

Read data from a queue and insert into the Badging App

Lab 2.2

120-150 min

Lab Objectives

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

- Read data from a queue.
- Pre-requisite: Download Badge Tool application on your system
- Insert data into the Badging App.

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

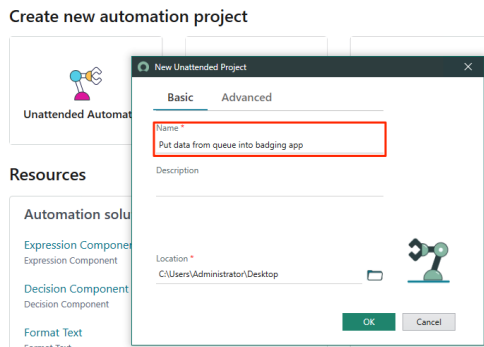
Requirements Summary

User Stories

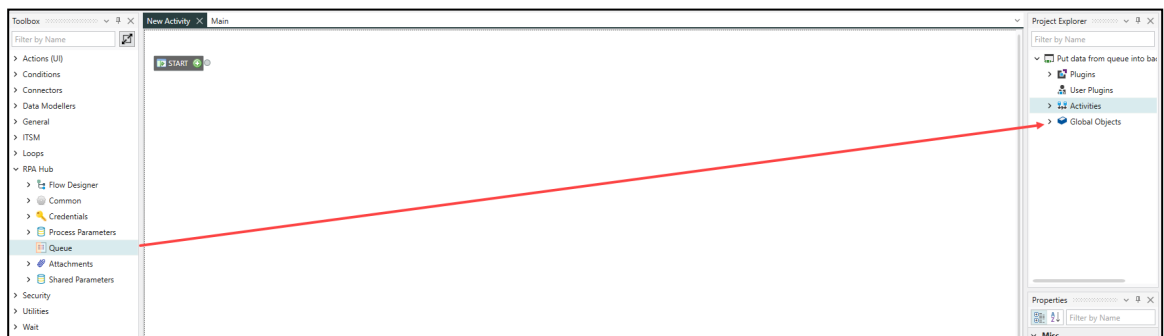
- A. Read data from a queue.
- B. Insert data into the Badging App.

A. Read data from the queue

1. Open RPA Desktop Design Studio.
2. From the Home tab, choose **Create** a new **Unattended Automation** project.
3. Enter the name of the project as **Put data from queue into badging app** and save it to the default location.

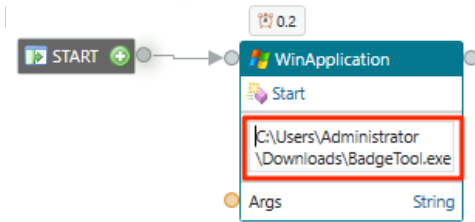


4. Click **OK**.
5. Verify RPA Desktop Design Studio is connected to your ServiceNow instance, if not, use the **Connection to instance** function in the Design ribbon.
6. In the **Toolbox** pane, navigate to **RPA Hub > Queue**.
7. Drag the Queue connector and drop it under the Global Objects.



8. Double click on the added Queue entry, in the Properties pane, enter **Badging Data** in the **QueueName** field.
9. From the **Toolbox**, drag the **Windows Connector** on to the **Global Objects** of the Project Explorer.
10. In the Project Explorer, double click on WinApplication, from the object explorer drag **Start** method on the design surface, connect it to **START** component

11. Double-click the File Path property of Start component and enter the exe file



12. Right click on the Start App component and select **Run From Here**.

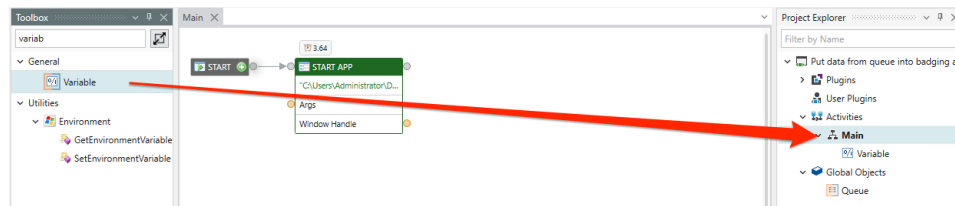
The BadgeTool application will launch, keep it open for later.

Note: If your component fails, make sure the BadgeTool.exe file is allowed to run. Microsoft Defender might block a downloaded executable. If this happens check the properties of the BadgeTool.exe and check **Unblock**.

13. To create a local variable with the name itemFound, do the following:

a. In the Toolbox pane, navigate to **General > Variable**.

b. Drag the **Variable** from **Toolbox** pane and drop it to the **Project Explorer > Activities > Main**. Or right click on Main under Project Explorer and select **Create a Variable**



c. Double-click the Variable and in the **Properties**, enter **itemfound** in the **Name** field.

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

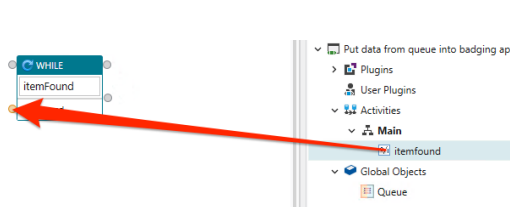
15. Double-click on the variable input field and enter **"True"**.

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

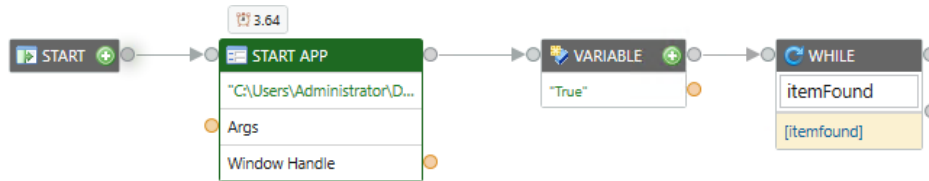
17. Drag the **While** component to the design surface.

18. Double-click the yellow condition property of the **While** component and enter **itemFound**.

19. Drag the variable on to the input port of the While component. This variable will be in the square brackets [itemFound]. We call this a data teleportation as you will see no yellow connection line.



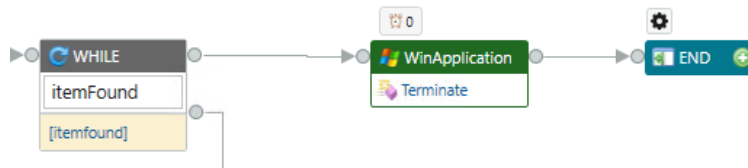
20. Connect your automation as shown below



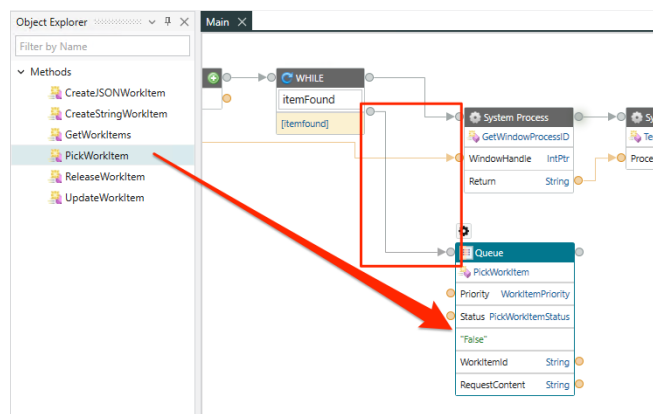
21. In the Project Explorer, double click on WinApplication, from the object explorer drag **Terminate** method on the design surface, connect it to **control** port of the **While** component

Note: Be careful not to use the **General > Terminate** component, this is used to terminate an automation whereas here we want to terminate the BadgeTool app.

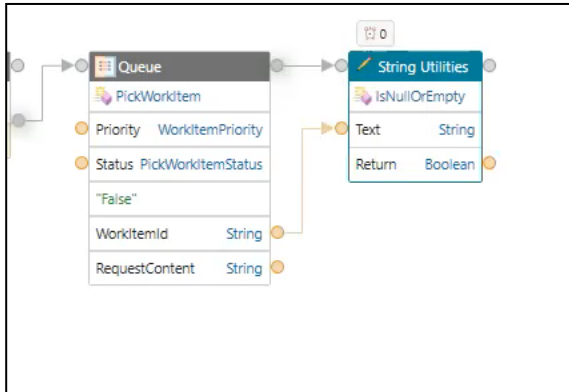
22. Connect the **Terminate** component to the **END** component.



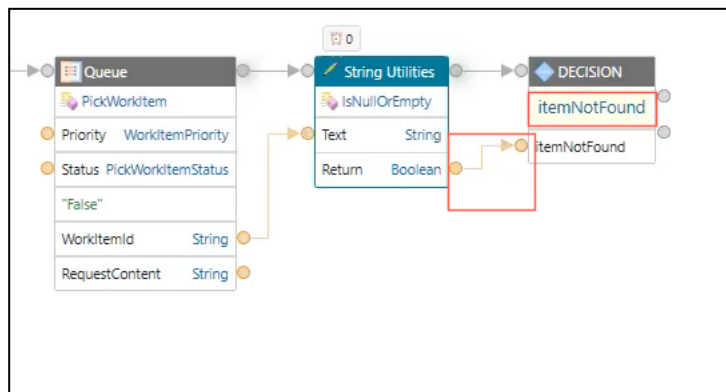
23. Double-click the **Queue** connector under the **Global Objects**, from Object Explorer drag the **PickWorkItem** method to the design surface and connect to the **While** components loop port.



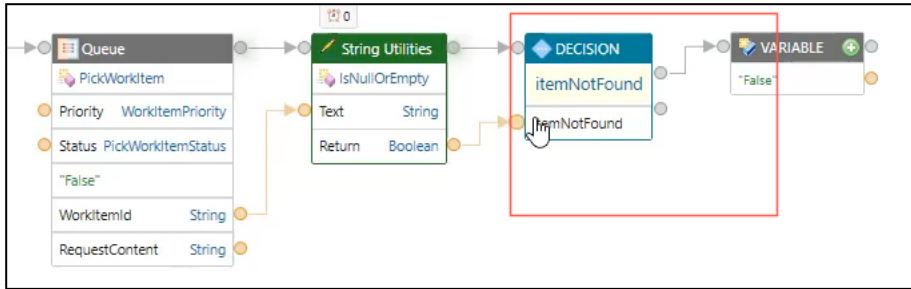
24. Double-click the Status property on the PickWorkItem Queue connector, in the Port Properties **dialog**, update the Read Data From selection to Static and select value as **Pending**, then click OK.
25. From the Toolbox pane, drag the **Utilities > String Utilities > IsNullOrEmpty** component to the design surface
26. Connect the **PickWorkItem** method to the **IsNullOrEmpty** component as shown in the below image.



27. From the **Toolbox**, drag the **Decision** component on to the design surface.
28. Double-click the **Decision** components condition and enter **itemNotFound**.
29. Connect the Decision component to the **IsNullOrEmpty** component as shown in the image below.

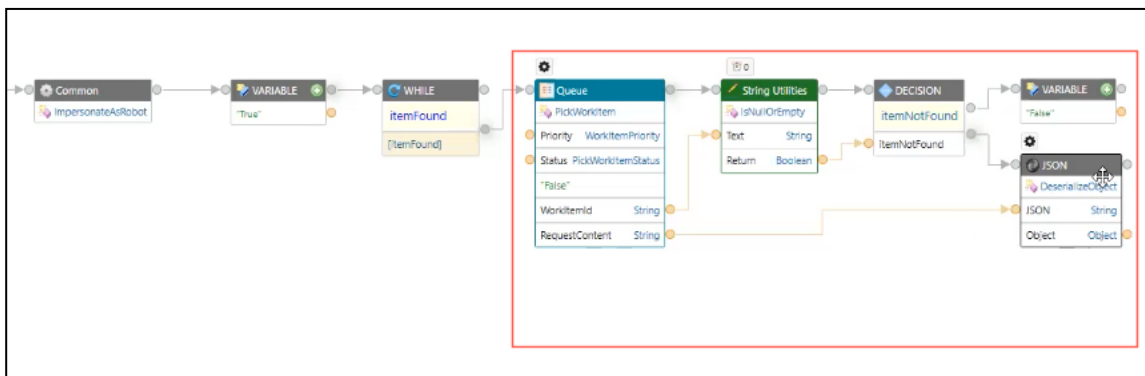


30. Drag the **itemFound** variable from the **Project Explorer** to the design surface. Enter the input as False
31. Connect the TRUE control output port of the **Decision** component to the **itemfound** variable as shown in the below image.

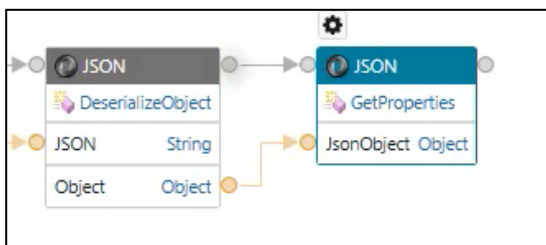


This will ensure we stop our while loop if there is no item received from the Queue in ServiceNow.

32. From the Toolbox pane, drag **Utilities > JSON > DeserializeObject** component to the design surface.
33. Connect the **RequestContent** data port of the **PickWorkItem** method to the data in port of the **DeserializeObject** component.
34. Connect the FALSE control out port of the **Decision component** to the control in port of the **DeserializeObject** component.



35. From the **Toolbox**, drag the **Utilities > JSON > GetProperties** component on to the design surface connected to the **DeserializeObject** component as shown in the below image.



36. On the **GetProperties** component bar, do the following:
 - a. Click the component settings icon (⚙️).
 - b. Click the add icon 5 times to add 5 JSON path notations.

c. Add entries as shown below.

SET PROPERTY CONFIG...

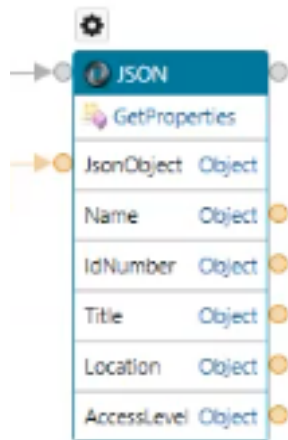
PROPERTIES

Name	Object	-
IdNumber	Object	-
Title	Object	-
Location	Object	-
AccessLevel	Object	-

OK

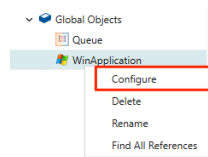
d. Click **OK**.

e. Verify that your output resembles the objects example shown below.



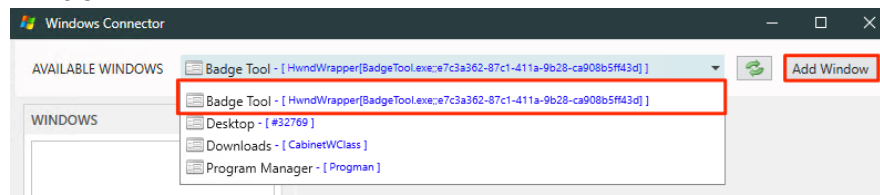
B. Insert data into the Badging App

1. To configure the Windows application, do the following:
 - a. Right-click the Windows application and select **Configure**.

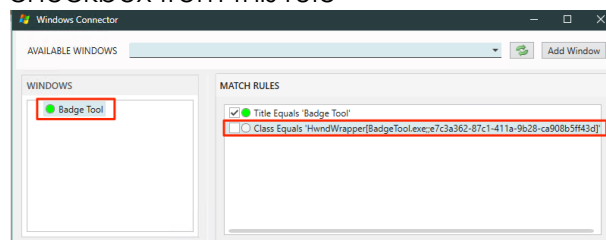


Note: Ensure the application is running or manually launch the app.

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



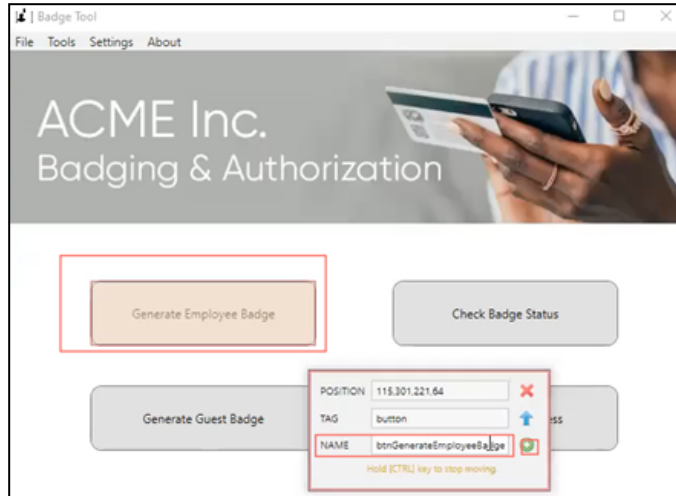
- a. In the Windows pane select the added **Badge Tool** entry, inspect the shown Match Rules
- b. As the class might change with every launch of BadgeTool, remove the checkbox from this rule



- c. In the Windows pane, right-click **Badge Tool** and select **Refresh**.
3. In the Windows pane, right-click **Badge Tool** and select **Add element** and perform the following:

Note: The Badge Tool application will be brought to front to allow selecting elements on it.

- a. Hover on Generate Employee Badge and enter **btnGenerateEmployeeBadge** in the Name field.
- b. Select the plus icon (+), the name field will be emptied as confirmation



- c. 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

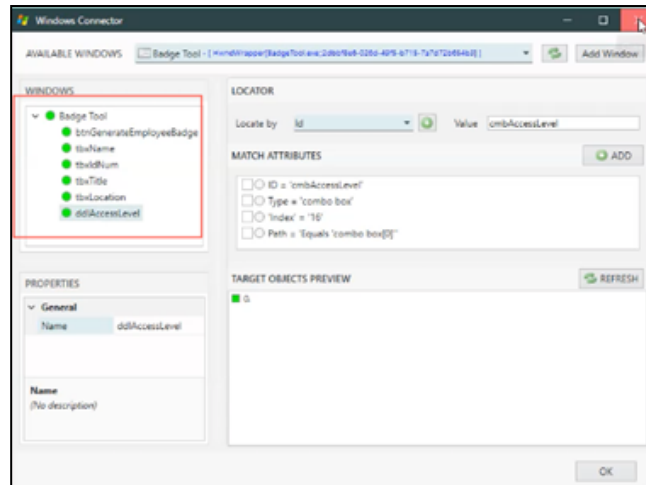


- a. Hover on **Name** text entry box and enter **tbxName** in the Name field, click the plus icon.
- b. Repeat for these fields:

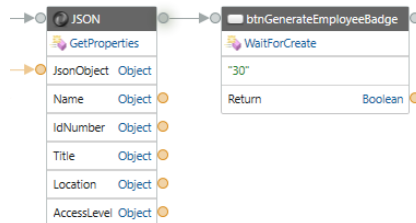
Text Field	Name
ID Number	tbxIdNum
Title	tbxTitle
Location	tbxLocation

- 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.

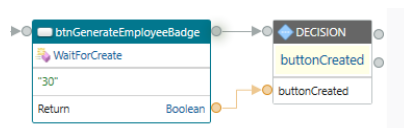
- e. Click the X icon to close the pop up.
- f. In the Windows tab, right-click the elements under **Badge Tool** and select **refresh**. Click ">" to see the list of added fields. Go to individual fields, right-click and select refresh. Ensure that all the field indicators turn green



- g. Click **OK** and close the window.
5. In the project explorer, expand the **WinApplication > Badge Tool** and double-click **btnGenerateEmployeeBadge**.
6. From the Object Explorer, drag **WaitForCreate** method on the design surface, connected to the **GetProperties** component.

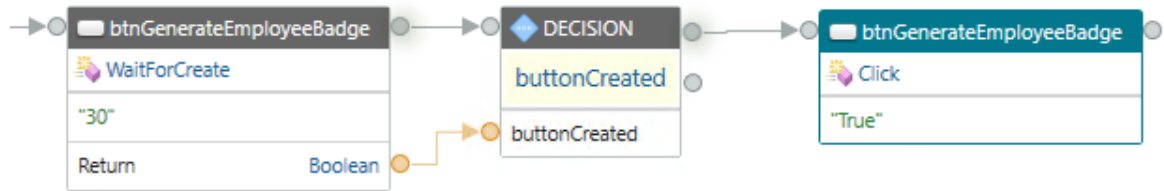


7. From the Toolbox pane, drag the **Decision** component on to the design surface, connected to the **WaitForCreate** method component.
8. In the **Decision** component, enter **buttonCreated** and connect the ports accordingly.



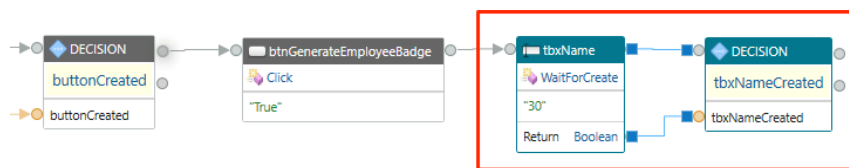
9. In the Project Explorer, navigate to **WinApplication > Badge Tool** and double-click **btnGenerateEmployeeBadge**.

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

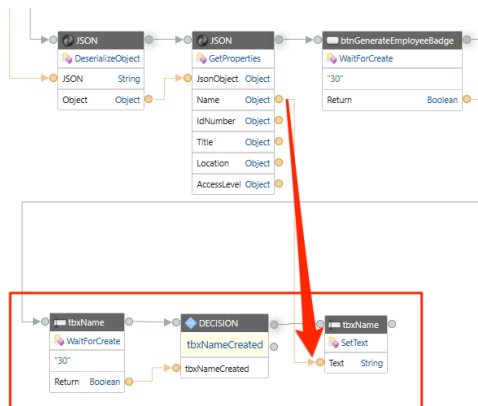


Note: We do a *WaitForCreate* to ensure the button is displayed on screen and use the decision to react if the button didn't show up after the timeout specified in the *WaitForCreate*. You can reuse this pattern to almost any screen interaction.

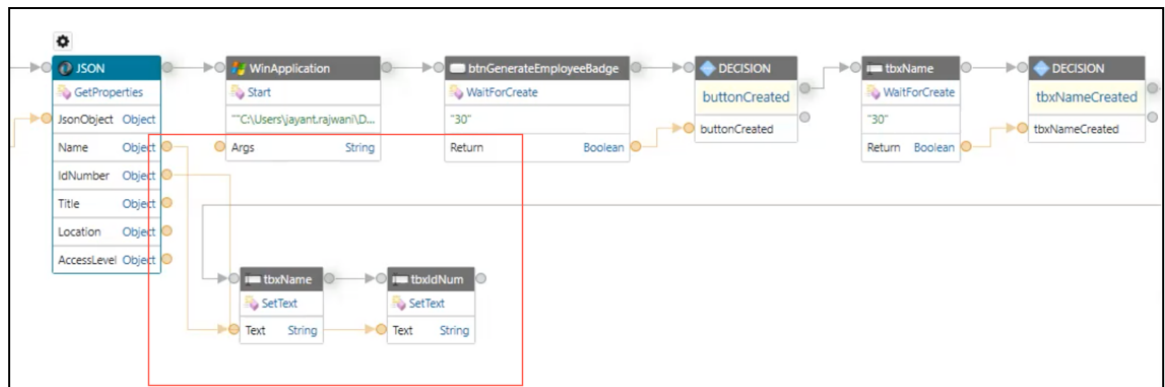
11. In the Project Explorer, navigate to **WinApplication > Badge Tool** and double-click **tbxName**.
12. From the Object Explorer, drag **WaitForCreate** method on the design surface, connected to the **Click** component.
13. From the Toolbox pane, drag **Decision** component on the design surface, connected to the **WaitForCreate** method component.
14. In the **Decision** component, enter **tbxNameCreated** and connect the ports accordingly.



15. In the Project Explorer, navigate to **WinApplication > Badge Tool** and double-click **tbxName**.
16. From the Object Explorer, drag **SetText** method on the design surface, connected to the **Decision** component. Connect the data ports of **JSON Name-Object** component and **tbxName-SetText** method.

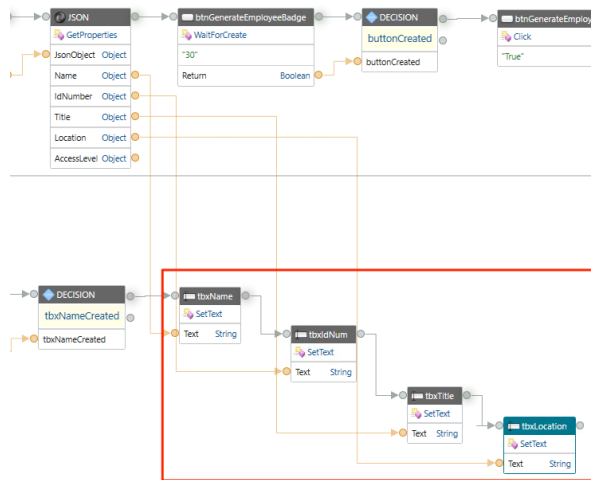


17. In the Project Explorer, expand the **WinApplication > Badge Tool** and double-click **tbxIdNum**.
18. From the Object Explorer, drag **SetText** method on the design surface, connected to the **tbxName-SetText** component. Connect the data ports of **JSON IdNumber-Object** component and **tbxIdNum-SetText** method.



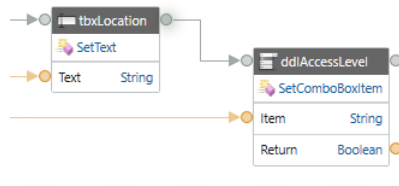
Note: We did not add a *WaitForCreate* this time as we expect all text fields to be on screen when the Name field is there which was checked before.

19. Repeat adding **SetText** components for Title and Location, connect them accordingly to the **GetProperties** outputs. The result will look like this:



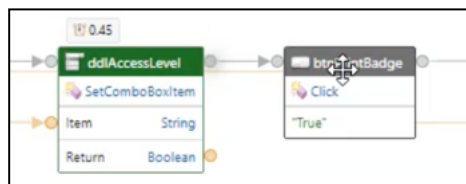
20. In the Project Explorer, navigate to **WinApplication > Badge Tool** and double-click **ddAccessLevel**.

21. Access level is a drop down list aka ComboBox, for this one we need the **SetComboBoxItem** method, drag it to the design surface connected to the **tbxLocation-SetText** component. Connect the data ports of **JSON AccessLevel-Object** component and **ddAccessLevel- SetComboBoxItem** method.



22. In the Project Explorer, navigate to **WinApplication > Badge Tool** and double-click **btnPrintBadge**.

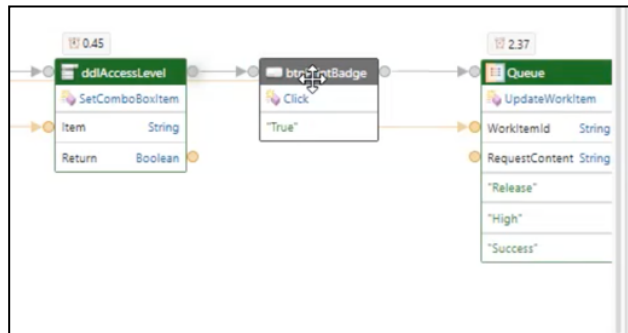
23. From the Object Explorer, drag the **Click** method on to the design surface, connected to the **ddlAccessLevel** component.



24. In the **Project Explorer**, double-click the **Queue** connector under the **Global Objects**.

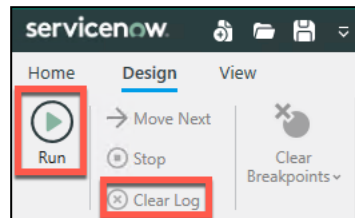
25. From the Object Explorer, drag **UpdateWorkItem** method on the design surface, connected to the **btnPrintBadge** component. Connect the data ports of the **PickWorkItem-Queue connector** (WorkItemId).

26. Double-click the In-progress value on the **UpdateWorkItem Queue** connector, in the **Port Properties** dialog, update the **Static Value** field to **Success** and click **OK**.



Lab Verification

1. Log in to your ServiceNow instance, go to the Badging Data queue, ensure you have items in state Pending. If not do the following:
 - a. Select all the work queue items and click **Reassign**.
 - b. Enter a comment and click **Reassign**.
2. On the windows machine, ensure Badge Tool application is not running.
3. In RPA Desktop Design Studio, do the following
 - a. Click **Assign bot process** and select your process and robot machine as before
 - b. **Clear the log** and select **Run**. Verify that your data updates in the work queue of your ServiceNow instance.



Note: Each executed component on the Main diagram should turn green. Failing components are marked by a red error icon, and more information for troubleshooting is available in the Execution Log.

5. In your ServiceNow instance, go to Badging Data queue and verify the Work Queue Items.

Details

Emp1 x

Badging Data

Purge After (days) 30Purge Status Success, Failure

Details

Work Queue Items (5)

Bot Process (1)

Work Queue Items 5

Just refreshed just now.

<input type="checkbox"/>	Name	Status	Stage	Created ▲	Started On	Completed On
	Emp1	Success	Stage_0	2022-04-12 22:39:54	2022-04-12 23:29:11	2022-04-12 23:29:15
	Emp2	Success	Stage_0	2022-04-12 22:39:56	2022-04-12 23:29:18	2022-04-12 23:29:22
	Emp3	Success	Stage_0	2022-04-12 22:39:59	2022-04-12 23:29:25	2022-04-12 23:29:29
	Emp4	Success	Stage_0	2022-04-12 22:40:01	2022-04-12 23:29:32	2022-04-12 23:29:36
	Emp5	Success	Stage_0	2022-04-12 22:40:04	2022-04-12 23:29:38	2022-04-12 23:29:43

Congratulations on completing the lab!