



Selectors

Selectors - Agenda



Definition, usage, examples

-  Tags and attributes



UiExplorer

UI frameworks and hierarchies

Full and partial selectors

-  Open Browser, Open Application
-  Element scope, Attach window

Advanced techniques

-  Wildcards
-  Finding elements

Best Practices

Q&A

Test

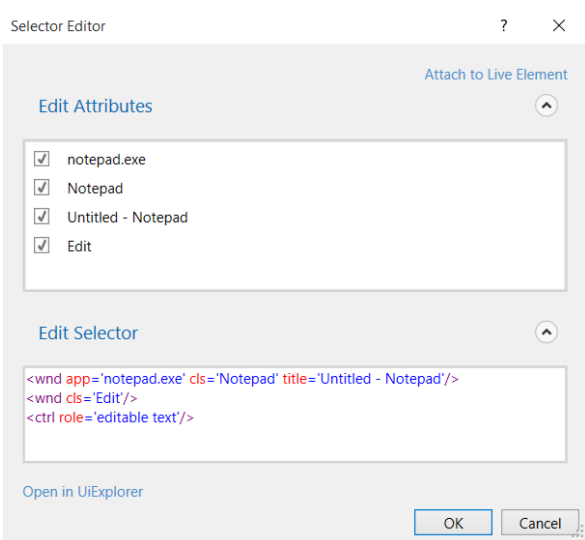
Online Resources

Definition and usage

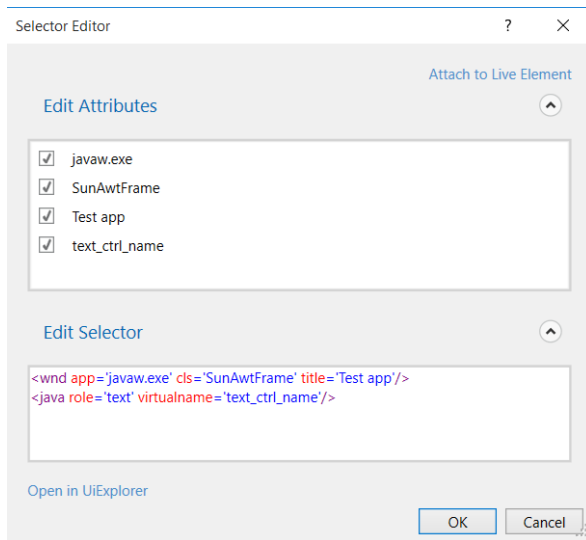
Selectors

- Used to find UI elements among the running applications
- Similar to postal addresses or OS file paths
- Made of 1 or more XML fragments with tags and attributes
- One line = One parent UI element needed to reach our target element
- UiPath generates the right selector automatically almost all the time

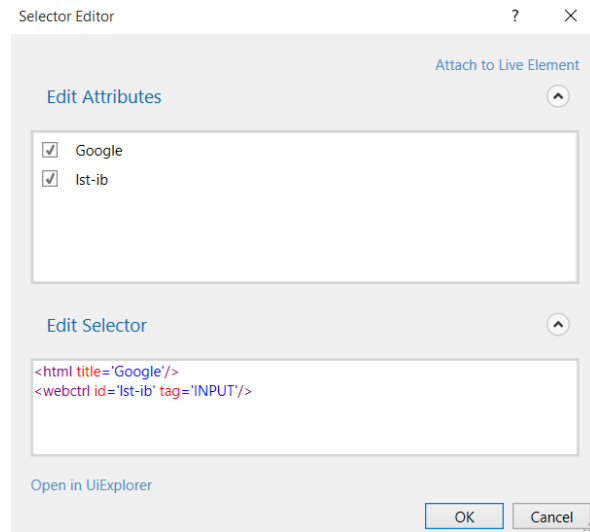
UI Frameworks



A desktop app element

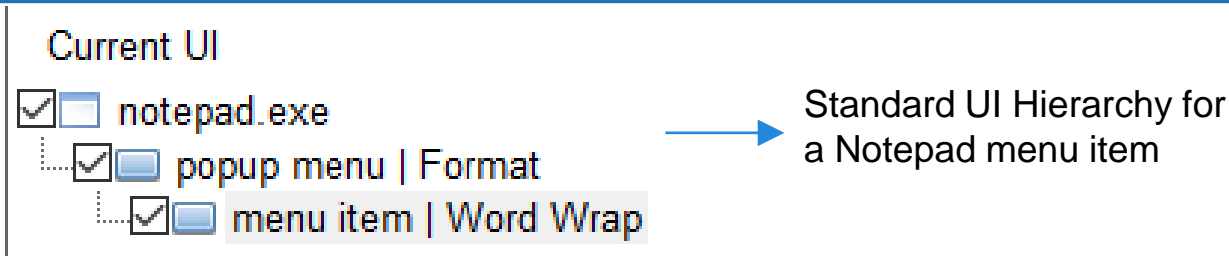


A java app element



A web app element

UI Hierarchies



The UI Hierarchy is consisting of **children** and **parents**, depending on what level of the selector you are looking at.

By default, UiPath will always try to capture **minimal** and **stable** selectors and that means the top and the bottom levels.

For getting a good selector, you can always **add more nodes** than your initial selector provided.

Meaning of the structure

<wnd app='winforms_ratchetx.mycrm_x86.exe' ctrlname='Form1' />

<wnd ctrlname='tabControl1' />

<wnd ctrlname='tabPagePeople' />

<wnd ctrlname='groupBox5' />

<wnd ctrlname='textBoxPeopleFirstName' />

The screenshot shows a window titled "My CRM (Sample App)" with a standard Windows interface (minimize, maximize, close buttons). Inside the window is a form with three tabs: "People", "Company", and "Other". The "People" tab is selected. Within this tab, there is a "Name:" section with two text boxes: "First:" containing "Oscar" and "Last:" containing "Madisox". Below these are radio buttons for "Male" (selected) and "Female". Further down is an "Address" section with a text box containing "1049 Park Avenue".

Arrows from the XML code on the left point to the following UI elements:

- A green arrow points from the `ctrlname='Form1'` attribute to the window's title bar.
- An orange arrow points from the `ctrlname='tabControl1'` attribute to the tab control area.
- A blue arrow points from the `ctrlname='tabPagePeople'` attribute to the "People" tab.
- A grey arrow points from the `ctrlname='groupBox5'` attribute to the "Name:" section.
- A red arrow points from the `ctrlname='textBoxPeopleFirstName'` attribute to the "First:" text box.

Tags and attributes

```
<webctrl parentid='slide-list-container' tag='A' aaname='Details' class='btn-dwnl' />
```

Tags

- nodes in the selector XML fragment
- represent a visual object on screen
- first node is the app window
- last one, the element itself

Might be:

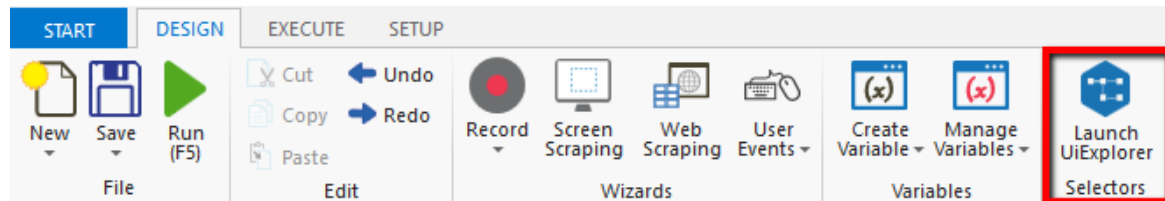
- **wnd** (Window)
- **html** (Web page)
- **ctrl** (Control)
- **webctrl** (Web page control)
- **java** (Java application control)

Attributes

- Every attribute has a name and a value
- Use only attributes with constant or known values

```
parentid='slide-list-container'  
aaaname='Details'  
tag='A'  
class='btn-dwnl'
```

UiExplorer



UiExplorer is a **visual** tool that

- helps you **refine** or improve a selector
- conveniently exposes the **hierarchy** of element nodes in the running apps
- shows all elements **properties** and their **values**
- helps you **validate** a selector in real time
- can sometimes even **repair** selectors automatically
- helps you **add / remove parent** elements from the selector

UiExplorer will always generate a full selector of your UI element.

Full and Partial Selectors

Full selector	Partial selector
Starts with the top level window	Excludes the top level window
Generated by Basic recorder	Generated by other recorder profiles
Usage - when dealing with one interaction inside a specific window	Usage – when dealing with many interactions inside a specific window.
Advantages: <ul style="list-style-type: none">- no need to add a container box just for an activity	Advantages: <ul style="list-style-type: none">- no need to repeat window selector- makes the automation process faster.

Exercise 1: types of selectors

Check how the selectors can combine and be generated in different ways

1. Open Basic recorder and perform a type into the editable field of a notepad file, click on the File menu item, and then click on Save.
2. Open Desktop recorder and do the same thing.
3. Observe the difference between the two generated recordings.

Wildcards

*

?

Sometimes, attributes change their values from time to time due to the fact that the application's UI changes. When this happens, part of the dynamic attributes often remain unchanged and we have to specify the variable parts somehow.

Means **zero or more characters** can be there. Example: title='* - Notepad'.

Means **exactly one character**. Example: title='Expl?rer'

Exercise 2: make a selector stable

Create a selector that will work on different uses of the same webpage

1. Open the calendar and record clicking on 15th of the current month
2. Parametrise the selector in order to click on a day number taken from the user
3. Try clicking on the 28th of current month instead of last month

Finding Elements

Most used activities for finding elements

Find Element

- finds a particular element and saves it
- returns a **UI Element** variable if the element is found
- throws an **error** if the element is not found

Element Exists

- checks if an element exists based on its selector
- returns a **Boolean** variable with the answer True / False

Find Children

- retrieves a **collection** of elements at once
- you can setup a flexible **filter** for elements properties

UI Automation

Element

Find



Anchor Base



Find Relative Element



Indicate On Screen



Element Exists



Find Element



Wait Element Vanish



Find Children



Get Ancestor



Element Scope

Best Practices

- Use only what seems to be a invariant value attribute. Use the * wildcard for invariable attribute values.
- Don't use idx attribute to your selector, unless it is a very small value like 1 or 2.
- If you are automating a web page, in UiExplorer you can highlight the specific UI element, then reload the web page and see if it is still highlighted.



Q&A



Test

Online Resources

- Ui [Selectors and UiExplorer](#)
- Ui [Ui Hierarchy, attributes, wildcards](#)
- Ui [Difference between basic and desktop recording](#)
- Ui [Building a stable and reliable selector](#)