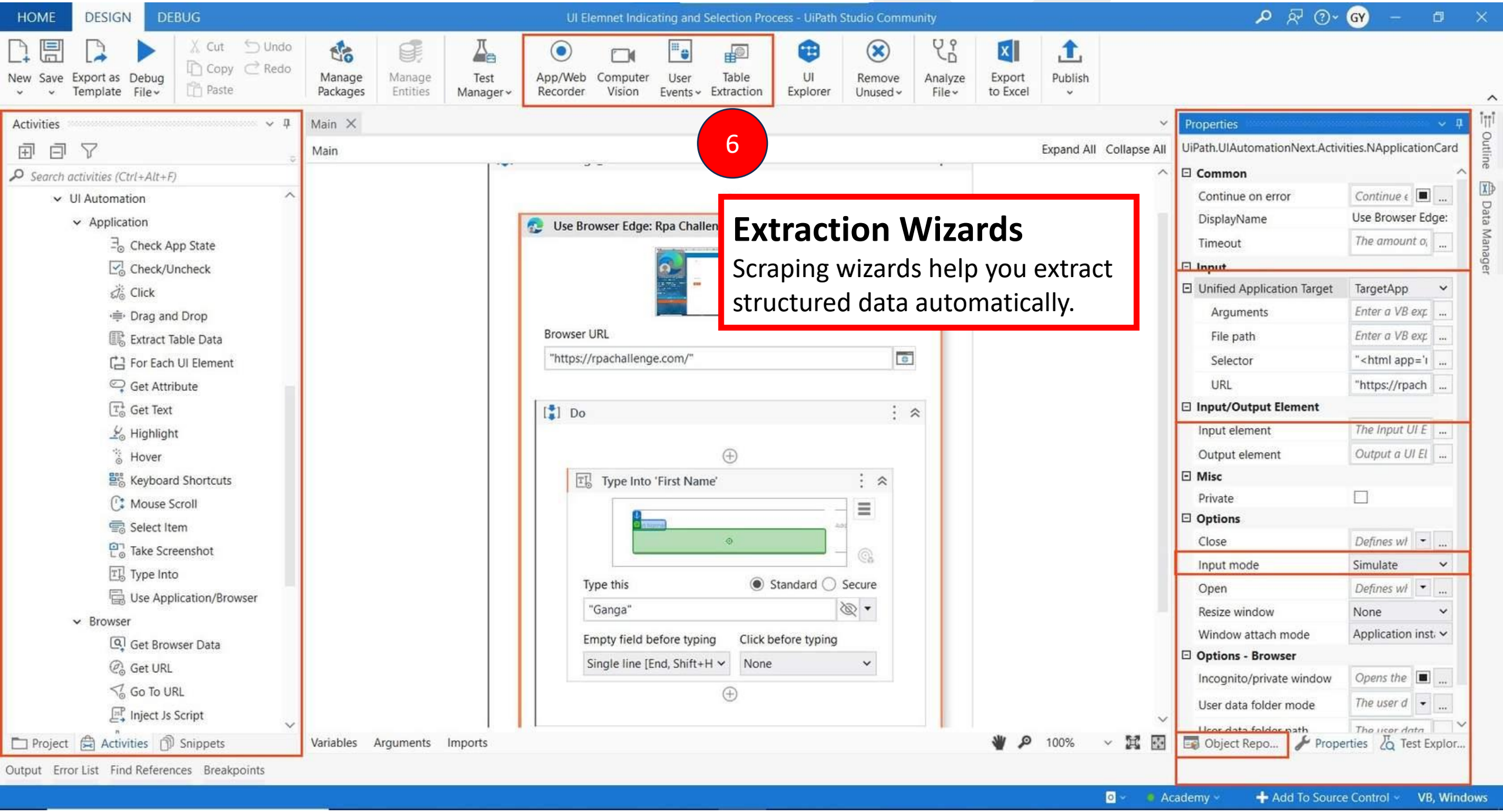


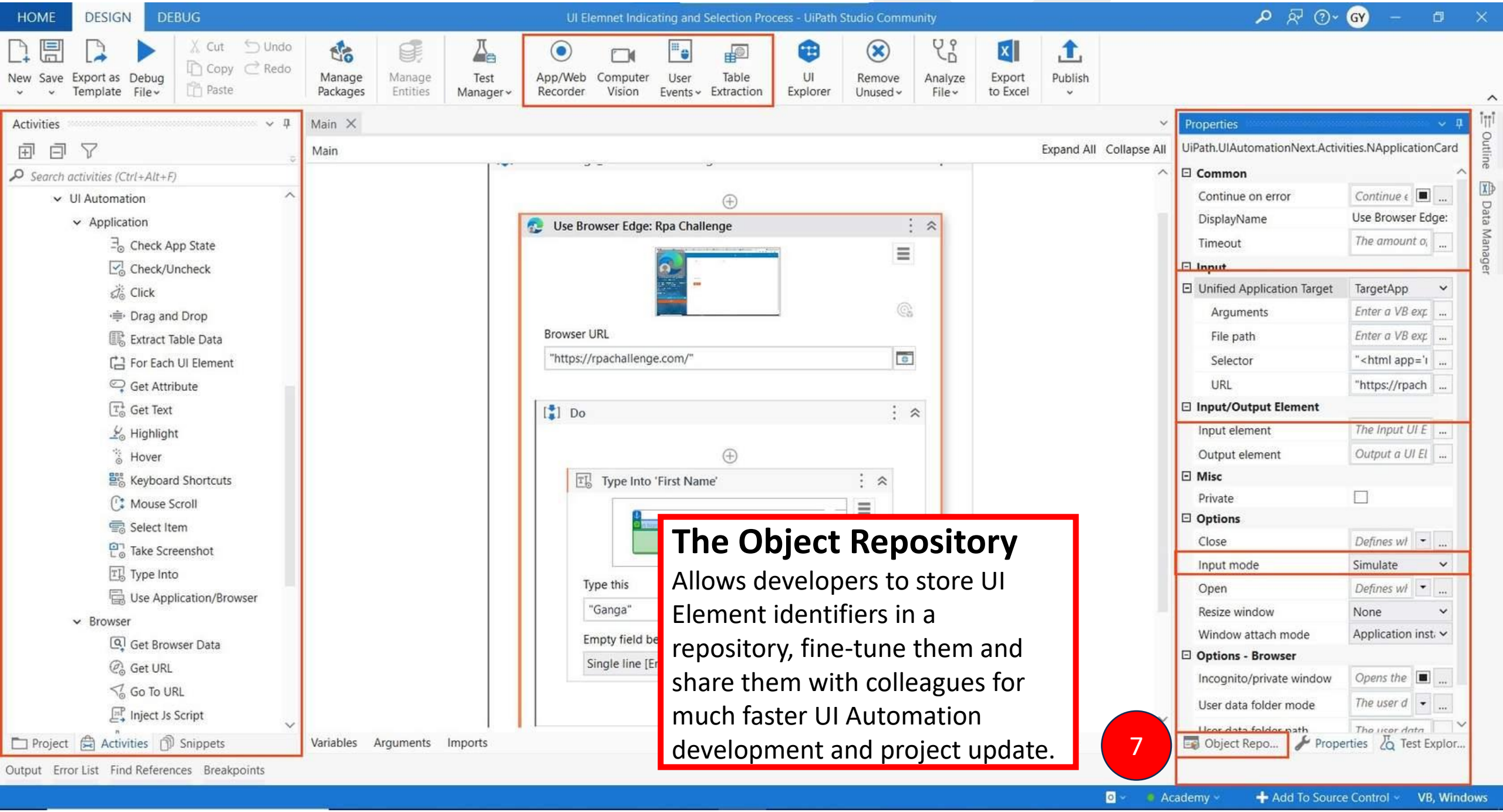
Input and Output Methods

Tell the Robot what technology to use to send input to or get output from an interface. They can make a huge difference in terms of how reliable your UI Automation is, how fast it works, and most importantly, if it can run only on a foreground window or a background window too

4

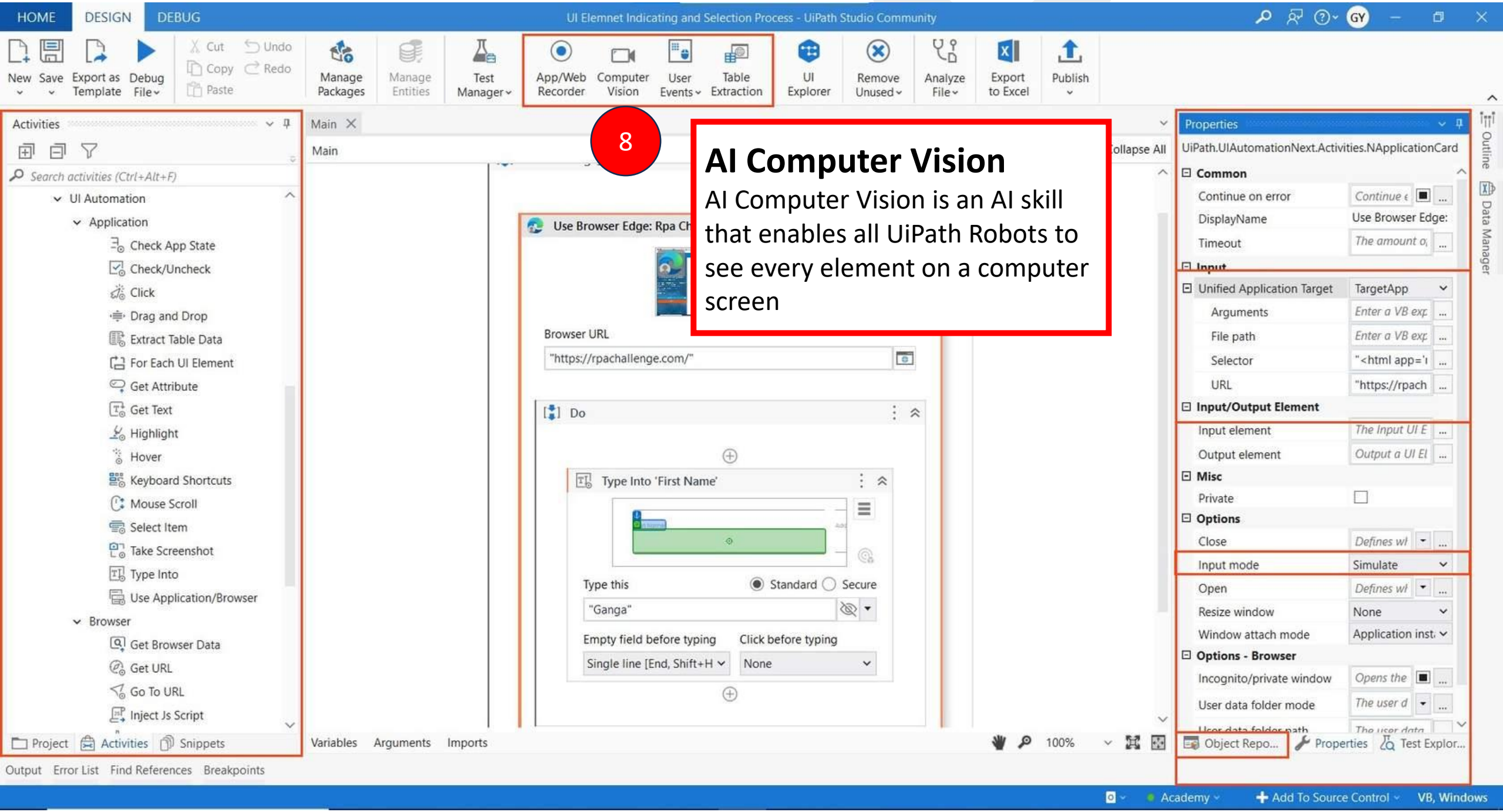

HOME
DESIGN
DEBUG
UI Elemnet Indicating and Selection Process - UiPath Studio Community
New
Save
Export as Template
Debug File
Cut
Copy
Paste
Undo
Redo
Manage Packages
Manage Entities
Test Manager
App/Web Recorder
Computer Vision
User Events
Table Extraction
UI Explorer
Remove Unused
Analyze File
Export to Excel
Publish
Activities
Search activities (Ctrl+Alt+F)
UI Automation
Application
Check App State
Check/Uncheck
Click
Drag and Drop
Extract Table Data
For Each UI Element
Get Attribute
Get Text
Highlight
Hover
Keyboard Shortcuts
Mouse Scroll
Select Item
Take Screenshot
Type Into
Use Application/Browser
Browser
Get Browser Data
Get URL
Go To URL
Inject Js Script
Project
Activities
Snippets
Main
Main
Collapse All
5
Recorders
Help you develop UI Automation much faster by following your manual interaction with a GUI step by step and translating it into a sequence of Studio activities.
Use Brows
Browser URL
"https://rpa
Do
Type Into 'First Name'
Type this
Standard
Secure
"Ganga"
Empty field before typing
Click before typing
Single line [End, Shift+H
None
Variables
Arguments
Imports
100%
Properties
UiPath.UIAutomationNext.Activities.NApplicationCard
Common
Continue on error
DisplayName
Use Browser Edge:
Timeout
Input
Unified Application Target
TargetApp
Arguments
File path
Selector
URL
"https://rpach
Input/Output Element
Input element
Output element
Misc
Private
Options
Close
Input mode
Simulate
Open
Resize window
None
Window attach mode
Application inst
Options - Browser
Incognito/private window
User data folder mode
Object Repo...
Properties
Test Explor...
Outline
Data Manager
Output
Error List
Find References
Breakpoints
Academy
Add To Source Control
VB, Windows


HOME
DESIGN
DEBUG
UI Elemnet Indicating and Selection Process - UiPath Studio Community
New
Save
Export as Template
Debug File
Cut
Copy
Paste
Undo
Redo
Manage Packages
Manage Entities
Test Manager
App/Web Recorder
Computer Vision
User Events
Table Extraction
UI Explorer
Remove Unused
Analyze File
Export to Excel
Publish
6
Activities
Search activities (Ctrl+Alt+F)
UI Automation
Application
Check App State
Check/Uncheck
Click
Drag and Drop
Extract Table Data
For Each UI Element
Get Attribute
Get Text
Highlight
Hover
Keyboard Shortcuts
Mouse Scroll
Select Item
Take Screenshot
Type Into
Use Application/Browser
Browser
Get Browser Data
Get URL
Go To URL
Inject Js Script
Project
Activities
Snippets
Main
Expand All
Collapse All
Use Browser Edge: Rpa Challen
Browser URL
"https://rpachallenge.com/"
Do
Type Into 'First Name'
Type this
Standard
Secure
"Ganga"
Empty field before typing
Click before typing
Single line [End, Shift+H
None
Extraction Wizards
Scraping wizards help you extract structured data automatically.
Variables
Arguments
Imports
100%
Properties
UiPath.UIAutomationNext.Activities.NApplicationCard
Common
Continue on error
DisplayName
Use Browser Edge:
Timeout
Input
Unified Application Target
TargetApp
Arguments
File path
Selector
URL
Input/Output Element
Input element
Output element
Misc
Private
Options
Close
Input mode
Simulate
Open
Resize window
None
Window attach mode
Application inst
Options - Browser
Incognito/private window
User data folder mode
Object Repo...
Properties
Test Explor...
Outline
Data Manager
Output
Error List
Find References
Breakpoints
Academy
Add To Source Control
VB, Windows

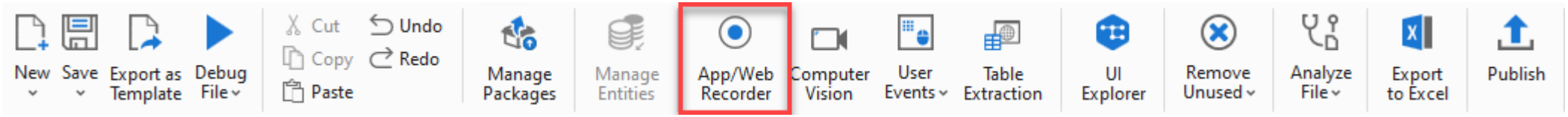


The Object Repository

Allows developers to store UI Element identifiers in a repository, fine-tune them and share them with colleagues for much faster UI Automation development and project update.


HOME
DESIGN
DEBUG
UI Elemnet Indicating and Selection Process - UiPath Studio Community
New
Save
Export as Template
Debug File
Cut
Copy
Paste
Undo
Redo
Manage Packages
Manage Entities
Test Manager
App/Web Recorder
Computer Vision
User Events
Table Extraction
UI Explorer
Remove Unused
Analyze File
Export to Excel
Publish
8
AI Computer Vision
AI Computer Vision is an AI skill that enables all UiPath Robots to see every element on a computer screen
Activities
Search activities (Ctrl+Alt+F)
UI Automation
Application
Check App State
Check/Uncheck
Click
Drag and Drop
Extract Table Data
For Each UI Element
Get Attribute
Get Text
Highlight
Hover
Keyboard Shortcuts
Mouse Scroll
Select Item
Take Screenshot
Type Into
Use Application/Browser
Browser
Get Browser Data
Get URL
Go To URL
Inject Js Script
Project
Activities
Snippets
Main
Use Browser Edge: Rpa Ch
Browser URL
"https://rpachallenge.com/"
Do
Type Into 'First Name'
Type this
Standard
Secure
"Ganga"
Empty field before typing
Single line [End, Shift+H
Click before typing
None
Variables
Arguments
Imports
100%
Properties
UiPath.UIAutomationNext.Activities.NApplicationCard
Common
Continue on error
DisplayName
Use Browser Edge:
Timeout
Input
Unified Application Target
TargetApp
Arguments
File path
Selector
URL
Input/Output Element
Input element
Output element
Misc
Private
Options
Close
Input mode
Simulate
Open
Resize window
None
Window attach mode
Options - Browser
Incognito/private window
User data folder mode
Object Repo...
Properties
Test Explor...
Outline
Data Manager
Output
Error List
Find References
Breakpoints
Academy
Add To Source Control
VB, Windows

App/Web Recorder



Recording can help you save a lot of time when automating your tasks.
The App/Web Recorder captures your actions as you perform them on the screen **and generates a Use Application/Browser activity** with a series of activities inside it based on your actions.

Automatically Recorded Actions	Actions That Must Be Selected Before Recording
• Clicking on buttons, links, and other clickable elements such as icons or images. A **Click** activity is generated. • Typing text in a text area such as a text box. A **Type Into** activity is generated. • Selecting or clearing a check box. A **Check/Uncheck** activity is generated. • Sending keyboard shortcuts using your keyboard. A **Keyboard Shortcuts** activity is generated. • Selecting an item from a drop-down. A **Select Item** activity is generated.	• Copying text using the **Get Text** activity. • Hovering over an element using the **Hover** activity. • Highlighting an element using the **Highlight** activity.

Modern & Classic Design Experiences

Two design experiences in UiPath Studio: Modern and Classic.

- Modern experience: User-friendly interface with new features.
- Classic experience: Traditional interface.
- Modern experience is the default in UiPath Studio.
- Users can switch to the Classic experience based on their preferences or requirements

Modern & Classic Design Experiences

Two design experiences in UiPath Studio: Modern and Classic.

- Modern experience: User-friendly interface with new features.
- Classic experience: Traditional interface.
- Modern experience is the default in UiPath Studio.
- Users can switch to the Classic experience based on their preferences or requirements

UiPath Documentation: Studio - Modern Design Experience (uipath.com)

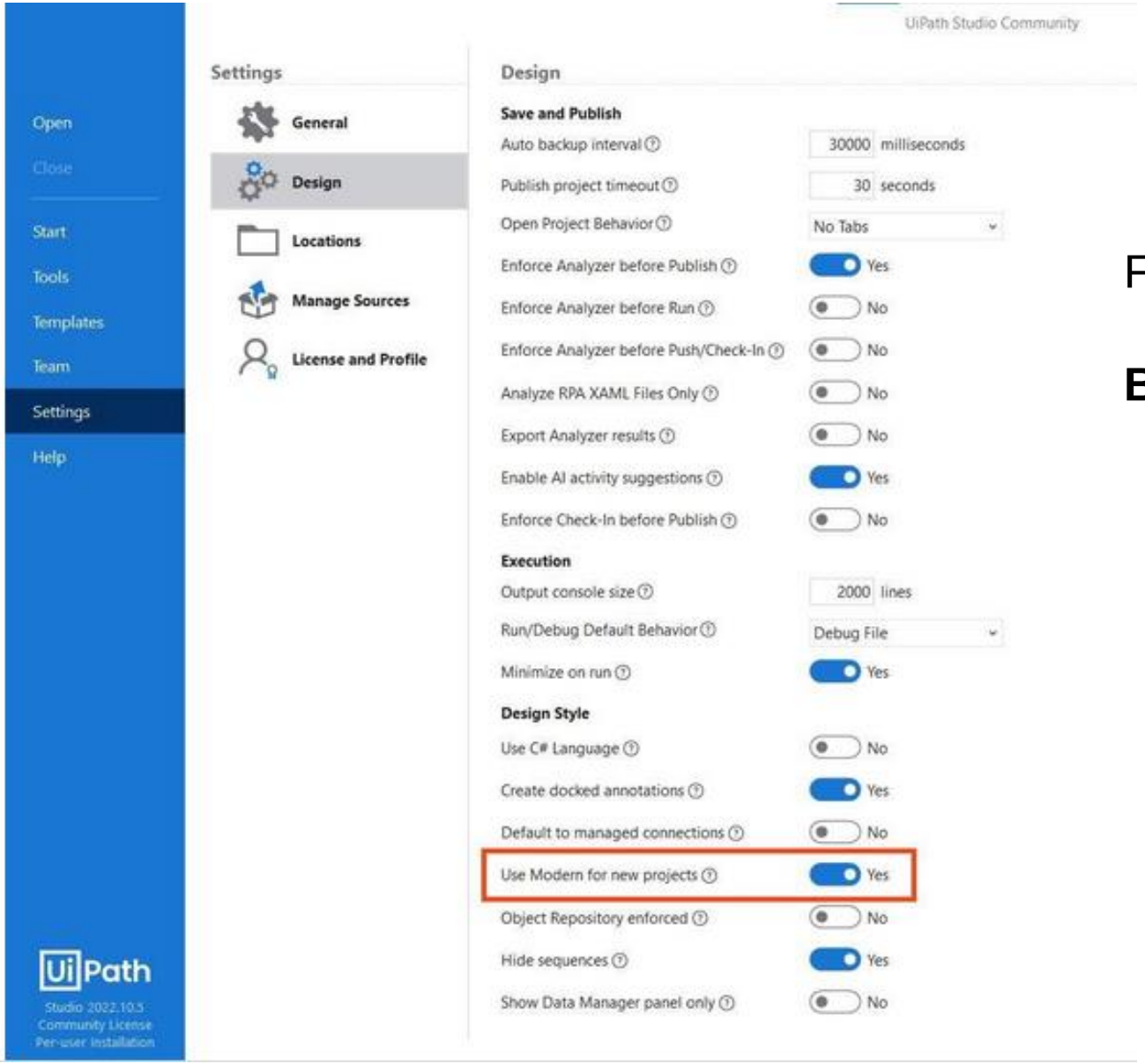
Changing Design Experiences
Modern <-> Classic

Change the user experience in UiPath Studio

1. At the Studio level

2. At the project level

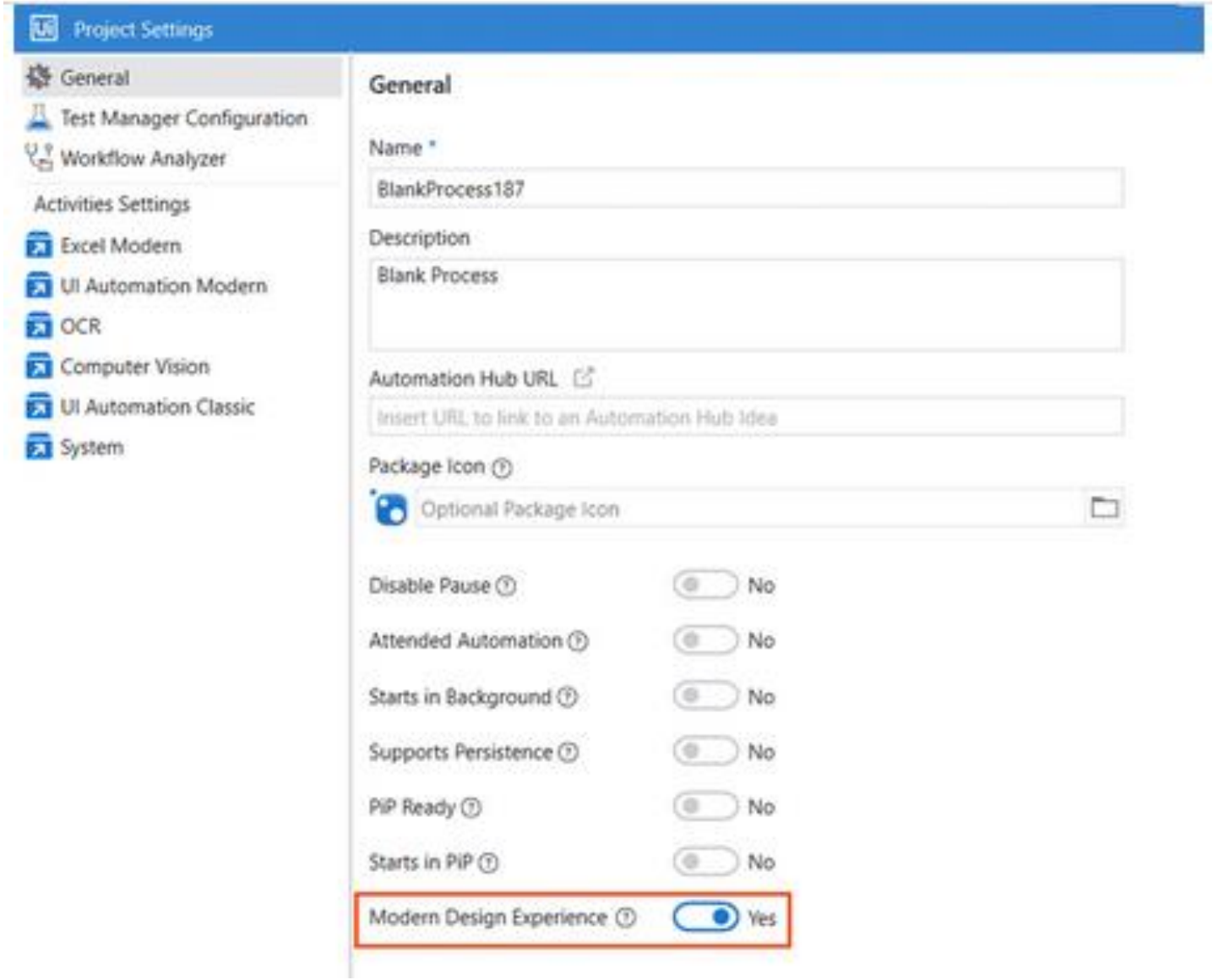
Change the default design experience at the Studio level



For all new projects, we can set this from

Backstage View > Settings > Design.

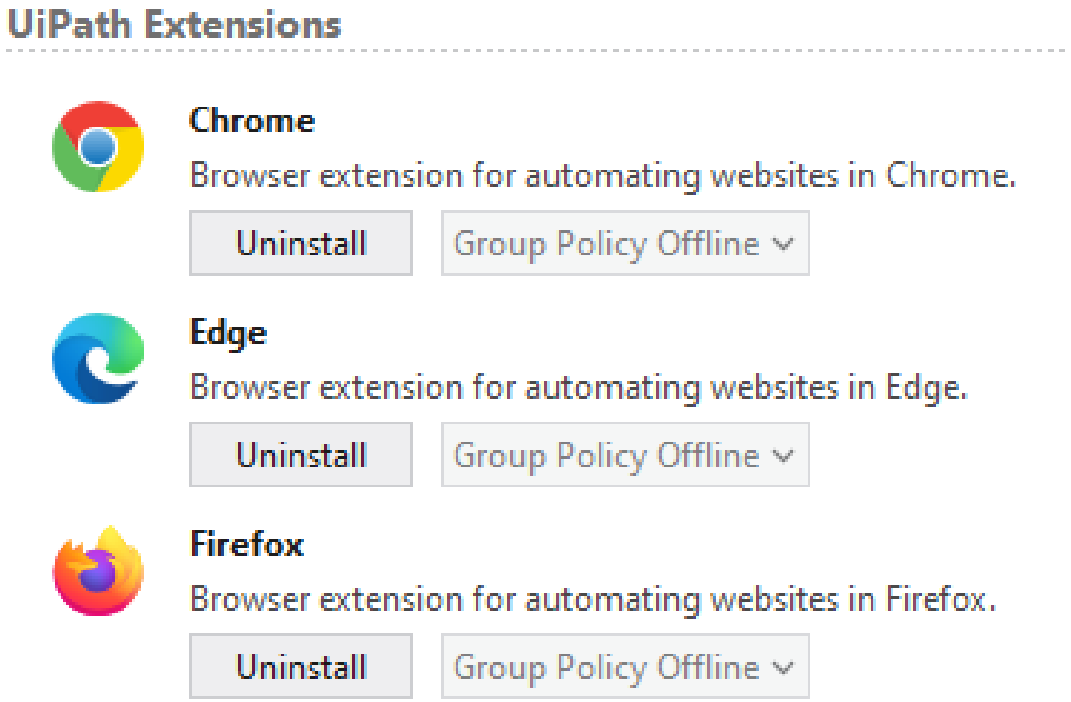
Change the default design experience at the Project level



The default design experience is set at project level from Project Settings or for all new projects from the

Project tab > Project Settings > General

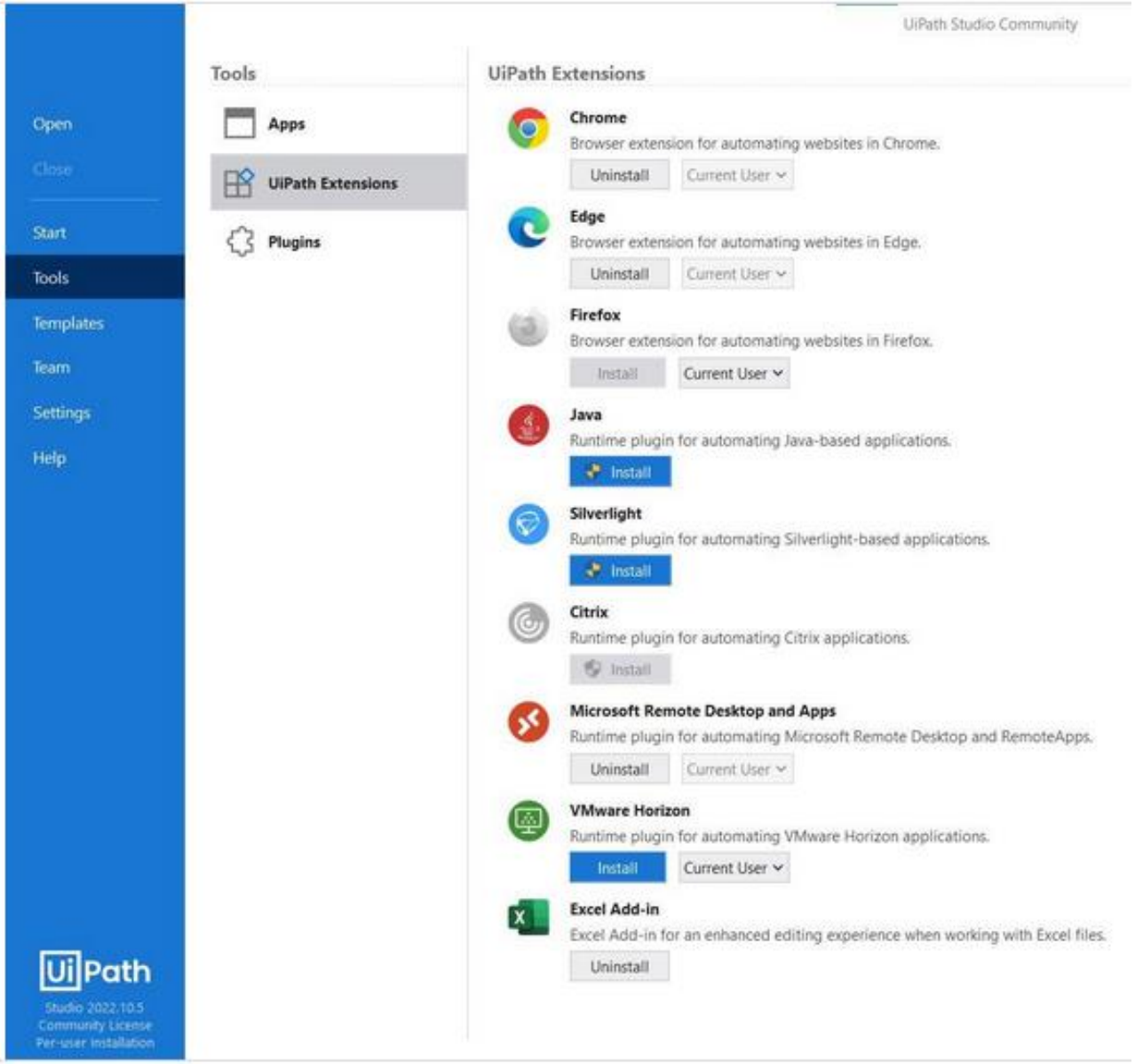
Browser Extensions



- Enable UiPath automation to interact with web application U
- Each browser has distinct extensions, akin to different keys for different doors.

If the UiPath browser extension is not installed and enabled, UiPath will not be able to interact with that browser

How to install and enable UiPath Extensions



1. Open UiPath Studio and click the (Home) "Tools" tab.
2. Navigate to the "UiPath Extensions" tab.
3. Choose the extension you want to automate from the list of available extensions.
4. Click on the "Install" button.
5. Wait for the extension to be installed. Once it's done, the status of the extension will change to "Installed".
6. Restart UiPath Studio for the changes to take effect.

After completing these steps, the extension will be installed and you'll be able to automate web applications in the selected browser.

https://docs.uipath.com/studio/docs/about-extensions

Targeting Methods: What is a Target?

A target refers to a **specific UI element that we want to interact with**.

It could be any component on the user interface, like a button, textbox, or dropdown list.

For example, if we have a login form, the "Submit" button and the "Username" and "Password" textboxes would be considered targets.

Targeting Methods: What is an Anchor?

An anchor **is an additional UI element that helps in identifying the target**.

Anchors are particularly useful when potential conflicts are detected in the selection window. By incorporating multiple anchors, we enhance the reliability and accuracy of our automation.

When working with modern UI automation activities, an anchor is automatically selected for you. However, you can also manually configure the anchor selection if needed

Note: You can add up to three anchors to a single target, maximizing the reliability of your automation.

Overview of the targeting methods

Like identifying a button on the screen with our eyes, the robot uses different targeting methods to "see" and "understand" elements on the screen.

The key targeting methods used in UiPath Studio modern design are:

1. Strict Selectors
2. Fuzzy Selectors
3. Image Recognition

Unified Targeting Method

The Unified Target method, a modern solution for UI automation, simplifies the interaction with user interfaces by incorporating various technologies and methods like Selectors, Fuzzy Selectors, and Images to identify UI elements.

By using a unified targeting method, the different methods within it complement and support each other, enhancing the reliability of UI automation. This means that even if one method encounters difficulties in selecting a UI element, other methods can step in to overcome the obstacle.

Key advantages of using the Unified Target method

One of the key advantages of using the Unified Targeting method is **its versatility and ease of use**. It provides a **ready-to-use solution** that seamlessly integrates into workflows without the need for complex configuration or considering individual method-specific details. This makes it **highly** convenient and efficient for developers and automation engineers.

To identify UI elements, the Unified Targeting method **uses a combination of UI frameworks**. By default, a proprietary framework is used, but alternative options like AA (Active Accessibility) and UIA (Microsoft UI Automation) are available based on the target application's type.

Unified Targeting Method

A Unified Target consists of at least one target, along with optional anchors (0 to 3)

Each UI element is located using a stack of targeting methods, including

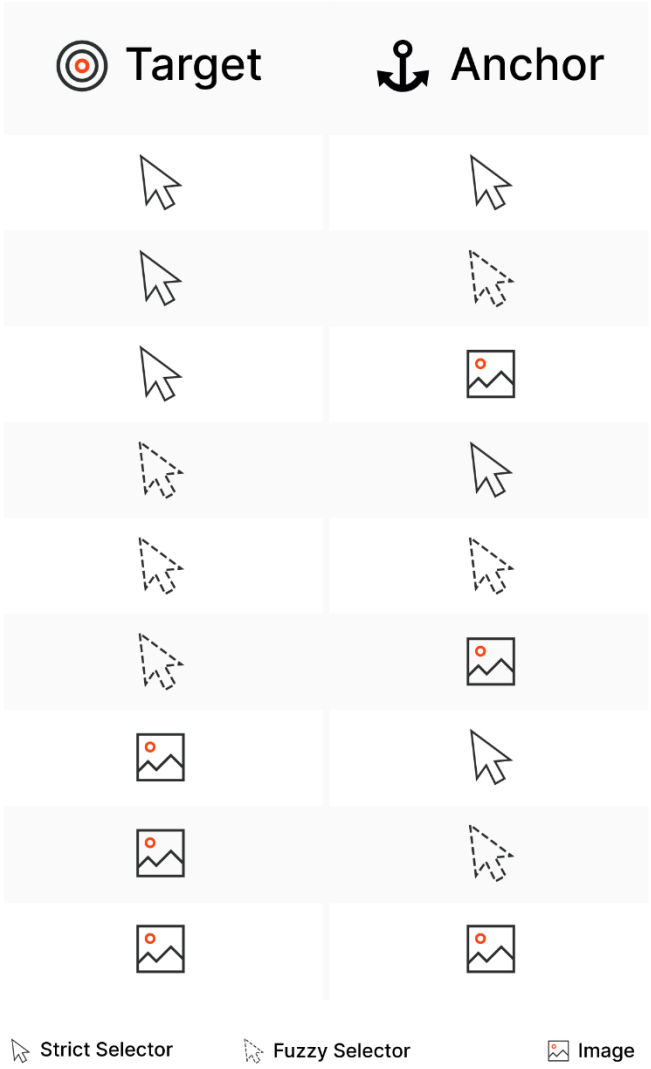
- Strict Selectors
- Fuzzy Selectors
- and Images

These targeting methods work **redundantly, simultaneously** attempting to identify the target element.
The **first method that successfully finds the target is considered**.

To customize the targeting methods used in the Unified Targeting method, you can access **the project settings** within the UI Automation Modern section.
From there, you can configure and adjust the specific targeting methods according to your needs and preferences.

Targeting Method : Descriptors

The Target and Anchor pair is known as a Descriptor



Under the hood, the robot generates these **nine unique combinations**, which are known as Descriptors.

Each Descriptor represents a unique combination of the identified Target and Anchor elements using their respective targeting methods.

Note: Keep in mind that the number of combinations depends on the number of anchors. A target can have up to three anchors associated with it.

A descriptor is a unique pair consisting of an anchor element and a target element. It's generated when both the anchor and target are successfully identified using their respective targeting methods.

UiPath Academy

Chapter 6:

User Interface Automation with the Modern Experience in Studio

Identifying UI Elements

Demo of the targeting methods

Demo, we'll create a simple automation process that

1. Opens the 'RPA Challenge' website, (www.rpachallenge.com)
2. Inputs a name in the first name field
3. Clicks the Submit button

Using the **Unified Targeting method** (one target & 0 to 3 anchors)
Includes selectors, fuzzy selectors, and images.
These methods can be applied to both targets and anchors and can be switched between in the settings, which can be accessed by hovering over them or directly from the selection options window

When the robot searches for a UI element, whether it be a target or an anchor, it uses all targeting methods **simultaneously** and **chooses the first one** that successfully locates the element.

Speed
The Selector method is the fastest and looks for elements that match specific attributes,
The Fuzzy method is more flexible in identifying elements.
The Image method looks for an exact image of the element and is recommended when other methods are not stable or accessible, but it can be slower and less reliable.

Overview of the targeting methods

- **Target** refers to the element in the UI that a UiPath robot interacts with, such as buttons, input fields, and dropdown menus.
- An **anchor** is a visual clue used to locate a target element and make interaction more reliable.
- You can add up to **three anchors** to a target.
- The key targeting methods in UiPath Studio modern design are **strict selectors, fuzzy selectors**, and **image recognition.**
- The provides tools to specify the target element and any anchors or attributes associated with it. **Selection Options window**
- **Indicate on Screen** is the easiest way to specify a target element by clicking on it in the UI.
- **Pause Configuration (F2)** Allows you to pause the selection of a UI element.
- **Hoverable Selection** allows you to indicate elements that only become visible when you hover over them.

https://docs.uipath.com/activities/docs/selection-options#indicating-an-element

Targeting Methods Resources

Topic	Link
About Targeting Methods	https://docs.uipath.com/activities/other/latest/user-guide/about-the-ui-automation-next-activities-pack#targeting-methods
About Targets and Anchors	https://docs.uipath.com/activities/other/latest/user-guide/about-the-ui-automation-next-activities-pack#about-targets-and-anchors
About Configuring the Descriptor	https://docs.uipath.com/activities/other/latest/user-guide/advanced-descriptor-configuration#configuring-the-descriptor
About Validating the Descriptor	https://docs.uipath.com/activities/other/latest/user-guide/advanced-descriptor-configuration#validating-the-descriptor

Overview of UI automation activities

Ui Activities can be divided into **four** categories

1. **Containers**
   These activities identify the browsers or applications that the process needs to interact with.
   The Container will execute all activities within it on the same application.
   These activities include Use Application/Browser

2. **Activities for Synchronization**
   These activities enable the Robot to perform specific actions when specific events on the machine occur based on the UI behavior

3. **Input Activities**
   These activities are used to input data into user interface elements.
   Other examples would be clicking, checking, typing into, or sending hotkey

4. **Output Activities**
   These activities are used retrieve information from GUI Elements.
   They can instruct the Robot to get text by using various methods, get structured data, or retrieve UI Elements containing images.

Containers – Some Examples

- **Use Application/Browser activity** is a UI automation activity in that opens a desktop application or web browser page to use in UI automation
- **Use Desktop Outlook App activity s**elects an account from the desktop Outlook application to use in your automation and enables Studio to integrate with Outlook
- **Use Excel File activity l**ets you select an Excel file to use in the automation and enables Studio to integrate with Excel. The data in the file is available to all the activities added inside Use Excel File
- **Use Gmail activity** selects a Gmail account to use in your automation and enables Studio to integrate with Gmail and Google Calendar
- **Use Outlook 365 s**elects an online Outlook 365 account to use in your automation and enables Studio to integrate with Outlook 365

‘Use Application/Browser’ example activity

With the "Use Application/Browser" activity, users can manage and automate software applications in a variety of ways.

Users can

- Launch and Close applications
- Retrieve window handles
- Manipulate Application Windows

The application settings can be customized and users can customize the activity to meet their individual needs.

‘Use Application/Browser’ activity

- The "Use Application/Browser" activity opens either a web browser page or desktop application for use in UI automation.
- Using a project-relative path is possible by copying the application folder to the project folder
- "Input" settings configure the target application
- "Arguments" pass parameters at startup, like opening a specific file
- "File path" specifies the executable file to open, clearing the URL
- "Open" chooses when to open the app: Never, IfNotOpen (default), or Always
- "Resize window" resizes the app at initialization: None, Maximize, Restore to current size, or Minimize
- "Window attach mode" chooses where inner activities search for their target elements: Application instance or Single window.
- "Close" chooses when to close the app: Never, IfOpenedByAppBrowser (default), or Always

https://docs.uipath.com/activities/docs/n-application-card#description

Introduction to UI Synchronization Activities

Check App State Activity

This activity allows you to check the state of an application, ensuring that it is in the expected state before proceeding with further actions

Verify Execution Feature

With this feature, you can verify that a specific action or activity has been successfully executed before moving on to the next step

Pick Branch Activity

The Pick Branch activity provides a flexible way to synchronize and execute different branches of activities based on specific conditions or events.

Activity properties

UI Automation activities have different sets of properties depending on the type of activity. Below is an illustration of the types of properties used in UI Automation activities.

Property	What it does
DelayBefore/ DelayAfter	How many milliseconds the robot waits before or after executing the activity.
ContinueOnError	Specifies if the automation should continue even when the activity throws an error.
Target	Provides several properties related to identifying the target UI Element.
Timeout (seconds)	How many milliseconds will the Robot try to perform an action on a UI element.
Input Mode	What input method do we use for input activities.
Output	It Stores the output of the activity in the form of variables.

Input Methods

Whenever we insert data into an application or send a command to a system to produce a change or continue, we perform an input action.

UiPath offers various ways to execute input actions.
These methods control a variety of factors, including the speed of execution, whether it runs on a background or hidden window, and whether it interferes with the user.

Modern design experiences use **four** input methods

1. Hardware Events
2. SendWindowMessages
3. Simulate
4. ChromiumAPI

Input Methods - Hardware Events

Hardware events are essentially the actions we perform using our keyboard or mouse, such as clicking or typing. These actions are captured by the operating system and can be used by UiPath to simulate user interactions with an application.

When we use the hardware events input method in UiPath, it directly interacts with the hardware device (mouse or keyboard) by sending messages to the operating system.
This means that when we click, the mouse cursor moves across the screen and when we type, the keyboard driver is used to type individual characters.

One important thing to note is that hardware events are compatible with all applications but **do not work in the background**. This means that the **attended user cannot touch the mouse or keyboard during the automation.**

Use case: this input method is useful for automating simple tasks that involve keyboard and mouse interactions, such as data entry or clicking buttons. It is also useful for automating applications that are not compatible with other input methods, as it can interact with any application.

Input Methods - SendWindowMessages

It simulate user interactions with an application.

When you use this input method, UiPath sends the same messages to an application that the application would receive if a user were to interact with it using a keyboard or mouse. This means that UiPath is essentially controlling the application in the same way that a user would.

The benefit of using Send Window Messages is that it **allows the automation to work in the background without interfering with the user's ability to use their computer.**

Use case : this input method is useful for automating applications that require a high level of accuracy or timing, as it can directly communicate with the application's window. It is also useful for automating applications that do not respond to other input methods.

Input Methods - Simulate

Simulate input method designed to mimic our actions in a way that's similar to how we would interact with the application. This means that instead of relying on low-level hardware events or window messages, Simulate can interact with UI elements directly.

By doing this, the robot can perform actions such as clicking a button or typing in a text box more accurately and consistently.

Additionally, using Simulate is often faster than using hardware events or window messages, and **allows the robot to interact with an application in the background while we perform other tasks**.

It's important to note, however, that **not all applications are compatible with Simulate**. In these cases, it's important to choose the input method that works best for the specific application being automated.

Use case: this input method is useful for automating tasks that involve interacting with graphical user interfaces (GUIs), such as clicking buttons or typing in text boxes. It is also useful for automating tasks that need to be performed quickly and accurately.

Input Methods - ChromiumAPI

ChromiumAPI is an input method used for browser automation, and it's **based on the Devtools protocol**. It's **compatible with all Chromium-based browsers**, such as Chrome or Edge. Basically, it can be **used for any website or application that runs inside the Chromium browser**.

ChromiumAPI supports several activities including Use Application/Browser, Click, Type Into, Hover, and Keyboard Shortcuts.

It uses **direct communication with the browser**, which means there are fewer communication channels and increased automation reliability.

One of the advantages of using ChromiumAPI is that it **works in the background**, allowing users to work on other activities while the automation is executing.

Use case: this input method is **useful for automating tasks that involve interacting with web applications**, as it can directly communicate with the browser's Devtools protocol. It is also useful for automating tasks that need to be performed in the background, such as web scraping or monitoring web pages.

Differences between input methods

Input Method	Compatibility	Background Execution	Speed	Hotkey Support	Auto Empty Field
Hardware Events	100% - all types of applications.	No	50%	Yes	No
SendWindowMessages	80%	Yes	50%	Yes	No
Simulate	99% - web apps 60% - desktop apps	Yes	100%	No	Yes
ChromiumAPI	100% - Chrome and Edge browsers	Yes	50%	Yes	Yes

Activities Resources

Topic	Link
UIAutomation Modern Activities	https://docs.uipath.com/activities/docs/ui-automation-modern-activities
UI Activities Properties	https://docs.uipath.com/studio/v2022.10/docs/ui-activities-properties
Input Methods	https://docs.uipath.com/studio/v2022.10/docs/input-methods

UiPath Academy

Chapter 6:

User Interface Automation with the Modern Experience in Studio

Input Methods and Input Activities Part 1

Foreground and Background processes

- **Foreground processes** are things you are currently using on your computer, such as documents or web browsers, for which you can see and interact with your screen

- **Background processes**, run on your computer without your involvement

Differences between background and foreground processes

Charcterstics	Foreground Process	Background Process
Example	Typing a document, browsing the internet	Antivirus scans, file backups
Visibility	Visible and interactive on the screen	Not visible or interactive on the screen
Resource usage	Uses computer resources while actively being used	Uses computer resources while running, but at a lower priority
Interruptions	Can be interrupted or paused by the user	Generally can't be interrupted or paused by the user
User attention	Requires user attention	Doesn't require immediate user attention

Background Mode

Background mode allows data to be input into an application or browser without requiring active user interaction, operating as a background process.

Background mode is **available only for Use application browser Activity**

Runs all the activities inside the "Do" block of the "Use application browser" card in the background, using either Simulate or ChromiumAPI as an input method

The following actions **aren't compatible** with the Background mode:

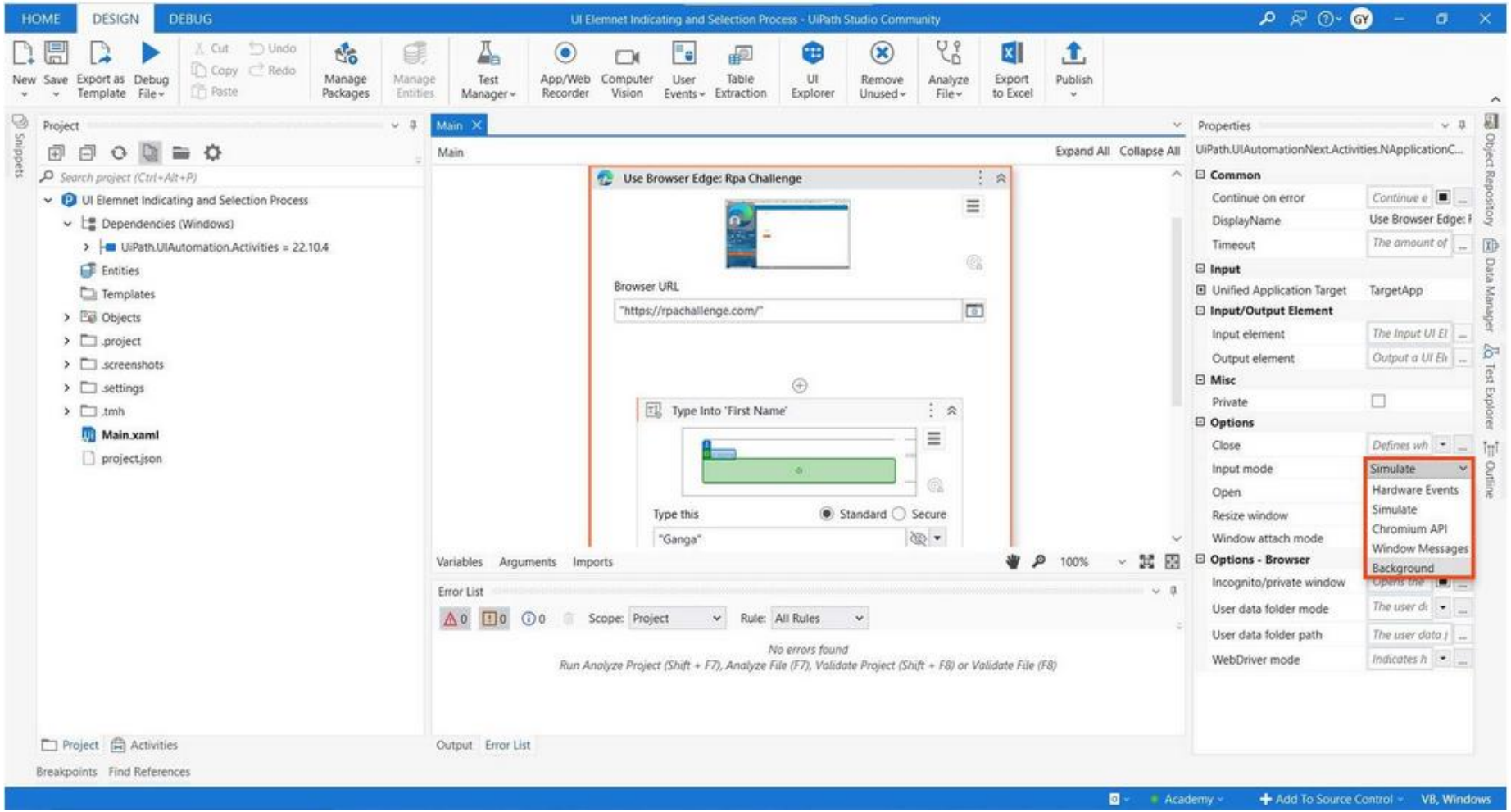
- Any activities using image as targeting method
- Native text automation
- Keyboard shortcuts
- Minimizing opened applications
- The Take Screenshot Activity

Note: If the automation includes incompatible actions for Background mode It'll run smoothly, **but incompatible activities will run in the foreground**. However, after the execution, it'll return to the background.

How to use Background Mode

From the **Properties panel ->Input mode option (**only for Use application browser Activity)

From the dropdown you can select the Background mode option



https://docs.uipath.com/activities/docs/n-application-card#options

Input Activities

The simple actions when working with UI elements in Studio are:

Click

Clicks a specified UI element

Type Into

Sends keystrokes to a UI element

Send Hotkey

Sends keyboard shortcuts to a UI element

Input Activities

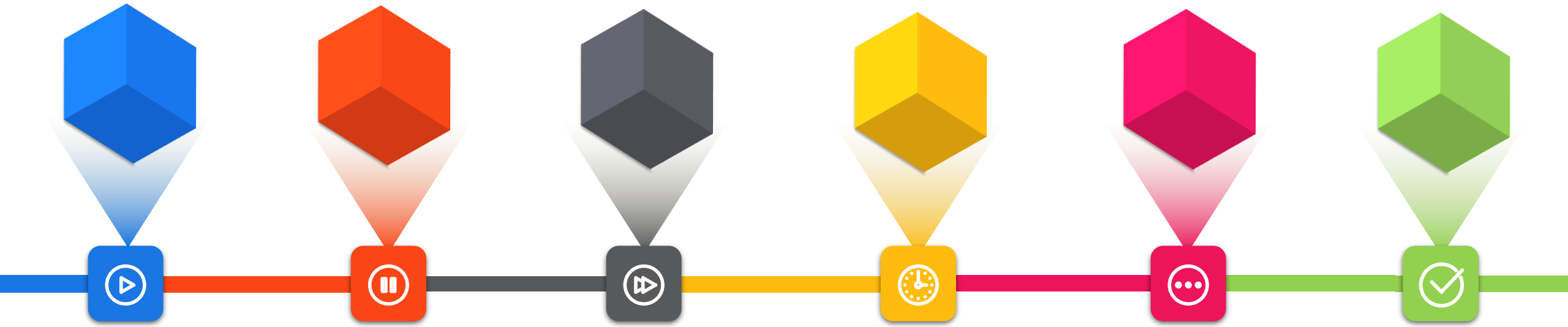
The input activities allow us to **insert data** into an application
or **send commands** to a system that produce a change or continuation

The main input activities are

- Check/Uncheck
- Click
- Go to URL
- Highlight
- Hover
- Keyboard Shortcuts
- Navigate Browser
- Select Item
- Type Into

Common Properties of UI Activities

The common properties of UI activities are:



ContinueOnError
Specifies whether the automation should continue if it shows an activity error

DelayAfter
Adds a pause after the activity, in milliseconds

DelayBefore
Adds a pause before the activity, in milliseconds

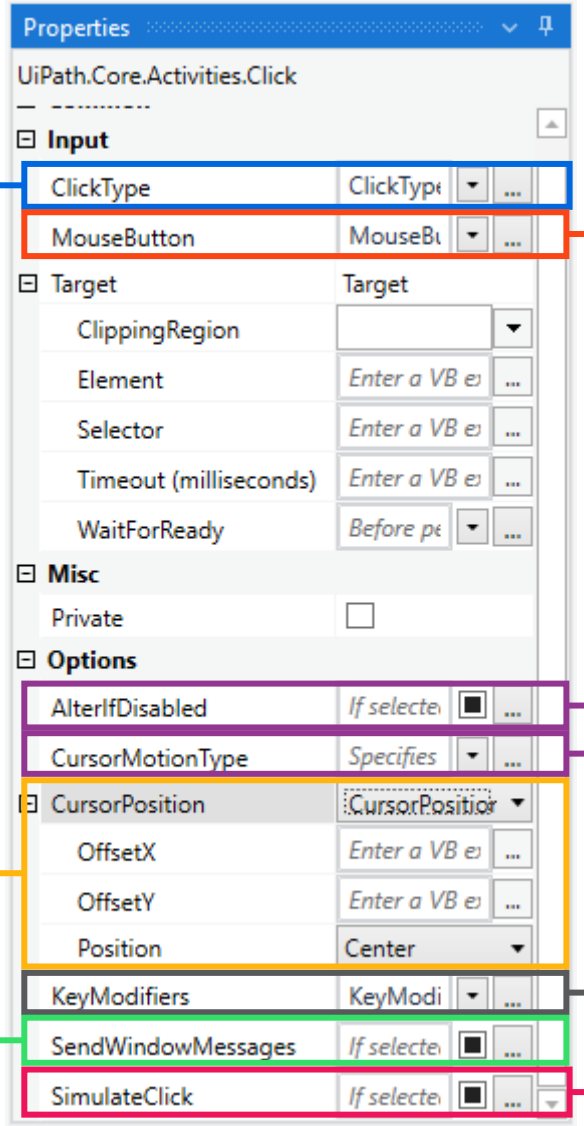
Timeout
Specifies the amount of time (in milliseconds) to wait for a specified element to be found

WaitForReady
Before performing the actions, waits for the target to become ready

Target
Identifies the UI element with which the activity works

Properties of Click

The properties of Click are:



ClickType:
Single, Double, Down, Up

CursorPosition:
Select the cursor option:

- Null
- CursorPosition
  - OffsetX
  - OffsetY
  - Position

SendWindowMessages:
If selected, the **click** is executed by sending a specific message to the target application

MouseButton:
Left, Middle, Right

AlterIfDisabled:
If selected, the simulated click is executed even if the specified UI element is disabled

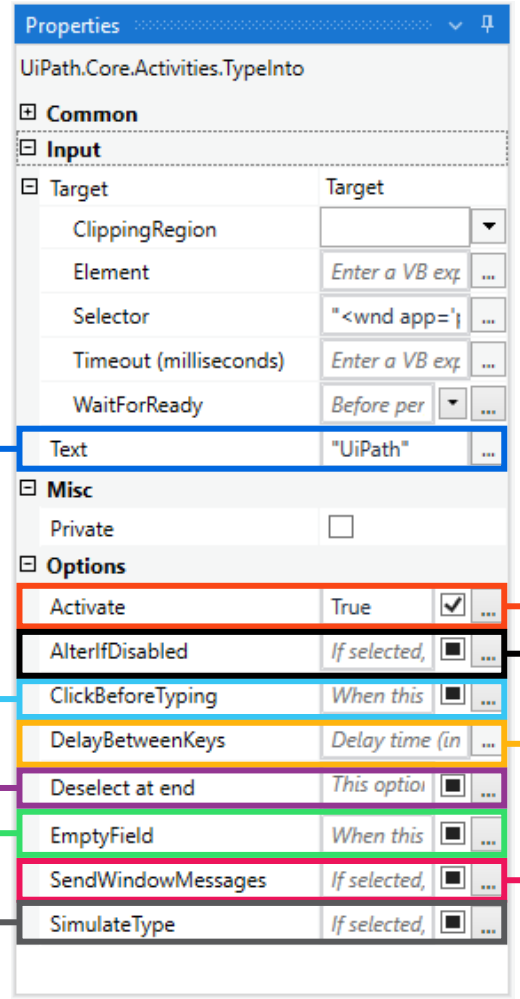
CursorMotionType:
Instant or Smooth

KeyModifiers: None, Alt, Ctrl, Shift, and/or Win

SimulateClick: If selected, it simulates the **click** by using the technology of the target application

Properties of Type Into

The properties of Type Into are:



Text: The text to be written in the specified UI element

ClickBeforeTyping: Click on the UI element

Deselect at end: Adds a Complete event after the text entry

EmptyField: Empty the UI element before typing

SimulateType: If selected, it simulates the **type** using the technology of the target application

Activate: Activate the UI element to be typed into

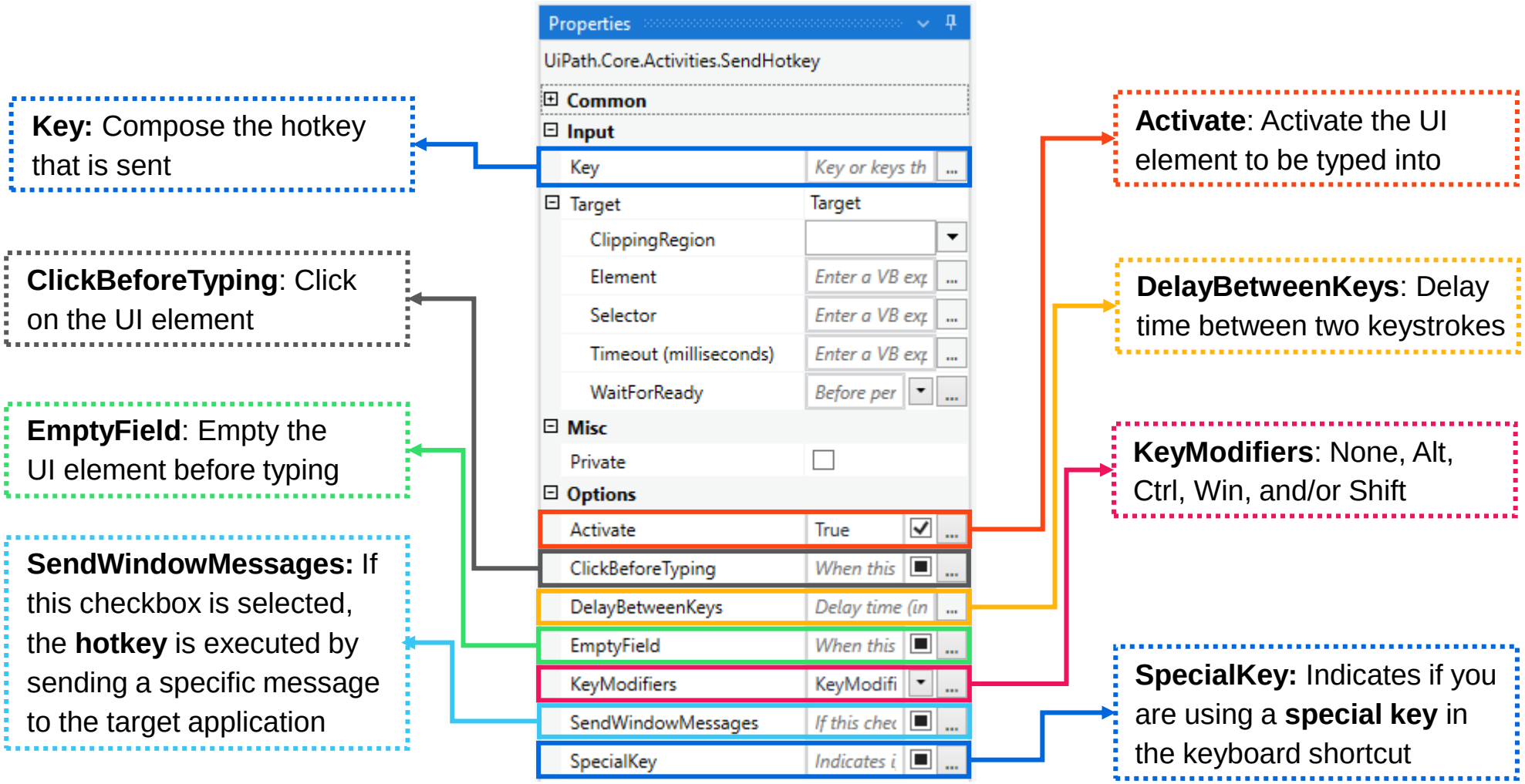
AlterIfDisabled: If selected, the simulated click is executed even if the specified UI element is disabled or read only

DelayBetweenKeys: Between each typed key

SendWindowMessages: If selected, the **type** is executed by sending a specific message to the target application

Properties of Send Hotkey

The properties of Send Hotkey are:



UiPath Academy

Chapter 6:

User Interface Automation with the Modern Experience in Studio

Input Methods and Input Activities Part 2

Input Activities

Here's a brief description of some important input activities in UiPath modern design:

1. **Click** - This activity is used to simulate a mouse click on a UI element. It can also be used to double-click, right-click, or click and hold on an element.

2. **Type Into** - This activity is used to simulate typing text into a text box, field, or any other UI element that accepts input.

3. **Select Item** - This activity is used to select an item from a drop-down menu or list box.

4. **Check/Uncheck** - This activity is used to check or uncheck a checkbox or radio button.

5. **Hover** - This activity is used to simulate a mouse hover over a UI element.

6. **Navigate Browser** - This activity is used to navigate to a specific URL or web page.

7. **Keyboard Shortcuts** - This activity is used to send keyboard shortcuts to the application.

Input Activities Resources

Topic	Link
Modern Activities	https://docs.uipath.com/activities/docs/ui-automation-modern-activities
Click Activity	https://docs.uipath.com/activities/docs/n-click
Type into Activity	https://docs.uipath.com/activities/docs/n-type-into
Hover Activity	https://docs.uipath.com/activities/docs/n-hover
Select Item Activity	https://docs.uipath.com/activities/docs/n-select-item
Check/Uncheck Activity	https://docs.uipath.com/activities/docs/n-check
Drag and Drop Activity	https://docs.uipath.com/activities/docs/n-drag-and-drop

Output Methods

Output methods are used to extract data from UI elements, such as text or images.

UiPath Studio offers **three main output** methods: **Full Text, Native Text, and OCR**.

Full Text extracts all the visible and hidden text from a UI element.

Native Text extracts only the visible text

OCR is used for non-text data (Images)

The output method you choose can depend on factors such as the speed required, whether the automation needs to run in the background, and whether text needs to be extracted from a hidden interface.

Output Methods

Output actions are used to **extract data**, generally in the form of text**, from UI elements**.
Output methods are how output actions extract data from UI elements.

Modern UI automation design typically employs **three main output methods**

Output method	
Full Text	❖ **Default method** and good enough in most cases ❖ It is the **fastest** ❖ It can **extract hidden text** ❖ It has **100% accuracy** ❖ Can **work in the background** ❖ Doesn't support virtual environments
Native Text	❑ **Compatible with applications that use Graphics Design Interface (GDI),** the Microsoft API used for representing graphical object ❑ **Doesn't extract hidden text** ❑ **Cannot work in the background** ❑ Doesn't support virtual environments
Optical Character Recognition (OCR)	➢ Only output method that works with virtual environments and with "reading" text from images. ➢ Its technology relies on recognizing each character and its position ➢ Cannot work in the background ➢ Cannot extract hidden text ➢ Its speed is by far the lowest

Differences between the Output Methods

Output Method	FULL TEXT	NATIVE	OCR
Default method and Compatibility	It is the **Default method** and good enough in most cases.	Compatible with applications that use **Graphics Design Interface (GDI)**, the **Microsoft API is** used for representing graphical objects.	OCR is the only output method that works with **virtual environments** and with **"reading" text from images**. Relies on recognizing each character and its position.
Automation Speed	**Fastest**	Somewhat **slower than Full Text.**	By far the **slowest**
Accuracy	**100%** accuracy.	**100%** accuracy on the **applications that support GDI.**	Varies from **one text to** another, by **changing settings** we can improve results.
Running in Background	**Works** in the background	**Cannot** work in the background	**Cannot** work in the background.
Hidden Text	**Can** extract hidden	**Cannot** extract hidden text	**Cannot** extract hidden text
Virtual Environment	**Doesn't** support virtual environments.	**Doesn't** support virtual environments.	**Works** with virtual environments and with **"reading" text from images**.
Text position and Formatting	**Doesn't capture** text position and formatting.	**Can** extract the text position and formatting (including text color)	Like the **Native** method, it also captures the text position.
Other	The method offers the **option to ignore the hidden message** and capture only the visible text	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 all the others	**Two default engines** that can be used alternatively: **Google Tesseract, Microsoft MODI.** There are additional OCR engines that can be installed free of charge (such as Omnipage and Abbyy Embedded) or paid (IntelligentOCR offered by Abbyy)

Output Methods Resources

Output or Screen Scraping Methods

https://docs.uipath.com/studio/v2021.10/docs/output-or-screen-scraping-methods

'Get Text' Activity

Extracts and copies the text from a UI element.
It should be **added inside a Use Application/Browser Activity**

This Activity supports the following scraping methods:

1. **Default**: tries all three scraping methods below, and the first one to return a text is used

2. **Text attribute**: uses the "text" attribute of the UI element.

3. **Fulltext**: offers the option to Ignore hidden text, which can be activated by selecting its respective check box.

4. **Native**: this method enables you to allow formatting and retrieve the screen coordinates of the words by selecting the check boxes for each of these features.

The Extraction Preview Wizard
Get Text Activity provides the Extraction Preview wizard.
This wizard enables you to choose the suitable output method based on the result displayed in the Extraction Preview wizard.

Output methods

Output methods are a way to extract text from a UI element, including

- full text
- native
- OCR
- text attributes

Get Text activity : The Get Text activity is used to extract the text from a UI element and store it in a variable.

Scraping Method : Using the "Scraping Method" property of the Get Text activity, you can specify how the text should be extracted from an element

UiPath Academy

Chapter 6:

User Interface Automation with the Modern Experience in Studio

Output Methods and Output Activities Part 3

Output Methods Resources

Get Text Activity

https://docs.uipath.com/activities/docs/n-get-text

Exercise - Extract Invoice Details Using Get text activity

Use-case/ problem statement - Extract Invoice Details

Develop a process using the Modern Design Experience that automates the extraction of invoice details.
The process should be capable of performing the following actions:

1. Read an input Excel file with the invoice numbers.
2. Sign in to the ACME website. (https://acme-test.uipath.com)
3. Navigate to the invoice search page.
4. For each invoice number in the Excel file, search for the invoice number.
5. Extract the invoice details, including Vendor Tax ID, Invoice Item, Total, and Date.
6. Write the extracted invoice details to an Excel file stored in the project folder.

Note: The invoice number on the ACME website keeps changing from one login to another

App/Web Recorder

App/Web Recorder allows users to capture their on-screen actions and convert them into activities in Studio, which saves them time in business processes by automating user interfaces and saving time.

- The App/Web Recorder captures our actions as we perform them on the screen and generates a Use Application/Browser Activity with a series of activities inside it based on our actions
- When using the recorder, we can choose the Input Methods as per the requirements
- If Full Configuration of Targets and Saved Values mode is enabled, it'll ask for confirmation as you indicate the UI element while recording
- After saving the recording, we can edit the generated activities at any time

UiPath Academy

Chapter 6:

User Interface Automation with the Modern Experience in Studio

Using the App/Web Recorder

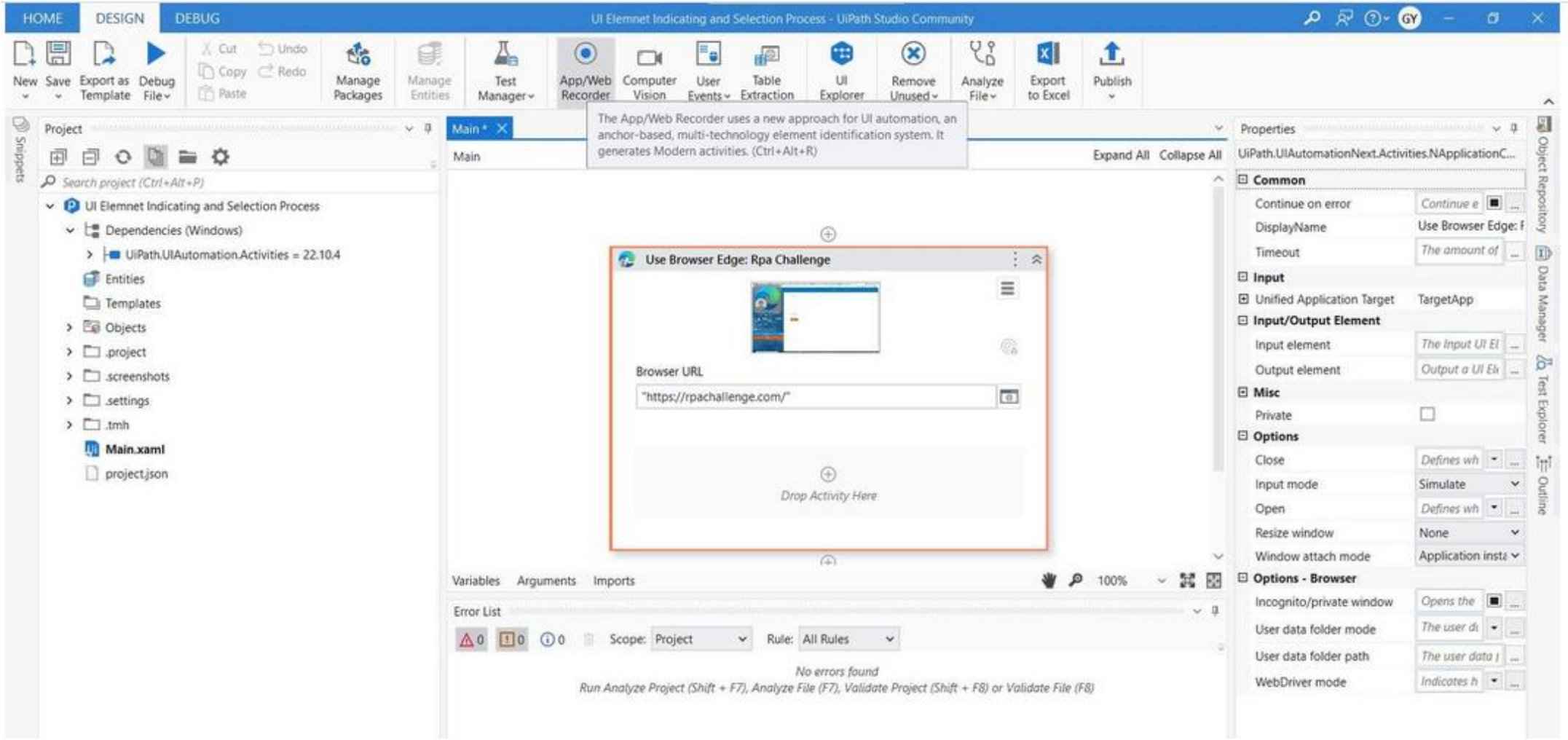
App/Web Recorder

The following table shows which actions are automatically recorded and the activities that are generated for them, as well as which actions must be manually selected before recording:

Automatically Recorded Actions	Actions That Must Be Selected Before Recording
➢ Clicking on buttons, links, icons, or images: generates a **Click activity**. ➢ Typing text in a text box or text area: generates a **Type Into activity**. ➢ Selecting or clearing a check box: generates a **Check/Uncheck activity**. ➢ Sending keyboard shortcuts using your keyboard: generates a **Keyboard Shortcuts** activity. ➢ Selecting an item from a drop-down: generates a **Select Item** activity.	➢ Copying text using the **Get Text** activity. ➢ Hovering over an element using the **Hover** activity. ➢ Highlighting an element using the **Highlight** activity.

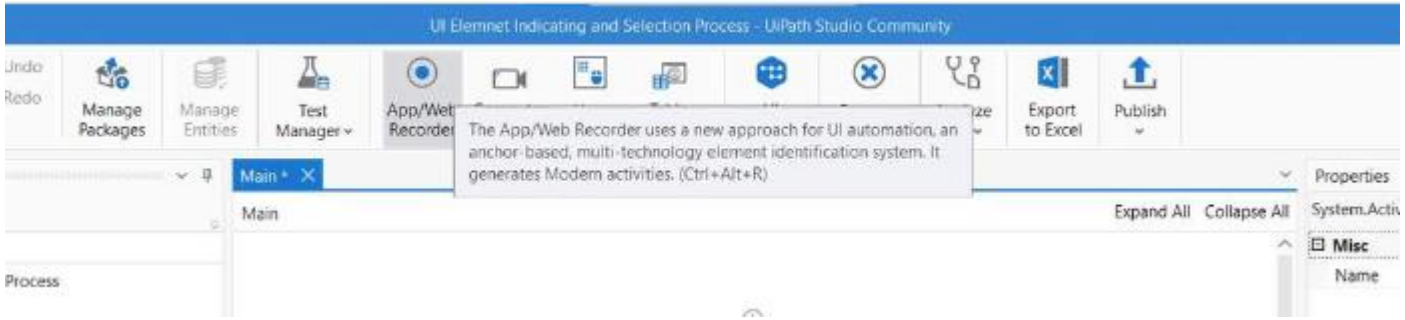
App/Web Recorder with a Use Application/Browser activity

You can only record actions on elements **within that application or browser window**.
This is useful when you want to create targeted automation workflows for specific applications or websites.



App/Web Recorder without a Use Application/Browser activity

You can capture actions **on all applications that are open on your computer**.
This is helpful for automating complex tasks that involve multiple applications,
as you can record actions across different applications and combine them into a single workflow.



App/Web Recorder Resources

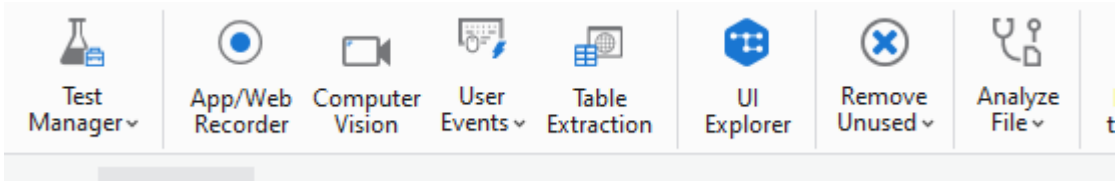
App/Web Recorder

https://docs.uipath.com/activities/docs/app-web-recorder

Table Extraction

Table Extraction automates the process of

- Extracting structured data from webpages and applications
- Converting it into a DataTable for efficient processing
- Reducing errors
- Increasing the accuracy of automation processes
- Increasing speed of automation processes



We can access the 'Table extraction' recorder from the design ribbon menu **when modern design is enabled** for the project.
The Table extraction recorder Wizard generates an **'Extract Table Data'** Activity in the workflow which outputs a 'Data Table'

UiPath Academy

Chapter 6:

User Interface Automation with the Modern Experience in Studio

Table Extraction

Table Extraction Wizard

The Table Extraction wizard provides a powerful tool for extracting structured data from tables by defining columns and selecting similar elements.

- **Add new column**: we can define a series of similar elements to be extracted as a table column
- **Extract text or extract URL**: allows to extract either the text or the URL associated with the selected data
- We can refine selection by selecting similar elements to the one we initially indicated
- **Column setting option**: it's represented by a cogwheel next to each column and allows to access this option to make individual edits or deletions to customize the final table to fit our needs.
  - Delete option: Delete the extracted column.
  - Parse data as: this option allows to select between three main types of data: **Text, Numbers, and Date & Time** and convert data from one type to another
  - Sort: this option allows to sort the data in columns by selecting 'Ascending' or 'Descending', depending on the requirement.
  - Sample after parsing: this option allows to preview the parsed data for a selected column.
- **Extract data from multiple pages**: by toggling this button to yes, you can indicate the next page navigation button or link to extract data from multiple pages.
- **Settings Section**: it allows to set a limit for extraction by either 'Max Pages' or 'Max Rows', and by default, it's set to 'No Limit.'

Table Extraction

Table Extraction

https://docs.uipath.com/activities/docs/table-extraction

Exercise – Table Extraction

Use-case/ Problem Statement - Table Extraction

Create a process which does the following actions using the Modern Design Experience:

1. Opens ACME web page (https://acme-test.uipath.com)
2. Asks the user for the Email and Password.
3. Enter the values provided into the fields.
4. Click the Login button.
5. Navigates to the Work Items page.
6. Extracts the data from the Work Items Table: WIID, Description, Type, Status and Date.
7. Stores the data in an Excel file using a Workbook Write Range activity.

Output: An Excel file stored in the project folder, containing the Work Items Table information.

Static Selectors / Descriptors

A Descriptor is **a target and anchor pair**

A Dynamic Selector/ Descriptor :

Is one that may change based on the state or content of the application.
Dynamic selectors often use wildcards or variables to accommodate changes in the target element.

Static Selector/ Descriptor :

A Static selector, in contrast, is one that **remains constant** regardless of changes in the application's state or content.
It **does not rely on variables or wildcards and remains the same every time it is used**.

Dynamic Descriptors

A Dynamic Descriptor :

Is one that may change based on the state or content of the application.
Dynamic selectors often use wildcards or variables to accommodate changes in the target element.

There are 3 main ways to create / modify dynamic descriptors

- Dynamic Text Target Option
- Fine-tuning Descriptors using Wildcards
- Fine-tuning Descriptors using Variables

Dynamic Text Target Option

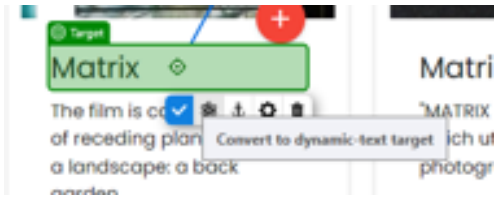
The dynamic text target option is a feature that enables automation to dynamically identify and interact with UI elements whose **text content changes** based on different scenarios or inputs.

This option allows the automation to handle **cases where the element's text changes frequently**, making it difficult to identify and interact with the correct UI element using a very specific descriptor.

Allows automation to dynamically identify and interact with UI elements whose text content changes based on different scenarios or inputs.

Doesn’t use wildcards or variables, just ignores the text & images in the target selectors for the element

For the activity (e.g. Click Activity) **Edit Target ->**
Hover Over Target ->
Select “Convert to Dynamic Text Target” in the floating menu



UiPath Academy

Chapter 4:

Descriptors in Studio

Using Dynamic Text Target Option

Fine-tuning Descriptors using Wildcards

A wildcard is a character or a set of characters that can be used to represent one or more characters in a string or a file path. It allows for more flexible and dynamic automation by allowing the automation program to match and manipulate multiple strings or files that share a similar pattern or naming convention.

The asterisk (*) is a wildcard that can represent **any sequence of characters (0 or more).**
So, if we have a list of files that are named "Document1", "Document2", "Document3", and so on, we can use the wildcard at the end of the word "Document" to match all of them. i.e**. “Document*“**

The Question Mark (?) represents a **single character.**
For example, if we have a list of files that are named "Report01.txt", "Report02.txt", "Report03.txt", and so on, we can use the wildcard "Report??.txt" to match all of them.
Note: this would fail for "Report1.txt", as it requires exactly 2 characters

We can modify the selector for a given UI element and replace a fixed text segment with either an **asterisk** or a **question mark** to get the desired result.

We can also use the “Repair” option in the **Selector Editor** to automatically update selectors with the appropriate wildcards

UiPath Academy

Chapter 4:

Descriptors in Studio

Fine-Tuning Descriptor Using Wildcards

Fine-tuning Descriptors using Variables

Variables provide the flexibility to make selectors more dynamic and adaptable to different scenarios.
When selectors contain variables, the values of these variables can be updated during runtime based on the input or data being processed.

This allows the same strict or fuzzy selector to be used for different inputs or data sets without having to create a new selector each time.

So rather than using wildcards, we can use variables in the target for the element

UiPath Academy

Chapter 4:

Descriptors in Studio

Fine-Tuning Descriptor Using Variables

Fine-tuning Descriptors

- Fine-tuning descriptors is crucial for improving reliability and adaptability in automation workflows
- Refining and adjusting descriptor configurations enhance accuracy and overcome challenges like unreliable selectors and dynamic UI elements.
- While manual editing of descriptors is generally not recommended, rare cases may require fine-tuning for the Unified targeting method.
- Techniques such as dynamic text targeting, wildcards, the repair function, variables, and adjusting image accuracy are effective for fine-tuning descriptors.
- Wildcards, specifically the asterisk (*) and question mark (?), play a significant role in selector customization. The asterisk (*) is used to replace zero or more characters, while the question mark (?) replaces a single character.
- Variables and arguments can be used in the Strict Selector or Fuzzy Selector to identify the desired target, either by creating new variables/arguments or selecting from existing ones in the context menu.
- The matching accuracy of the Fuzzy Selector can be adjusted using the Accuracy slider to optimize its performance in selecting targets.
- The Image targeting method can be fine-tuned by adjusting the Image Accuracy using its corresponding slider. This allows for precise control over the accuracy of image-based targeting, ensuring reliable identification of target elements based on visual cues.

Dynamic Descriptors

Topic	Link
Advanced Descriptor Configuration	https://docs.uipath.com/activities/docs/advanced-descriptor-configuration#selectors
Dynamic Selectors	https://docs.uipath.com/studio/v2022.10/docs/dynamic-selectors
About Wildcards	https://docs.uipath.com/studio/v2022.10/docs/selectors-with-wildcards
About Configuring the Descriptor	https://docs.uipath.com/activities/other/latest/user-guide/advanced-descriptor-configuration#configuring-the-descriptor

Exercise – Fine-tuning the Descriptor

Use-case/ Problem Statement – Get IMDB Movie Rating

1. Implement a user input prompt to capture a movie title from the user.

2. Use the movie title as input to search for the movie on the movie search page of the IMDB website. (https://www.imdb.com)

3. Extract the IMDB rating of the searched movie from the website.

4. Display the retrieved IMDB rating through a message box.

Evaluate Descriptors - Show All Matches

The "Show all matches" feature helps to identify all the possible matches for the selected element within the automated application.

This option is useful for debugging UI descriptors or selections, allowing you to ensure the accuracy of your automation setup.

Evaluate Descriptors : Show All Matches

The 'Show all matches' feature allows us to quickly identify all the matching UI elements for any targeted method. To use this feature, simply **click on the eye icon** next to each targeting method.

The validation process provides the correct functioning of all combinations of search methods for the target.

The Validate button on the Selection screen is used to validate your selection.

Evaluate Descriptors – Target Element Validation

Target element validation allows you to assess the effectiveness of the current selection in accurately identifying the target. It inspects the generated descriptors for the target using all the selected methods and anchors to provide a validation result, which is displayed at the top of the Selection Options window. If adjustments are required, a message will be shown with guidance on how to improve the selection.

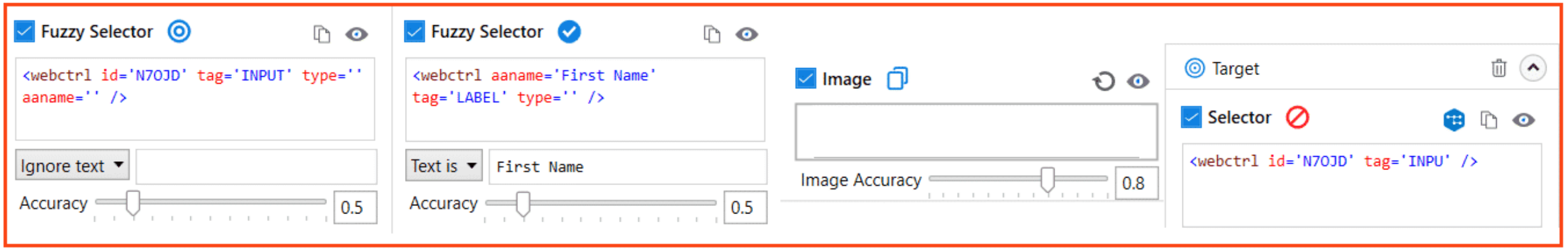
In the Advanced Settings section of the Selection Options window, one of the following icons represents the performance of each method.

Method was first to identify element

Method successfully identified element

Method identified duplicates. At least one Anchor working

Method failed to identify an element



UiPath Academy

Chapter 4:

Descriptors in Studio

Target Element Validation

Evaluate Descriptors : Target Element Validation

The Validate button on the Selection screen is used to validate your selection.

During validation, **duplicates in the target application are highlighted**.
Icons are displayed next to the targeting methods to indicate the status of each method.

- **Blue bullseye**: signifies the fastest targeting method.
- **Checkmark**: indicates that the targeting method worked but wasn't the fastest.
- **Duplicate icon**: represents the presence of duplicates.
- **Stop sign icon**: appears when a targeting method fails to identify any elements.

Target Element Validation

Topic	Link
About Validating the Descriptor	https://docs.uipath.com/activities/other/latest/user-guide/advanced-descriptor-configuration#validating-the-descriptor
Screen Selection	https://docs.uipath.com/activities/docs/selection-options
Descriptor Configuration	https://docs.uipath.com/activities/other/latest/user-guide/n-click