WebAutomation:

**How can I make the activity wait for an element to appear before performing an action?**

When your webpage is slow or takes time to respond and load all the elements, there is an option in the activities called “Wait for Ready.” By default, this is set to "None." Apart from "None," there are two other options: "Interactive" and "Complete."

Interactive – This option identifies the element as soon as it appears on the page, even if the webpage is still loading.

Complete – This option identifies the element only when the webpage is fully loaded and all the elements have appeared.

**REPLY Q: What is the best practice you recommend here?**

The recommended option is “Complete,” even if your webpage is interactively responsive. This ensures you wait for the element and also accounts for dynamically loaded webpages based on Ajax calls, allowing actions to be performed promptly.

**Reply Q: Is there any time limit for the activity to wait until the element appears if I select "Complete" for "Wait for Ready"?**

Yes, there is a “Wait Time” property that specifies the time limit for the browser to completely load and for the elements to appear.

2. Attributes –

Question: Hello everyone, I'm unsure about which attributes to choose after spying on elements on the website. I have tried using “ID”, “Tag”, “InnerText”, “ParentID”, and “Parent Tag”, but I'm still unable to click the elements correctly. What is the recommended attribute to use consistently?

Hi User,

The choice of attribute depends on the behavior of the website. If the website elements are dynamically changing or if AJAX loaders are used based on the input element, we recommend utilizing the “Dynamic Resilience” option. This option allows you to choose and detect surrounding elements to identify and interact with the native element.

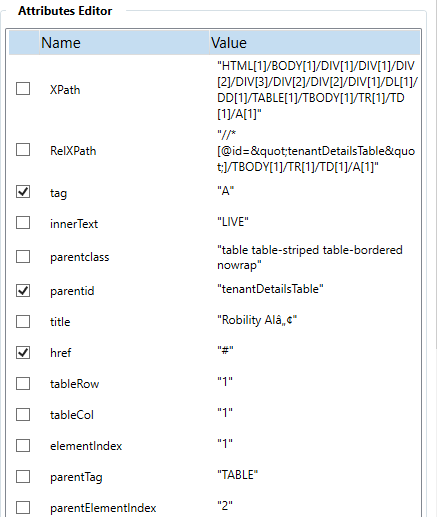
Additionally, using “XPath” can help you navigate to the root and identify elements more easily.

Please let me know if you need further details.

Reply to Question: Thanks for the advice. Since the website will be dynamically changing, is it recommended to use “XPath” with surrounding elements enabled?

No, if the website is dynamically changing, you should avoid using “XPath” as the primary attribute since it can also be affected by these changes. Instead, you can use surrounding elements along with “InnerText” or another stable attribute to precisely identify the value.

You can choose the attributes from the below that opts for the solution.



3. Is there any alternative methods for retrieving the attribute, such as using different activities or custom scripts in WebAutomation?

Yes, alternative methods for retrieving attributes can be considered if the "Get Attributes" activity does not meet your needs.

1. CheckAppState - This activity helps you to check whether the element is available or not on the specified website. And returns the result in Boolean values. This is similar to “Element Exist” activity, but in CheckAppState activity you do not need additional “IF, Else” condition activity.

It provides you with two branches that will lead to perform actions based on the availability of the element.

1. ExtractData - This activity helps you to extract data from web applications that have a similar layout and structure across multiple pages. It helps in retrieving the value with URL from the website as output.
2. GetText – This activity helps you in extracting the text value from the website.

If you are searching for any additional input, please provide more information.

Reply Question: Hey, thanks for the suggestions! I am particularly dealing with a dynamically changing website. Is there any other activity or approach that can handle such scenarios more effectively?

To handle such scenarios, you can use “Retry” scope. This activity helps you to execute a loop and delay with specified time limit to wait for the element to become available. You can checkout there documentation available on the Docs portal to view the detailed information.

ReplyQuestion: Thanks once again, on the comparative case with “Robility6” and “Robility Enterprise” version activity, there is one such activity as “Inject JS”. Will this activity help me out on the aspect mentioned above?

As you said, on the comparative case, the “Robility Enterprise” is more advanced than the “Robility6” and which is why “Inject JS” activity has been deprecated in the enterprise version. When you keep on injecting your custom scripts, some client’s websites might not have compatibility to pull those scripts and load everything. This actually can result in timeout and slowness issues.

On the other hand, for the best practice, it is recommended to use “Robility Enterprise” solutions for your automation needs.

Reply question: Ok, understood.

4. Hi, Does “WebAutomation” support nested elements within the HTML table? And if supported, how are the elements extracted ?

Hello User, yes, the “HTMLTabletDatatable” activity will help you to extract the elements from the nested table but it will not extract the elements from another element at the same time. You will have to repeat the process to get the nested elements from the table.

Reply Question: Thanks, If there any merged cells (<td> with rowspan or colspan attributes), will that affect the data extraction?

ReplyQuestion: If that’s the case, I am facing the below issue as provided in the image, my table contains merged cells, but after extraction when I view it in “Table viewer”, the columns and rows collapse. Can you help me with it?

