



Blue Prism Labs

Lab 7: MS Excel VBO

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 7: MS Excel VBO	5

Introduction

Blue Prism is packaged with many Visual Business Objects, or VBOs, which provide feature rich extended capabilities to your automations, in addition to the standard Blue Prism drag and drop features.

The purpose of having Objects separated from Processes is to maximize their reusability. With that in mind, there are many VBOs that come pre-built for no additional charge (Blue Prism never charges for anything other than the license) so that users don't have to reinvent the wheel. These pre-built VBO's include integrations with MS Excel, Word, Outlook, extending web services functionality and utilities for file, image, date/time manipulation, etc.

Additional VBO's from 3rd party providers (business partner or technology providers) are also available in the Digital Exchange.

In this lab, you will be introduced to the MS Excel VBO.

Lab 7: MS Excel VBO

Prep: Excel must be downloaded and installed prior to working these steps. Please install Excel using this link and your MS office account: https://support.office.com/en-us/article/download-and-install-or-reinstall-office-365-or-office-2019-on-a-pc-or-mac-4414eaaf-0478-48be-9c42-23adc4716658#InstallSteps=Install_on_a_PC

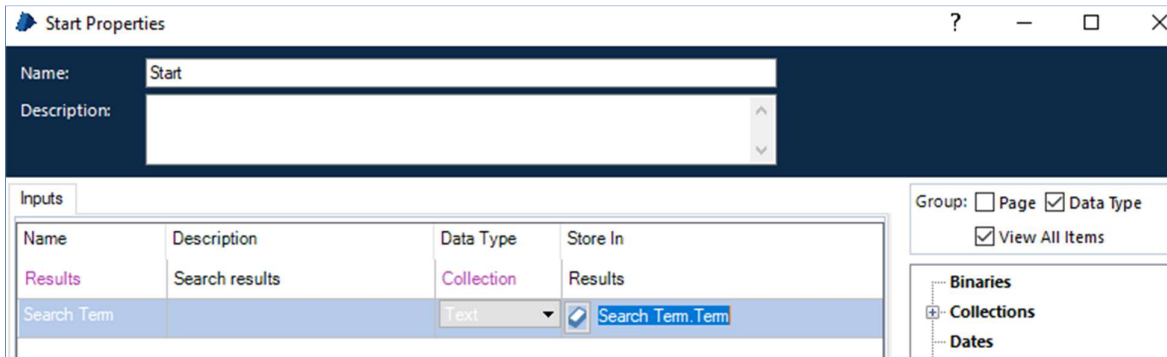
- 1) Now that we have an action to read the search results, let's make a way to record them. Open up the Process called "Lab 7 Process Start - BP" and make a new page called "Record in Excel".
- 2) Download the attached Excel sheet "Results.xlsx" and store it in "C:\temp\Results\". If you would prefer to save it someplace else, that's fine, just make a note of the file path. You will need it in the next step!
- 3) On the new page you just created, create a Data Item with type "Number" and name it "Handle". Next, create 4 more Data Items with type "Text". Name them "File Path", "Cell Reference", "Workbook Name", and "Worksheet Name" with initial values of "C:\temp\Results\Results.xlsx", "A1", "Results", and "Results" respectively. Finally, create 2 Collections and name them "Results" and "Search Term"
- 4) Open the properties of "Search Term". In the bottom right, click "Add Field". Set the name to "Term" and the type to "Text".

Name	Type	Description
Term	Text	

Click the "Initial Values" tab and set an initial value by clicking "Add" in the bottom right hand corner and adding any value in the added row (It doesn't matter what, as it will be overwritten). Press "OK" to close.

Term (Text)
Something Awesome!

- 5) Open the properties of "Start" and add 2 inputs named "Results" and "Search Term". Set their types to "Collection" and "Text" respectively and store them in "Results" and "Search Term.Term".



Start Properties

Name: Start

Description:

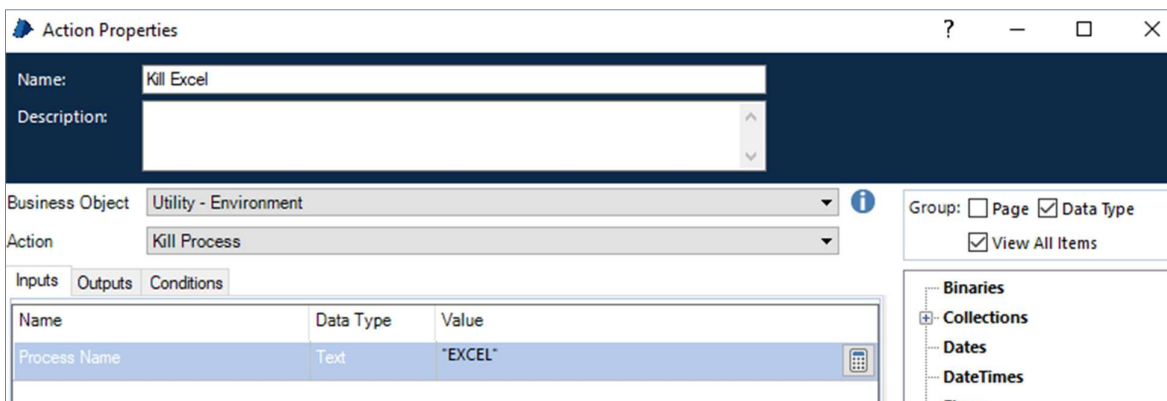
Inputs

Name	Description	Data Type	Store In
Results	Search results	Collection	Results
Search Term		Text	Search Term.Term

Group: ☐ Page ☒ Data Type
☒ View All Items

Binaries
 Collections
 Dates

- 6) Now that we have the basics set up, it's time to get to the good stuff! The purpose of this Lab is to give you a taste of the Excel VBO, so go ahead and add 10 actions and link them all together. 9 of these 10 actions will be calling an action from the excel VBO!
- 7) Before we dive into Excel, we want to make sure there aren't any instances already open. Open the first action's properties and name it "Kill Excel". For "Business Object" select "Utility – Environment" and for "Action" select "Kill Process". Set the Value of "Process Name" to "EXCEL". Don't forget the quotation marks! Press "OK" to close.



Action Properties

Name: Kill Excel

Description:

Business Object: Utility - Environment

Action: Kill Process

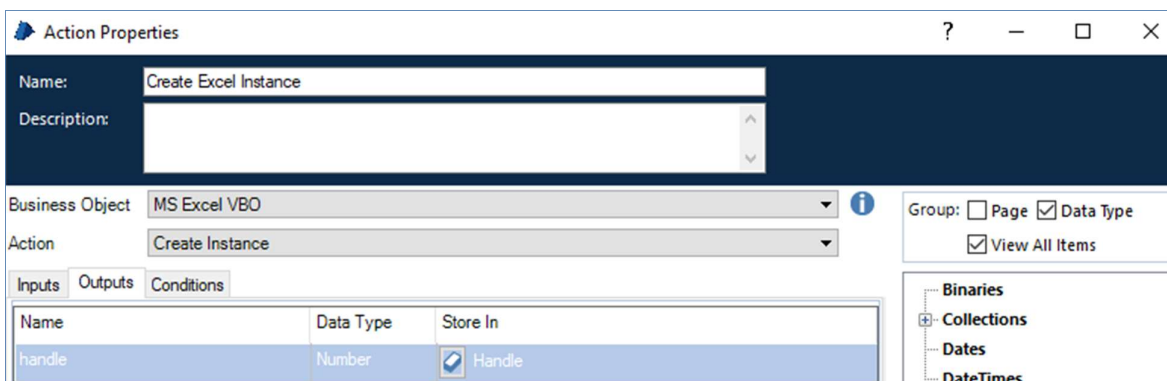
Inputs

Name	Data Type	Value
Process Name	Text	"EXCEL"

Group: ☐ Page ☒ Data Type
☒ View All Items

Binaries
 Collections
 Dates
 DateTimes

- 8) Next, we want to create an Excel instance. Open the second action's properties and name it "Create Excel Instance". For "Business Object" select "MS Excel VBO" and for "Action" select "Create Instance". Set the Value of "Enable Elements" input to "False". Set the output "handle" to be stored in "Handle". Press "OK" to close.



Action Properties

Name: Create Excel Instance

Description:

Business Object: MS Excel VBO

Action: Create Instance

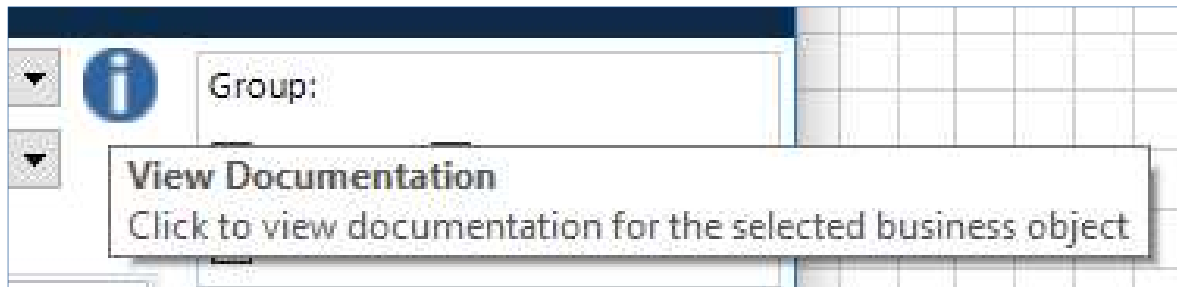
Inputs

Name	Data Type	Store In
handle	Number	Handle

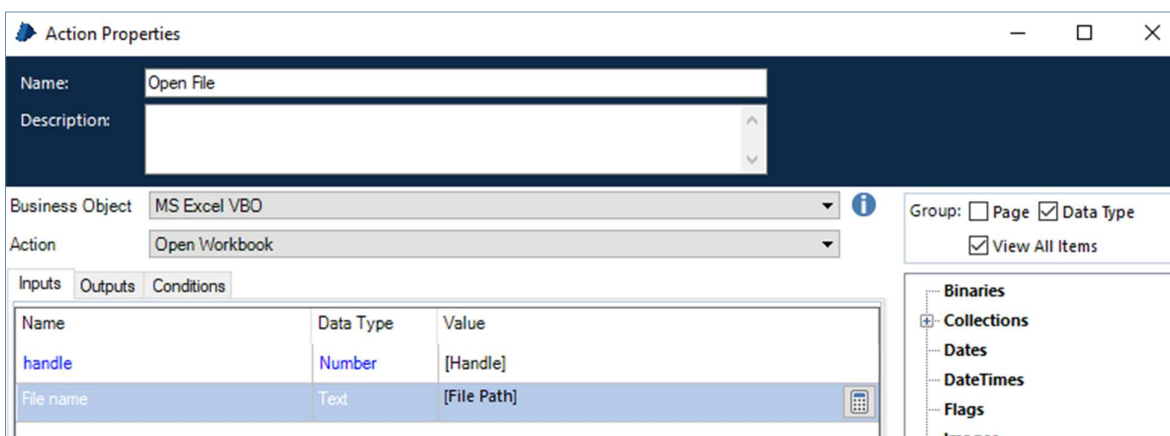
Group: ☐ Page ☒ Data Type
☒ View All Items

Binaries
 Collections
 Dates
 DateTimes

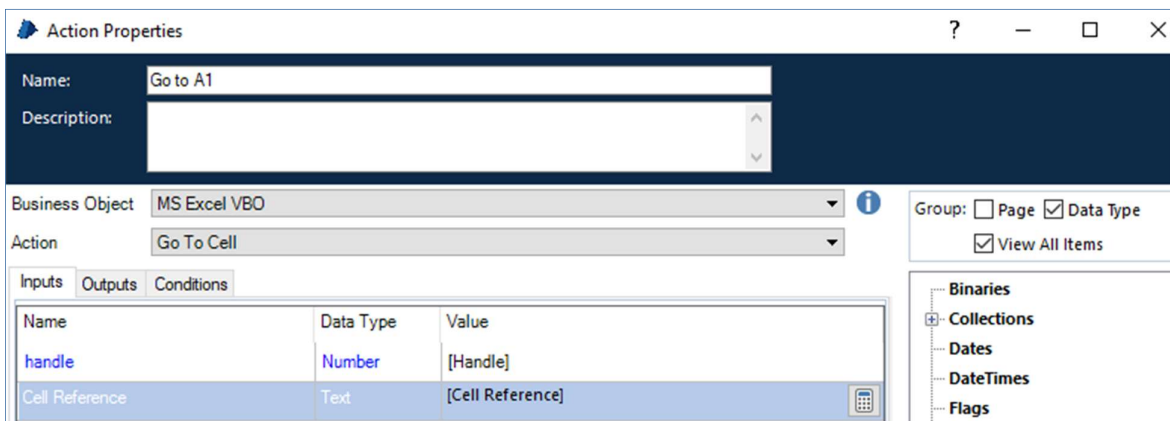
- 9) Following instructions is fine for a lab, but what about when you are creating your own processes? How will you figure out what inputs to send to Objects that came preloaded? Fortunately, there is excellent documentation to help! Next to the Business Object selection, you will notice a blue circle with the letter “i” in it. Click that. A browser should open with information on all the Actions included in the MS Excel VBO. Leave this open, as it will be a helpful reference as you continue through this lab!



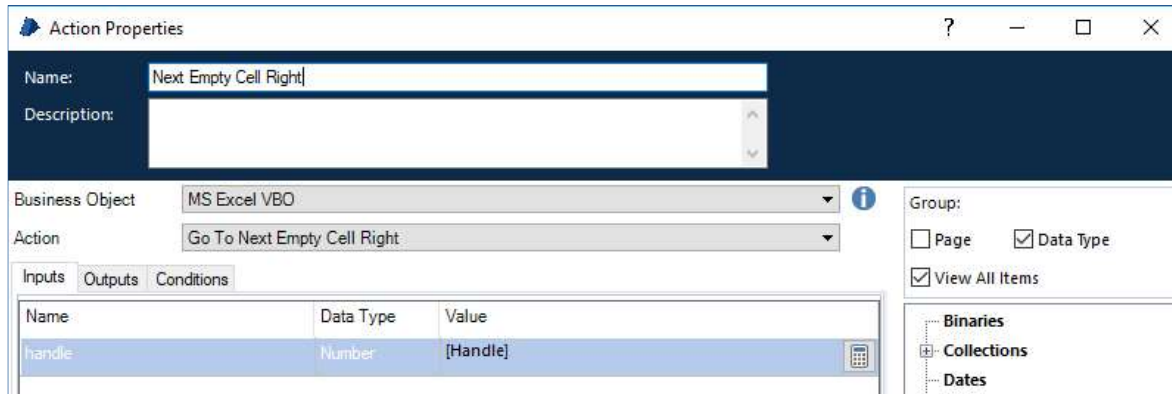
- 10) Next, we want to open the workbook. Open the third action's properties and name it "Open File". For "Business Object" select "MS Excel VBO" and for "Action" select "Open Workbook". Set the input Values of "handle" and "File Name" to "[Handle]" and "[File Path]" respectively. Set the output "Workbook Name" to be stored in "Workbook Name". Press "OK" to close.



- 11) Open the fourth action's properties and name it "Go to A1". For "Business Object" select "MS Excel VBO" and for "Action" select "Go To Cell". Set the input Values of "handle" and "Cell Reference" to "[Handle]" and "[Cell Reference]" respectively. Press "OK" to close.

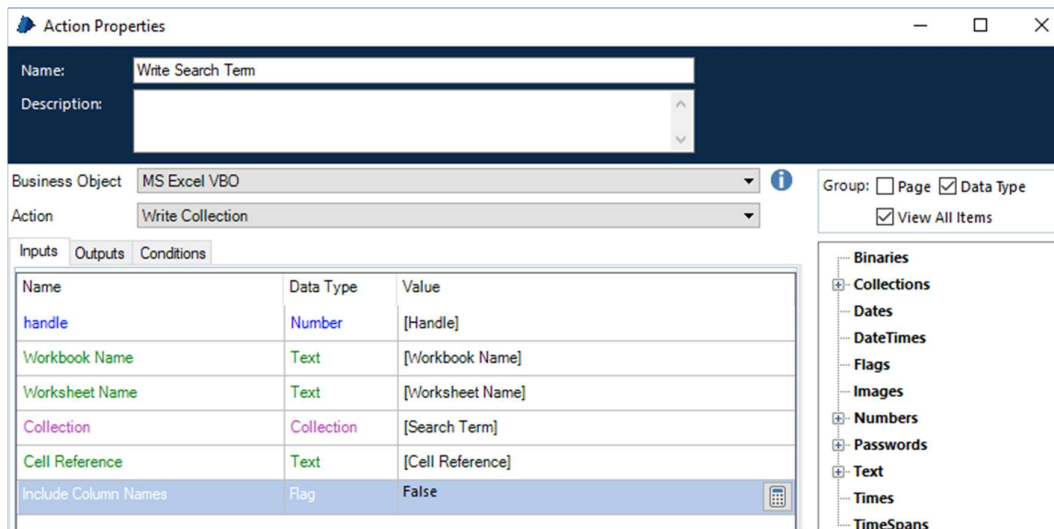


- 12) Next, we want to navigate to the next open column. Open the fifth action's properties and name it "Next Empty Cell Right". For "Business Object" select "MS Excel VBO" and for "Action" select "Go To Next Empty Cell Right". Set the input Value of "handle" to "[Handle]". Set the output "Cell Reference" to be stored in "Cell Reference". Press "OK" to close.



The screenshot shows the 'Action Properties' dialog for the action 'Go To Next Empty Cell Right'. The 'Name' field is 'Next Empty Cell Right'. The 'Business Object' is 'MS Excel VBO'. The 'Action' is 'Go To Next Empty Cell Right'. The 'Inputs' tab is selected, showing a table with one input: 'handle' of type 'Number' with value '[Handle]'. The 'Outputs' tab is also visible, showing 'Cell Reference' as an output. The 'Conditions' tab is empty. On the right, the 'Group' section has 'Data Type' and 'View All Items' checked. A list of categories (Binaries, Collections, Dates) is shown on the far right.

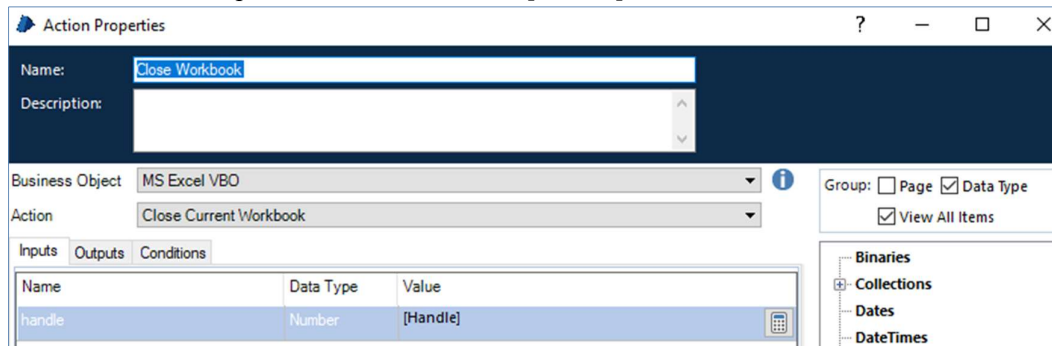
- 13) Next, we write the search term to the workbook. Open the sixth action's properties and name it "Write Search Term". For "Business Object" select "MS Excel VBO" and for "Action" select "Write Collection". There are a lot of Inputs for this one, but by now you should be getting the hang of this. Fill out the Values as shown below and then press "OK" to close.



The screenshot shows the 'Action Properties' dialog for the action 'Write Collection'. The 'Name' field is 'Write Search Term'. The 'Business Object' is 'MS Excel VBO'. The 'Action' is 'Write Collection'. The 'Inputs' tab is selected, showing a table with six inputs: 'handle' (Number, [Handle]), 'Workbook Name' (Text, [Workbook Name]), 'Worksheet Name' (Text, [Worksheet Name]), 'Collection' (Collection, [Search Term]), 'Cell Reference' (Text, [Cell Reference]), and 'Include Column Names' (Flag, False). The 'Outputs' tab is also visible. The 'Conditions' tab is empty. On the right, the 'Group' section has 'Data Type' and 'View All Items' checked. A list of categories (Binaries, Collections, Dates, DateTimes, Flags, Images, Numbers, Passwords, Text, Times, TimeSpans) is shown on the far right.

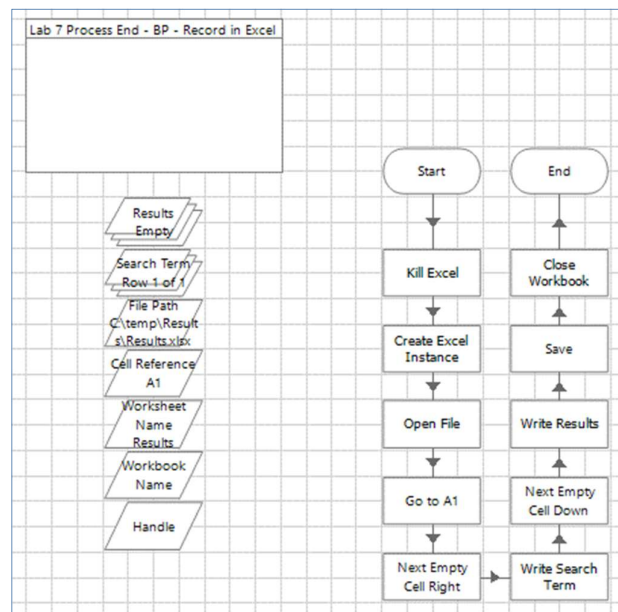
- 14) Next, we want to navigate to below the search term. Open the seventh action's properties and name it "Next Empty Cell Down". For "Business Object" select "MS Excel VBO" and for "Action" select "Go To Next Empty Cell Down". Set the input Value of "handle" to "[Handle]". Set the output "Cell Reference" to be stored in "Cell Reference". Press "OK" to close.
- 15) It's finally time to write the data to Excel! Open the eighth action's properties and name it "Write Results". For "Business Object" select "MS Excel VBO" and for "Action" select "Write Collection". This will be very similar to Step 13 (in fact, copying the Action from Step 13 could save you time). Fill out the Values as shown in Step 13 but enter "[Results]" for the Value of "Collection" rather than "[Search Term]". Press "OK" to close. If you are unsure about any of the inputs, remember the documentation from Step 9!

- 16) With the data entered, we need to save the workbook. Open the ninth action's properties and name it "Save". For "Business Object" select "MS Excel VBO" and for "Action" select "Save Current Workbook". Press "OK" to close.
- 17) Finally, it's time to close the workbook. Open the tenth and last action's properties and name it "Close Workbook". For "Business Object" select "MS Excel VBO" and for "Action" select "Close Current Workbook". Set the input Value of "handle" to "[Handle]". Press "OK" to close.

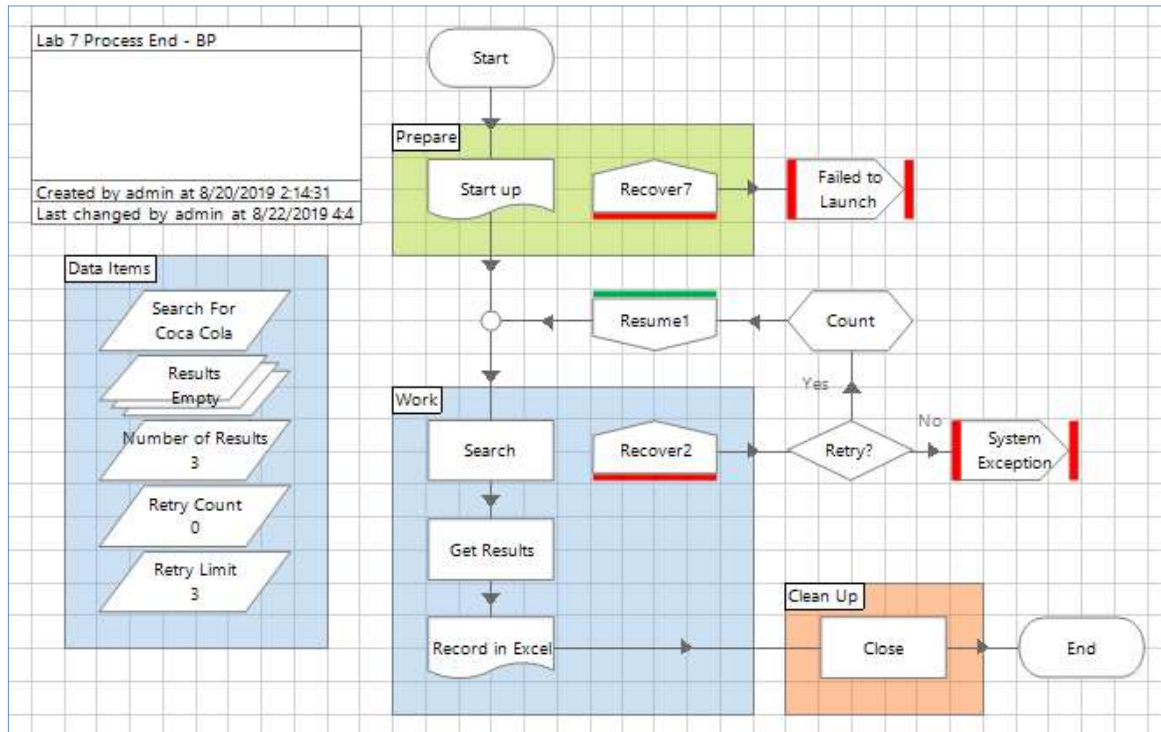


The screenshot shows the 'Action Properties' dialog box for the 'Close Workbook' action. The 'Name' field is set to 'Close Workbook'. The 'Business Object' is 'MS Excel VBO' and the 'Action' is 'Close Current Workbook'. The 'Inputs' tab is selected, showing a table with one input: 'handle' of type 'Number' with value '[Handle]'. The 'Outputs' and 'Conditions' tabs are also visible. On the right, there are checkboxes for 'Page', 'Data Type', and 'View All Items', and a list of 'Binaries' including 'Collections', 'Dates', and 'DateTimes'.

- 18) Confirm that your page looks like this:



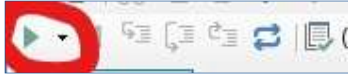
- 19) Go back to the Main Page and add a Page referencing “Record in Excel” (the page you just finished!). Make sure to set your inputs using your Data Items by double clicking the Page Action! Your Main Page should look like the image below. Don’t worry too much about proper exception handling for now, as we’ll wrap that up in Lab 10.



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



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



22) Watch your process run. When it's done, open the Excel file to view the results! Feel free to close both the browser and the process.

Note: In this lab you used actions from an Object you didn't have to build. Many common applications, such as MS Excel, have Objects that have already been built, so there's no need to reinvent the wheel! You can find these Objects by checking the Digital Exchange and by asking your partner. With many Objects built, either by your organization or by others, modelling Processes is as simple as putting the Lego blocks together!
