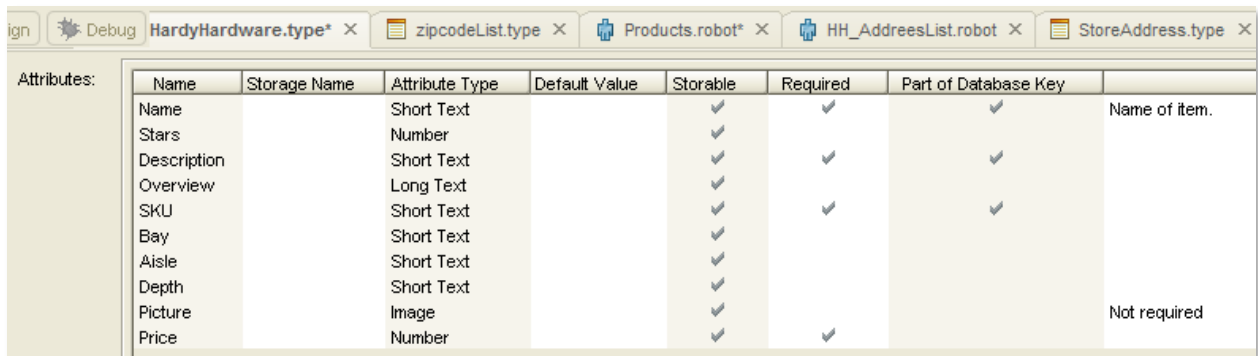# Lab 10-1
Lab – Advanced Repeat/Next Loops

In this lab, you will add two new attributes to your HardyHardware.Type file to accommodate the additional product data you will extract. Specifically, you'll extract the product rating (stars) and the depth of the product in inches. You will create a new Robot because you want to extract that data for ALL products on the HardyHardware web site. This will involve setting up more complex Repeat/Next loops. And you will write that data to the development database.

1. Before you develop your Robot, go to the HardyHardware web site using your browser and note how you, as a human user accomplish what you want to do. Note that to search show products, you have to click on the "Products listing" link on the left of the page. Note too that this brings up a full page of products and that you have to click on the number link at the bottom of the page to bring up the next page of items (there is no [Next] button. Item detail is displayed by clicking on the individual product. You will use this knowledge to build your Robot.

2. If not already running, open Design Studio. Go to the Types folder you created in the Projects panel, select "HardyHardware" (click on its tab if its already open or double-click to open it if it's not).

3. For your HardyHardware.Type, add two new attributes:

   a. Stars   Attribute Type: Number   Required: No   Part of database key: No

   b. Depth   Attribute Type: Short Text   Required: No   Part of database key: No

   When you have completed, reorder your Attributes and review their properties. The list you end up with should look like this:

Name	Storage Name	Attribute Type	Default Value	Storable	Required	Part of Database Key	
Name		Short Text		✓	✓	✓	Name of item.
Stars		Number		✓			
Description		Short Text		✓	✓	✓	
Overview		Long Text		✓			
SKU		Short Text		✓	✓	✓	
Bay		Short Text		✓			
Aisle		Short Text		✓			
Depth		Short Text		✓			
Picture		Image		✓			Not required
Price		Number		✓	✓		

4. Since you will be writing the extracted to the development database, update the database table for the Type. Go to the Tools menu, select "Create Database Table. Select the "objectdb" database and the HardyHardware Type and generate and execute the resulting SQL statement.

5. It's time to build your Robot. But this time, to save one page load, you'll use the Product info web page on the HardyHardware website as the starting URL. Create a new Robot called "Products.robot." For the URL, enter:
   http://class.kofax.com/hardyhardware/index.php/hikashop-menu-for-products-listing

6. Add a variable to the robot using the HardyHardware type.

7. Select the first item. Make sure the green select box includes the *entire* item like this:
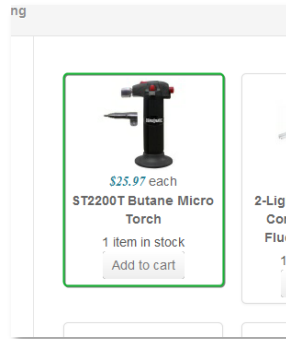
8. Add a "For Each Tag with a Class" Loop step to loop through each item on the page. The Class you'll select in this step is "hikashop_subcontainer."

9. Add a Click step to select the first item (the butane torch). Rename the step to "Click item."

10. *Clicking brings up the item detail where you can set up your extraction. To give you a more advanced challenge, you'll extract the number of Stars for the item as well as the product's Depth (in inches) which is shown in the table. Then you can add the ExtractData snippet you created in an earlier lab to get the rest of the data. Let's start with the product rating or "stars."* To accomplish this, you'll perform your extraction from the HTML source code itself rather than from the picture. But to get your cursor in the right place, start by click on the rating (stars) near the top of the page. Using the arrows on the toolbar of the browser panel, expand the selected area until it look like this:

11. Examine the HTML code in the source panel below. Look for a line (probably just above what is currently selected) that says:

```
<input type="hidden" name="hikashop_vote_rating" data-type="product" data-max="5" data-ref="1701" data-rate="3"
id="select_id_1701_hikashop_main_div_name" value="3">
```

Notice it contains the value of the stars (value="3"). Click on that line of the code to select it. Then right mouse-click and select "Extract," "Attribute," "value," and "HardyHardware.Stars."

12. Go to the "Basic" tab of your new Extract Tag Attribute Step and rename it "Extract Stars." Then go to the "Action" tag enter the "Tag Attribute Name" which will be "value." (If you examine the HTML code in Step 10 above you'll notice that "value=3," exactly what you want to return).

13. *Now you want to extract the assembled depth. But that data exists in a table and it could conceivably move up or down in the table or possibly not be there at all. Since it is associated with a label in the preceding column of the table, "Assembled Depth," you'll want to set that as a Named Tag.* Find and select "Assembled Depth (in.)" in the browser panel, right mouse-click and select "Other" and "Set as Named Tag" from the context

menu. Change the step's name on the Basic tab to "Assembled Depth" to make it obvious what the step is finding.

14. Go to the "Finders" tab of your Assembled Depth step. In the "Tag Pattern" box enter the following pattern as a Value: **.*?Assembled Depth.*?**

15. On the "Action" tab, set the "Tag Name" to "Named" from the dropdown. And enter "Assembled Depth" as the "Name."

16. Now you'll add the extract step. Click on the dimensions for Assembled Depth, in this case "3.0 in" to select it. Right mouse-click and select "Extract," "Text" and "HardyHardware.Depth" from the context menu. Now select your new "Extract Depth" step in the robot view and examine the properties created for you on the "Finders" tab. Notice the data is being extracted from the same row as the named tag...exactly what you want!

17. You also need to clear the named tag so it will not be used in subsequent steps. Add a new Action Step just before the end step. Go to the "Action" tab for the new unnamed step and from the Action dropdown select "Other" and "Clear Named Tags/Ranges."

18. But what if there is no assembled depth in the detail shown in the table? Let's assume you want your Robot to continue with extraction without generating an error. So add a "Try" step just before the "Assembled Depth" step.

19. That will automatically add a branch to your robot with its own end step. Select the end step of the branch and join it to the main end step by dragging it.

20. Go to the "Assembled Depth" step. Select the "Error Handling" tab. Select "Try Next Alternative."   By doing this, if the Robot is not able to find the assembled depth in the table, it will go to the branch step, skipping the extraction and continue on.

21. Finally, for simplicity, group these steps together into a Group called "Extract Depth."
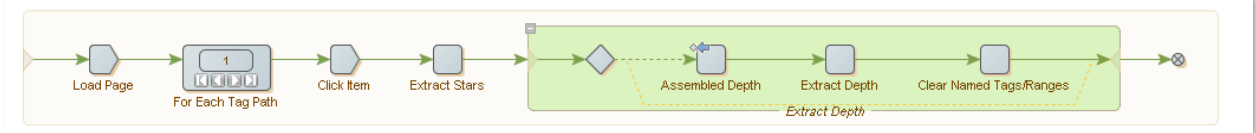

Then, collapse your group.

22. The rest of your extraction has already been created and saved as a Snippet. Select your Robot's end step and add the ExtractData Snippet just before the end step.

23. Selecting the end step, examine the Variables panel. You should successfully extracted data for all variables. And use the "For Each Tag Path" loop to cycle through two or three iterations and make sure extraction is happening for those products as well.

24. Add a "Return Value" Action Step just before the end step. Save your Robot.

25. Go to the Debug tab and run the Robot in debug mode. It should return detail for all products on the first Products listings page of the web site. *But there are many pages, and you want to return results for ALL products. What's next?*

26. *You'll want to set up a Repeat/Next loop to loop through the pages. This one will be a little more challenging than the last one you did. There is no [Next] button at the bottom of the page…just numbers: 1 2 3 4 5 etc. You'll begin the same*: Add a new "Repeat" loop step just before the "For Each Tag Path" step.

27. Then add a branch immediately following the Repeat loop. It should look like this:
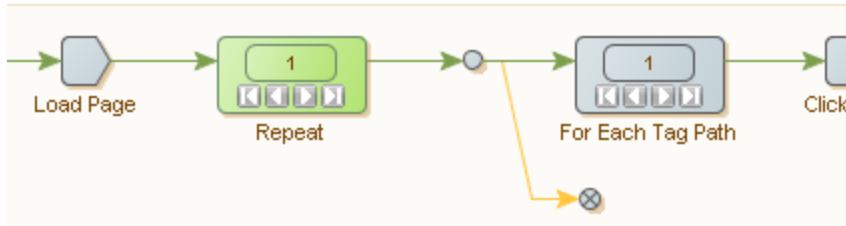
28. *You'll now need to create a method that takes a number and increments it by 1 each time the loop repeats. That will be followed by a "Click" step to click on that number…and that will be followed by a "Next" step to go back to the main branch and perform the extraction.* To begin, add the "Next" step just before the end step on the new branch you just created.

29. You'll need a variable to contain the page number. Go to the Variables panel and add a new simple Variable called "PageNumber." Check the "Global" checkbox. Choose the "Integer" type and "1" as the value.
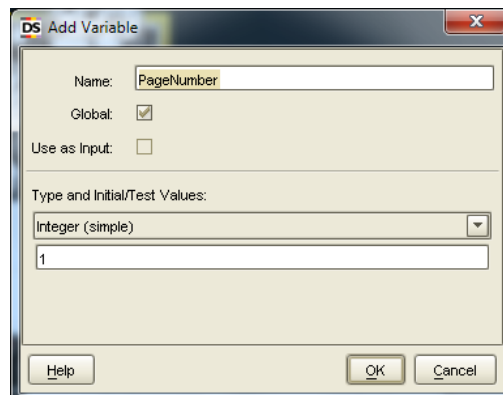
    Move it to the top of the Variables list.

30. Back in the robot view panel add an "Assign Variable" action step BEFORE the "Next" step on the new branch you created. Go to the "Action" tab of this newly created step and select "Expression" from the Value dropdown list. Enter "PageNumber + 1" in the Value Box. Select "PageNumber" from the Variable dropdown list. *This will accomplish increasing the page number by one each time the loop repeats.*

31. Now you need to add a "Click" action step between "Assign Variable" and "Next. Select the "Next" step in the robot view panel. Then in the browser view panel, go to the bottom of the page and click on the number "2." This is the first link available to advance to the next page. Right mouse-click and select "Click" from the context menu. This inserts the click step in the correct place in your Robot.

32. On the "Basic" tab of the new "Click" step, change the name to "Click next page."

33. On the "Finders" tab:

    a. Verify that the Attribute Name is "class."

    b. Verify that the Attribute Value is "Equals Text."

    c. Verify that the Text is "pagenav."

    d. From the Tag Pattern dropdown list, select "Expression." Click on the [Edit] button and enter the following expression: **>>.*?<< +PageNumber + >>.*?<<** This is an example of concatenating everything in the tag before and after the page number (which will be incremented by 1 for each iteration of the Repeat/Next loop).

34. On the "Error Handling" tab, uncheck API Exception and Log as Error and select "Break Loop" from the "Then" dropdown list. This will break the loop if there is no additional page number found.

35. Your completed Robot should look like this:
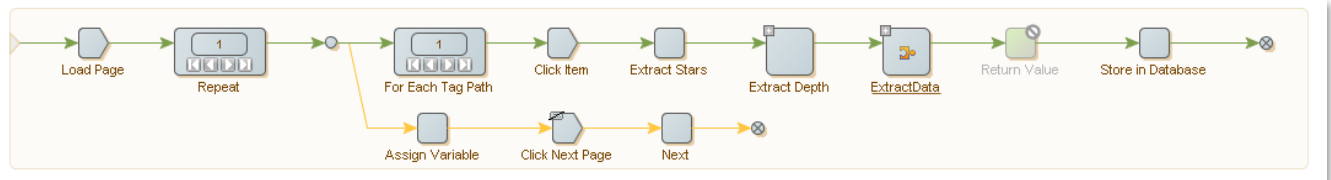
36. Save your Robot.

37. Go to the "Debug" tab and run your Robot in debug mode for 3-5 minutes to ensure that it's returning what you want it to without errors. Once you are satisfied, STOP the Robot. Then use the ⟲ (restart debug) to return your Robot to the beginning.

38. Click on the "Design Mode" tab. Add a "Store in Database" step just after the "Return Value" step. Select the "objectdb" database and the "HardyHardware" variable.

39. Disable the "Return Value" step (you can do this by right mouse-clicking on the step and selecting "Disable" from the context menu).

40. Then, upload your Robot to the Management Console

41. Open the Management Console and based on what you learned in an earlier module, create a weekly schedule to run the robot.

42. Also, try running the Robot manually. Once you're satisfied it's running satisfactorily, stop the robot.