



Blue Prism Labs

Lab 6: Advanced Spying

Document Revision 1.0



Trademarks and copyrights

The information contained in this document is the proprietary and confidential information of Blue Prism Limited and should not be disclosed to a third party without the written consent of an authorised Blue Prism representative. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying without the written permission of Blue Prism Limited.

© Blue Prism Limited, 2001 – 2019

®Blue Prism is a registered trademark of Blue Prism Limited

All trademarks are hereby acknowledged and are used to the benefit of their respective owners.
Blue Prism is not responsible for the content of external websites referenced by this document.

Blue Prism Limited, 2 Cinnamon Park, Crab Lane, Warrington, WA2 0XP, United Kingdom
Registered in England: Reg. No. 4260035. Tel: +44 870 879 3000. Web: www.blueprism.com

Contents

Trademarks and copyrights	2
Contents	3
Introduction	4
Lab 6: Advanced Spying.....	5

Introduction

In Lab 1, we learned how to spy elements of an application using Application Modeller. Application details are drawn from the application as a list of attributes. These attributes are not data created by Blue Prism; attributes are characteristics of the element provided to Blue Prism by the application. Once captured, this information can enable a Business Object to “remember” an element, and it does this by using a selection of attributes as the “fingerprint” of an element. This fingerprint is then used to identify and interact with the element from the Object Studio diagram.

Application Modeller retrieves all available attributes from the application but typically only some of these are used to form the element fingerprint. The selected attributes are shown in the “Match” column and can be checked or un-checked as required. The initial selection is a default choice made when the element is first spied, and although this first choice is often correct, it may well require manual adjustment to make a unique fingerprint.

There are many types of element (buttons, text boxes, etc.) used by many kinds of applications (thick-client, thin-client, mainframe, etc.), each with a different set of attributes. Blue Prism cannot anticipate the construction of every kind of application it will ever encounter, and as such it is important to realize the initial selection of attributes made by Application Modeller is a suggestion and not an instruction.

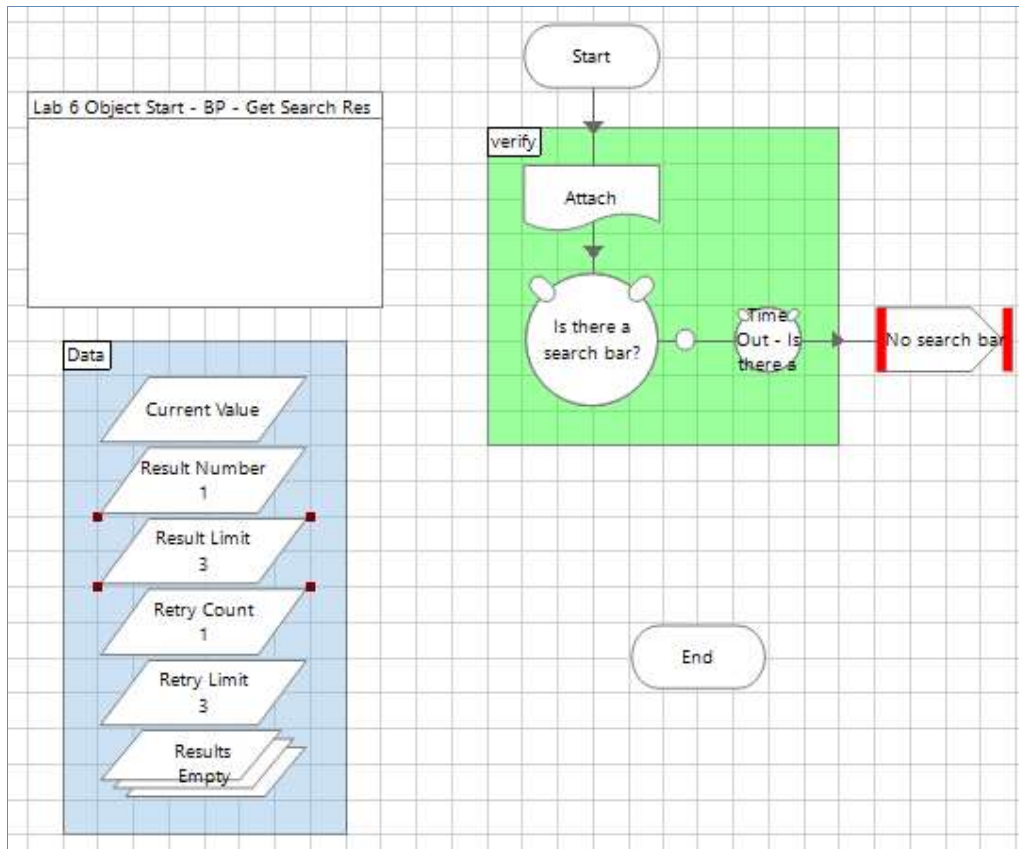
Application Modeller cannot know for certain which combination of attributes will form a unique element fingerprint but can posit a suggestion based on the type of element and type of application. The responsibility for finding the unique combination ultimately rests with you, the creator of the Business Object.

Spying items inside the Application Modeller is best done at the most native format as the application will allow. However, you don’t always want to use the attributes that are checked by default when you identify an element.

In this Lab we will go over some more details on Spying and attribute selection.

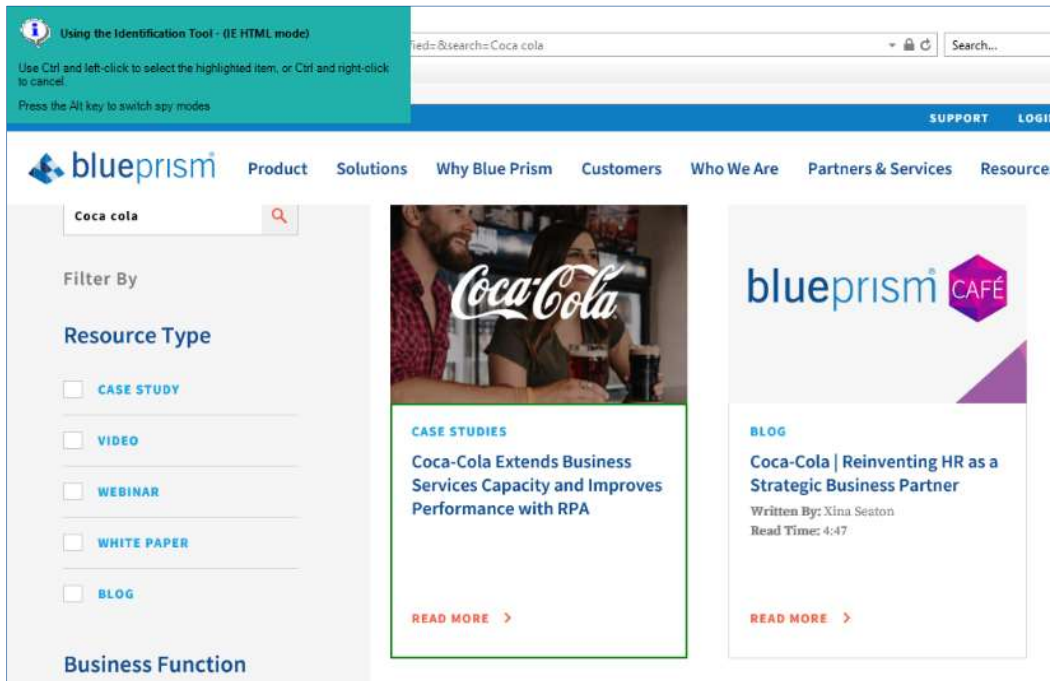
Lab 6: Advanced Spying

- 1) Open the Object called “Lab 6 Object Start - BP” and make a new page called “Get Search Results”.
- 2) Create a Data Item with type “Text” and name it “Current Value”. Next, create 4 more Data Items with type “Number”. Name them “Result Number”, “Result Limit”, “Retry Count”, and “Retry Limit” with initial values of 1, 3, 1, and 3 respectively. Finally, create a Collection and name it “Results”.
- 3) In accordance with best practices, add the “Verify” section after “Start”. To save time, you can copy this from the “Search” Page and paste it here. Your action should now look like this:

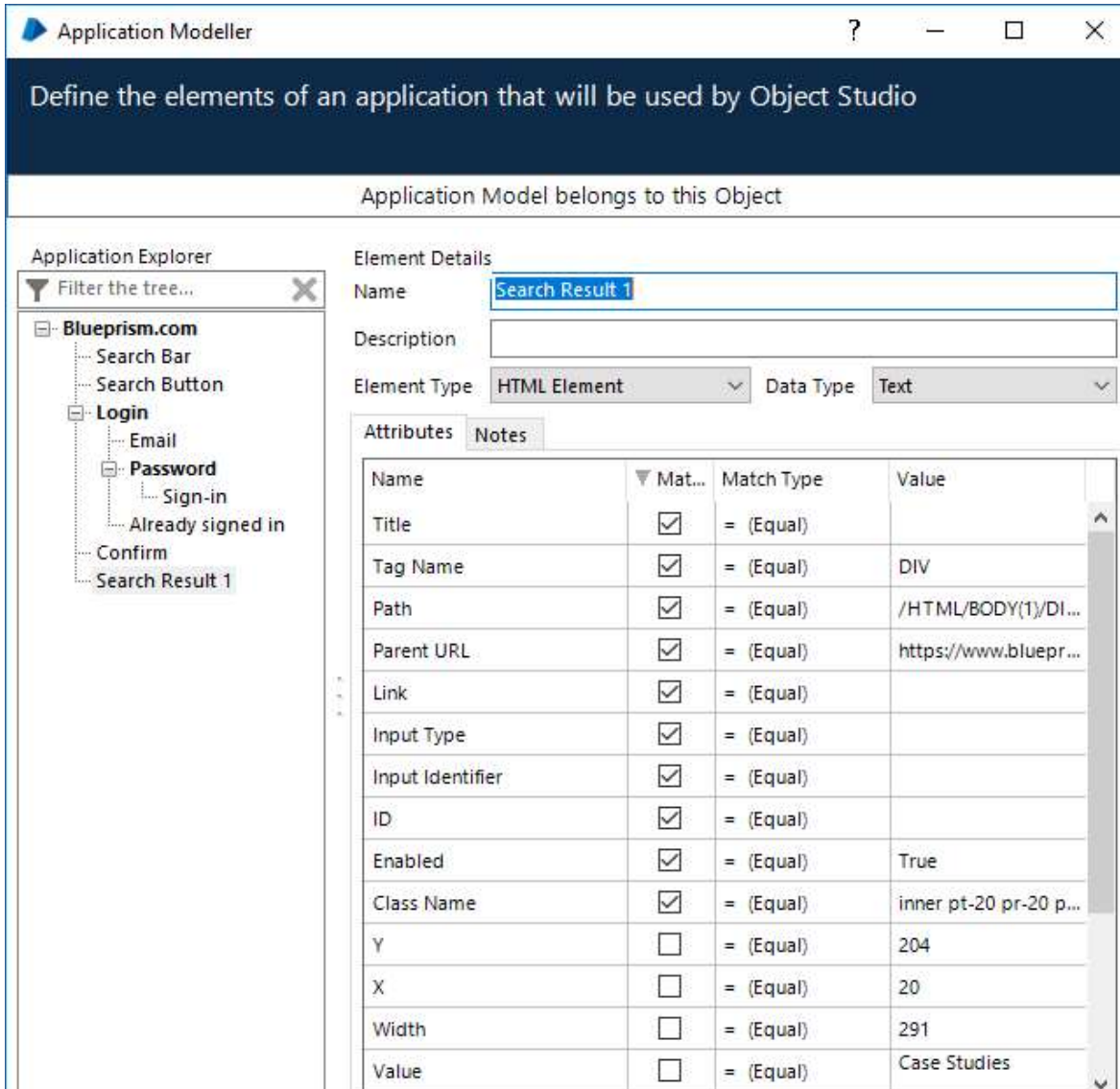


- 4) Now that we have the basics set up, it's time to get to the good stuff! The purpose of this action is to read the search results, and in order to do that we will need to Spy them in the Application Modeler. That said, before we can use Application Modeler we need the application open. Go to the “Launch” page and the “Search” Page and run them each by pressing “Reset” and then “Run” (upper left corner). When that completes, go back to the “Get Search Results” Page we were working on previously.

- 5) Open the Application Modeler. Add a new Element and name it “Search Result 1”. Use the “Identify” button to Spy the block panel of the first search result. When you’re done, click “Highlight” to confirm that your Spying worked.



- 6) Click “Match”, located near the middle of the Application Modeler screen, to quickly see which Attributes you’re using to identify this element. Notice that Values for some of these, such as “Title”, are blank. Since the value is blank, they cannot help you in identification. Uncheck them.



The screenshot shows the 'Application Modeller' window with the title 'Define the elements of an application that will be used by Object Studio'. The 'Application Explorer' on the left shows a tree structure for 'Blueprism.com' with elements like 'Search Bar', 'Search Button', 'Login', 'Email', 'Password', 'Sign-in', 'Already signed in', 'Confirm', and 'Search Result 1'. The 'Element Details' panel on the right shows the details for 'Search Result 1'. The 'Name' field is 'Search Result 1', the 'Description' is empty, the 'Element Type' is 'HTML Element', and the 'Data Type' is 'Text'. The 'Attributes' tab is selected, showing a table of attributes with their match status and values.

Name	Mat...	Match Type	Value
Title	<input checked="" type="checkbox"/>	= (Equal)	
Tag Name	<input checked="" type="checkbox"/>	= (Equal)	DIV
Path	<input checked="" type="checkbox"/>	= (Equal)	/HTML/BODY{1}/DI...
Parent URL	<input checked="" type="checkbox"/>	= (Equal)	https://www.bluepr...
Link	<input checked="" type="checkbox"/>	= (Equal)	
Input Type	<input checked="" type="checkbox"/>	= (Equal)	
Input Identifier	<input checked="" type="checkbox"/>	= (Equal)	
ID	<input checked="" type="checkbox"/>	= (Equal)	
Enabled	<input checked="" type="checkbox"/>	= (Equal)	True
Class Name	<input checked="" type="checkbox"/>	= (Equal)	inner pt-20 pr-20 p...
Y	<input type="checkbox"/>	= (Equal)	204
X	<input type="checkbox"/>	= (Equal)	20
Width	<input type="checkbox"/>	= (Equal)	291
Value	<input type="checkbox"/>	= (Equal)	Case Studies

- 7) We want to make this object generic. With that in mind, we cannot use the Parent URL as an identifier. Uncheck it as well. Click “Highlight” again to show that it still works.

Note: Other attributes you will almost always want to make sure are not checked are “Visible”, “Screen Visible”, “Parent Visible”, and “Enabled” to make your spying more resilient.

- 8) Repeat steps 5-7 for Search Result 2, identifying the second search result.
- 9) We don’t know how many results there will be, so Spying each search result wouldn’t make sense. Instead, let’s compare the Attributes from “Search Result 1” to the Attributes from “Search Result 2”. Look at all the Attributes with Values, not just the Attributes that are checked. What differences can you find?

- 10) You should notice several differences. For example, the “Value” Attribute contains the text of the result. Since we are trying to determine this, we cannot use it as an identifier. The “Path” Attribute will be slightly different, and could be used to identify the Element, but parsing through the Paths is more work than we need to do here. Uncheck it. With only “Tag Name” and “Class Name” left checked, there is no way to differentiate which result we’re looking at. Press “Highlight” and watch as all the results are highlighted. You should get an error saying there are multiple matches. Look at the “Match Index” Attribute. This Attribute describes the order elements are found. Let’s try and use this!
- 11) Check “Match Index”. Next, uncheck “Path” (if checked). You should only have 2 boxes checked now, as shown below. Click “Highlight” to confirm that it’s still working. Note, it may take a while the first time.

Attributes			
Notes			
Name	Mat...	Match Type	Value
Tag Name	<input checked="" type="checkbox"/>	= (Equal)	DIV
Match Index	<input checked="" type="checkbox"/>	= (Equal)	1
Class Name	<input checked="" type="checkbox"/>	= (Equal)	inner pt-20 pr-20 p...
Y	<input type="checkbox"/>	= (Equal)	204
X	<input type="checkbox"/>	= (Equal)	20

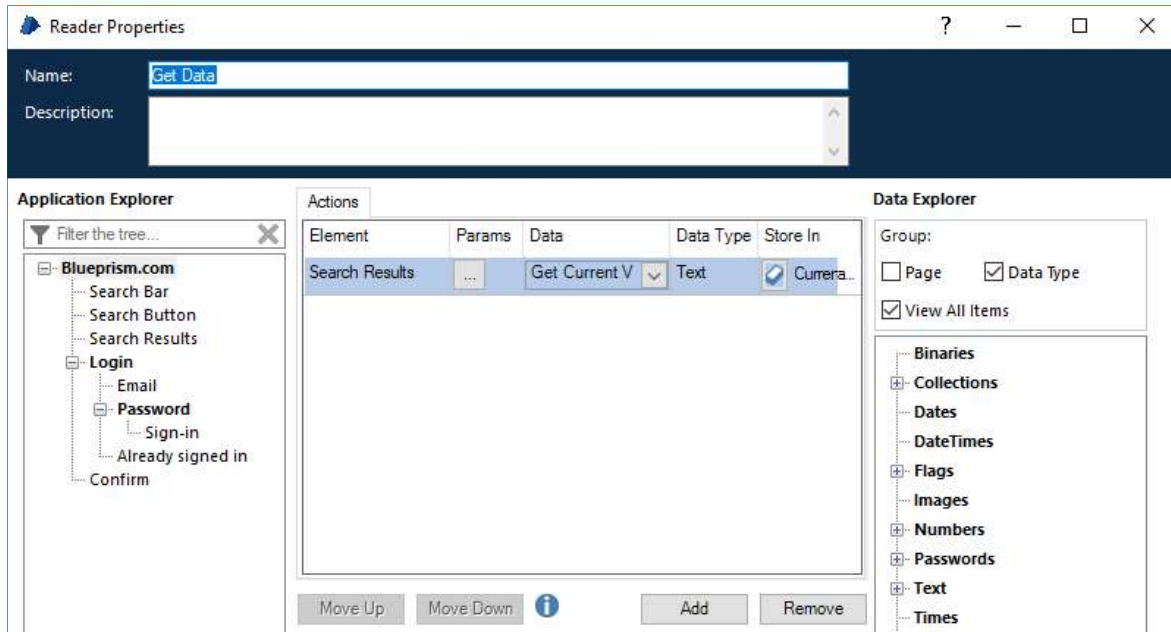
Note: In general, it is best practice to always check “Match Index” as it will help speed up your automations.

- 12) Try changing the Value of “Match Index”. All you have to do is click on it. Change it from a “1” to a “2” and then click “Highlight”. Notice that the SECOND result is highlighted! Try changing it again to a “3”. Now notice that the THIRD result is highlighted.
- 13) We want to be able to use this! Click on the “Match Type” for “Match Index” and change it to “Dynamic” as shown below. This will allow you to change the Value from the Object as needed in your business logic or action.

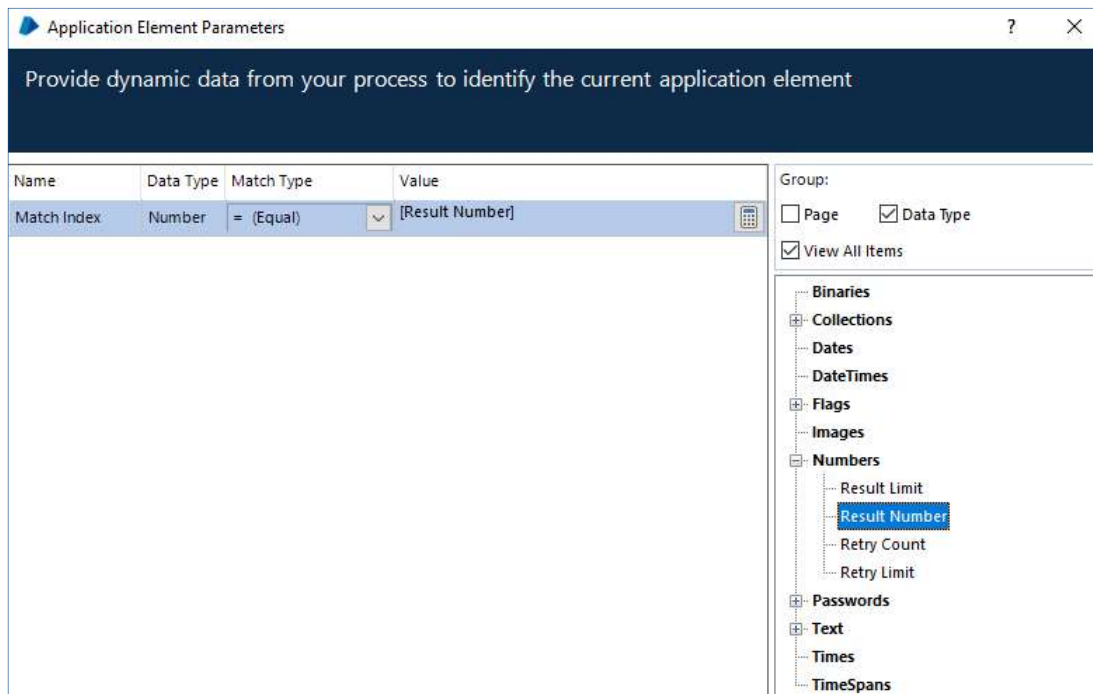
Attributes			
Notes			
Name	Mat...	Match Type	Value
Tag Name	<input checked="" type="checkbox"/>	= (Equal)	DIV
Match Index	<input checked="" type="checkbox"/>	Dynamic	1
Class Name	<input checked="" type="checkbox"/>	= (Equal)	inner pt-20 pr-20 p...
Y	<input type="checkbox"/>	= (Equal)	204
X	<input type="checkbox"/>	= (Equal)	20

- 14) We only need the one element now, so change its name to “Search Results”. Next, right click the other Search Result and select “Delete”. Press “OK” to close the Application Modeller.

- 15) Add a Read stage and open its properties. Name it “Get Data” and set it up as shown below:

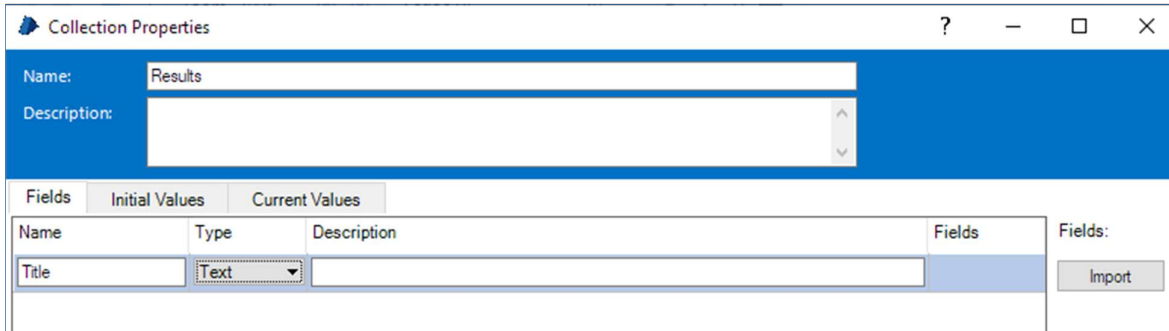


- 16) This will get the “Current Value” Attribute, which we saw in step 10 was the result title, and store it in the Data Item “Current Value”. However, we still have to specify which result we want it to read. Click on the three dots (...) under where it says “Params”. This will open a new window where you can specify the value of “Match Index” to be used. Set it to be the Data Item “Result Number”.



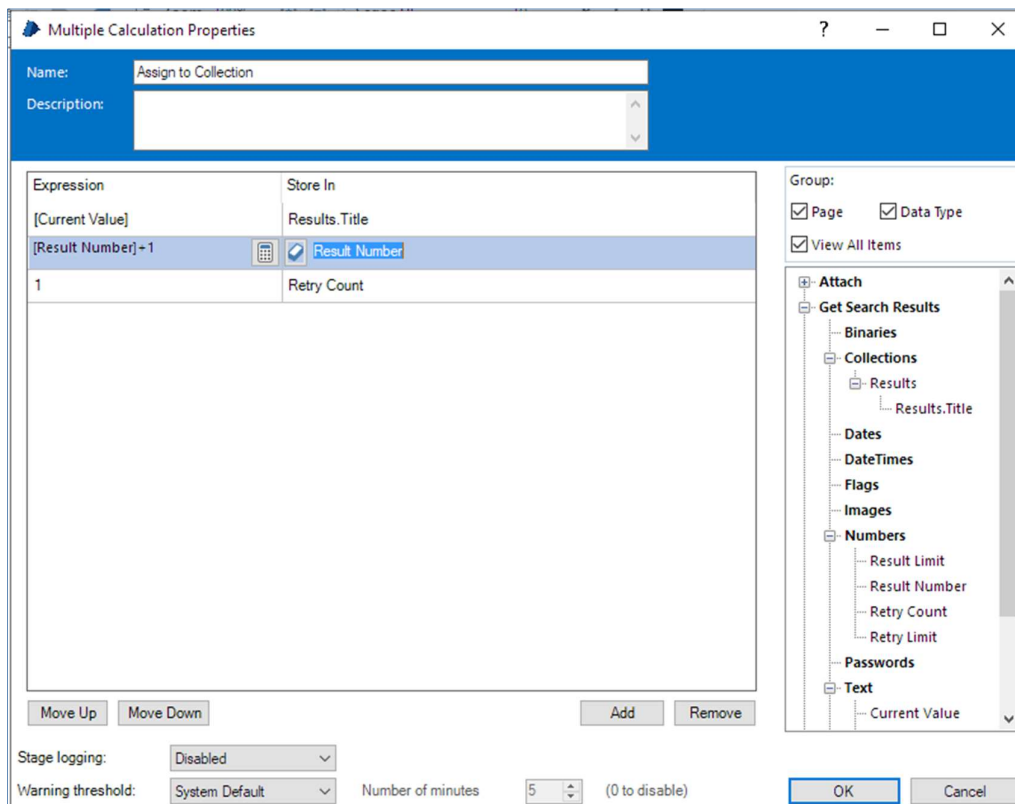
- 17) Press “OK” to exit this screen, then press “OK” to exit the Read stage properties.

- 18) Before we can record the results to the collection, we need to add a row to the collection. Add an Action and open its properties. Name it "Add Row". For "Business Object", scroll to the bottom and select "Collections". For "Action", select "Add Row". Finally, you must specify what collection to add a row to. Under "Value" enter "Results", which is the name of your collection. Don't forget the quotes! Press "OK" to exit.
- 19) Open the properties of the collection "Results". Click "Add Field", located in the bottom right, and add a "Text" field called "Title" as shown below. Press "OK" to exit.



Name	Type	Description	Fields
Title	Text		

- 20) Add a "Multi Calc" stage and name it "Assign to Collection". Set it up to store "Current Value" in the collection and to also update the "Result Number". (Don't worry about the third line in the picture below, Retry Count, it related to step 23!)



Expression	Store In
[Current Value]	Results.Title
[Result Number] + 1	Result Number
1	Retry Count

Group: ☒ Page ☒ Data Type ☒ View All Items

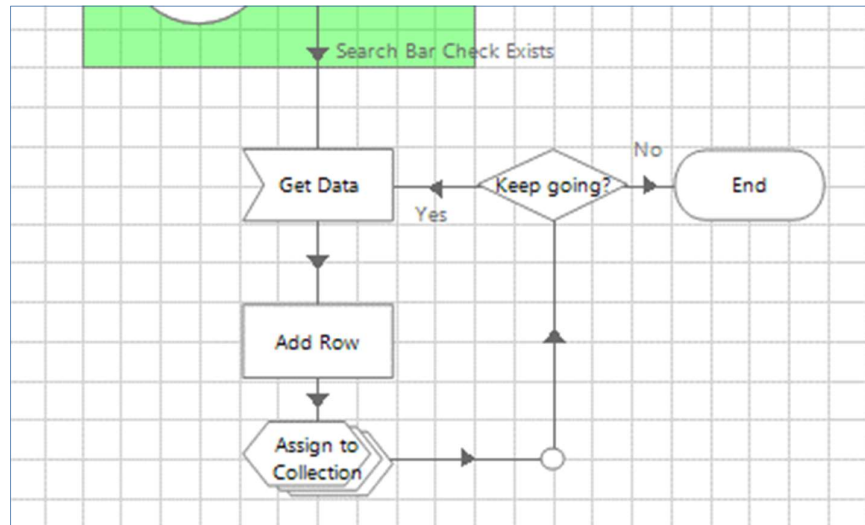
Attach: ☒ Get Search Results ☒ Binaries ☒ Collections ☒ Results ☒ Results.Title

Text: ☒ Current Value

Stage logging:

Warning threshold: Number of minutes: 5 (0 to disable)

- 21) Add a Decision stage and name it “Keep Going?”. Set its expression to “[Result Number]<=[Result Limit]”. Add an anchor and link it all together as shown below.



- 22) Open the properties of “End” and add “Results” as an output.

End Properties

Name: End2

Description:

Outputs

Name	Description	Data Type	Get Value From
Results		Collection	Results

Group:

- ☒ Page
- ☒ Data Type
- ☒ View All Items
- ☒ Attach
 - Get Search Results
 - Binaries
 - Collections
 - Results
 - Results.Title
 - Dates
 - DateTimes
 - Flags
 - Images
 - Numbers
 - Result Limit
 - Result Number
 - Retry Count
 - Retry Limit
 - Passwords
 - Text
 - Current Value

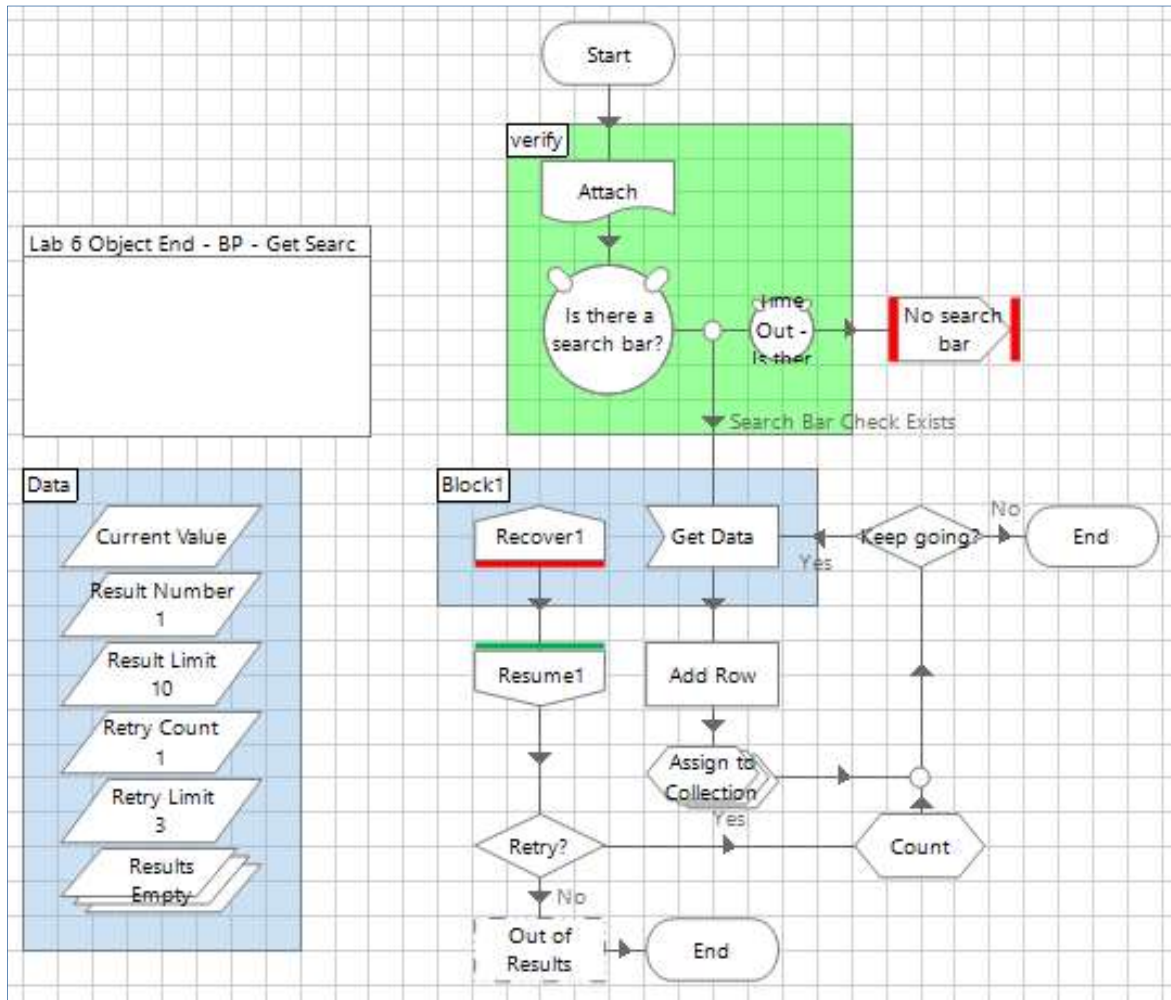
Stage logging: Disabled

Warning threshold: System Default

Number of minutes: 5 (0 to disable)

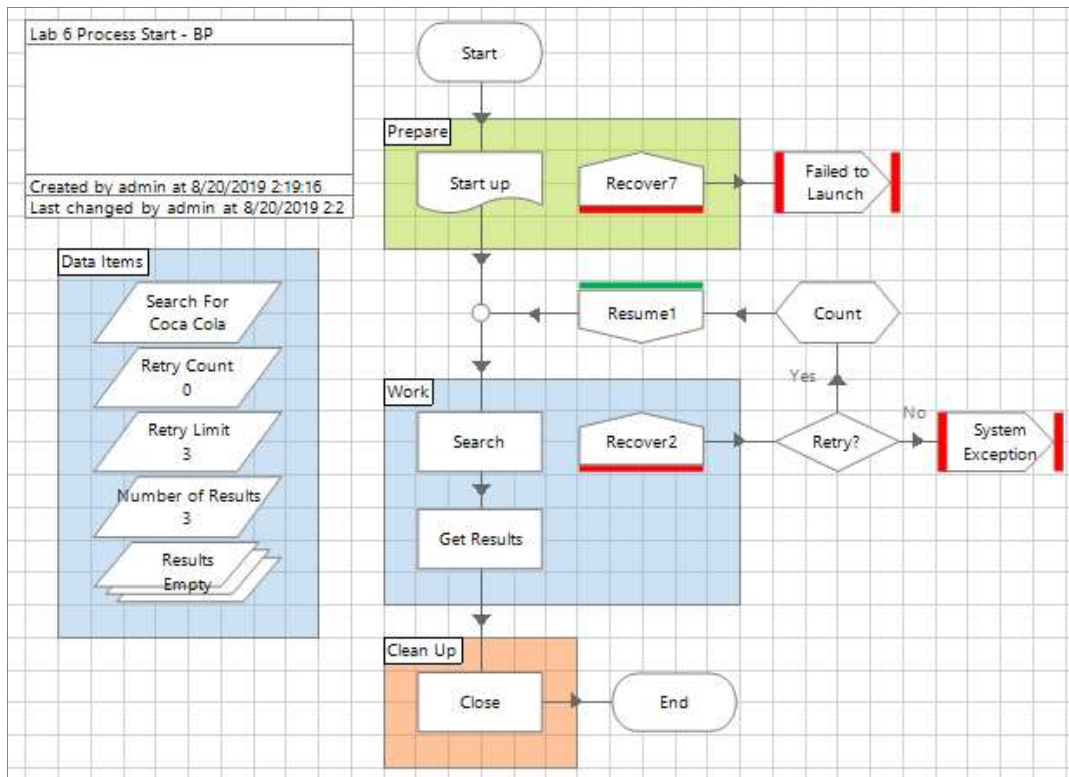
OK Cancel

- 23) At this point the Action will work, returning the top 3 search results. However, what if there are only 2 results? Currently, the Action would crash. Since you are an expert in retry loops after Lab 4, set up a retry loop as shown below. Don't forget to reset "Retry Count" to "1" when you advance to the next result! Hint: Just add a calculation to the Multi Calc stage that stores "1" in "Retry Count"



- 24) Open the properties of "Start" and set "Result Limit" as an input. This way you can tell the Object how many results you want to grab.
- 25) Publish the action by right clicking on the page name and selecting "Publish". This is necessary to call the action from your process. Save and close the object.
- 26) Back in the Process (Lab 6 Process Start – BP), add a number Data Item called "Number of Results" and give it an initial value of 3. Next, add a collection called "Results". Finally, add an action to the "Work" Block after "Search". Name it "Get Results" and set it to call "Get Search Results". Link it together. Set the input and output to the Data Item and Collection you just created.

27) Your process should now look like this:



28) Click the “Reset” button in the upper left:



29) Click the “Go” button in the upper left:



30) Watch your process run. When it’s done, check the collection to view the results! Feel free to close the browser, the object, and the process.

Note: You have now learned how to modify attributes when spying to identify the “fingerprint” of elements in your target applications. You also used the dynamic nature of attributes to loop through the results from Blueprism.com and capture the number desired which you can set in the business logic of your process. This makes your object very flexible and can be used with any process – and each process can control how many results it needs to capture!