

Create a Catalog Item, Get Data from Catalog Item and insert into Queue then trigger a Robot and Process the Data

Lab 2.3

120 -150 min

Lab Objectives

In this lab you will work in RPA Desktop Design Studio to automate the following tasks:

- Create a Request for badge
- Auto Trigger an Unattended Robot
- Process the request data using an Unattended Robot

 **Lab Dependency:** Requires completion of Lab 2.1 & 2.2

Scenario

When a Catalog Item is ordered on ServiceNow a flow should insert the data in a Queue and then call an Unattended Robot which should pick the work item from the queue and process it. In this case a badge creation request data needs to be put into a thick client badging app.

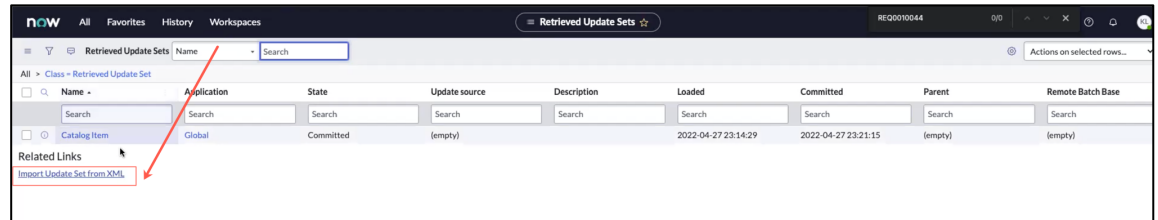
Requirements Summary

User Stories

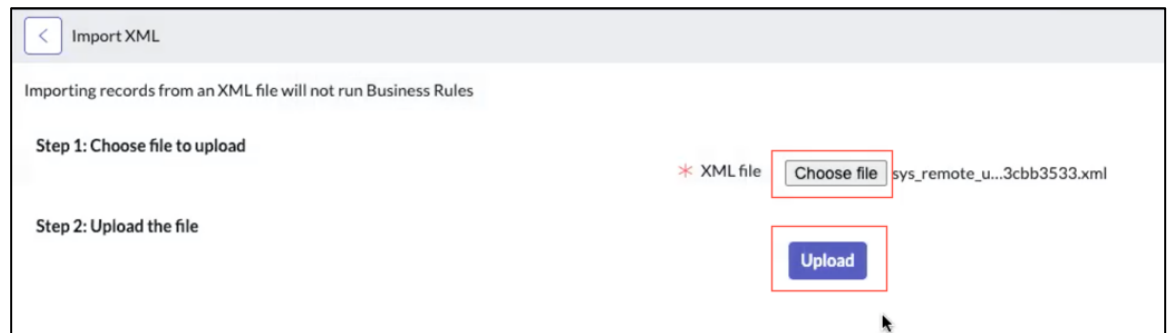
- A. Prepare instance and RPA Hub for new process
- B. Read data from queue, parse data and download attachment
- C. Insert data into Badging App
- D. Activate trigger to start robot automatically with fulfillment

A. Prepare Instance and RPA Hub for new process

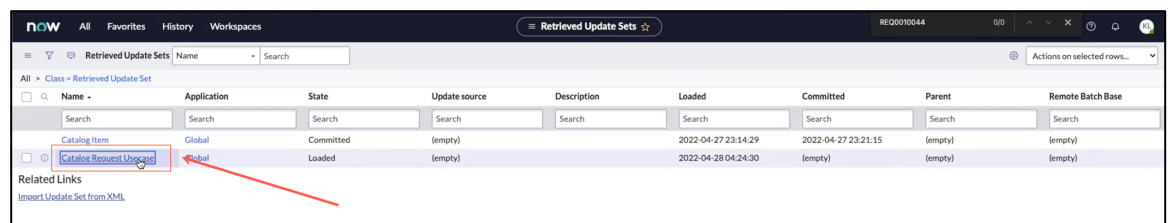
1. Log into your ServiceNow instance.
2. Navigate to **All > System Update Sets > Retrieved Update Sets**.
3. Perform the following steps to import the attached xml in the list view of the table:
 - a. Click **Import Update Set from XML**.



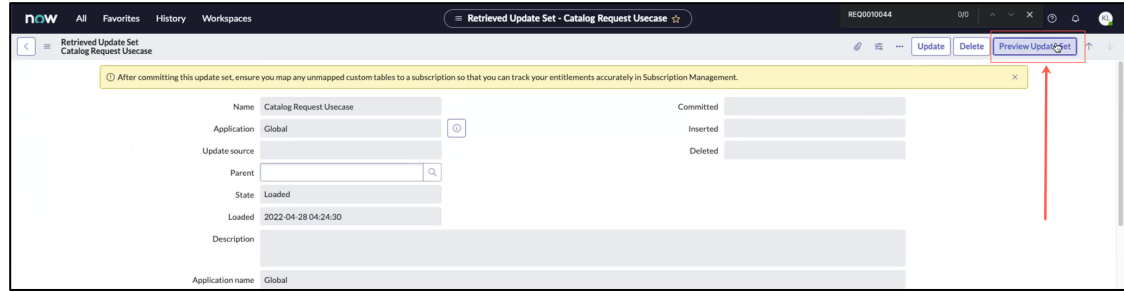
- b. Click **Choose file** to select an xml file to upload.
(sys_remote_update_set_f227570a31006d507f44c82e2168db15)
- c. Click **Open**.
- d. Click **Upload**.



4. Once imported, open the update newly imported update set.



5. Click **Preview Update Set**.



6. If there are no errors, then click **Commit update set**.

7. If you encounter errors, perform the following:

- On the **Update Set Preview Problems** tab, for all the error types, click **Accept remote update** from the **Available Actions** column.



- Click **Commit Update Set**.

8. To create a new package, perform the following steps:



- Navigate to **All > Robotic Process Automation > RPA Hub > Build > Packages**.
- Click **New**.
- Fill out the form

Field	Value
Name	Insert Data from Queue to Badging App
Package	Unattended

- Click **Save**.

9. Create a bot process, perform the following steps in RPA Hub:

- a. Create a bot process by doing one of the following actions.

Option	Action
From the list icon	Click the list icon (). On the Lists tab, under Build , click Bot Process . Click New .
From the plus icon	Click the plus icon (). Select New Process .

- b. Fill out the form

Field	Value
Name	Insert Data from Queue to Badging App
Process Type	Unattended
Trigger Mode	API
Package	Insert Data from Queue to Badging App

- c. Click **Save**.

10. To create a credential set within a bot process, do the following:

- a. Go to the **Credential Sets** tab and click **New**.

- b. Fill out the form

Field	Value
Name	Unique name of the credential set.
Windows Username	Username of the Windows machine to be used by the assigned robot. Enter the username in the DOMAIN \Username format.
Windows Password	Password for the Windows machine to be used by the assigned robot.

- c. Click **Save**.

11. To assign a robot to this bot process, do the following:

A robot can only be assigned to bot processes of the same trigger type. If you completed the previous labs, your robot is currently assigned to a bot process with a schedule-based trigger. You need to unlink this to proceed. Follow these steps:

- a. Go to **Lists > Bot Processes**
- b. Open the previously created Bot Process *Insert Badging data into Queue*
- c. Open related list **Assigned Robots**
- d. Select (checkbox) your robot and click **Remove**

Now continue, with the new bot process displayed assign the robot.

- a. Go to the **Assigned Robots** tab and click **Add**.
- b. On the **Available Robots** tab, select the required robot.
- c. Click **Add**.
- d. Click **Save**.

12. To map process robot credentials, do the following:

- a. Go to the **Process Robot Credentials** tab and click **New**.
- b. On the **Process Robot Credential** form, fill in the fields.

Field	Description
Credential Set	Name of the credential set to be associated with the robot.
Robot	Name of the assigned robot.

Note: Notice there is only one credential set to be selected even though multiple credential sets are now available in the platform. Credential sets are linked to a bot process and are only valid within that context.

- c. Click **Save**.

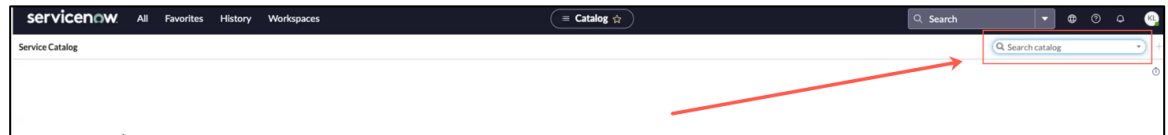
13. Perform the following steps to associate a queue to this bot process.

- a. Navigate to **Lists > Build > Queues**.

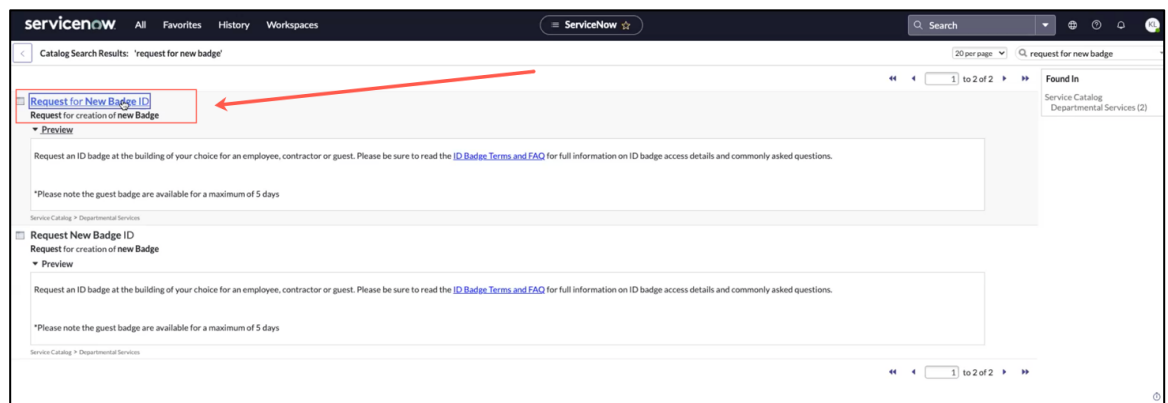
- b. Open the *Badging data* queue.
 - c. On the **Bot Process** tab, click **Add**.
 - d. On the **Available Bot Processes** tab, select *Insert Data from Queue to Badging App*.
 - e. Click **Add**.
 - f. Click **Save**.
14. The Update Set includes a Service Catalog entry, we need to modify the fulfillment flow to match your RPA Hub setup. For this navigate to **All > Process Automation > Flow Designer**.
15. From the **Flows** tab, open **Move data from Catalog Item to RPA Hub Queue**
16. Expand the 2nd step **Add WorkItem to Queue**
17. Notice the second parameter *Queue Name* is empty, select the **Badging data** Queue
- Note:** *Inspect the Content parameter, our robot automation will need to work with the data and structure defined here.*
18. Click **Done**
19. Click **Save**
20. Click **Activate**

Lab Verification

1. Navigate to **All > Self-Service > Service Catalog**.
2. In the **Search catalog** field on the right-hand side of the page, enter "Request for new badge ID".



3. Select **Request for new badge ID**.



4. Enter details as follows:

Request for creation of new Badge

Request an ID badge at the building of your choice for an employee, contractor or guest. Please be sure to read the [ID Badge Terms and FAQ](#) for full information on ID badge access details and commonly asked questions.

*Please note the guest badge are available for a maximum of 5 days

Access Level
Employee

Name
John Doe

Employee ID
A74533

Title
Mr.

Location
New York

5. Click the attachments icon.



6. In the Attachments dialog box, click **Choose file**, select a file, and click **Open**. Ensure the file type is jpeg or png.
7. Close the Attachments dialog box.

8. Click **Order Now**.
9. Navigate to **All > Robotic Process Automation > RPA Hub > Build > Queues**.
10. Open **Badging Data**
11. Open the Work Queue Items tab, refresh if needed and open the latest record. Its name is the number of the requested item just ordered.

Badging data

Purge After (days) 30Purge Status Success, Failure

Save

...

DetailsWork Queue Items (6)Bot Process (2)

Work Queue Items 6

Last refreshed 1m ago.

Mark as Complete

Reassign

<input type="checkbox"/>	Name	Status	Stage	Created ^	Started On	Completed On	Robot	Priority
	Emp3	Success	Stage_0	2022-12-28 09:13:17	2022-12-28 11:21:31	2022-12-28 11:21:35	My virtual PC	High
	Emp1	Success	Stage_0	2022-12-28 09:13:17	2022-12-28 11:21:35	2022-12-28 11:21:38	My virtual PC	High
	Emp4	Success	Stage_0	2022-12-28 09:13:17	2022-12-28 11:21:38	2022-12-28 11:21:41	My virtual PC	High
	Emp5	Success	Stage_0	2022-12-28 09:13:17	2022-12-28 11:21:41	2022-12-28 11:21:43	My virtual PC	High
	Emp2	Success	Stage_0	2022-12-28 09:13:17	2022-12-28 11:21:43	2022-12-28 11:21:46	My virtual PC	High
	RITM0010002	Pending		2022-12-30 07:56:21			(empty)	Low

12. Scroll down to the Content area, verify the content in the **Request Content** field.

Details

Type

Last Started Time

YYYY-MM-DD HH:mm:ss

Stage

Deferred Till

YYYY-MM-DD HH:mm:ss

SLA

YYYY-MM-DD HH:mm:ss

Completed By

Robot

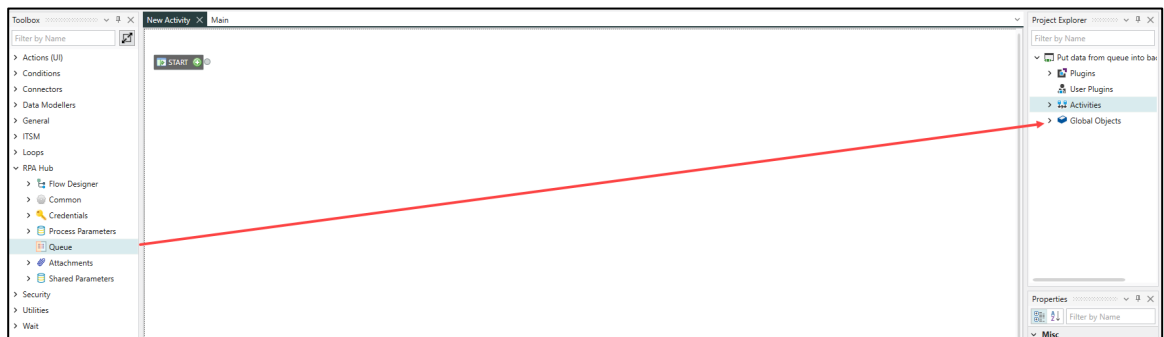
Content

Request Content

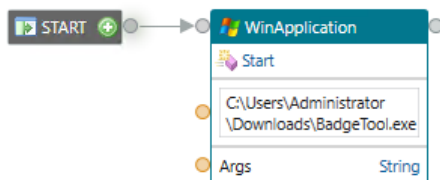
Employee|
John Doe|
A74533|
Mr.|
New York|
2b1a078e318c2d507f44c82e2168dbb1

B. Read data from queue, parse data and download attachment

1. Open **RPA Desktop Design Studio**.
2. From the Home ribbon, choose to create a new **Unattended Automation** project.
3. Enter the name of the project as **Insert data from queue to badging app** and save it to the default location.
4. Click **OK**.
5. Validate RPA Desktop Design Studio is still connected to your ServiceNow instance. If not use the Connection Manager and restore the connection
6. From the **Toolbox** pane, drag the **RPA Hub > Queue** connector to Global Objects.



7. Set the Queue Name in the properties pane to **Badging Data**.
8. From the Toolbox, drag **Windows Connector** component to Global Objects
9. Double-click the **WinApplication** in Project Explorer, from Object Explorer drag **Start** to the design surface, connected to **START**
10. Double-click the Start component and enter the exe file path of the badging app.



11. To create a local variable, do one of the following:

Option	Action
--------	--------

From Toolbox	Navigate to General > Variable Drag the Variable from Toolbox pane and drop it to the Project Explorer > Activities > Main .
From Project Explorer	Navigate to Activities > Main Right click on Main -> Create a Variable

12. Double-click the Variable, in the **Properties**, enter **itemFound** in the **Name** field.

13. Drag the variable **itemFound** to the design surface.

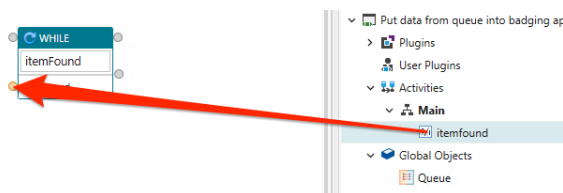
14. Double-click the variable and enter "**True**".

15. In the Toolbox pane, navigate to **Loops**.

16. Drag the **While** component to the design surface.

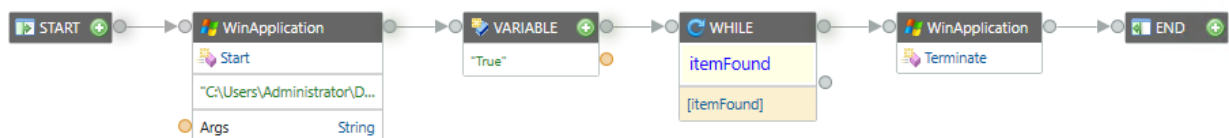
17. Double-click the **While** component and enter **itemFound** on the input port of while component

18. Drag the variable on to the data input port of the *While* component.

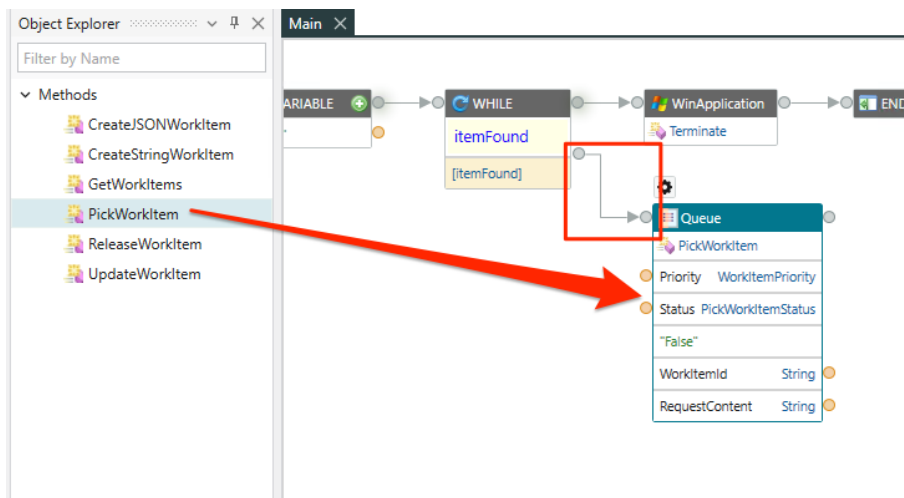


19. Double-click the **WinApplication** in the Project Explorer, drag **Terminate** from the Object Explorer and connect it to the Control out port of the *While* component

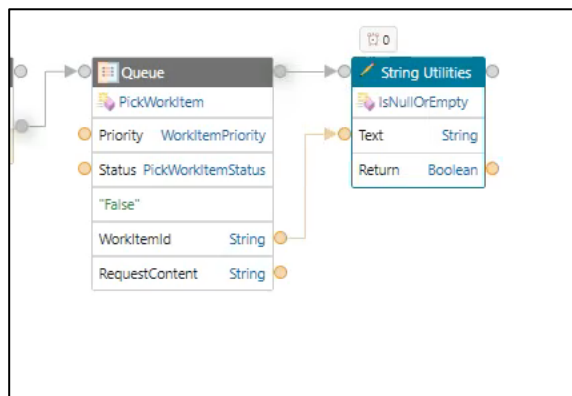
20. Connect the **Terminate** component to the **END** component.



21. Double-click the **Queue** connector under the *Global Objects*, from the **Object Explorer** pane drag **PickWorkItem** method to the design surface and connect to the **While** component.

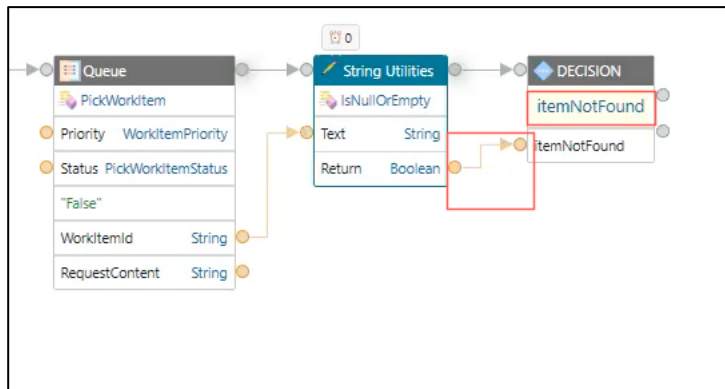


22. Drag the **IsNullOrEmpty** component from Toolbox **Utilities > String Utilities** to the design surface.
23. Connect the **PickWorkItem** method to the **IsNullOrEmpty** component as shown in the below image.



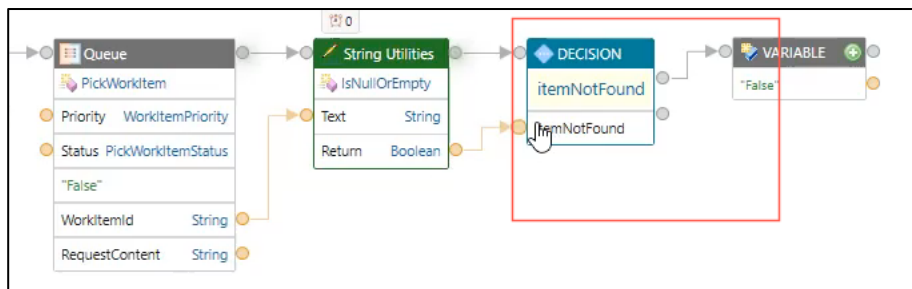
24. From the **Toolbox**, drag the **Decision** component on to the design surface.
25. Double-click the **Decision** component and enter **itemNotFound**.

27. Connect the Decision component to the **IsNullOrEmpty** component as shown in the below image.



28. Drag the **itemFound** variable from the **Project Explorer** to the design surface.

29. Connect the **Decision** component to the **itemFound** variable as shown in the below image.



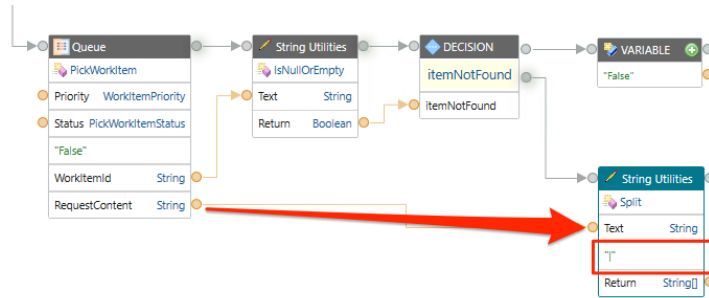
30. Double-click the variable and enter "**False**".

This will be our exit condition for the while loop. If no item is in the Queue, the robot will exit.

31. Now let's build the processing of the catalog request. The information in the queue will have a string with all values required concatenated by a pipe (|) character. We need to split them into separate attributes.

From the **Toolbox**, drag the **Split** component on to the design surface.

- Connect the Request Content Data out port to the Data in port of the Split component.
- Double click on the separator of the Split component and enter |.

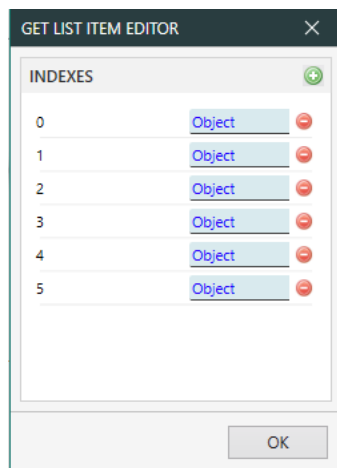


32. From the **Toolbox**, drag the **GetListItem (Utilities >Reflection)** component on to the design surface, connect to the Split component.

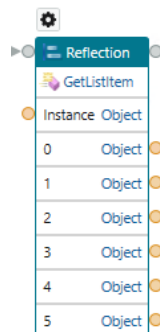
33. On the *GetListItem* component bar, do the following:

- Click the component settings icon (⚙️).
- Click the add icon 6 times to add 6 Indexes.

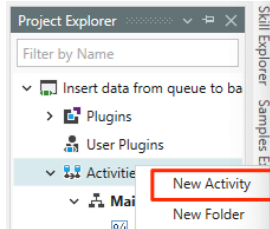
Note: The count will be from 0 to 5



Click **OK**. Output should look like the example shown below.



34. In Project Explorer, do a right-click on **Activities** and select **New Activity**



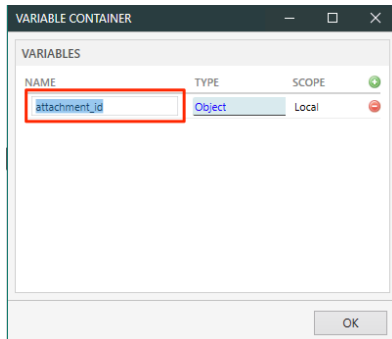
35. Double-click the new activity, a new design surface tab will open

36. In the *Properties* pane rename the activity to **DownloadAttachment**

The next steps focus on downloading this attachment for the Badge application.

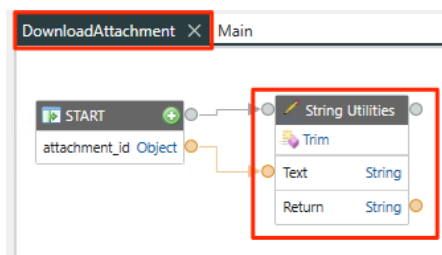
37. Open the settings dialog on the *START* component

a. Add a new input variable and name it '**attachment_id**'



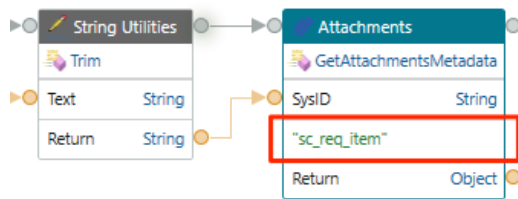
b. Click **OK**

38. From the Toolbox, drag the **Trim (Utilities > String Utilities)** component on to the design surface and connect the *START* component. Also connect the *attachment_id* data out port to the *Text* data in port.



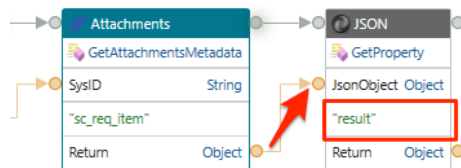
39. From the Toolbox, drag the **GetAttachmentsMetadata (RPA Hub > Attachments)** component on to the design surface and connect to the *Trim* component.

a. On the *GetAttachmentsMetadata* component double-click the table name property and enter "**sc_req_item**".



40. From the Toolbox, drag the **GetProperty (JSON)** component on to the design surface and connect to the *GetAttachmentsMetadata*.

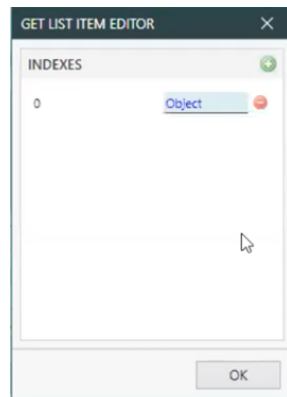
- On the *GetProperty* component, double-click on the property name field and enter "result".
- Connect the *GetAttachmentsMetadata* return data port to the *JsonObject* data in port of *GetProperty*



41. From the Toolbox, drag the **GetListItem (Reflection)** component on to the design surface and connect to the **GetProperty (JSON)**.

On the *GetListItem* component bar, do the following:

- Click the component settings icon (⚙️).
- Click the add button to add 1 Index.
- Add entry as shown in the below image.



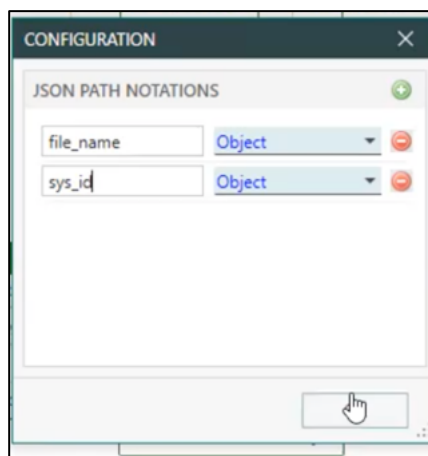
- Click **OK**.

- e. Connect the Return data out port of *GetProperty* to Instance data in port of *GetListItem*

42. From the Toolbox, drag the **GetProperties (JSON)** component on to the design surface and connect to the *GetListItem*.

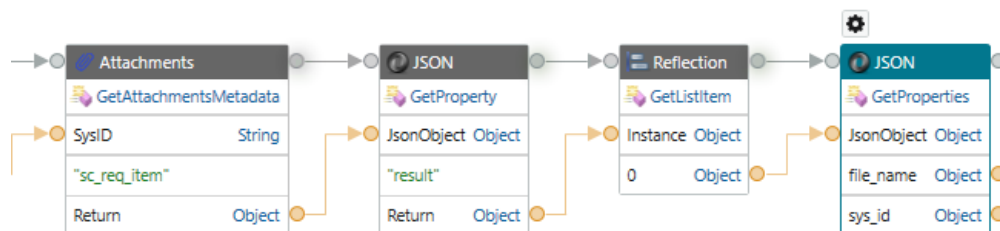
On the *GetProperties* component bar, do the following:

- a. Click the component settings icon (⚙️).
- b. Click the add button to add 2 Indexes.
- c. Add entry as shown in the below image.

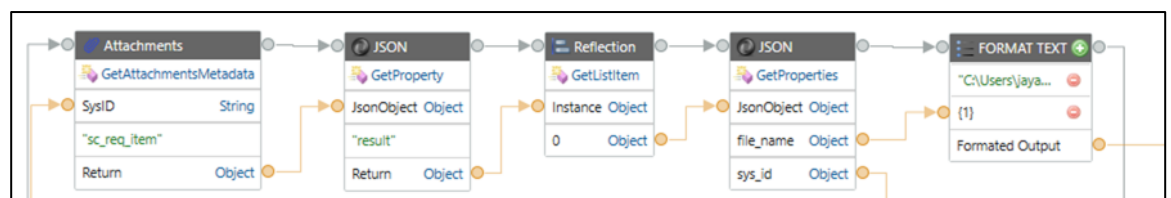


- d. Click **OK**.

The sequence will look like this:

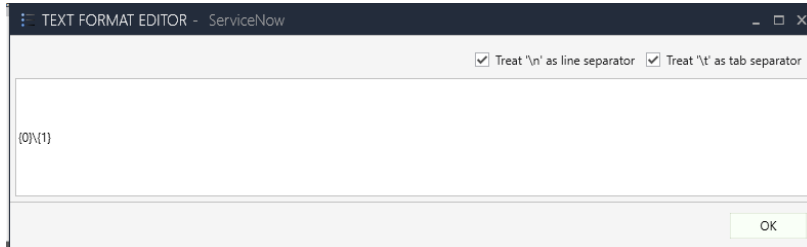


43. From the *Toolbox*, drag the **Format Text** component on to the design surface and connect to the *GetProperties* component.



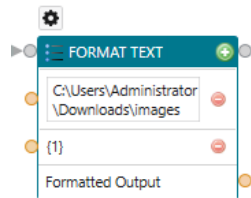
On the *Format Text* component bar, do the following:

- Click the add icon twice, this will add {0} and {1} as input data ports
- Click the component settings icon (⚙️).
- Enter '{0}\{1}' as follows:

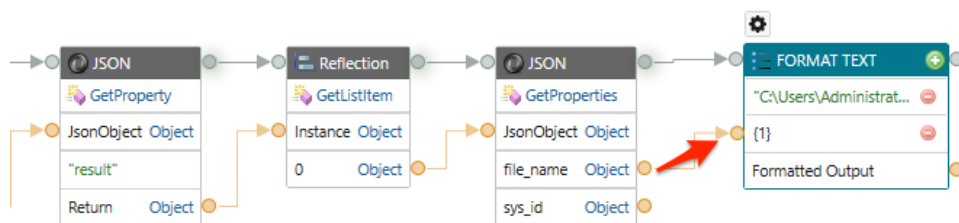


This defines how the input values will be formatted into an output string.

- Click **OK**.
- Enter a path which can be used to download the image {0}. We used a newly created folder 'images' within the users download directory



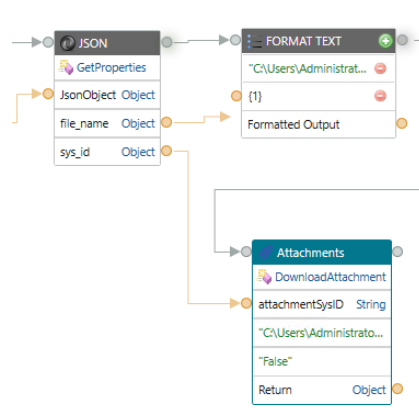
- Connect the *file_name* Data out port of *GetProperties* component to the {1} of Format Text.



The purpose of these sequence is to know the filename of the attachment which will be used later to upload to the Badge Tool application.

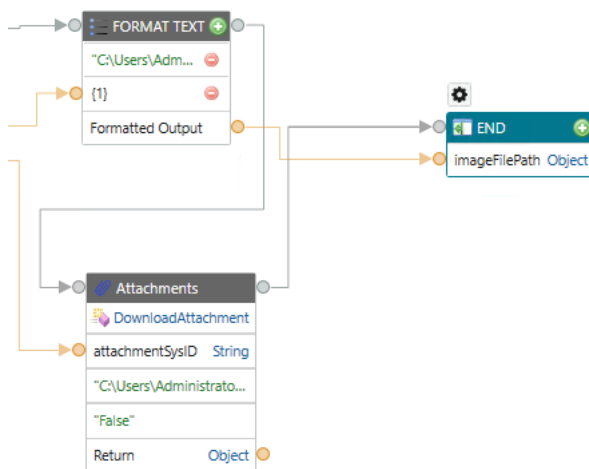
- From the **Toolbox**, drag the **DownloadAttachment (RPA Hub > Attachments)** component on to the design surface and connect to the **Format Text** component.
 - Double-click on the *folderPath* and enter the same image path as used on the FormatText component.

- b. Connect `sys_id` data out port of `GetProperties` to `attachmentSysID` input port of `DownloadAttachment` component

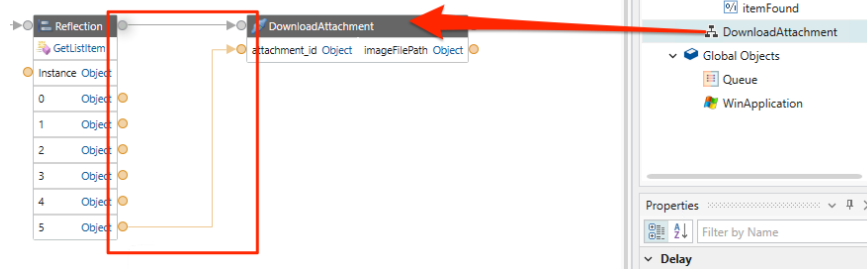


Challenge: Use a variable to store the image path and reuse it for both components.

45. Open the settings dialog on the `END` component
 - a. Add a new output variable and name it '**imageFilePath**'
46. Connect the `END` as shown to `DownloadAttachment` and `Format Text`



47. Back in the Main Activity, drag the **DownloadAttachment** activity to the design surface, connect it to `GetListItem`.
 - a. Connect the **5th data out port** from `GetListItem` to `attachment_id` on `DownloadAttachment`



C.Insert data into Badging App

1. To configure the Windows application, do the following:

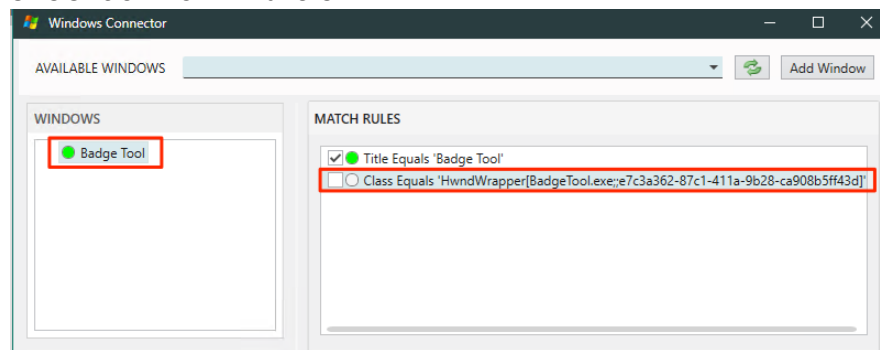
a. Right-click the Windows application and select **Configure**.

Note: Note: Ensure the application is running or manually launch the app.

b. In the Available Windows select the BadgeTool application and click **Add Window**

c. In the *Windows pane* select the added **Badge Tool** entry, inspect the shown *Match Rules*

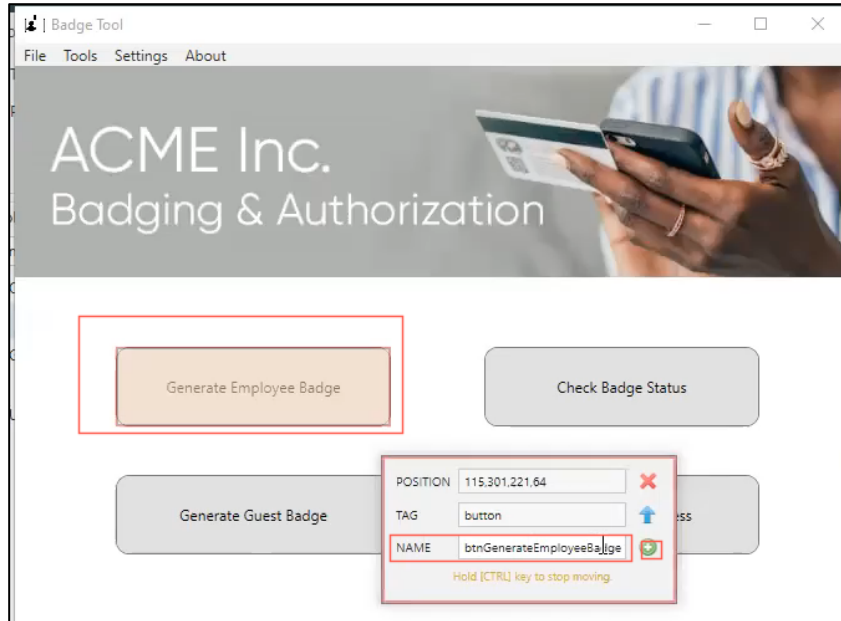
d. As the **class** might change with every launch of BadgeTool, remove the checkbox from this rule



2. In the Windows pane, right-click **Badge Tool** and select **Add element** and do the following:

a. Hover on Generate Employee Badge and enter **btnGenerateEmployeeBadge** in the "Name" field of the popover box.

b. Select the plus icon (⊕), the name field will be emptied as confirmation



3. Click the X icon to close the pop up.
4. Click the Generate Employee Badge button manually on the Badge Tool app. Then - in the Windows Connector dialog – repeat the Add Element function to capture more elements as below


Note: For all data entry fields ensure the Tag is edit in the pop-up window. If you see the tag as Pane, click the up arrow (↑) till you see edit

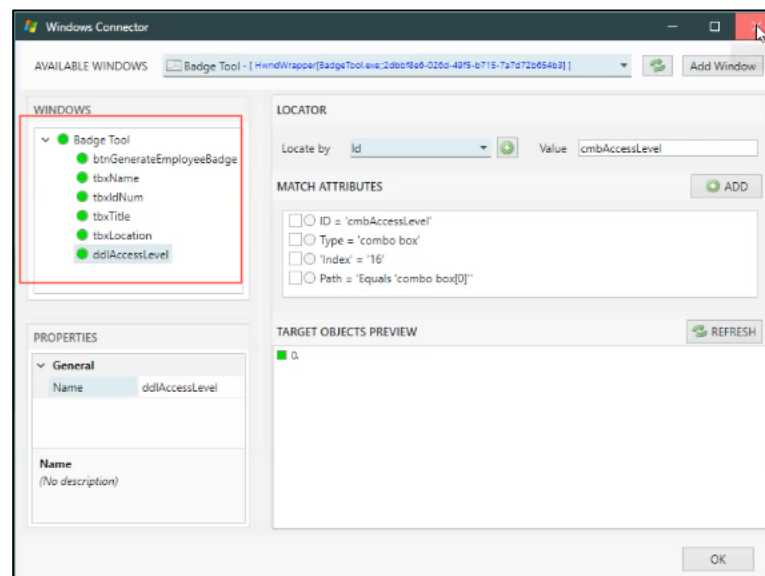


- a. Hover on **Name** text entry box and enter **tbxName** in the Name field, click the plus icon.
- b. Repeat for these fields:

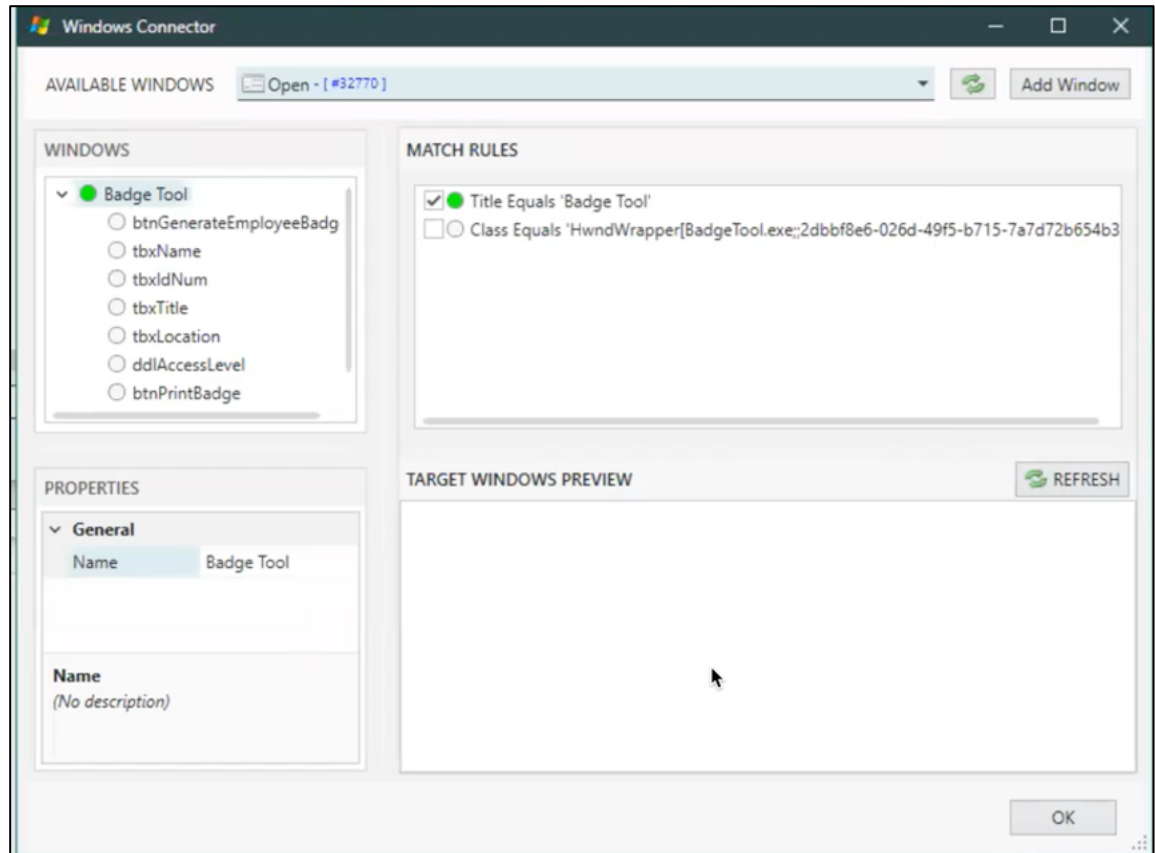
Text Field	Name
ID Number	tbxIdNum
Title	tbxTitle
Location	tbxLocation

- c. Hover on **Access Level** field and enter **ddlAccessLevel** in the Name field. Ensure the **Tag** is **combo box** in the pop-up window for this field, click the plus icon.

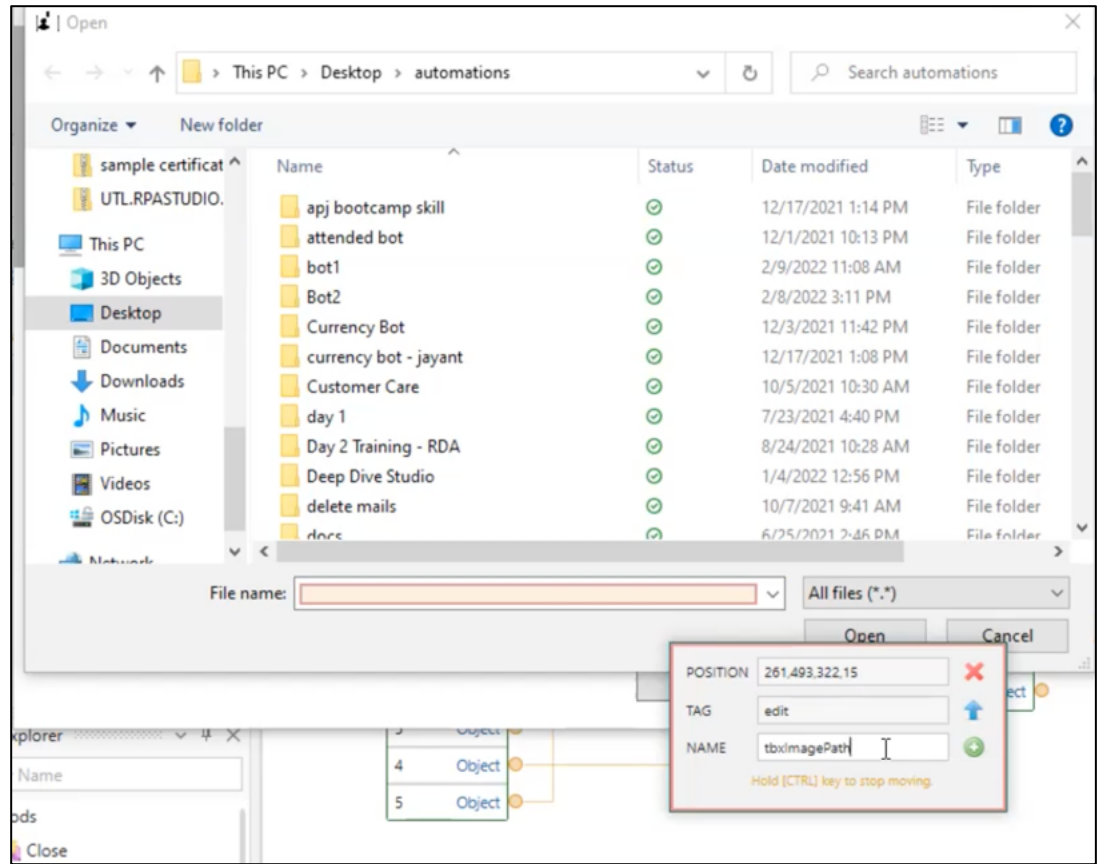
- d. Hover on **Print Badge** button and enter **btnPrintBadge** in the Name field.
Ensure the **Tag** is **button** in the pop-up window for this field, click the plus icon.
 - e. Hover on **Upload** field and enter **btnUpload** in the Name field.
Ensure the **Tag** is **button** in the pop-up window for this field.
 - f. Click the plus icon ().
 - g. Click the X icon to close the pop up.
5. In the Windows tab, right-click the elements under **Badge Tool** and select **refresh**.
Click ">" to see the list of added fields. Go to individual fields, right-click and select refresh. Ensure that all the field indicators turn green



6. Click Upload button in the badging app to open the file selection window.
7. In the Available Windows select the "Open" application if you do not see Open click refresh.

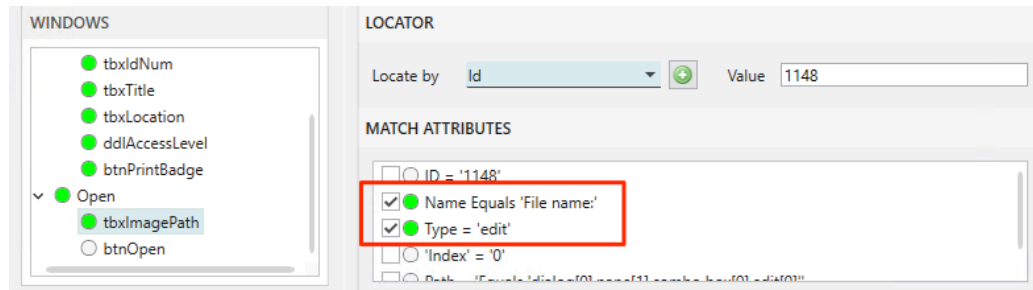


- a. Click **Add Window**
 - b. As the **class** might change with every time the Open dialog launches, remove the checkbox from this rule
8. In the Windows tab, right-click **Open** and select **Refresh**.
 9. In the Windows tab, right-click **Open** and select **Add element** and do the following:
 - a. Hover on the space in the File name field and enter **tbximagePath** in the Name field.



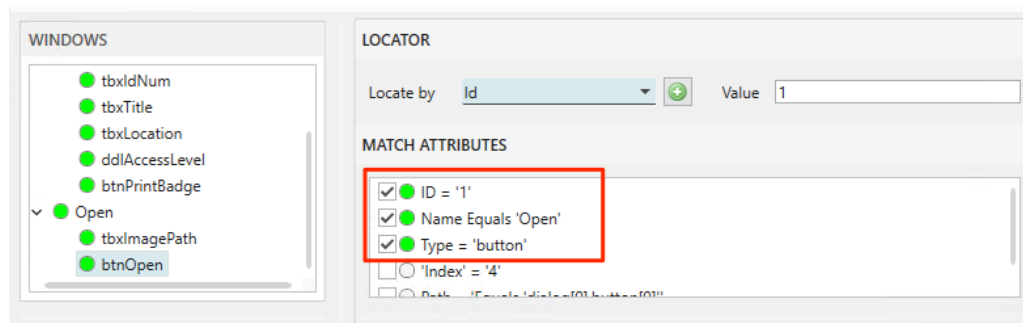
- Ensure the **Tag** is **edit** in the pop-up window for this field.
- Click the plus icon (⊕).
- Hover on **Open** field and enter **btnOpen** in the Name field.
Ensure the **Tag** is **button** in the pop-up window for this field.
- Click the plus icon (⊕).
- Click the X icon to close the pop up.

10. Right-click **tbxImagePath**, select **Refresh** and ensure the following attributes are checked as per the below image:

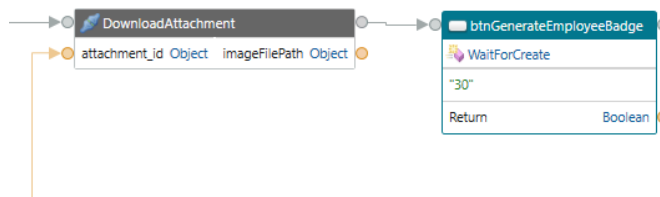


Note: With these two attributes selected, there is only one text box in the dialog. Without these you will see a red '[2]' on refreshing the element noting that the locator is not unique.

11. Right-click **btnOpen**, select **Refresh** and ensure the following attributes are checked as per the below image are done:

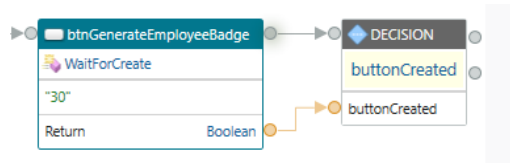


12. Click OK to close the Window Connector.
13. In the Project Explorer, expand the **WinApplication > Badge Tool** and double-click **btnGenerateEmployeeBadge**.
14. From the Object Explorer, drag **WaitForCreate** method on the design surface, connected to the **DownloadAttachment** component.

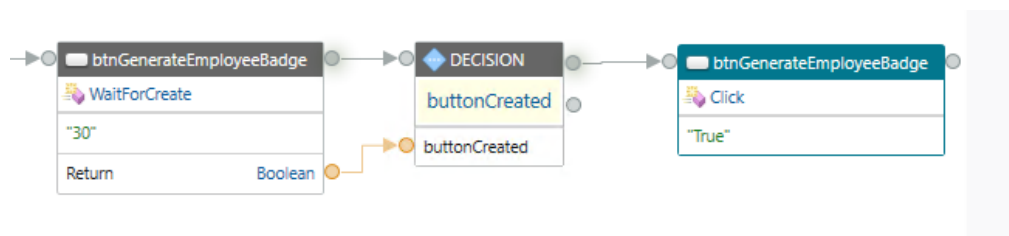


15. From the Toolbox pane, drag **Decision** component on the design surface, connected to the **WaitForCreate** method component.

16. In the **Decision** component, enter **buttonCreated** and connect the ports accordingly.



17. In the Project Explorer, expand the **WinApplication > Badge Tool** and double-click **btnGenerateEmployeeBadge**
18. From the Object Explorer, drag **Click** method on the design surface, connected to the **Decision** component.



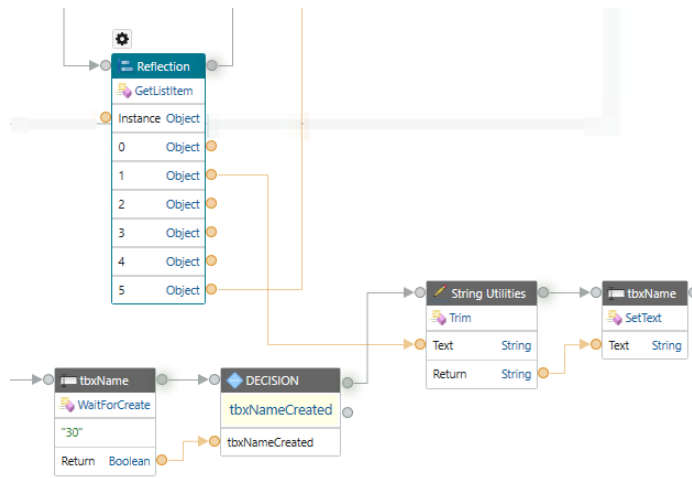
Note: We do a **WaitForCreate** to ensure the button is displayed on screen and use the decision to react if the button didn't show up after the timeout specified in the **WaitForCreate**. You can reuse this pattern to almost any screen interaction.

19. In the Project Explorer, expand the **WinApplication > Badge Tool** and double-click **tbxName**.
20. From the Object Explorer, drag **WaitForCreate** method on the design surface, connected to the **Click** component.
21. From the Toolbox pane, drag **Decision** component on the design surface, connected to the **WaitForCreate** method component.
22. In the **Decision** component, enter **tbxNameCreated** and connect the ports accordingly.



23. From the Toolbox pane, drag **Trim** component on the design surface, connected to the **Decision** component.

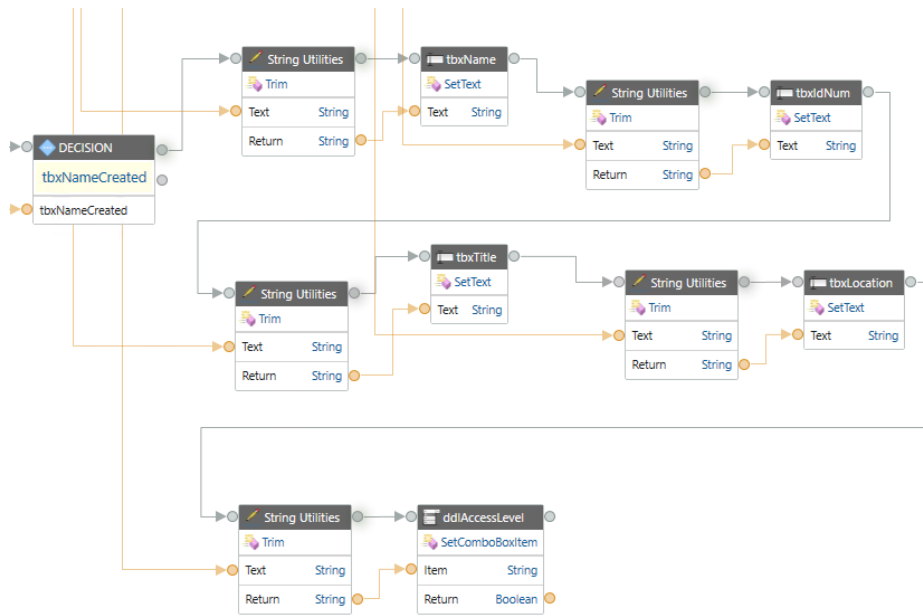
24. Connect the **1 Object Data out port** of **GetListItem** component to the Data in port of this **Trim** component.
25. In the Project Explorer, expand the **WinApplication > Badge Tool** and double-click **tbxName**.
26. From the Object Explorer, drag **SetText** method on the design surface, connected to the **Trim** component.



27. Repeat adding Trim's and SetText / SetComboBoxItem components for these attributes

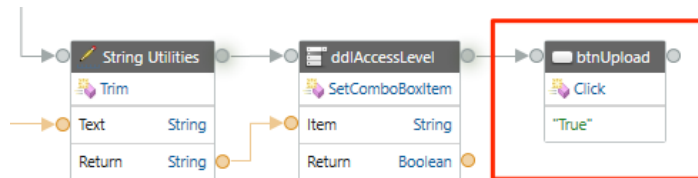
Windows Connector Element	Data output port of GetListItem	Component used to set value
tbxIdNum	2	SetText
tbxTitle	3	SetText
tbxLocation	4	SetText
ddlAccessLevel	0	SetComboBoxItem

The automation looks similar like this:



28. In the Project Explorer, expand the **WinApplication > Badge Tool** and double-click **btnUpload**.

29. From the Object Explorer, drag Click method on the design surface, connected to the **ddlAccessLevel** component.



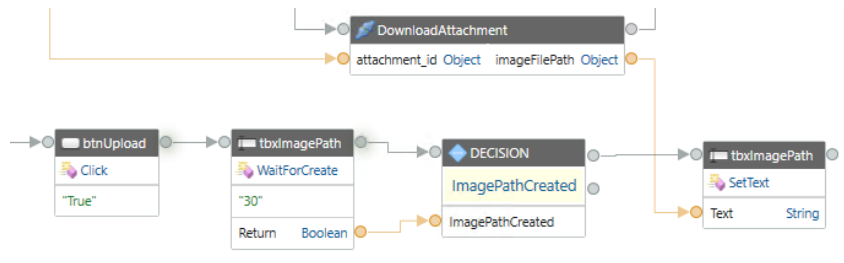
30. In the Project Explorer, expand the **WinApplication > Open** and double-click **tbxImagePath**.

31. From the Object Explorer, drag WaitForCreate method and SetText on the design surface.

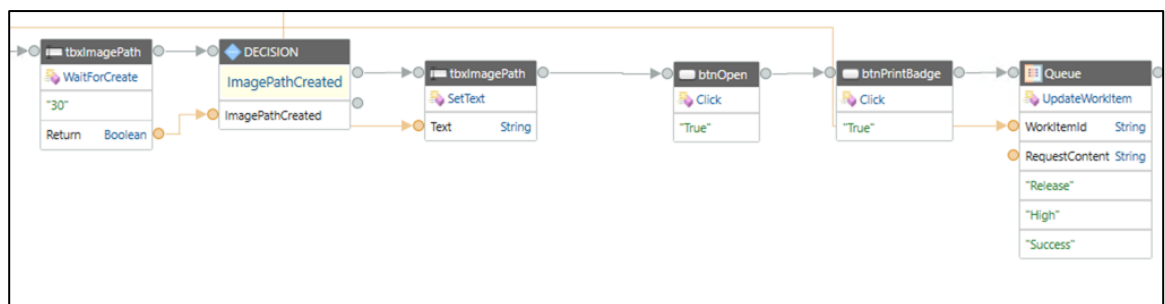
32. Connect the *imageFilePath* Data out port of *DownloadAttachment* to the Data In port of **tbxImagePath-SetText**.

33. From the toolbox, drag Decision component to the design surface. Double-click the component and enter "ImagePathCreated".

34. Connect the components as follows:



35. In the Project Explorer, expand the **WinApplication > Open** and double-click **btnOpen**.
36. From the Object Explorer, drag **Click** method on the design surface and connect to the SetText component
37. In the Project Explorer, expand the **WinApplication > Badge Tool** and double-click **btnPrintBadge**.
38. From the Object Explorer, drag **Click** method on the design surface, connected to the **btnOpen** component.
39. In the **Project Explorer**, double-click the **Queue** connector under the **Global Objects**.
40. From the Object Explorer, drag **UpdateWorkItem** method on the design surface, connected to the **btnPrintBadge** component. Connect the data ports of the **PickWorkItem-Queue** connector (WorkItemId).



41. Double-click the In-progress value on the **UpdateWorkItem Queue** connector, in the **Port Properties**, update the **Static Value** field to **Success** and click **OK**.
42. **Optional:** Test your automation. To do so follow these steps:
 - a. **Assign bot process** to allow studio to access the queue
 - b. Click **Run** to test your automation.

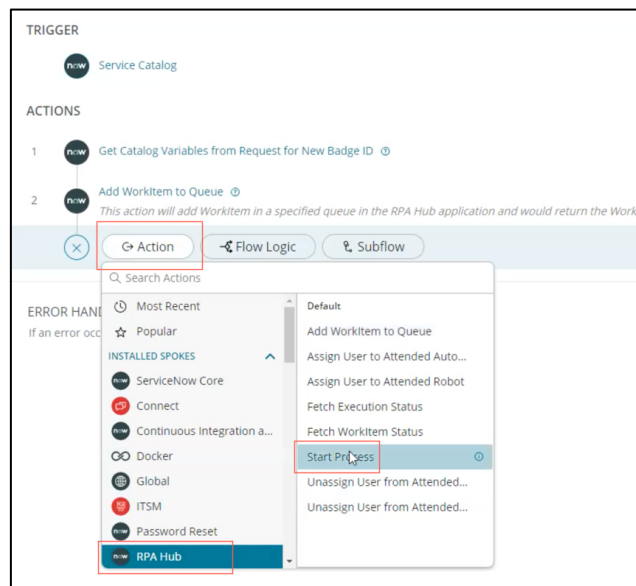
Note: If you created a queue record in the previous part of this lab you can now test you automation. If not, you will need to go back first and request a new badge from the Service Catalog. If your automation has errors debug and retest. You can use the **Reassign** UI action on the work queue item to process it again in the robot.

43. After the automation is completed, click **Publish**.
44. Select **New Version (to existing package)** and select Package as "Insert Data from Queue to Badging App", provide a comment and click **Publish**.

Note: *In the RPA Desktop Design Studio, on the design surface use Shift+Arrows keys to scroll through the design surface.*


D. Activate trigger to start robot automatically with fulfillment

1. Navigate to in RPA Hub to **Build > Bot Process**.
2. Select the bot process – **Insert Data from Queue to Badging App**
3. Select the *package version* that you published.
4. Click **Save**.
5. Click **Publish**.
6. Navigate to **All > Process Automation > Flow Designer**.
7. From the **Flows** tab, open **Move data from Catalog Item to RPA Hub Queue**
4. Click **Add an Action, Flow Logic, or Subflow**.
5. Click **Action > RPA Hub > Start Process**.






6. Select the **Process Name** field as **Insert Data from Queue to Badging App**.



TRIGGER




 Service Catalog




ACTIONS

- 1  Get Catalog Variables from Request for New Badge ID ⓘ
- 2  Add WorkItem to Queue ⓘ
This action will add WorkItem in a specified queue in the RPA Hub application and would return the WorkItem ID and Unique Name information.
- 3  Start Process ⓘ
This action will trigger a specific Bot process (API based trigger) in RPA Hub and return status.

Action:

* Process Name:  

▼ Robots   

Robot:   

Note: If you do not see this bot process, verify that the bot process Trigger Mode is API, and the Bot Process is Published.

7. Click **Done**.
8. Click **Save**.
9. Click **Activate**.

Lab Verification

Request a new badge from the Service Catalog and watch your virtual machine. If your unattended robot is connected it should wake up and enter the data into the Badge Tool. If it does not work, remember this is the sequence how the automation works:

- A flow is triggered, the flow creates an entry in Work Queue and starts the robot.
- Unattended robot starts, reads information from Work Queue Item.
- This robot opens the Badging App and inserts data into badging app.

Congratulations on completing the lab!