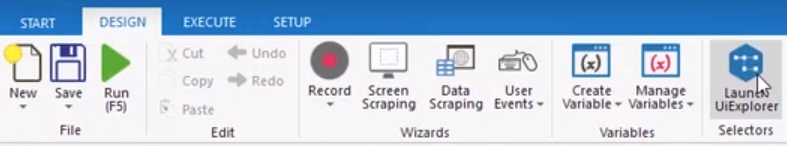
**简介：**Selectors 是识别UI元素的方式。它是一个**XML字符串**，其中包含一些属性可以唯一地定义指定的元素。

**学习大纲：**

* 什么是Selector
* 什么是稳定的Selector
* 如何在Selector里插入通配符
* 如何在Selector里插入变量

**1. UI Exploer中的Selector**

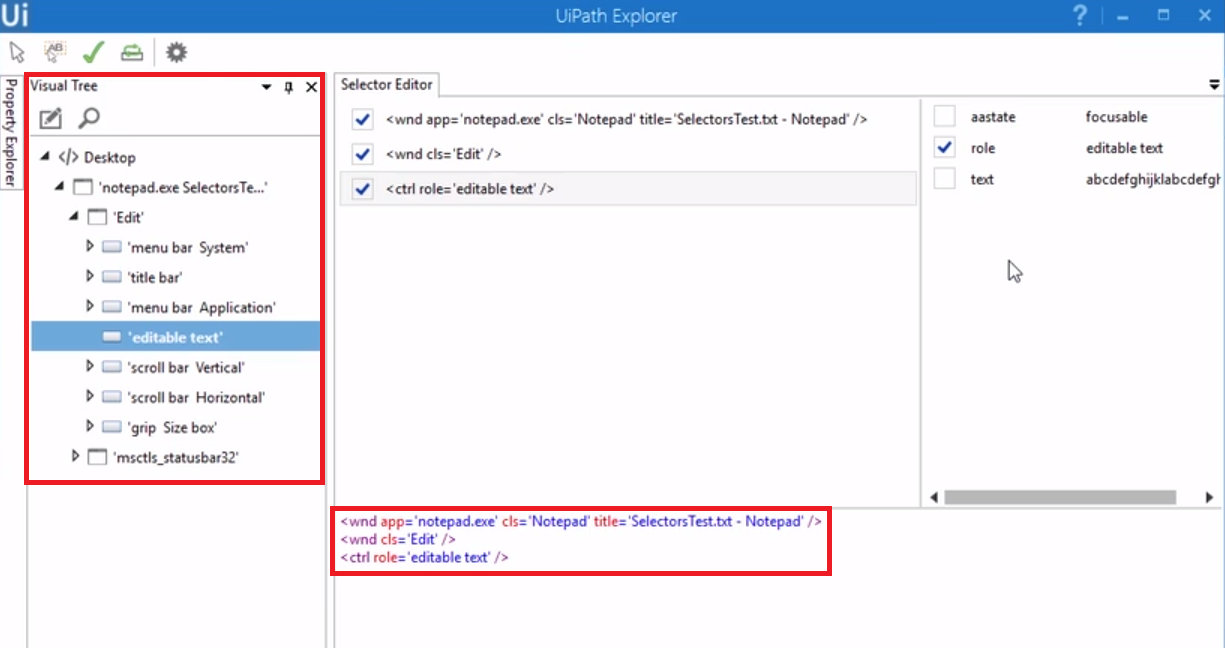
UI就是很多个互相嵌套的容器，使用**UI Explorer**工具可以帮助我们了解UiPath怎么去识别某个UI中的元素，了解Selector的工作原理。



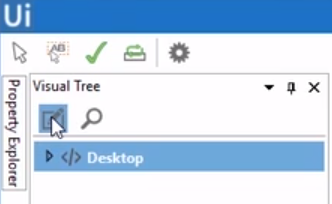
1) 点击工具栏上的Select Target Elements，再点击UI界面中的某个元素。



2) 该UI元素的Selector显示在下面的模块(Selector Editor Panel)里，而左边的模块(Visual Tree Panel)里是一个完整的UI树状图，包含了UI所有的元素。

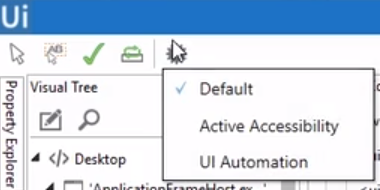


3) 点击**Highlight**工具，然后在UI Tree里选中某个元素，此时UI中对应的元素会被黄色的矩形框起来，方便查看。



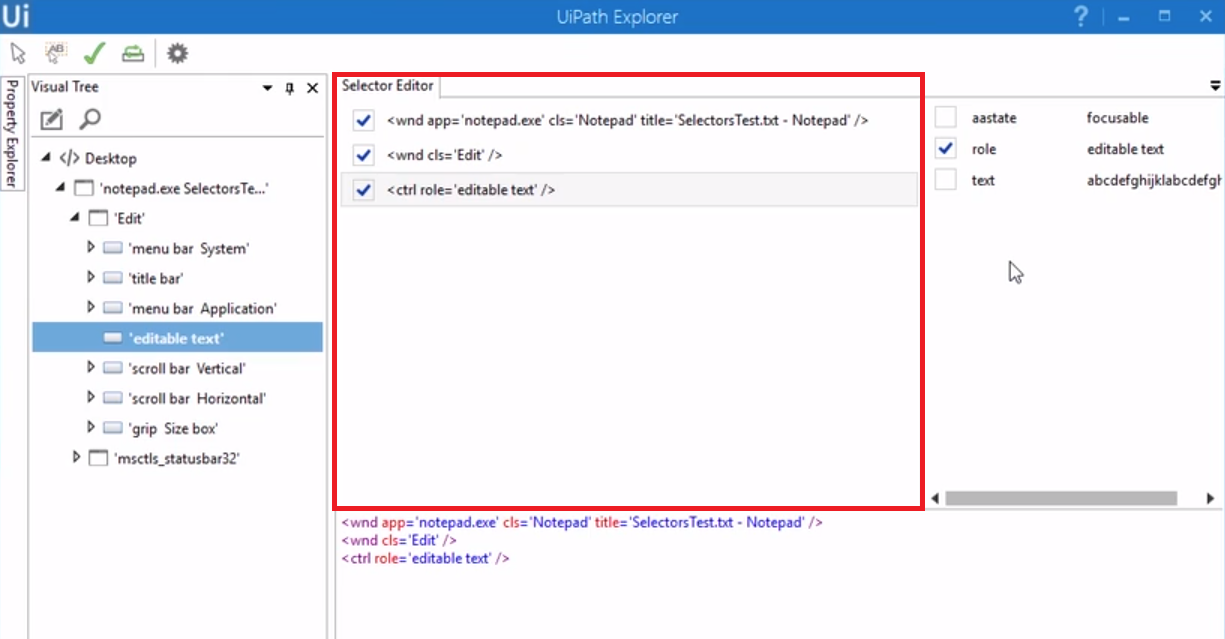
4) 如果某个元素未被识别（在选取这个元素时没有出现蓝色的矩形框），可以更改UiFrameworks的选项，再重新指定元素。

* UiFrameworks用于更改确定UI元素及其Selector的技术
  + Default：UiPath的专有方法。通常适用于所有类型的用户界面。
  + Active Accessibility：Microsoft早期提供的一种使应用程序可访问的解决方案。当Default方法不起作用，而软件是比较旧的版本时，推荐尝试这种方法。
  + UI Automation：Microsoft改良过的访问模式。当Default方法不起作用，而软件是比较新的版本时，推荐尝试这种方法。



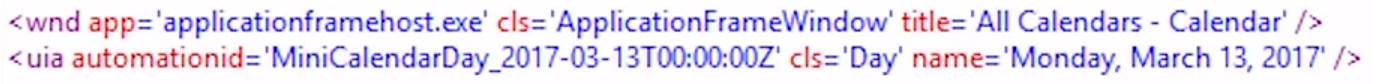
5) 上面的模块(Property Explorer Panel)显示的是未优化的原始Selector，它包含了目标元素的完整路径。下面的模块则是优化过的最终Selector。

* UiPath默认使用第一层和最后一层容器来构成最终的Selector，因此优化过的Selector里的元素会少一些。
* 选择或者取消选择某一行，可以在最终的Selector里添加或删除元素。

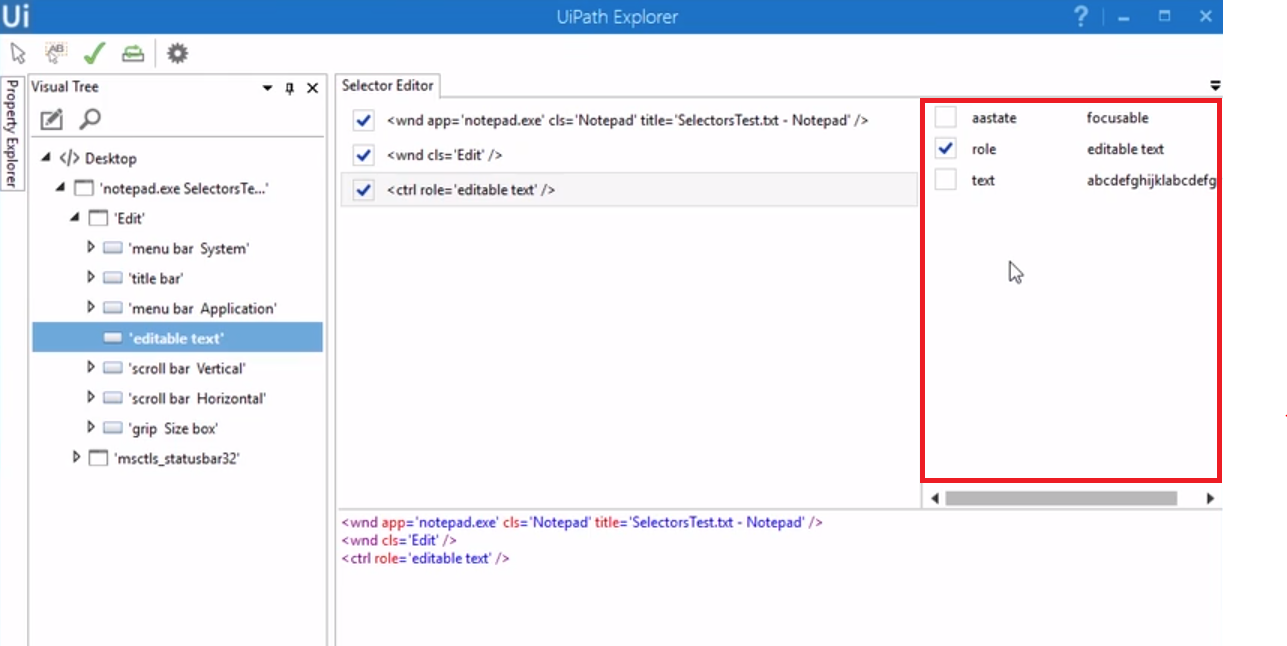


6) 优化后的Selector是标准的XML代码，它由两部分构成：

* 元素类型：wnd和uia。元素的类型一目了然，比如window，control，HTML，Java等等。
* 属性 (Attributes)：元素类型后面的部分，它是Selector识别特定元素的关键。
  + 下列代码的第一行表示名为applicationframehost.exe的应用程序，该程序的Class是ApplicationFrameWindow，标题是All Calendars - Calendar。如果任一条件不匹配，就找不到这个应用程序。



7) 属性显示在右边的模块(Selector Attributes Panel)。选择或者取消选择某个属性，会相应的更改Selector。



**总结：**Selector就是UI元素和相关属性的列表，UiPath根据这些信息去匹配屏幕上已有的元素。

**注意：**在一些情况下，当UiPath和目标应用程序时用不同的权限打开的，Selector可能无法正常作用。推荐总是用相同的权限打开程序。

**2. Selector的调试**

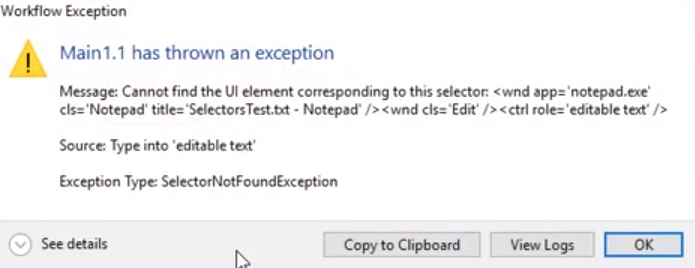
**实例：**用Basic方式录制一个Sequence，在名为SelectorsTest.txt的文本文件里录入一段文字。

**分析：**

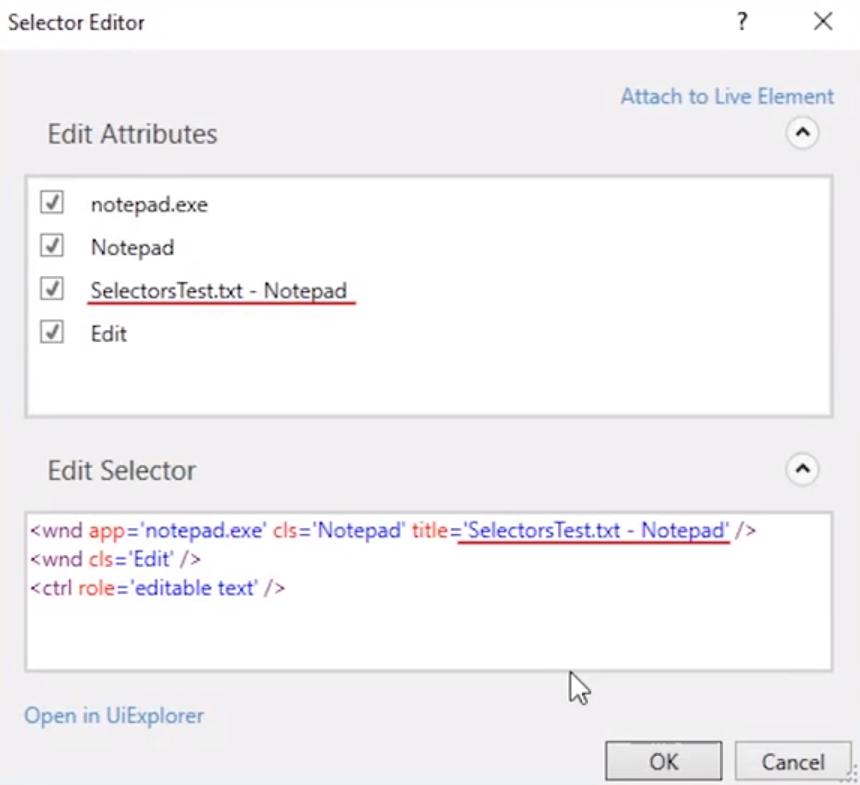
1) 查看Type Into活动的属性面板，打开Selector Editor。

2) 关掉SelectorsTest.txt，打开五个新记事本并运行Sequence，30秒后UiPath将报错。

\* 这种提示大部分都是Selector出现问题。



3) 问题出在title属性上。由于指定了标题，当SelectorsTest.txt没有打开时，程序就无法正常运行。



4) 考虑到你是想让这个程序在任何文本文件里都可以运行，还是只在指定的文件里运行，这可以是一个bug，也可以不是。

5) 如果想要程序在任何文本文件里运行，只要删除title属性就可以了。

6) 重新运行程序。虽然打开了五个记事本，但程序只会在最上层的记事本里运行。

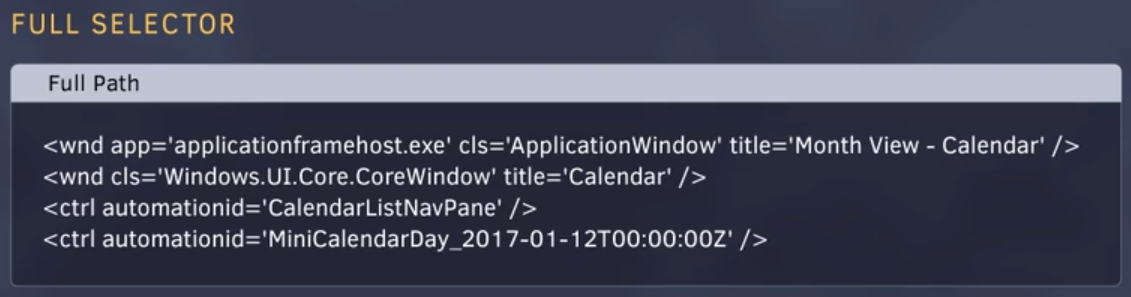
\*Selector Editor和UI Explorer有所不同。在Selector Editor的Edit Attributes模块里，该Selector包含的所有UI元素的相关属性都会列出来。而在UI Explorer里，只会显示选中的那个UI元素的相关属性。

点击左下角的Open in Explorer可以在UI Explorer里查看和修改更多的UI元素和属性。你也可以在Selector Editor的Edit Selector模块里手动修改代码。

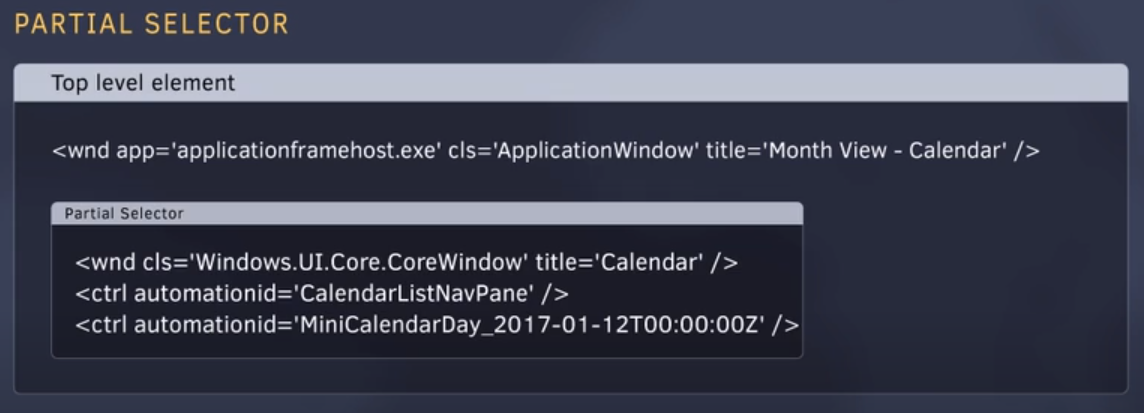
**3. 自定义Selector**

1) Full Selector和Partial Selector

Full Selector：包含了确定该UI元素位置的所有必要信息，其中包括顶级窗口的信息。



Partial Selector：不包括顶级窗口的信息，但是嵌套在一个包含了顶级窗口信息的容器中，比如Attach Window活动，Open Application活动。



除了是否包含顶级窗口的信息，Full Selector和Partial Selector没有什么区别。**在同一个窗口执行多个操作时**建议使用Partial Selector；**需要在多个窗口之间切换时**建议使用Full Selector。如果可能受到**其他应用程序或窗口的干扰**，最好使用Partial Selector。

\*Full Selector也可以在Attach Window或者Open Application等的容器中使用。

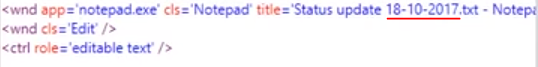
\*录制功能的Basic模式使用的都是Full Selector，Desktop模式使用的都是Partial Selector。

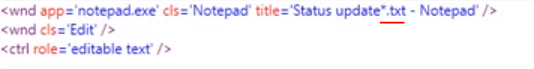
2) 动态Selector：**通配符**

* \* (星号)：代替任意个字符。
* ? (问号)：代替任意一个字符。

**实例1：**如何让UiPath识别出一个名为“Daily Report - ”加上当天的日期的文本文件。

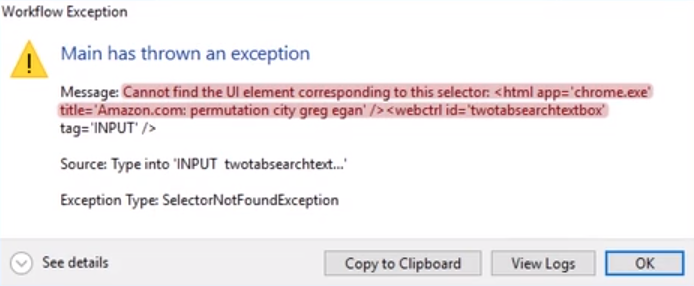
**方法：**将动态的部分，在这里即日期，用\*表示。



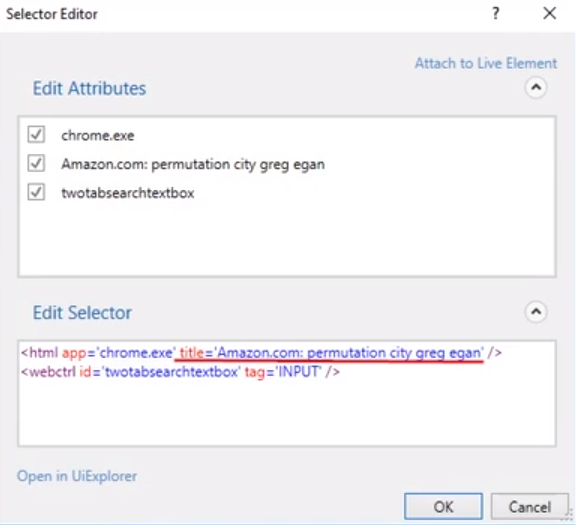


**实例2：**首先在亚马逊首页查询书A的价格，接着用Basic模式录制下列操作：选中并清空搜索栏，输入书B的名称并查询，读取书B的价格并用Message Box显示。

运行程序30秒钟后，将收到下面的错误信息。



**分析：**由于首先查询了书A，在录制过程中浏览器的标题包含书A的名称。但是在录制结束后，浏览器的标题已经变成了书B，因此UiPath无法识别出浏览器窗口来运行程序。

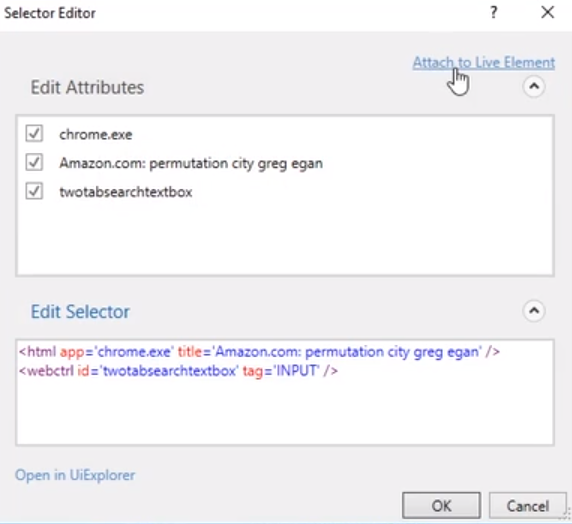


**方法：**可以直接去掉title属性。但是更好的方法是，把书A的名称替换为通配符\*。GetText活动的Selector也需要相应的修改。

3) 动态Selector：**Attach to Live Element**

如果不使用通配符的方法修改Selector，还可以使用Attach to Live Element。

找到出错的Selector，点击Selector Editor右上角的Attach to Live Element，然后再次点击UiPath运行时没能识别出的元素，比如实例2中的搜索栏。UiPath会将之前的Selector和这一次的进行对比，用\*代替其中不同的部分。

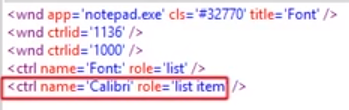


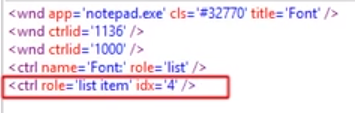
4) 动态Selector：**Index**

Index告诉UiPath应该选择第几个UI元素。

**实例：**在记事本中录制一个更改字体的Sequence，并把它改成选择第四种字体。

**方法：**录制完成后，从Selector Editor中打开UI Explorer。如果我们去掉字体名称的属性，它将会被自动替换为idx=4。





\*如果在一个Selector中出现了Index属性，**这是一个信号，说明现有的属性不足以让UiPath识别出目标UI元素**，因此需要依靠Index属性来找到它。但是这样的Selector是不可靠的，我们应该尝试添加其他上级元素或者属性而避免使用Index。

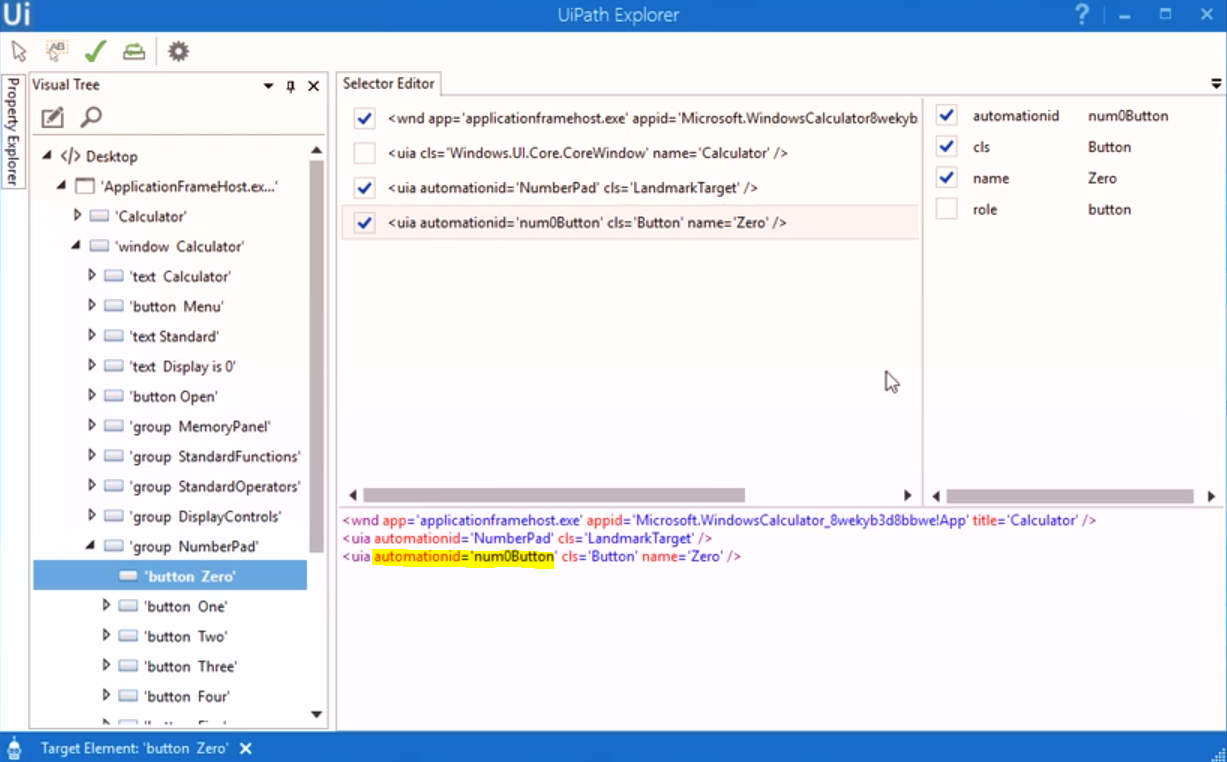
\*应尽量避免使用大于2的Index属性，尝试添加其他的属性来确保Selector的稳定性。

5) 动态Selector：**变量**

**实例：用户输入一个0-9的数字，在Windows自带的计算器中标亮这个数字。**

**方法：**

* 添加一个Input Dialog方法，要求用户输入0-9的数字，输出变量digit
* 添加一个Highlight方法，然后随便指定一个计算器中的数字，这里指定0。
* 用UI Explorer查看计算器的UI元素，找出哪一个属性与数字有关。





* 包含有数字的属性只有Automationid，需要保留。Name属性与数字的英文有关，它是动态的，必须去掉。稳妥起见，可以把cls='Button'属性和整个NumberPad元素去掉。最终的Selector如下图所示。



* 将最终得到的Selector复制到Highlight活动的Selector里，然后在**属性面板**里把Selector里修改为<uia automationid='num**" + dight + "**Button'/>。
  + 一定要在属性面板里修改，因为Selector Editor里的Selector是不带引号的。
  + 另一个思路是声明一个**String型的变量**，然后用**Replace方法**把Selector里的0替换为该变量。

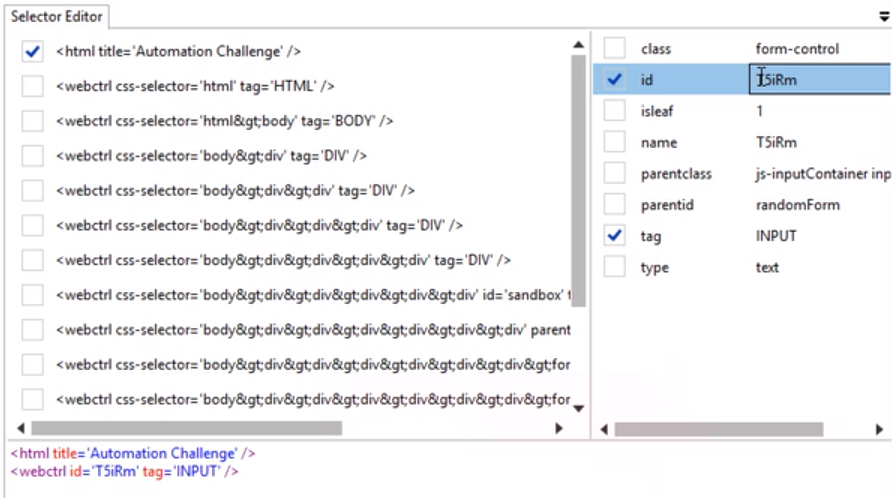
**4. 创建更可信的Selector**

**实例：**

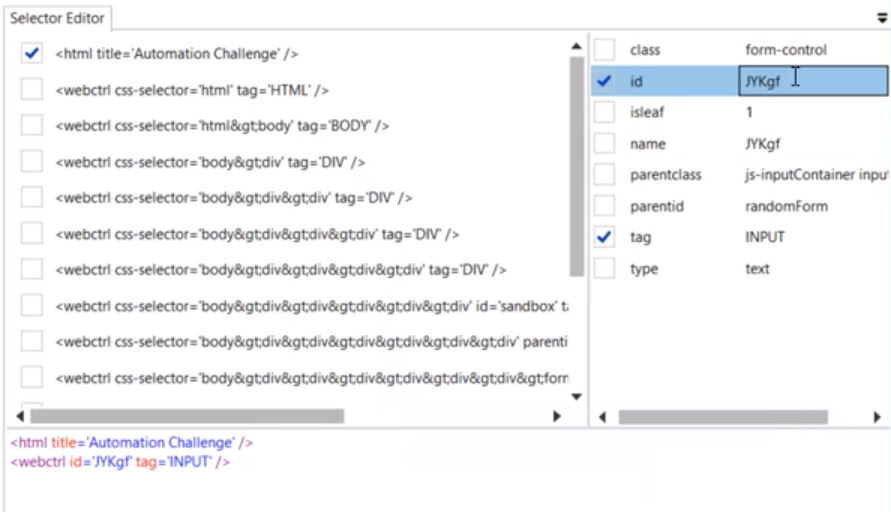
**http://rpachallenge.com** 网站提供了一个关于RPA的练习。在这个网站上，每点一次Submit，信息的排列顺序和位置都会改变。如何让UiPath总能定位到First Name的文本框并填入Mack的名字？

**分析：**

1) 首先使用UI Explorer分析First Name文本框的Selector，该Selector有一个id属性是唯一的。

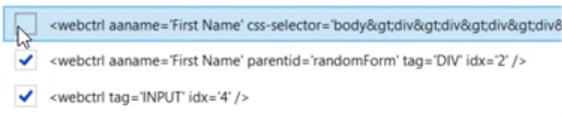


2) 点击网页上的Submit按钮进入下一页，新的First Name文本框的id属性值和上一个不同。这说明id属性是动态的，无法每次都定位到First Name文本框。

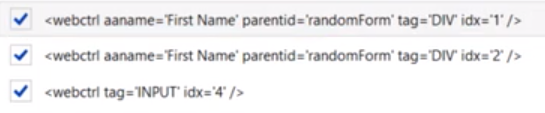


3) 文本框元素里没有任何一个属性与First Name有关。但是它的上级元素有一个aaname属性提到了First Name，我们把这个元素添加到Selector里。

4) 该元素里的css-selector属性是唯一的，但是只要网页设计稍作调整，这个属性就会发生变动，因此要把它去掉。



5) 去掉css-selector属性后，UiPath会自动为该元素添加idx='2'属性，这说明现有的属性可能不足以确定该元素的位置。其原因是：除了css-selector属性外，该元素和它的上级元素的其他属性(aaname/ parentidid/ id)是相同的。只要把该元素的上级元素也选上，idx='2'就会消失。记得去掉新添加的元素的css-selector属性。



6) 这个Selector是能用的，但是它存在很大的缺陷：一是该Selector仅能在特定的网页结构下运行；二是为Selector添加过多的上级元素会让它变得不稳定，因为上级元素只要稍有变化就会发生错误。

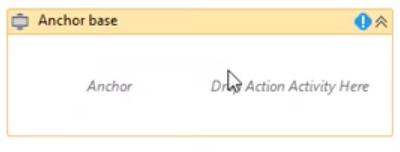
\*点击**对号按钮**可以检查当前Selector的有效性。



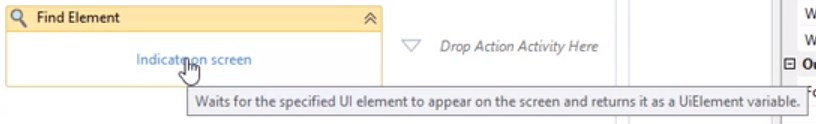
**方法：**此处提供两种更稳妥的方法解决上述问题。

1) **使用Anchor Base方法**：

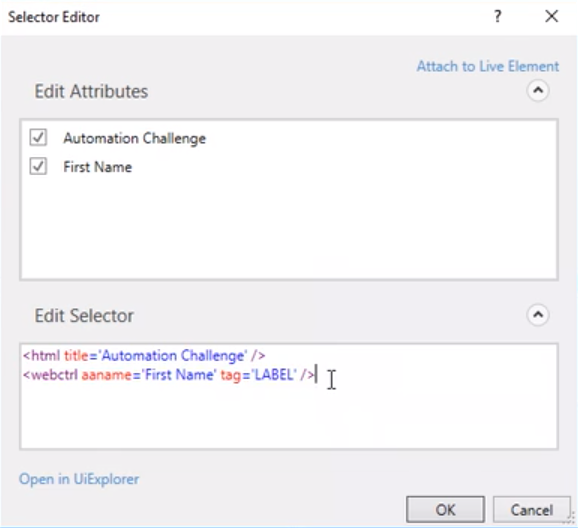
* 在该方法的容器中，左边放锚点(anchor)活动，右边放要运行的其他活动。



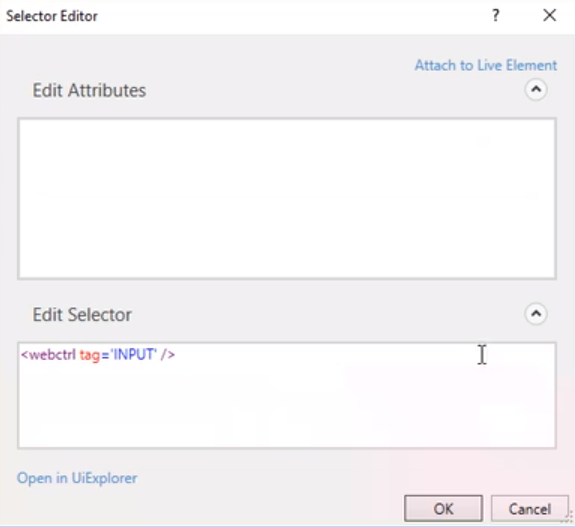
* **Find Element**活动或**Find Image**活动可以作为**锚点**。Find Element活动同样通过Target里的各种属性来定位UI元素，但不同的是它会输出一个**UIElement对象**。稍后，我们将用这个对象创建一个更可信的Selector。



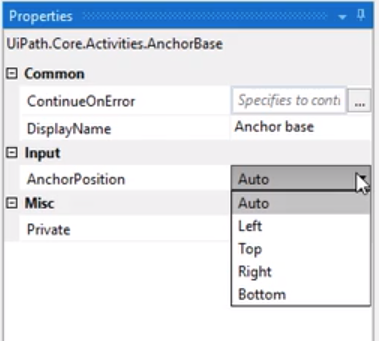
* 先指定网页上的First Name标签(Label)作为锚点，标签的Selector通常非常稳定。



* 在Anchor Base容器的右边，添加**Type Into**活动用来在文本框里输入Mack的名字。指定First Name标签右边的文本框作为TypeInto活动的目标元素，可以看到它的Selector里只有一个tag属性，并没有id等动态属性。
* **Anchor Base活动的原理**是UiPath将以指定的锚点作为参考，选取屏幕上离它最近的符合Selector条件(webctrl tag='INPUT')的UI元素，这样做缩小了搜索范围。



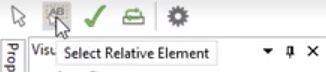
* Anchor Base活动有一个非常有用的属性**Anchor Position**。它的默认值是Auto，但是如果锚点相对于目标元素的位置总是固定的，就可以把它改为其他值。在本例中它不是固定的，因为有时候文本框在标签的左边，有时候在下边。



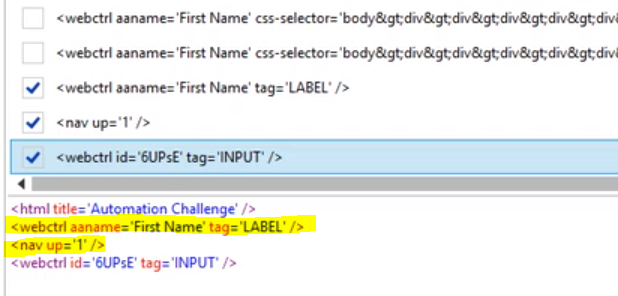
* 由于Anchor Base活动通过锚点和目标元素在屏幕上的相对位置来进行定位，因此它**无法在后台运行**。

2) 使用**Select Relative Selector**

* 打开UI Explorer，先指定First Name文本框。点击左上角工具栏里的Select Relative Element。



* 此时进入选取模式，选取First Name的标签文字。这样做会在列表中添加新的关于First Name标签的元素：webctrl和nav。nav的up属性可以导航到父元素，再从它的子元素列表里选择目标元素(...that locates the First Name label, navigates up to the parent element level and then selects the input in the list of children elements.)。同时别忘了去掉文本框的id属性。
  + nav支持的其他属性有**next**和**prev**，分别表示前往结构树的下一个或者上一个元素(...which go to the next or the previous elements in the tree respectively.)。



* 添加一个**Type Into活动**，把得到的Selector 复制到该活动的Selector属性里，大功告成。
* 使用Select Relative Selector功能，程序是**可以在后台运行**的，因为该功能基于**应用程序的内部结构**而非屏幕上的位置。

**5. 练习**

**要求：**

1) 通过**FakeNameGenerator.com**生成随机的数据，提取姓名，电话和公司名。

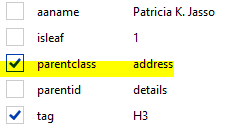
2) 将提取的信息填入RPAChallenge.com的对应区域。

3) 尝试分别使用Anchor Base活动和Select Relative Element选项来获取可靠的Selector。

**思路：**比较简单，略。

**注意：**

* 浏览器的标题可能产生变化，用**通配符**来避免该变化的影响。
* FakeNameGenerator.com生成的姓名找不到合适的anchor或relative element，可以从UI Explorer里为它添加其他属性保证Selector的可靠性。



**\*本课使用过的新活动、方法、函数等：**

**Highlight**

**Anchor Base**

**Find Element**

**Find Image**