LECTURE 06. SELECTORS

Robotic Process Automation
[07 November 2023]

Elective Course, 2023-2024, Fall Semester

Camelia Chisăliţă-Creţu, Lecturer PhD
Babeş-Bolyai University

Acknowledgements

This course is presented to our Faculty with the support of UiPath Romania.

UiPath
Robotic Process Automation

Contents

Selectors. Motivation. Details

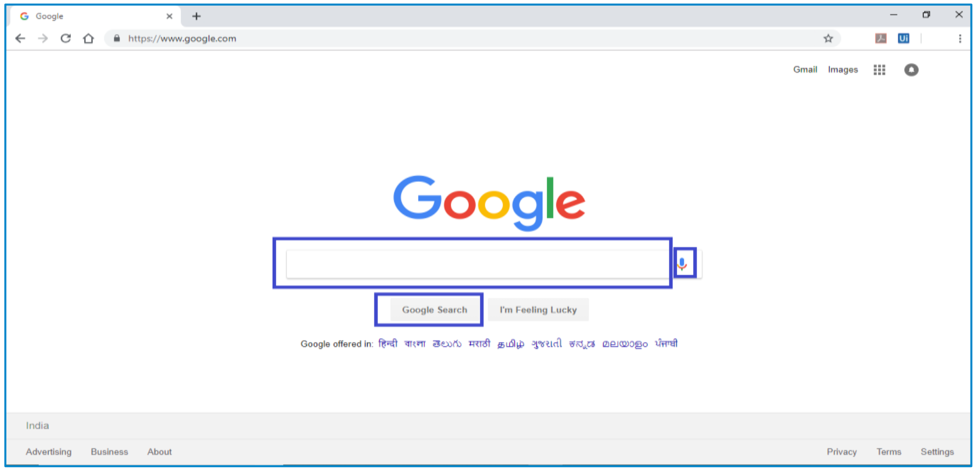
- UI interaction requires the use of
  - **buttons, test fields, drop-down list, windows** and
  - **advanced features** which require combination of **selectors**;
- **Selectors** indicate
  - the **address** of an element in UiPath Studio;

- characteristics:
  - they are a fundamental part of UiPath Studio being used to recognize the objects on the screen;
  - they allow to **uniquely identify UI elements** on the screen, among multiple applications;
  - **they are described as XML strings that consists of properties that uniquely identify a specified element.**

UI Element. Details

- **UI Elements** indicates
  - **all graphical user interface pieces** that construct an application;

- E.g.:
  - a search bar (Text Field);
  - the Search Button;
  - a microphone shaped image for audio search.



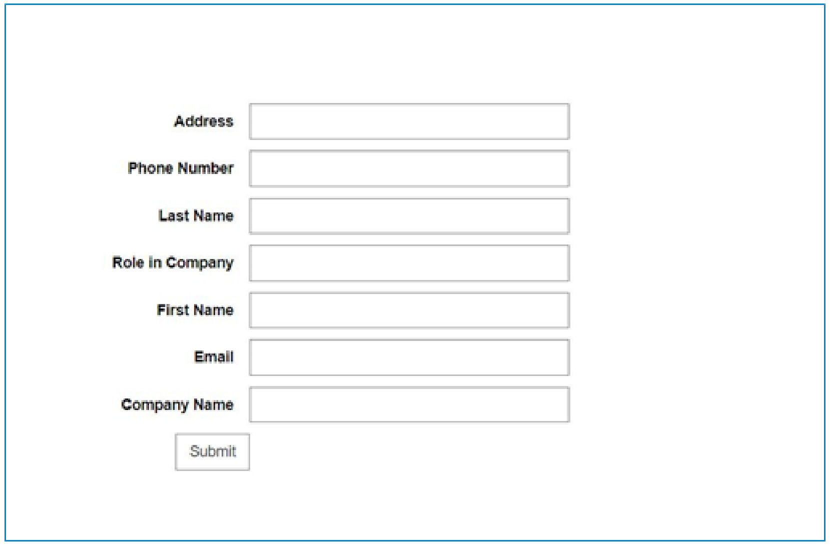
UI Interface. Types

- **UI Interface** is
  - a container that holds all UI elements that construct an application.

- **Static User Interface**
- **Dynamic User Interface**

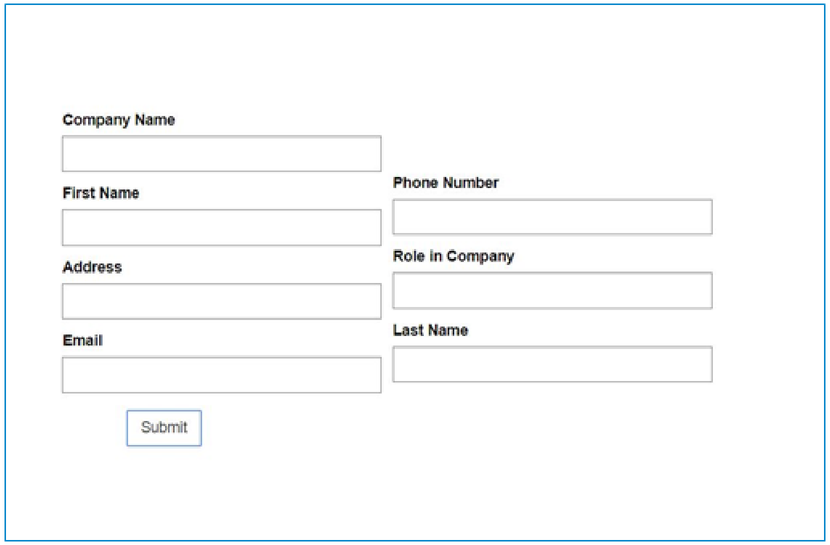
UI Interface Types. Details

- **Static** Interface Scenario



- the UI element named "Address" will always be found at this exact pixel coordinate in the left hand side of the web page;
- if **the layout does not change**, the selector will remain valid throughout its operations.

- **Dynamic** Interface Scenario



- **the layout is changed** although it contains the same UI elements;
- the selector from the previously identified would become invalid as the pixel positioning of the "Address" element has changed.

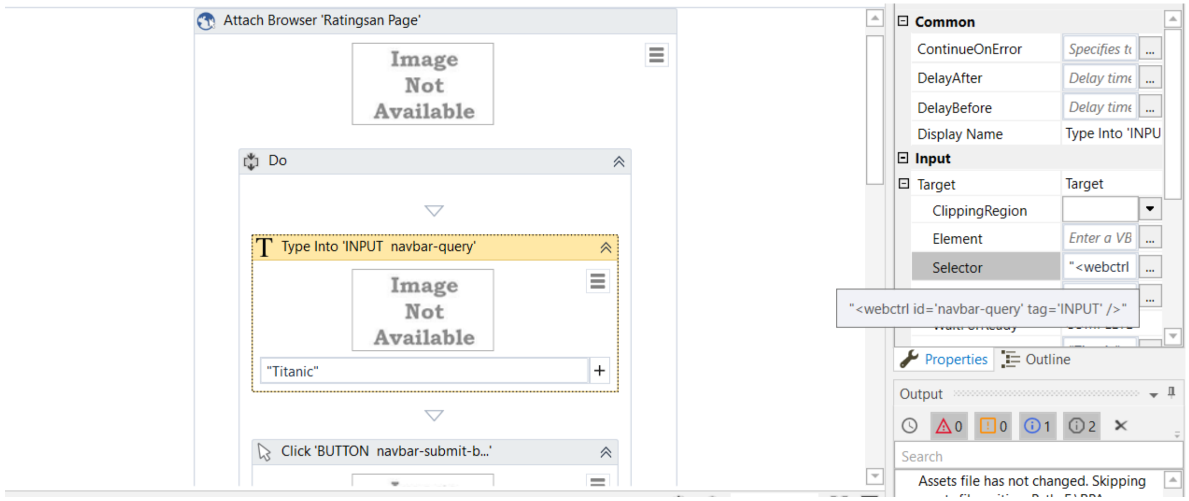
Selectors in UI Interface. Details

- **selectors can store the attributes** and **characteristics of a GUI element along with all its parents in the shape of an XML fragment**;
- most of the time, selectors are automatically generated by UiPath Studio and no additional input is required from the user, especially if the automated application is a static UI.

Selector Editor. Details

- **Selector Editor** window enables
  - **to see** the automatically generated selectors and **to edit** their attributes;

- Steps to access the Selector Window:
  - access **Workflow Designer** panel;
  - **click** on the **activity** that exposes the **selector** to be edited;
  - in the **Properties** panel, **click** on the option **TARGET**.



ContinueOnError	Specifies t
DelayAfter	Delay time
DelayBefore	Delay time
Display Name	Type Into 'INPU
Target	Target
ClippingRegion	
Element	Enter a VB
Selector	"<webctrl

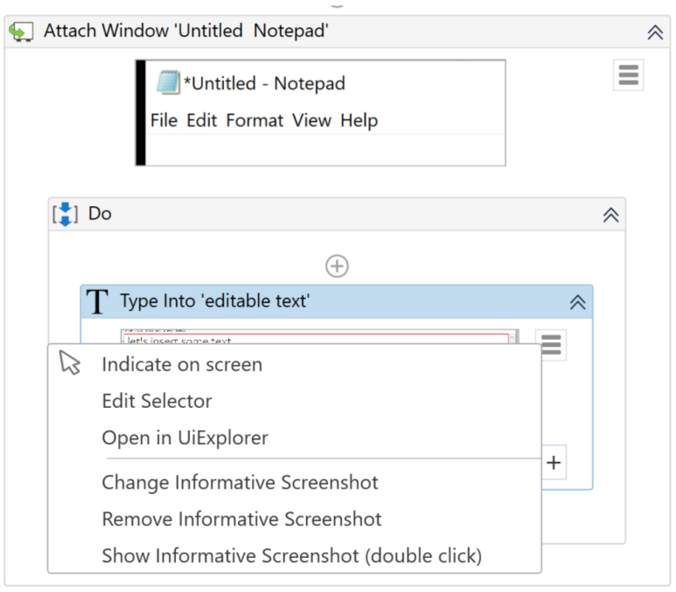
"<webctrl id='navbar-query' tag='INPUT' />"

Assets file has not changed. Skipping

see **Demo1A–Selector Editor**

Selector Editor. Details (2)

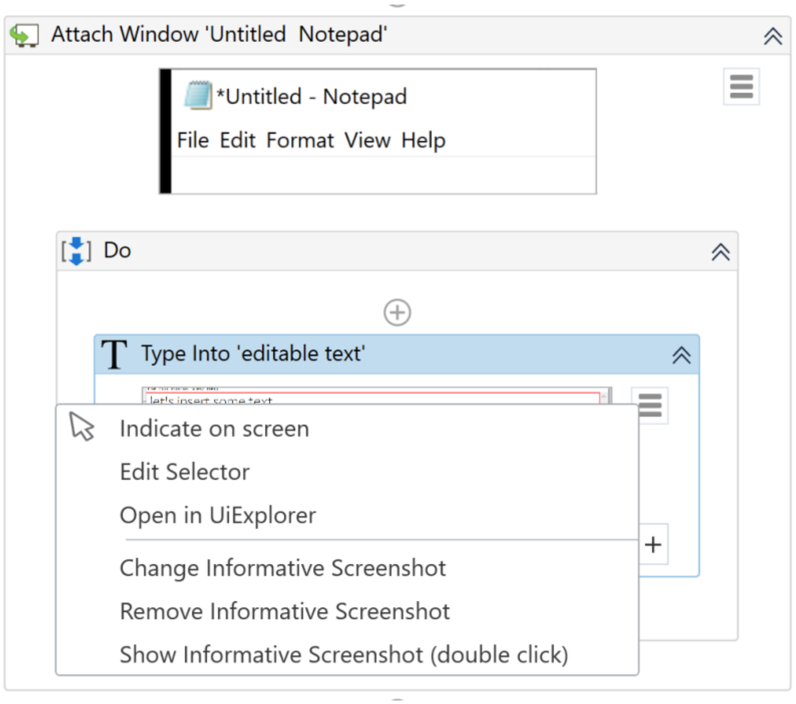
- Steps to access the **Selector Window**:
  - *from the activity itself:*
    - **click** the **hamburger menu button** placed o the activity;
    - **click** on **Edit Selector**;
  - from the **Properties** panel:
    - **click** on the **…** button next to the **Selector** property.


Attach Window 'Untitled Notepad'
*Untitled - Notepad
File Edit Format View Help
Do
Type Into 'editable text'
Indicate on screen
Edit Selector
Open in UiExplorer
Change Informative Screenshot
Remove Informative Screenshot
Show Informative Screenshot (double click)

see **Demo1A–Selector Editor**

Selector Editor. Properties (1)

- Options of the **Hamburger Menu**:
  - **Indicate on Screen:**
    - it simplifies the automation where the target element is changed;
    - *the selectors are automatically created;*
  - **Edit Selector:**
    - it allows to edit the selectors already created;
    - if the selectors do not work due to any reason, *they can be edited by using this option;*
  - **Open in UI Explorer:**
    - it allows to *edit the selector by using the UI Explorer tool*;



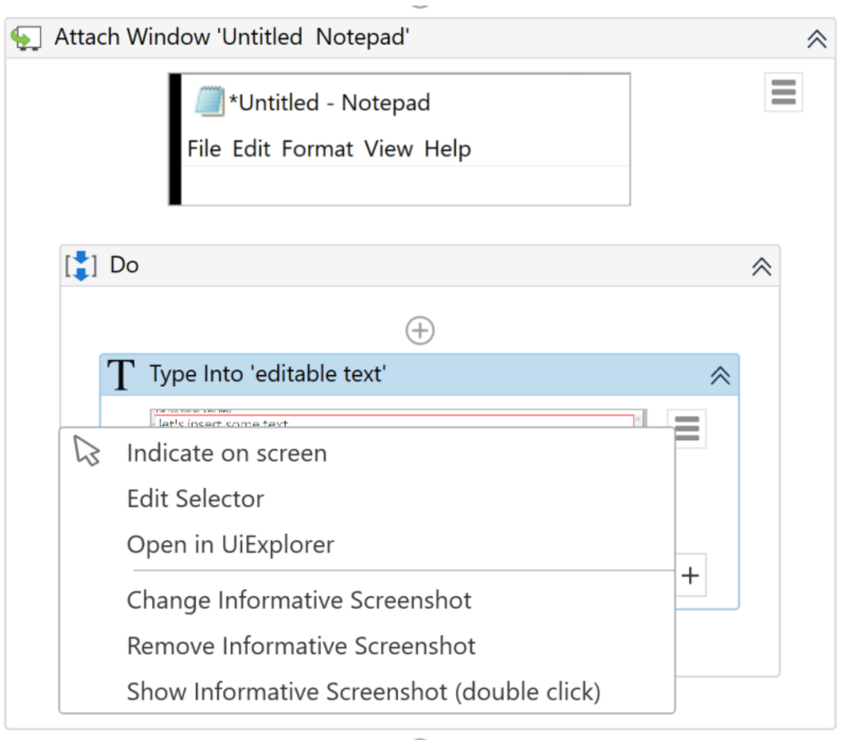
Attach Window 'Untitled Notepad'
*Untitled - Notepad
File Edit Format View Help
Do
Type Into 'editable text'
Indicate on screen
Edit Selector
Open in UiExplorer
Change Informative Screenshot
Remove Informative Screenshot
Show Informative Screenshot (double click)

see **Demo1A–Selector Editor**

Selector Editor. Properties (2)

- Options of the **Hamburger Menu** :
  - **Change Informative Screenshot:**
    - when any element is selected,

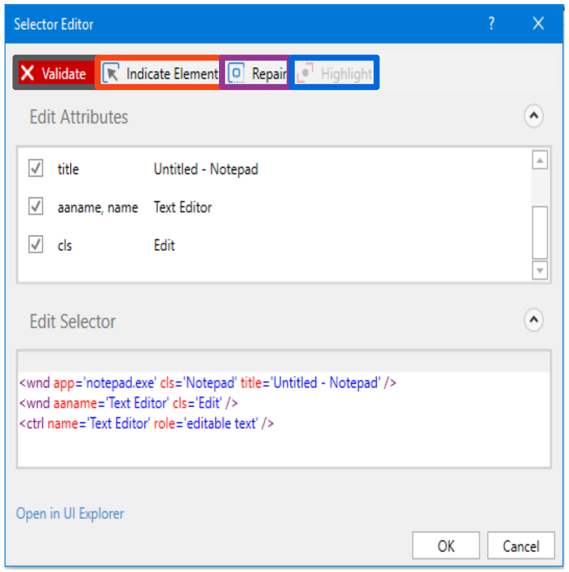
    a sample screenshot is created;
    - in case the user wants to change the screenshot, this option can be used.
  - **Remove Informative Screenshot:**
    - to be used to delete an auto-generated screenshot;
  - **Show Informative Screenshot (Double Click):**
    - to be used to show the informative screenshot or image.


Attach Window 'Untitled Notepad'
*Untitled - Notepad
File Edit Format View Help
Do
Type Into 'editable text'
let's insert some text
Indicate on screen
Edit Selector
Open in UiExplorer
Change Informative Screenshot
Remove Informative Screenshot
Show Informative Screenshot (double click)

see **Demo1A–Selector Editor**

Selector Status. Details

- the **Selector Status** can be viewed in the **Selector Editor** window;
- the **Selector Status** colors:
  - **valid:** green;
  - **to be validated:** grey;
  - **invalid:** red;
  - **changed and not validated yet:** yellow.



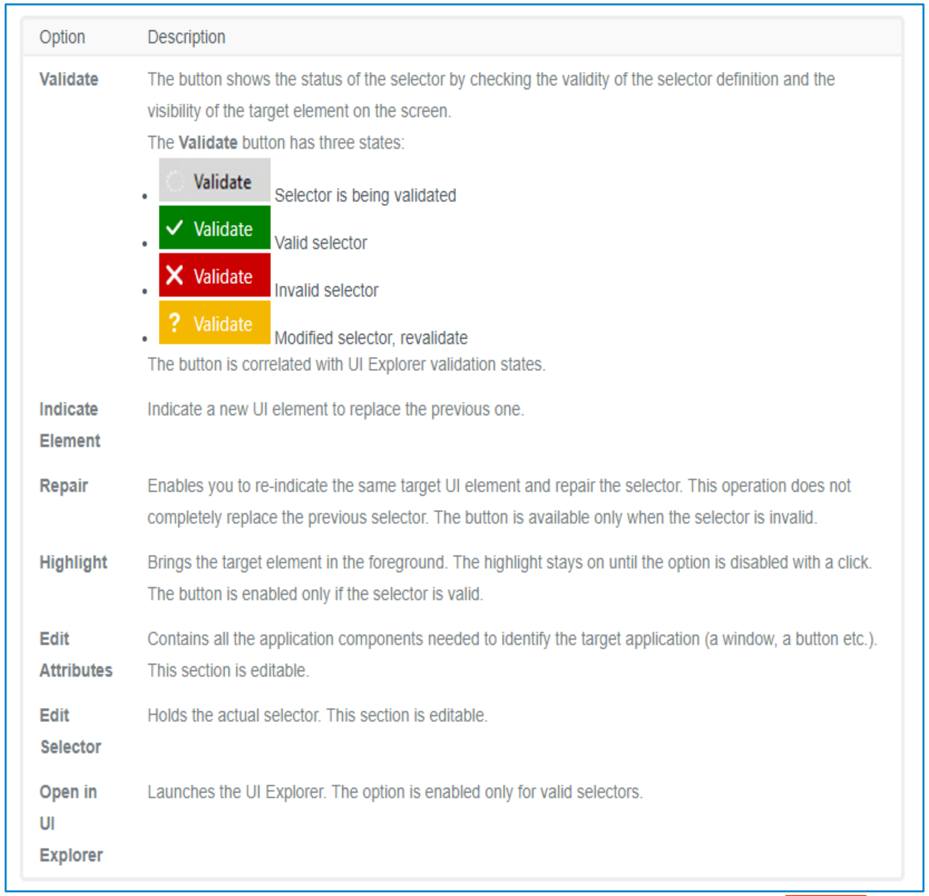
Selector Editor

Edit Attributes

✓	title	Untitled - Notepad
✓	aaname, name	Text Editor
✓	cls	Edit

Edit Selector

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



Option	Description
Validate	The button shows the status of the selector by checking the validity of the selector definition and the visibility of the target element on the screen. The **Validate** button has three states: • Validate – Selector is being validated • ✓ Validate – Valid selector • ✗ Validate – Invalid selector • ? Validate – Modified selector, revalidate. The button is correlated with UI Explorer validation states.
Indicate Element	Indicate a new UI element to replace the previous one.
Repair	Enables you to re-indicate the same target UI element and repair the selector. This operation does not completely replace the previous selector. The button is available only when the selector is invalid.
Highlight	Brings the target element in the foreground. The highlight stays on until the option is disabled with a click. The button is enabled only if the selector is valid.
Edit Attributes	Contains all the application components needed to identify the target application (a window, a button etc.). This section is editable.
Edit Selector	Holds the actual selector. This section is editable.
Open in UI Explorer	Launches the UI Explorer. The option is enabled only for valid selectors.

see **Demo1A–Selector Editor**

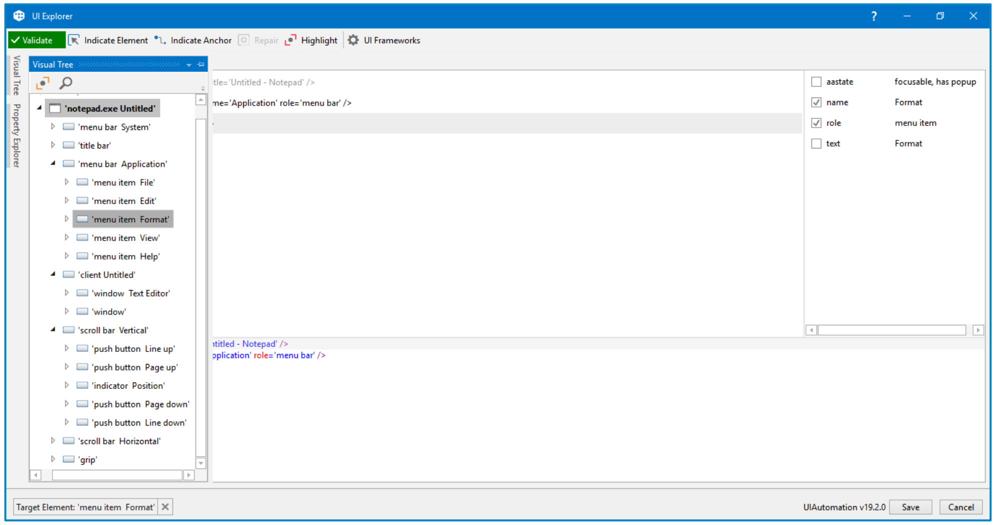
Demo 1A. Selector Editor

- **Use the Basic recorder to create a process that performs the following actions:**
  - **1. *open* the Notepad Application;**
  - **2. *type* in Notepad "Let's see some selectors at work today in Notepad!";**
  - **3. *change* the Font to 'Corbel';**
  - **4. *select* the Font Style to 'Bold';**
  - **5. *set* the Font Size to 20;**

- **Perform the following tasks:**
  - Inspect in **Selector Editor** window the selectors associated to the UI elements used during automation;
- **Discuss the followings:**
  - *What tags are available?*
  - *What attributes do they have?*
  - *Can we change the attributes?*
  - *Are all valid selectors?*

UI Explorer. Details

- **UI Explorer** is
  - **a tool** that provides flexibility to customize the selector;
- ways to access **UI Explorer**:
  1. in the **Design** panel;
  2. click on the **Hamburger Menu button**, using **Edit Selector** option and clicking the **open in UI Explorer** option in the **Selector Editor** window;
  3. from **Home** -> **Tools** -> **UI Explorer** tool option.



```
tle='Untitled - Notepad' />
me='Application' role='menu bar' />
```

'notepad.exe Untitled'
- 'menu bar System'
- 'title bar'
- 'menu bar Application'
  - 'menu item File'
  - 'menu item Edit'
  - 'menu item Format'
  - 'menu item View'
  - 'menu item Help'
- 'client Untitled'
  - 'window Text Editor'
  - 'window'
- 'scroll bar Vertical'
  - 'push button Line up'
  - 'push button Page up'
  - 'indicator Position'
  - 'push button Page down'
  - 'push button Line down'
- 'scroll bar Horizontal'
- 'grip'

☐ aastate	focusable, has popup
☑ name	Format
☑ role	menu item
☐ text	Format

```
ntitled - Notepad' />
pplication' role='menu bar' />
```

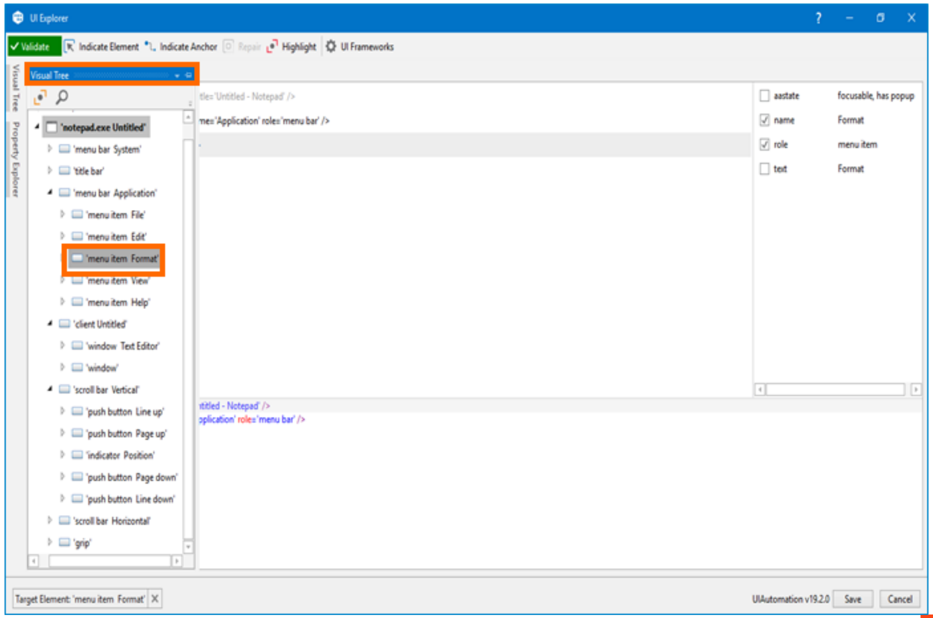
Target Element: 'menu item Format'

UIAutomation v19.2.0

see **Demo1B–UI Explorer**

UI Explorer. Visual Tree

- **Visual Tree** is
  - **a list of containers** from the parent container to the **Target** UI element;
- it is located on the left-hand side of **UI Explorer**;
- E.g.:
  - to change the format of text written in Notepad, click on “menu item Format” button; in this case the defined interaction in a tree will look like:
    - **Container 1:** Notepad;
    - **Container 2:** Menu bar;
    - **Container 3:** Font.



'notepad.exe Untitled'
- 'menu bar System'
- 'title bar'
- 'menu bar Application'
  - 'menu item File'
  - 'menu item Edit'
  - 'menu item Format'
  - 'menu item View'
  - 'menu item Help'
- 'client Untitled'
  - 'window Text Editor'
  - 'window'
- 'scroll bar Vertical'
  - 'push button Line up'
  - 'push button Page up'
  - 'indicator Position'
  - 'push button Page down'
  - 'push button Line down'
- 'scroll bar Horizontal'
- 'grip'

```
tle='Untitled - Notepad' />
me='Application' role='menu bar' />
```

☐	aastate	focusable, has popup
✓	name	Format
✓	role	menu item
☐	text	Format

```
ntitled - Notepad' />
pplication' role='menu bar' />
```

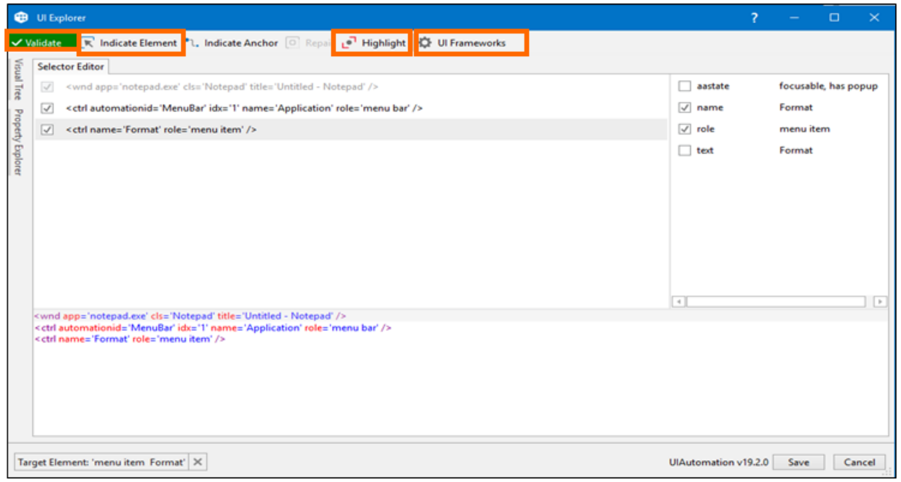
Target Element: 'menu item Format'

UiAutomation v19.2.0

see **Demo1B–UI Explorer**

UI Explorer. Features

- Features included in **UI Explorer** tool:
  - **Validate:**
    - it has different colors to indicate if a selector is correct or not; *this is already defined;*
  - **Indicate Element:**
    - to indicate a particular UI element; *this is already defined;*
  - **Highlight:**
    - it is used to highlight the UI Element that is currently edited;
  - **UI Frameworks:**
    - is used when individual elements are not recognized;
    - values: *Default, Active Accessibility, UI Automation*.



```
<wnd app='notepad.exe' cls='Notepad' title='Untitled - Notepad' />
<ctrl automationid='MenuBar' idx='1' name='Application' role='menu bar' />
<ctrl name='Format' role='menu item' />
```

☐ aastate	focusable, has popup
☑ name	Format
☑ role	menu item
☐ text	Format

```
<wnd app='notepad.exe' cls='Notepad' title='Untitled - Notepad' />
<ctrl automationid='MenuBar' idx='1' name='Application' role='menu bar' />
<ctrl name='Format' role='menu item' />
```

Target Element: 'menu item Format'

UIAutomation v19.2.0

see **Demo1B–UI Explorer**

Demo 1B. UI Explorer

- **Make a copy of Demo 1A and name it Demo 1B:**
- **Perform the following tasks:**
  - Inspect in **UI Explorer** tool the selectors associated to the UI elements used during automation;
    - Change/select some attributes;
    - Remove/add some selectors;
    - Perform validation on selectors after changes;
- **Discuss the followings:**
  - *What tags are available?*
  - *What attributes do they have?*
  - *Can we change the attributes?*
  - *Can be other attributes added?*
  - *Are all valid selectors?*

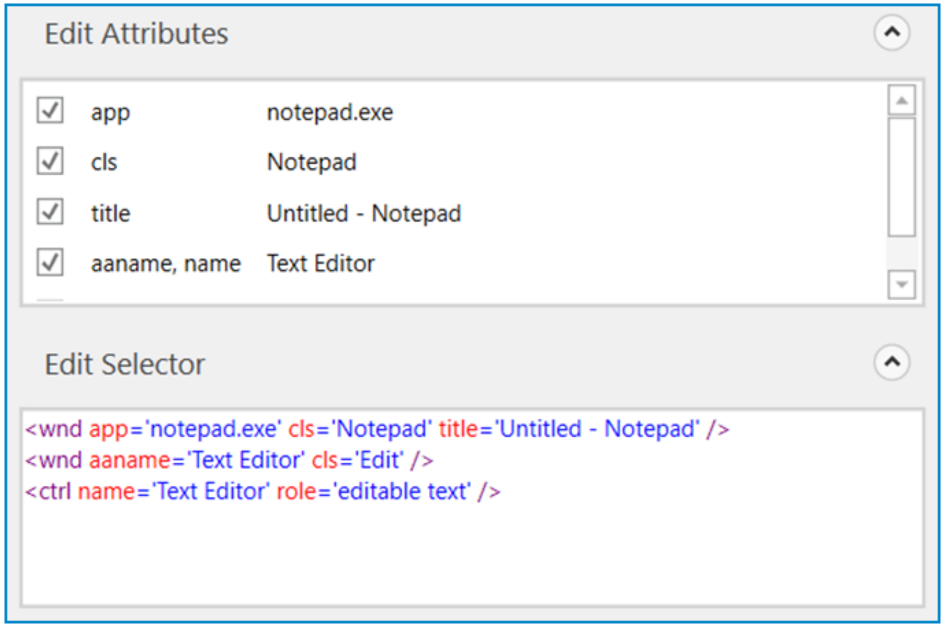
Selectors. Types

- the selectors are defined by looking at the element they target to perform their specific activity to;
- types of selectors:
  - **Full Selectors:**
    - they contain **all the required elements to identify a UI element**;
    - they are generated by the **Basic Recorder**;
  - **Partial Selectors:**
    - they contain **only some elements (and attributes) to uniquely identify a UI element**;
    - they are mainly generated by the **Desktop Recorder**;
  - **Dynamic Selectors:**
    - their **attributes values can be changed** based on **a selected variable**.

see **Demo1C–FullSelectors, Demo1D–PartialSelectors, Demo1E–DynamicSelectors**

Selectors. Full Selectors

- **Full selectors:**
  - contain all the required elements to identify a UI element, including the top-level window;
  - best suited when the Robot performs **actions that require switching between multiple windows**, i.e., the use of containers would add unnecessary complexity;
  - the **Editor** and the **Explorer** are **not grayed** out and are displaying the full selector;



✓	app	notepad.exe
✓	cls	Notepad
✓	title	Untitled - Notepad
✓	aaname, name	Text Editor

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

see **Demo1C–FullSelectors**

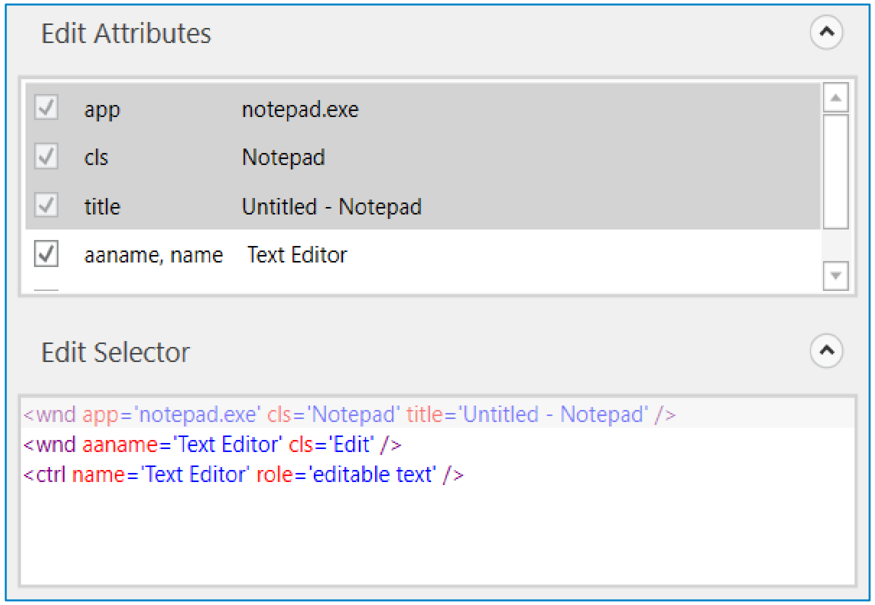
Demo 1C. Full Selectors

- **Make a copy of Demo 2A and name it Demo 2C:**
- **Change the workflow by inserting the following steps:**
  - **1. *open* the Notepad Application;**
  - **2. *open* the Wordpad Application;**
  - **3. *type* in Notepad "Let's see some selectors at work today in Notepad!";**
  - **4. *type* in Wordpad "Let's see some selectors at work today in Wordpad!";**

- **Perform the following tasks:**
  - Inspect in **Selector Editor** window the **Full Selectors** associated to the UI elements used during automation;
- **Discuss the followings:**
  - *Are all selectors enabled?*
  - *Can we change the attributes?*
  - *Does the automation interfere between windows?*

Selectors. Partial Selectors

- **Partial selectors:**
  - are mainly generated by the **Desktop Recorder**;
  - do not contain information about the top-level window; it **is grayed** out (and read-only) in the **Editor** and the **Explorer** section;
  - the user **can edit only elements belonging to the partial selector**;
  - best suited **for performing multiple actions in the same window**;



✓	app	notepad.exe
✓	cls	Notepad
✓	title	Untitled - Notepad
✓	aaname, name	Text Editor

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

see **Demo1D–PartialSelectors**

Demo 1D. Partial Selectors

- **Use the Desktop recorder to create a process that performs the following actions:**
  - **1. *open* the Notepad Application;**
  - **2. *type* in Notepad “Let’s see some selectors at work today in Notepad!”;**
  - **3. *change* the Font to ‘Corbel’;**
  - **4. *select* the Font Style to ‘Bold Italic’;**
  - **5. *set* the Font Size to 16;**

- **Perform the following tasks:**
  - Inspect in **Selector Editor** window the **Partial Selectors** associated to the UI elements used during automation;
- **Discuss the followings:**
  - *How many **containers** are required?*
  - *Are all selectors enabled? Are all valid selectors?*
  - *Can we change the attributes?*

Selectors. Dynamic Selectors

- **Dynamic selectors:**
  - can change specific attribute values based on the selected variable;
  - best suited for situations in which **the targeted element can constantly change its value.**

- E.g.:. A calendar on a web page; we want to click a specific date and receive this action as user input;
  - it can be used the dynamic selector to click the specified date by the user;
  - the input date from the user is stored in a variable;
  - the variable is placed inside a selector;
  - the robot will receive the date, day, and month, identify the specific element from the calendar GUI and perform the required action.

see **Demo1E–DynamicSelectors**

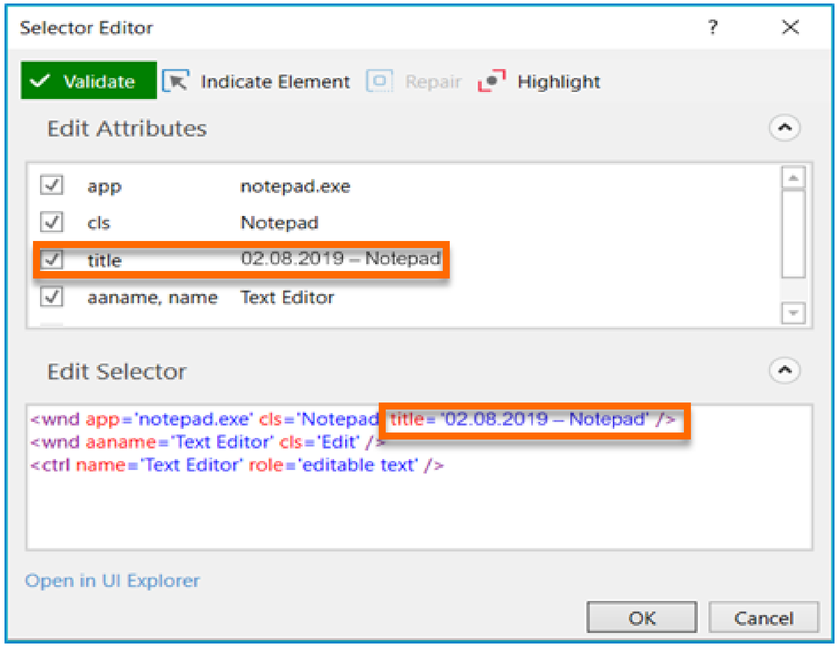
Demo 1E. Dynamic Selectors

- **Use Desktop recorder to create a process that performs the following actions:**
  - **1. *open* the Windows Calendar Application;**
  - **2. *enter* a day D;**
  - **3. *place* an event on day D with the message “Practical exam!”;**
  - **4. *save* the event;**
  - **5. *close* the application;**

- **Perform the following tasks:**
  - Inspect in **Selector Editor** window the **Dynamic Selectors** associated to the UI elements used during automation;
- **Discuss the followings:**
  - *How the selector looks like after using a **variable**?*
  - *Can we change the attributes?*

Customizing Selectors. Details

- Customizing selectors allows
  - to adapt the values of some attributes in order to increase their usage;
- their default setting contains some preset attributes that can easily change them to make the selectors more reliable or tailoring them to the required needs;
- the level of customization usually changes during the debugging phase.
- E.g.:
  - default selector:
    - `<webctrl id="targetElem-212345">`
    - **Target Element** = `'targetElem'`
    - **Variable value** = `'212345'`



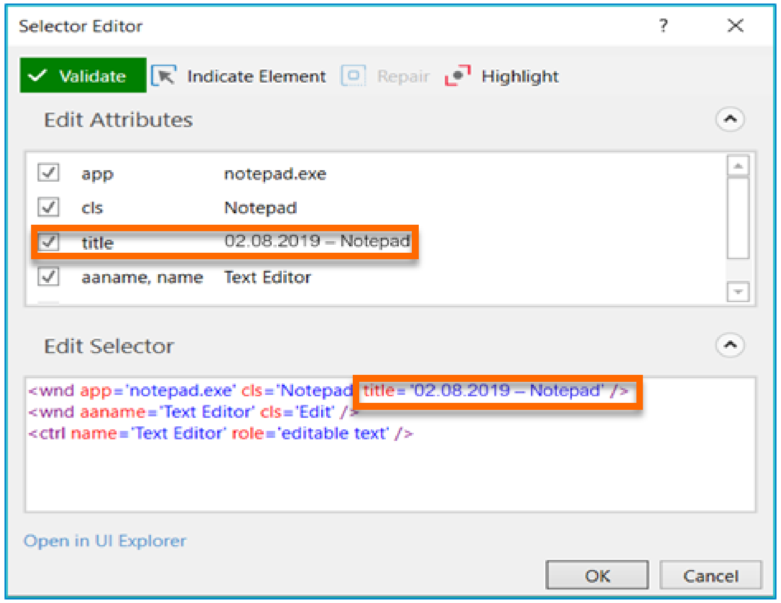
✓	app	notepad.exe
✓	cls	Notepad
✓	title	02.08.2019 – Notepad
✓	aaname, name	Text Editor

```
<wnd app='notepad.exe' cls='Notepad' title='02.08.2019 – Notepad' />
<wnd aaname='Text Editor' cls='Edit' />
<ctrl name='Text Editor' role='editable text' />
```

see **Demo1E–DynamicSelectors**

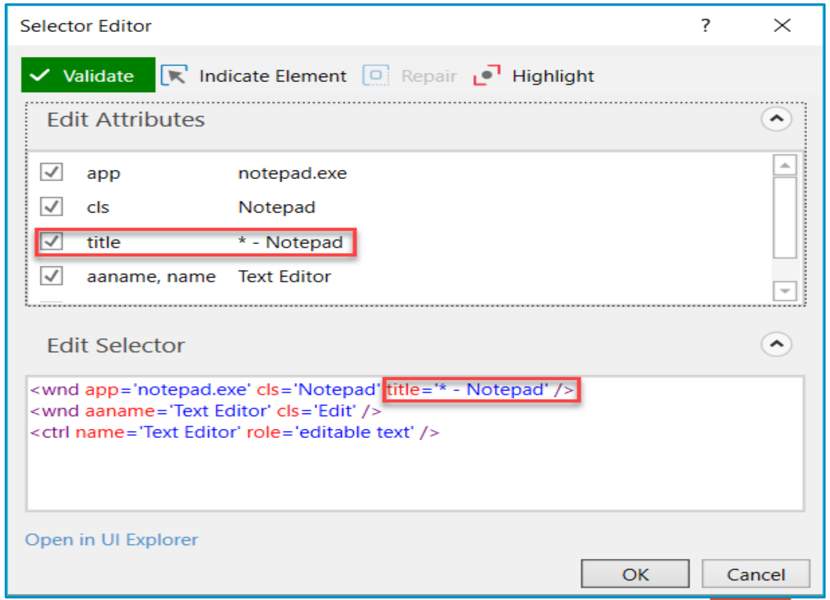
Customizing Selectors. Example

- E.g.: we can use a file for which the file name changes often: every day to display the current date;
  - in this case, a static selector does not work after the limited time period such as one day;
  - the solution is to replace the dynamic part of the selector with an asterisk (*), i.e., replace the name of the file from the selector with a *wildcard*.



✓	app	notepad.exe
✓	cls	Notepad
✓	title	02.08.2019 – Notepad
✓	aaname, name	Text Editor

```
<wnd app='notepad.exe' cls='Notepad' title='02.08.2019 – Notepad' />
<wnd aaname='Text Editor' cls='Edit' />
<ctrl name='Text Editor' role='editable text' />
```



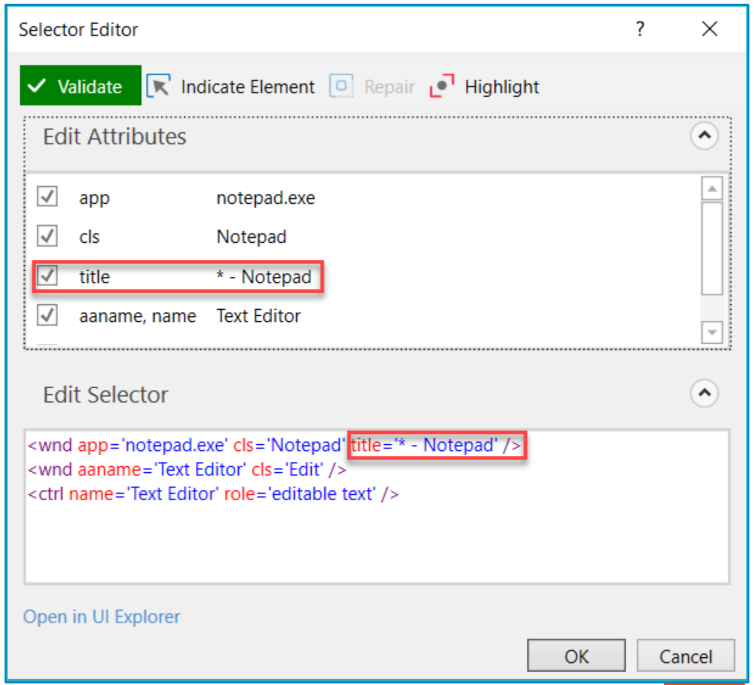
✓	app	notepad.exe
✓	cls	Notepad
✓	title	* - Notepad
✓	aaname, name	Text Editor

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

see **Demo1F–Wildcards**

Wildcards. Details

- A **wildcard** is
  - a special character that can replace the dynamic part of a selector;
- the customized selector that contains a wildcard replaces certain number of characters;
- adding a variable in between selectors can be called as **making selectors to be dynamic** or **customizing selectors**;
- E.g.:
  - default selector:
    - `<webctrl id="targetElem-*">`
    - Target Element = `'targetElem'`
    - Part that changes (may vary)= `'*'`



✓	app	notepad.exe
✓	cls	Notepad
✓	title	* - Notepad
✓	aaname, name	Text Editor

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

see **Demo1F–Wildcards**

Wildcards. Types

- there are two types of wildcards:

?

Question mark

- Replaces 1 character;

*

Asterisk

- Replaces 0..n characters.

see **Demo1F–Wildcards**

Demo 1F. Wildcards

- **User Basic/Desktop recorder to create a process that performs the following actions using 2 text files "File1.txt", "File2.txt", "File21.txt":**
  - **1. choose a file name from the followings: "File1.txt", "File2.txt", "File21.txt";**
  - **2. *open* the chosen file in Notepad Application;**
  - **2. *type in Now +* "logging some activity…";**

- **Perform the following tasks:**
  - Customize the selector to be able to write in any file named as **"File*.txt"**;
  - Inspect in **Selector Editor** window the **Dynamic Selectors** associated to the UI elements used during automation;
- **Discuss the followings:**
  - *How the selector looks like after using a **wildcard**?*
  - *Can we change the attributes?*

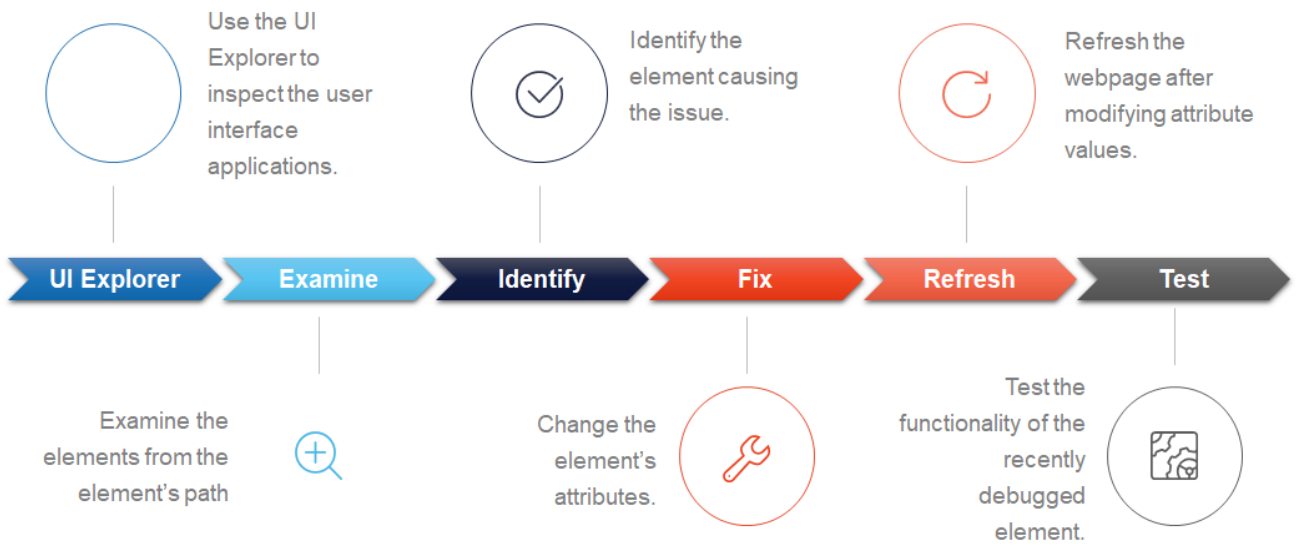
Debugging Selectors. Details

- **Debugging** is
  - the process of identifying and removing errors from a given project;
  - *a hit and trial method to identify the error and help in finding the correct selectors until the desired action is achieved;*
- debugging may be coupled with logging and this results in a powerful functionality that offers information about the project and step-by-step highlighting, increasing confidence in project quality;

- **UI Explorer**
  - is a tool for **checking**, **customizing** and **debugging selectors**;
  - enables to inspect all the attributes that could be used in identifying the element causing the issue.

Debugging Process. Details

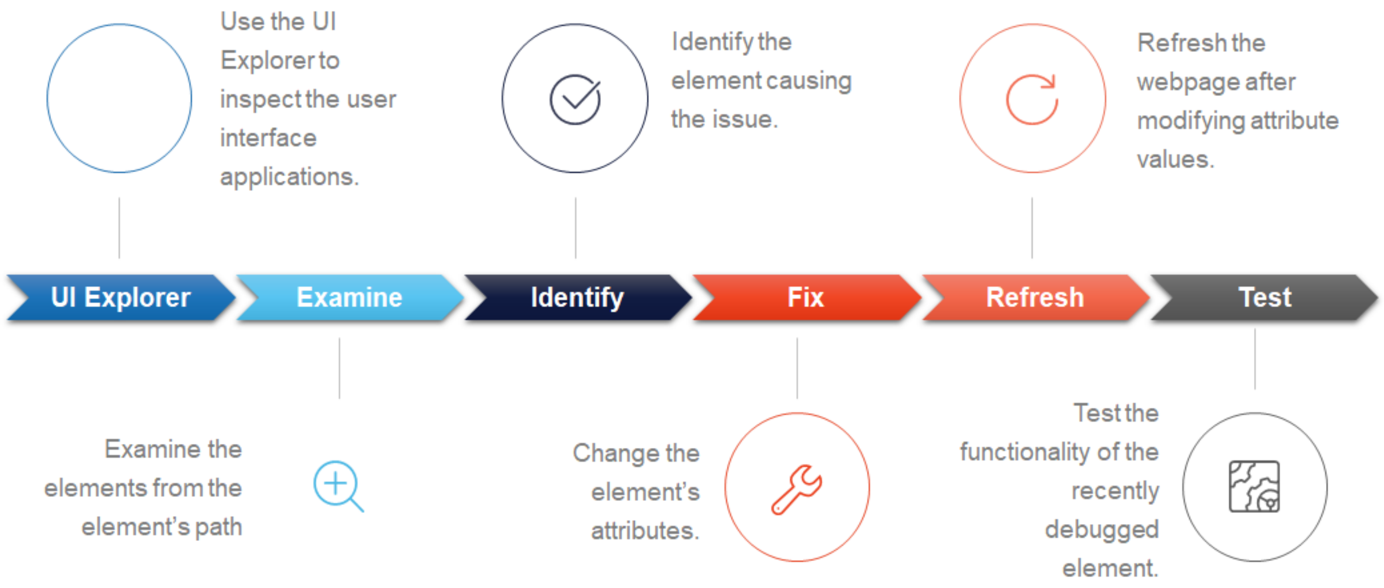
- the **debugging process**:
  - starts once the element has been identified;
  - involves **changing** element’s attributes, either **adding** or **removing** them and **using wildcards** where specific attributes have variable values inside them;
  - after each change, the application (or webpage) must be refreshed, and the selector verified for accuracy;
  - is not always done in the same way for each selector and the amount and type of debugging varies for every selector;



Use the UI Explorer to inspect the user interface applications.
Identify the element causing the issue.
Refresh the webpage after modifying attribute values.
UI Explorer
Examine
Identify
Fix
Refresh
Test
Examine the elements from the element’s path
Change the element’s attributes.
Test the functionality of the recently debugged element.

Debugging Process. Functionalities

- There are several functionalities useful during the debugging process:
  - **Find Element;**
  - **Element Exists;**
  - **Find Children;**
  - **Get Attributes.**


Use the UI Explorer to inspect the user interface applications.
Identify the element causing the issue.
Refresh the webpage after modifying attribute values.
UI Explorer
Examine
Identify
Fix
Refresh
Test
Examine the elements from the element's path
Change the element's attributes.
Test the functionality of the recently debugged element.

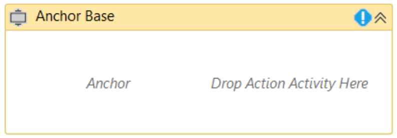
Handling Dynamic UI Elements. Details

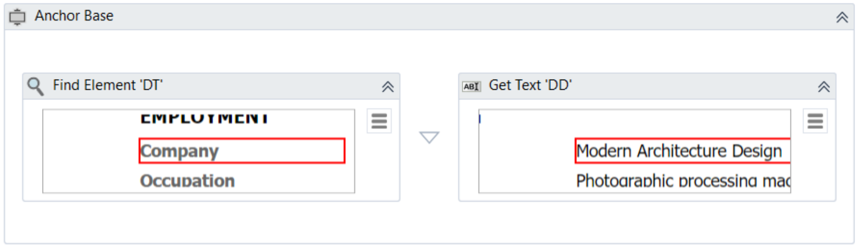
- in UiPath there are specific tools that help dealing with UI Elements that change their id frequently;
- extensive use of CSS selectors causes errors at the slightest change in any parent;
- when **selectors are not reliable** there are several solutions:
  - **Anchor Base** activity container:
    - **useful when the UI Element position is not fixed;**
    - the identification is based on the **position on the screen** of the anchor and the target element;
  - **Relative Selectors**:
    - useful to identify UI Elements that is relative to another element;
    - The identification is based on another element (anchor) that is found in a **specific position in the structure tree**.

see **Demo1G–RPAChallenge**

Anchor Base Activity. Details

- **Anchor Base** activity container consists of two components:
  - **anchor:** an UI Element that is used later as reference;
    - **Find Element** or **Find Image** activities are used to identify the anchor;
    - E.g.: a label may be used as anchor, its selector does not change often, it's stable;
  - **action:** an action on some UI Element;
    - activities as **Click**, **Type into**, **Get Text**, etc.;
    - the selector may have only the **tag** attribute;
    - Ui Path does not use other attributes that are normally dynamic;



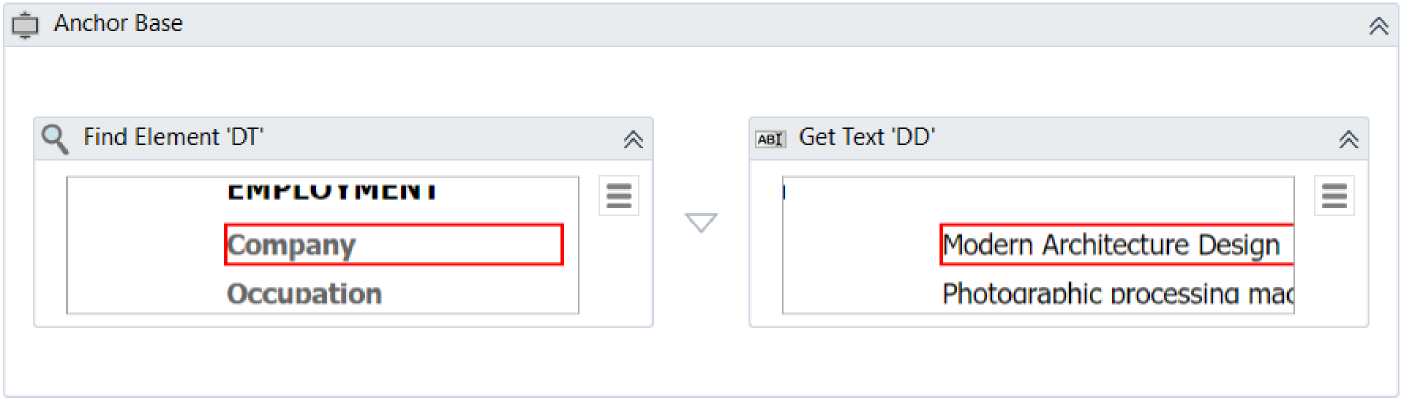


see **Demo1G–RPAChallenge**

Anchor Base Activity. Properties

- **Anchor Base** activity container consists of two components:
  - **Anchor Position** property:
    - **Values: Auto (default), Left, Right, Top, Bottom;**
    - useful if the position of the anchor relative to the target element is always fixed, otherwise 'Auto' value should be used;

- **the robot finds the anchor [anchor component] and uses it as reference to perform the action [action component] on the closest element on the screen that matches the selector;**

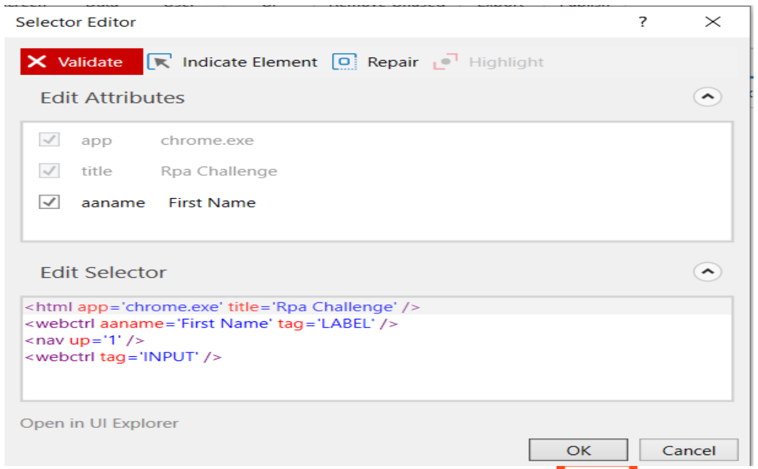


see **Demo1G–RPAChallenge**

Relative Selectors. Details

- **Relative selectors** allows to
  - identify UI Elements that is relative to another element;
- Steps:
  - 1. indicate the target element;
  - 2. indicate the anchor;
  - 3. customize the selector so it includes the position in the UI structure tree;
    - use the **nav** tag to state the relationship with the anchor: **up, prev, next**;
  - 4. copy the selector into the activity associated to the target element;

- **the robot identifies the target element based on another element (anchor) that is found on a specific position in the structure tree;**



✓ app	chrome.exe
✓ title	Rpa Challenge
✓ aaname	First Name

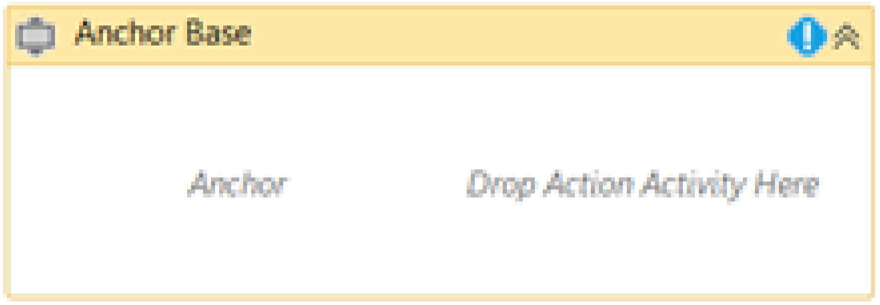
```
<html app='chrome.exe' title='Rpa Challenge' />
<webctrl aaname='First Name' tag='LABEL' />
<nav up='1' />
<webctrl tag='INPUT' />
```

see **Demo1G–RPAChallenge**

Anchor Base Activity vs Relative Selectors

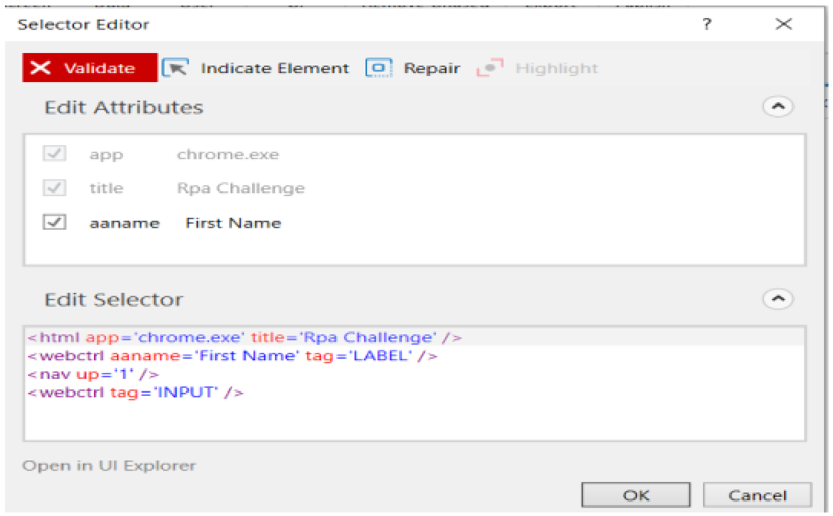
Anchor Base Activity

- it does **not** work in background;
- it uses the **screen position** of the anchor and the target element.



Relative Selectors

- it works on background;
- it uses the **internal structure of the application** to identify the target element.



✓	app	chrome.exe
✓	title	Rpa Challenge
✓	aaname	First Name

```
<html app='chrome.exe' title='Rpa Challenge' />
<webctrl aaname='First Name' tag='LABEL' />
<nav up='1' />
<webctrl tag='INPUT' />
```

see **Demo1G–RPAChallenge**

Demo 1G. RPA Challenge

- **Automate the following process:**
  - **1. *open* the FakeNameGenerator.com website in Chrome browser;**
  - **2. *generate* input data based on given *name set, country* and *gender*;**
  - **3. *extract* values for Name, Phone number and Company name;**
  - **4. *type into* RPAChallenge.com website in Chrome browser the values in the First Name, Phone Number and Company Name fields.**
    - ***use UI Explorer to build reliable selectors;***
    - ***try the* Anchor Base *activity and* Select Relative *element option in UI Explorer to get the target element relative to its label.***
  - **5. repeat steps 2..4. for 5 times.**

Next lecture

- **Robotic Enterprise Framework**

References

- UiPath Docs - https://docs.uipath.com/
- UiPath Studio Docs - https://docs.uipath.com/studio/standalone/2023.4
- UiPath Forum - https://forum.uipath.com/
- UiPath Academy - https://academy.uipath.com/