

Module 3 – Business Logic in Model-Driven Apps / Dataverse

Goal

To apply business logic and automate processes against records in Dataverse.

You will learn how to:

- Configure Business Rules
- Create real-time Dataverse Workflows
- Use the Dataverse connector in Power Automate
- Author and execute Dataverse Functions
- Build a Business Process Flow (BPF)

Table Of Contents

Configure Business Rules for Inspection	2
Create a Dataverse Real-Time Workflow.....	13
Create a Business Process Flow	24
Use Power Automate Flows with Dataverse	44
Create a Dataverse (Power Fx) Function	63
Execute a Dataverse Function from Power Automate	71

Configure Business Rules for Inspection

We will use the Business Rule designer in Dataverse to conditionally show or hide fields on the Inspection table.

1. In the Power Apps maker portal, navigate to the Inspections table, then select Business rules.

The screenshot shows the Power Apps maker portal interface. On the left, the 'Objects' sidebar is open, showing 'Inspection' selected. Under 'Inspection', 'Business rules' is highlighted with a red box. The main content area shows the 'Inspection' table properties: Name (Inspection), Primary column (Inspection Number), Type (Last modified), Standard (2 weeks ago). To the right, there are sections for Schema, Data experiences, and Customizations. The 'Customizations' section has a 'Business rules' item, which is also highlighted with a red box. At the bottom right, there are buttons for 'Update forms and views' and 'Edit'.

2. You will see all business rules configured for this table. Click + New business rule.

Note: A business rules must be **On** (activated) or they will not function.

3. Enter the business rule name: **BR - Inspection - Show/Hide Inspection Times**

4. Enter a description: Hides or shows the start/completion/pick up times if the drop off time contains data.

The screenshot shows the 'Inspection: New business rule' screen. The 'Business rule name' field contains 'BR - Inspection - Show/Hide Inspection Times', which is highlighted with a red box. The 'Description' field contains the text 'Hides or shows the start/completion/pick up times if the drop off time contains data.', which is also highlighted with a red box. Below the form are standard toolbar buttons: Add, Cut, Copy, Paste, Delete, and Snapshot.

5. In the top right, select the **Scope** you want this rule to apply. Leaving the default to **Entity** is fine here.

The screenshot shows the 'Business rule' screen with the 'Scope' dropdown menu open, showing 'Scope : Entity'. The rest of the screen is mostly obscured by a large black redaction box.

6. Select the **Condition** to show the **Properties** panel and enter in the following details:

- a. Display Name: **Drop Off Time Contains Data**

b. Entity: **Inspection**

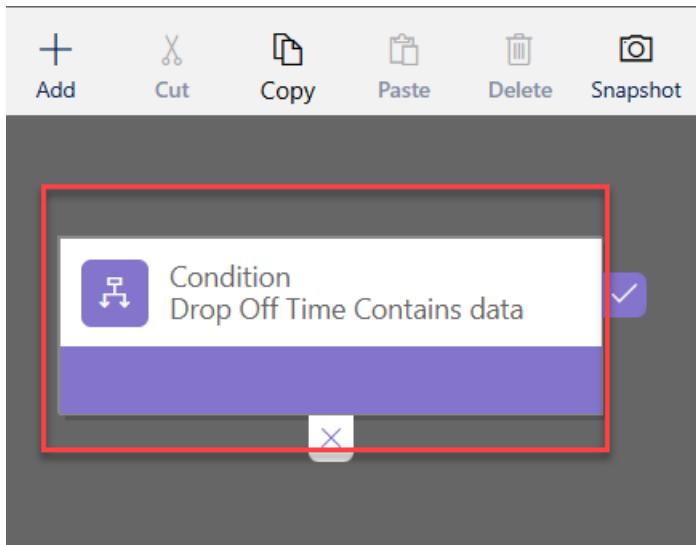
c. Rules:

i. Source: **Entity**

ii. Field: **Drop Off Time**

iii. Operator: **Contains data**

d. Click **Apply** (you may need to scroll) before clicking anywhere else otherwise it doesn't save.



Components Properties

Condition

Display Name
Drop Off Time Contains data

Entity
Inspection

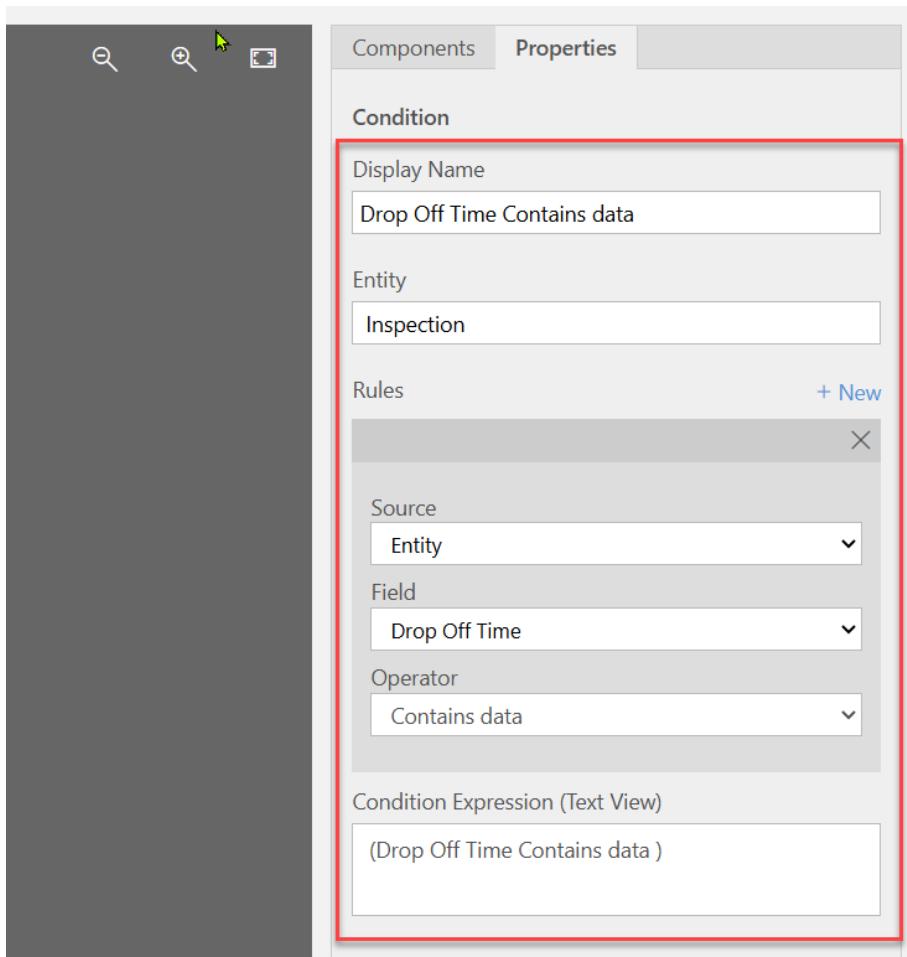
Rules + New

Source Entity

Field Drop Off Time

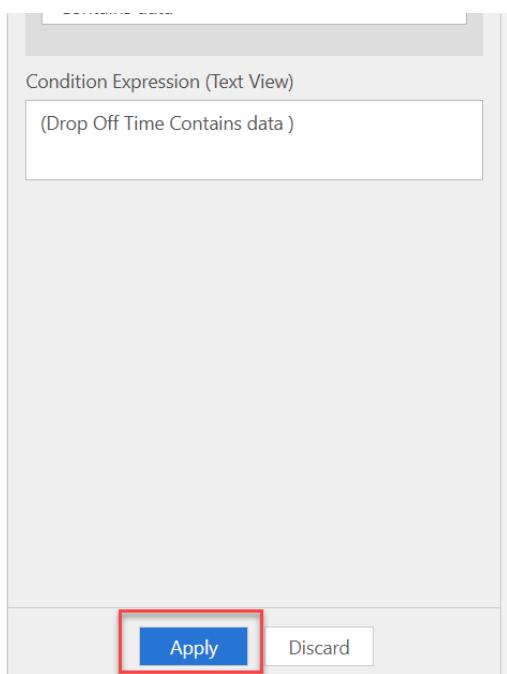
Operator Contains data

Condition Expression (Text View)
(Drop Off Time Contains data)

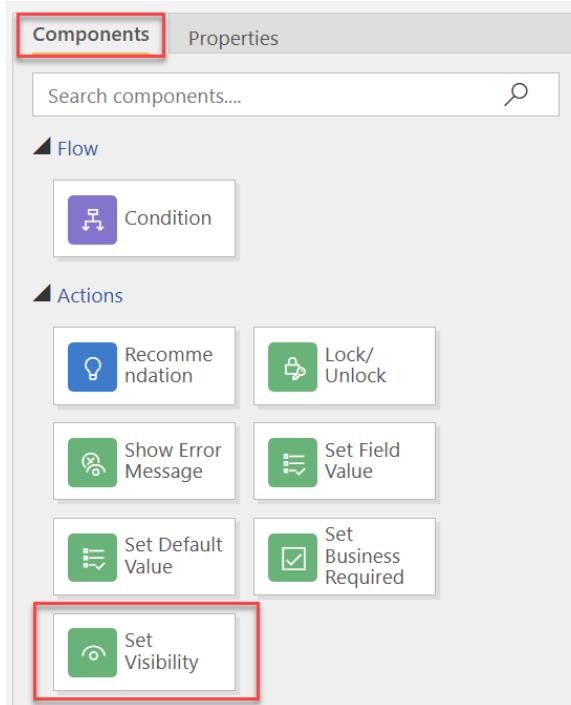


Condition Expression (Text View)
(Drop Off Time Contains data)

Apply Discard



7. Select the **Components** tab to see available **Actions** to perform.



8. Drag the Set Visibility action next to condition branch that evaluates to **true**

The screenshot shows the Power Apps canvas with a business rule named 'BR - Inspection - Show/Hide Inspection Times'. The rule has a single condition branch: 'Drop Off Time Contains data'. To the right of the branch, a 'Set Visibility' action is placed. A red box highlights the 'Set Visibility' action. At the top of the canvas, there are standard toolbar icons for Add, Cut, Copy, Paste, Delete, and Snapshot.

9. Enter in the **Properties** for the action (remember to hit **Apply** afterward):

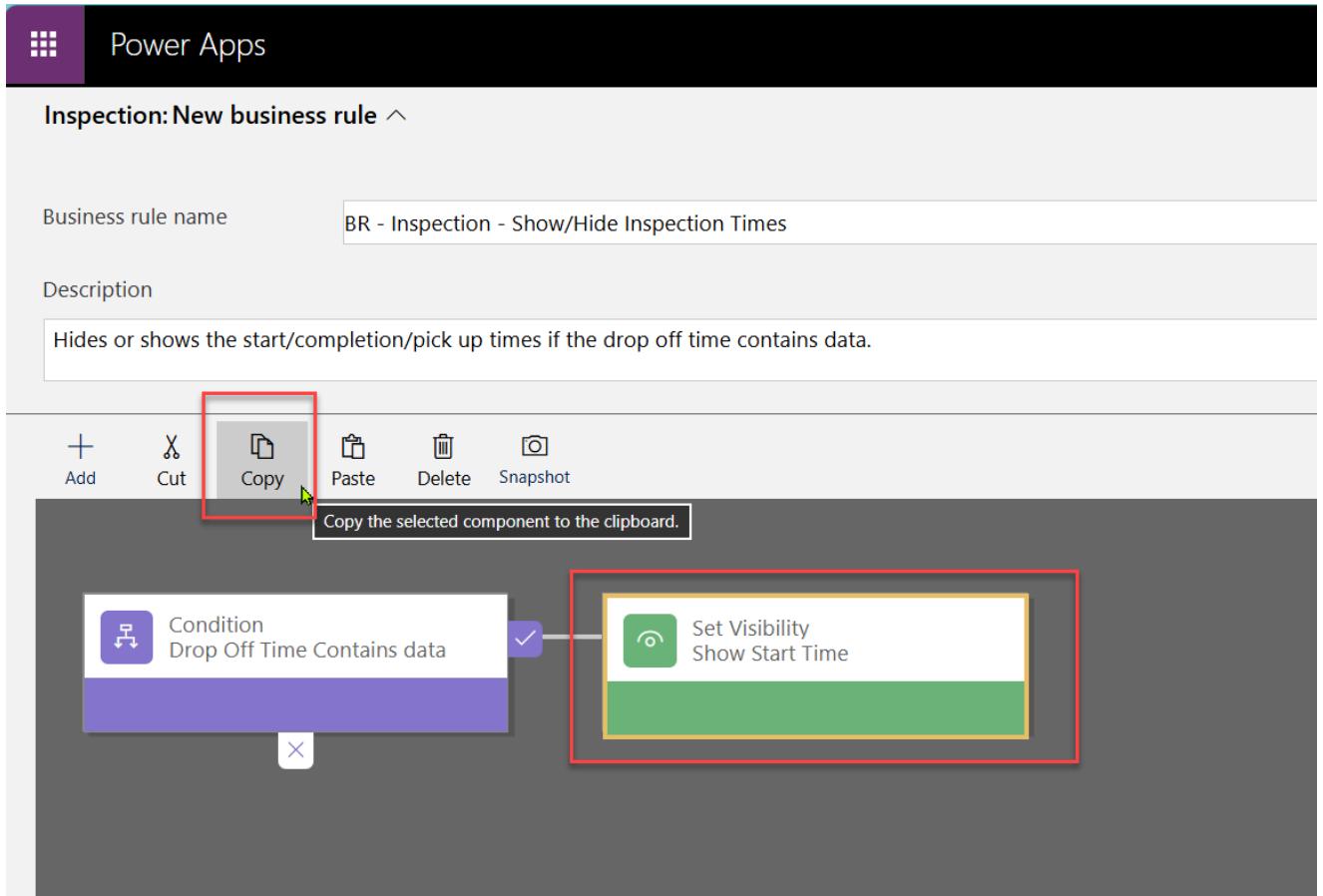
- a. Display Name: **Show Start Time**
- b. Visibility: Select the **Start Time** column and select **Yes**
- c. Click **Apply**

The screenshot shows a 'Properties' dialog box with tabs for 'Components' and 'Properties'. The 'Properties' tab is selected. Under 'Set Visibility', the 'Display Name' is set to 'Show Start Time' and the 'Entity' is set to 'Inspection'. In the 'Visibility' section, the 'Start Time' column is selected, and the 'Yes' option is chosen. A message at the bottom states 'Your changes haven't been applied yet.' with 'Apply' and 'Discard' buttons.

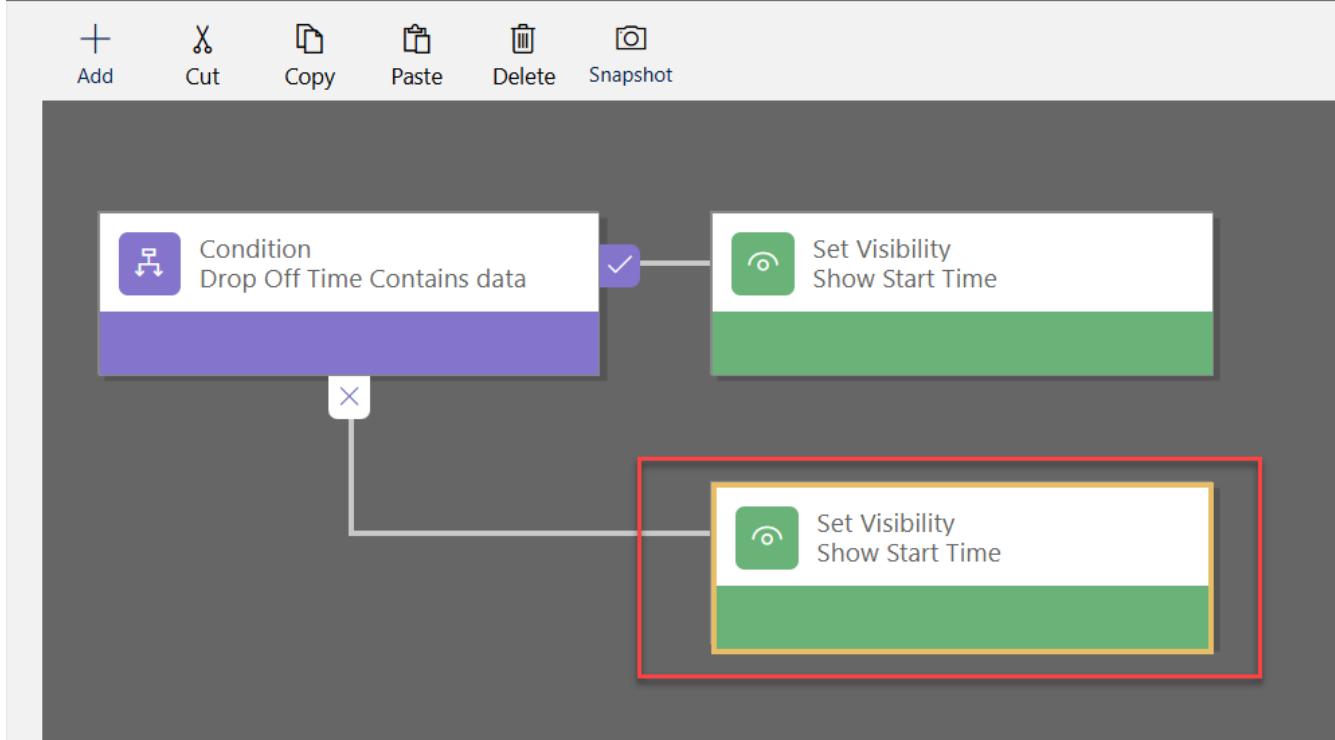
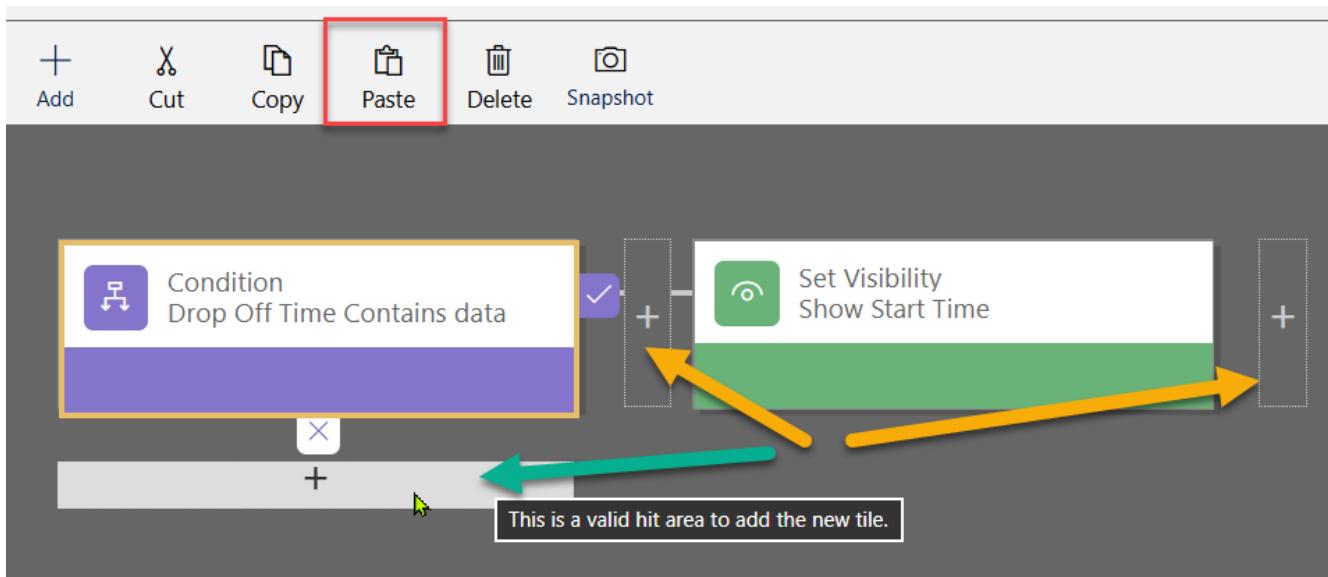
Setting	Value
Display Name	Show Start Time
Entity	Inspection
Start Time	Yes

Your changes haven't been applied yet.

10. Instead of dragging another component, select the Set Visibility component then click **Copy**



11. Click the **Paste** button, select a valid area (**grey rectangle with +**) to place it. In this case place it under the condition which is the false branch

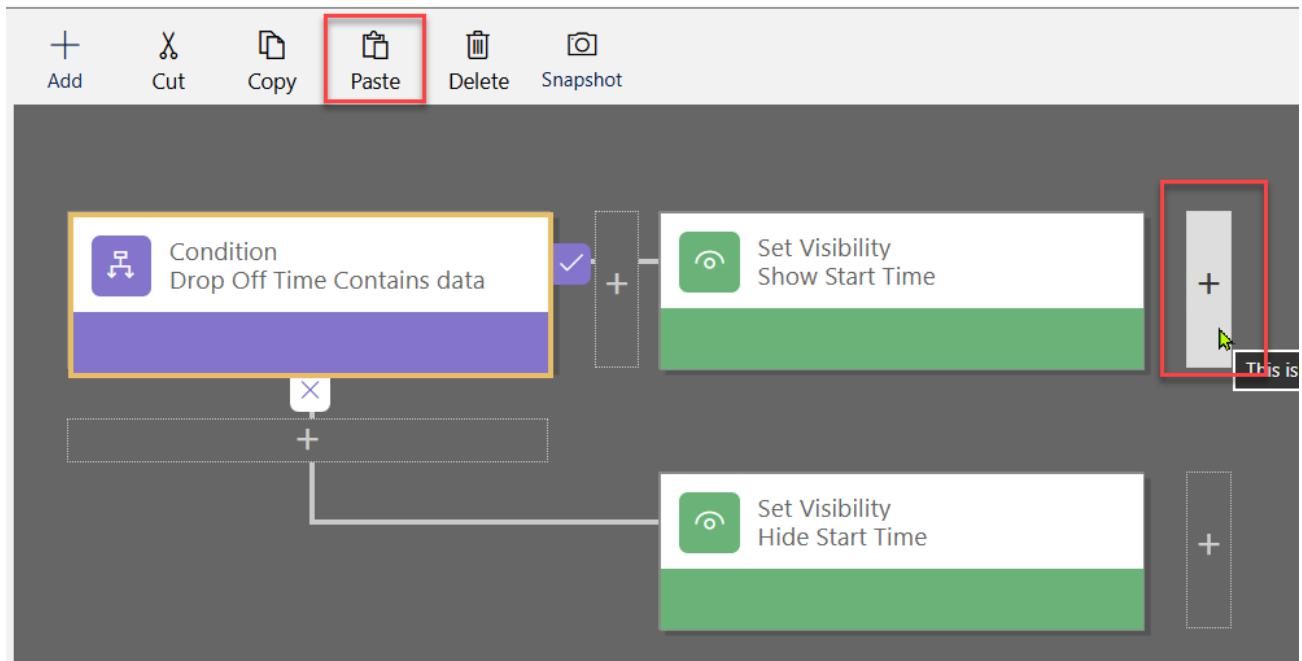


12. Select the Action that was pasted and update its properties:

- Display Name: **Hide Start Date**
- Visibility: Change to **No**
- Click **Apply**

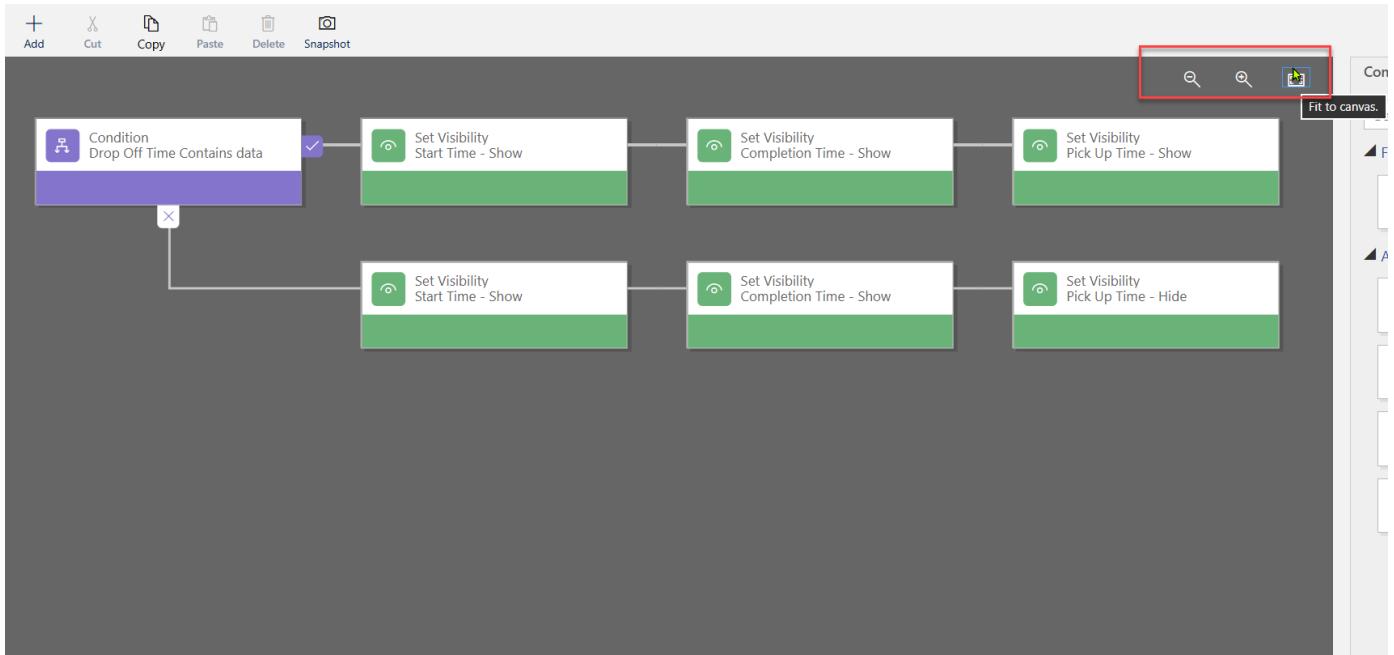
13. Click **Paste** again and place it on the top branch next to Show Start Time. Update the Properties:

- Display Name: **Show Completion Time**
- Visibility: Change column to **Completion Time**
- Click **Apply**



14. Click **Paste** again and place it on the top branch next to Show Completion Time. Update the Properties:
 - a. Display Name: **Show Pick Up Time**
 - b. Visibility: Change column to **Pick Up Time**
 - c. Click **Apply**
15. **Copy** the Hide Start Time action
16. **Paste** the action on the bottom branch next to Hide Start Time. Update the Properties:
 - a. Display Name: **Hide Completion Time**
 - b. Visibility: Change column to **Completion Time**
 - c. Click **Apply**
17. **Paste** the action on the bottom branch next to Hide Start Time. Update the Properties:
 - a. Display Name: **Hide Completion Time**
 - b. Visibility: Change column to **Completion Time**
 - c. Click **Apply**
18. Click **Paste** again and place it on the bottom branch next to Hide Completion Time. Update the Properties:
 - a. Display Name: **Hide Pick Up Time**
 - b. Visibility: Change column to **Pick Up Time**
 - c. Click **Apply**

19. You can use **zoom in**, **zoom out**, or **fit to canvas** to resize the view at any time.



20. In the bottom of the designer canvas, the **Business Rule (Text View)** should look like this:



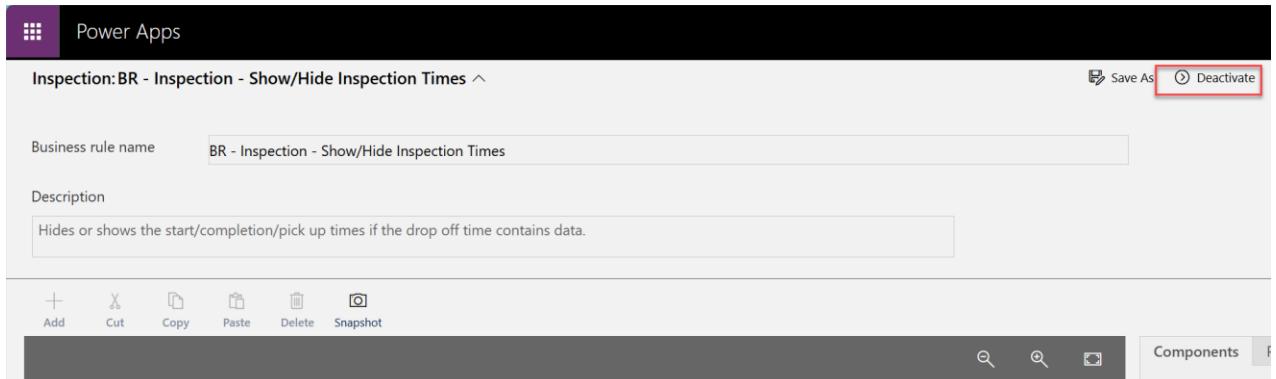
21. Click **Validate** to ensure your business rule is filled out correctly with no missing details.

22. Click **Save** to create the business rule.

23. Click **Activate** to turn the rule on.



24. If you need to make modifications to the rule, you need to first **Deactivate** it. Once you are done making changes, **Activate** it again.



25. You can also turn the rule **Off (Deactivate)** or **On (Activate)** in the maker portal screen.

The screenshot shows a list of business rules under 'Vehicle Inspections > Tables > Inspection > Business rules'. The first rule, 'BR - Inspection - Set Always Required Fields', has its status set to 'On'. The second rule, 'BR - Inspection - Show/Hide Inspection Times', has its status set to 'On' and is selected, indicated by a checked checkbox in the first column. The 'Turn off' button for this selected rule is highlighted with a red box. Other columns include 'Name', 'Status', 'Scope', and 'Customized'.

26. Open the **Vehicle Inspection Management** app.

27. Navigate to **Inspections** and click **+New** to open the form for a new record.

28. Select the **Timeline** tab.

The screenshot shows the 'New Inspection' form. On the left is a navigation sidebar with 'Home', 'Recent', 'Pinned', 'Inspections', 'Inspections' (selected), 'Inspection Checklist I...', 'Repair Quotes', and 'Customers'. The main area has tabs for 'General', 'Timeline' (highlighted with a red box), 'Checklist', and 'Admin'. Under 'Inspection Times', there are fields for 'Scheduled Time' and 'Dropped Off Time', each with a date picker icon.

29. The Start Time, Completion Time, and Pick Up Time columns are **hidden**

30. Enter the current date in the **Dropped Off Time**

31. The Start Time, Completion Time, and Pick Up Time columns **appear**.

Inspection Times

Scheduled Time	---	...
Dropped Off Time	10/7/2025	8:00 AM
Start Time	---	...
Completion Time	---	...
Pick Up Time	---	...

32. Delete the value from Dropped Off Time.

33. The Start Time, Completion Time, and Pick Up Time columns become **hidden** again.

Inspection Times

Scheduled Time	---	...
Dropped Off Time	---	...

34. The business rule is behaving as expected.

35. Click the back arrow or navigate away and click **Discard Changes** when prompted.

Create a Dataverse Real-Time Workflow

We will now create a classic real-time workflow to automatically set the primary name column of the Vehicle table to show a composite of the vehicle year/make/model and Customer name. This will make it easily find and identify an existing customer's vehicle.

36. Navigate to the **Processes** page where you'll see all **Workflows, Business Process Flows, and Actions**.

Like business rules, a Workflow must be turned **On (Active)** to execute.

The screenshot shows the Power Apps interface with the 'Processes' page selected. The left sidebar lists various objects like Agents, Apps, and Processes. The main area displays two workflows: 'Inspection Type Area: Set Name' and 'Vehicle Inspection Process'. Both workflows are listed as 'Process (Workflow)' type, managed by Chris Piatecki, and were last modified 2 months ago. The 'Status' column indicates both are 'On'. A red box highlights the 'Processes (2)' link in the sidebar.

37. To create a new Workflow, click **+New > Automation > Process > Workflow**.

The screenshot shows the 'New' menu in the Power Apps interface. The path selected is: + New > Automation > Process > Workflow. Each step in the path is highlighted with a red box. The final step, 'Workflow', has a green hand cursor icon pointing at it, indicating the target for creation.

38. Enter the following details into the side panel:

- a. Display name: **Vehicle: Set Name**
- b. Table: **Vehicle**
- c. Run workflow in the background: **Uncheck**
- d. Click **Create**

New workflow

X

Display name *

Vehicle: Set Name

Table *

Vehicle

Run workflow in the background
(recommended)

Start from

Blank

Template

39. The classic workflow designer will open in a new browser tab. Select the following **Options for Automatic Processes:**

- a. Scope: **Organization**
- b. Start when: **After**
 - i. Record is **created**
 - ii. Record fields **change** - click select to pick the following columns:
 1. **Contact**
 2. **Make**
 3. **Model**
 4. **Year**
- c. Execute as: **The user who made changes to the record**

Process: Vehicle: Set Name

Information

Common

- Information
- Audit History
- Catalog Assignments
- Comments
- SLA Items
- PM Recordings
- Copilot components
- Card

Process Sessions

- Process Sessions

General Administration Notes

Hide Process Properties

Process Name *

Activate As

Available to Run

Run this workflow in the background (recommended)

As an on-demand process

As a child process

Options for Automatic Processes

Scope

Start when:

After Record is created

After Record status changes

Before Record is assigned

Record fields change

Execute as:

The owner of the workflow

The user who made changes to the record

Entity

Category

Workflow Log Retention

Keep logs for workflow jobs that encountered errors

Select Fields

Select the fields that the process will monitor for changes.

<input type="checkbox"/>	Display Name ▲	Name	Type
<input type="checkbox"/>	Color	ppcc_color	Option Set
<input checked="" type="checkbox"/>	Contact	ppcc_contact	Lookup
<input type="checkbox"/>	Created By	createdby	Lookup
<input type="checkbox"/>	Created By (Delegate)	createdonbehalfby	Lookup
<input type="checkbox"/>	Created On	createdon	Date and Time
<input type="checkbox"/>	License Plate #	ppcc_licenseplate	Single Line of Text
<input checked="" type="checkbox"/>	Make	ppcc_makeid	Lookup
<input checked="" type="checkbox"/>	Model	ppcc_modelid	Lookup
<input type="checkbox"/>	Modified By	modifiedby	Lookup
<input type="checkbox"/>	Modified By (Delegate)	modifiedonbehalfby	Lookup
<input type="checkbox"/>	Modified On	modifiedon	Date and Time
<input type="checkbox"/>	Name	ppcc_name	Single Line of Text
<input type="checkbox"/>	Owner	ownerid	Owner
<input type="checkbox"/>	Owning Business Unit	owningbusinessunit	Lookup
<input type="checkbox"/>	Record Created On	overriddencreatedon	Date and Time
<input type="checkbox"/>	Registration Renewal Date	ppcc_registrationrenewal...	Date and Time
<input type="checkbox"/>	Status	statecode	Status
<input type="checkbox"/>	Status Reason	statuscode	Status Reason
<input type="checkbox"/>	Trim	ppcc_trimid	Lookup
<input type="checkbox"/>	VIN	ppcc_vin	Single Line of Text
<input checked="" type="checkbox"/>	Year	ppcc_yearid	Lookup

40. It's a good idea to update the Notes section to describe the workflow purpose and any changes made over time.

Process: Vehicle: Set Name

Information

Common

- Information
- Audit History
- Catalog Assignments
- Comments
- SLA Items
- PM Recordings
- Copilot components
- Card

Process Sessions

- Process Sessions

Notes

Enter a note

Title: Note created on 10/1/2025 10:20 PM by Chris Piasecki
Sets the name of the record to a composite of: [Year] [Make] [Model] ([Contact Full Name]).

Chris Piasecki 10/2/2025 4:58 PM

41. Click Add Step to add an **Update Record** step to the workflow.

Options for Automatic Processes

Scope: Organization

Start when:

- After: Record is created
- Before: Record status changes
- After: Record is assigned
- Before: Record fields change
- Before: Record is deleted

Execute as:

- The owner of the workflow
- The user who made changes to the record

Select

Add Step ▾

Stage

Check Condition

Conditional Branch

Default Action

Create Record

Update Record

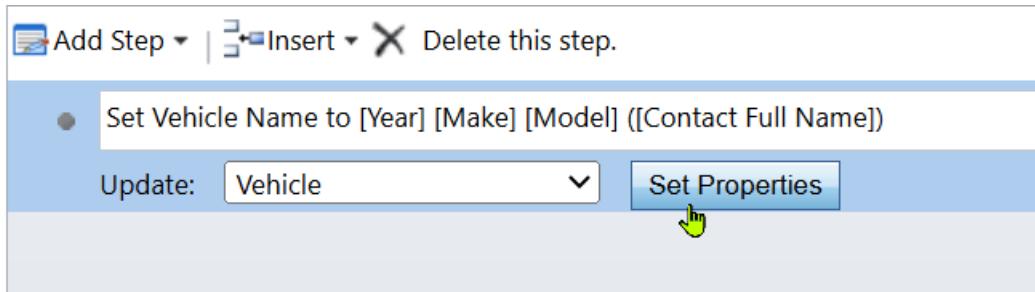
Assign Record

Send Email

Start Child Workflow

42. Paste in the following description:

Set Vehicle Name to [Year] [Make] [Model] ([Contact Full Name])



43. Clicking **Set Properties** will open the step configuration in a new browser window.

44. You'll see fields laid out based on the primary main form.

45. Collapse the General and Inspections tabs to bring the **Admin** tab within view where you will find the **Name** field. **Select** it to bring focus.

Note: Any fields that are not on the primary main form will be listed under Additional Fields.

Process: Vehicle: Set Name
Update Vehicle

General

Inspections

Admin

RECORD INFO

Name

Owner

Created On

Created By

Modified On

Modified By

Status

Additional Fields

Form Assistant

Dynamic Values

Operator: Set to

Look for: Vehicle

Color

Add

OK

46. Use the Form Assistant on the right hand side to set the column value.

47. Set the Operator as **Set to**

48. First we want populate the **Vehicle year**:

- Look for: **Vehicle**
- Select the **Year** column
- Click **Add**
- Click **OK**

Form Assistant

Dynamic Values

Dynamic Values

Operator:

Set to

Look for:

Vehicle

Year

Add

X |

Year(Vehicle)

Default value:

OK

49. Select the Name field to put focus and **add a space** after the Year

Process: Vehicle: Set Name

Update Vehicle

- ▶ General
- ▶ Inspections
- ◀ Admin

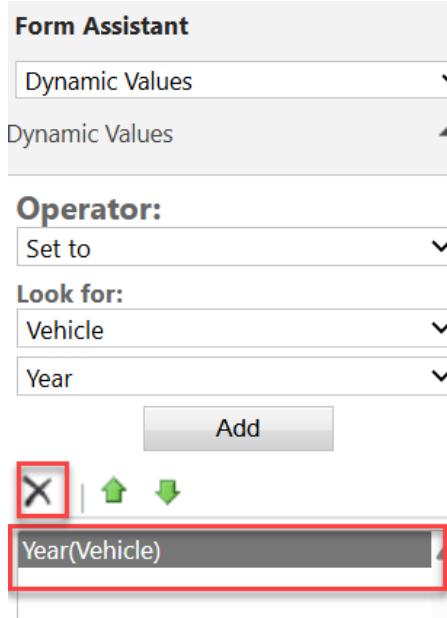
RECORD INFO

Add a space

Name

Owner

50. In the Form Assistant, remove Year from the added fields.

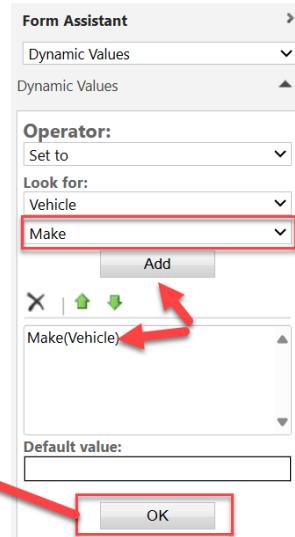


51. Then add the **Make** column:

- Select the **Make** column
- Click **Add**
- Click **OK**

Process: Vehicle: Set Name
Update Vehicle

The screenshot shows the 'Update Vehicle' form. On the left, there are sections for 'General', 'Inspections', and 'Admin'. Under 'Admin', there is a 'RECORD INFO' section. The 'Name' field is highlighted with a yellow box. Other fields include 'Owner', 'Created On', 'Created By', 'Modified On', 'Modified By', and 'Status'. At the bottom, there is a 'Additional Fields' section.



52. Select the Name field to put focus and **add a space** after the Make.

53. In the Form Assistant, remove Make from the added fields

54. Then add the **Model** column:

- Select the **Model** column
- Click **Add**
- Click **OK**

55. Select the Name field to put focus and **add a space** after the Model (you may need to use right arrow key to move the cursor if you don't see the entire text).

56. Also add an **open bracket** “(“ (without the quotes)

57. In the Form Assistant, remove Model from the added fields

58. Then add the **Full name** column from Contact:

- Change **Look for** to **Contact (contact)**
- Select the **Full Name** column
- Click **Add**
- Click **OK**

Process: Vehicle: Set Name

Update Vehicle

General

Inspections

Admin

RECORD INFO

Name **(e)) ({Full Name(Contact (Contact)))}**

Owner

Created On

Created By

Modified On

Modified By

Status

Additional Fields

Form Assistant

Dynamic Values

Operator: Set to

Look for: Contact (Contact)

Full Name

Add

Full Name(Contact (Contact))

Default value:

OK

```
{Year(Vehicle)} {Make(Vehicle)} {Model(Vehicle)} ({Full Name(Contact (Contact)))}
```

59. Select the Name field to put focus and add a **closing bracket “)”** (without the quotes)

60. The final result should look like below:

```
{Year(Vehicle)} {Make(Vehicle)} {Model(Vehicle)} ({Full Name(Contact (Contact)))})
```

61. Click **Save and Close**

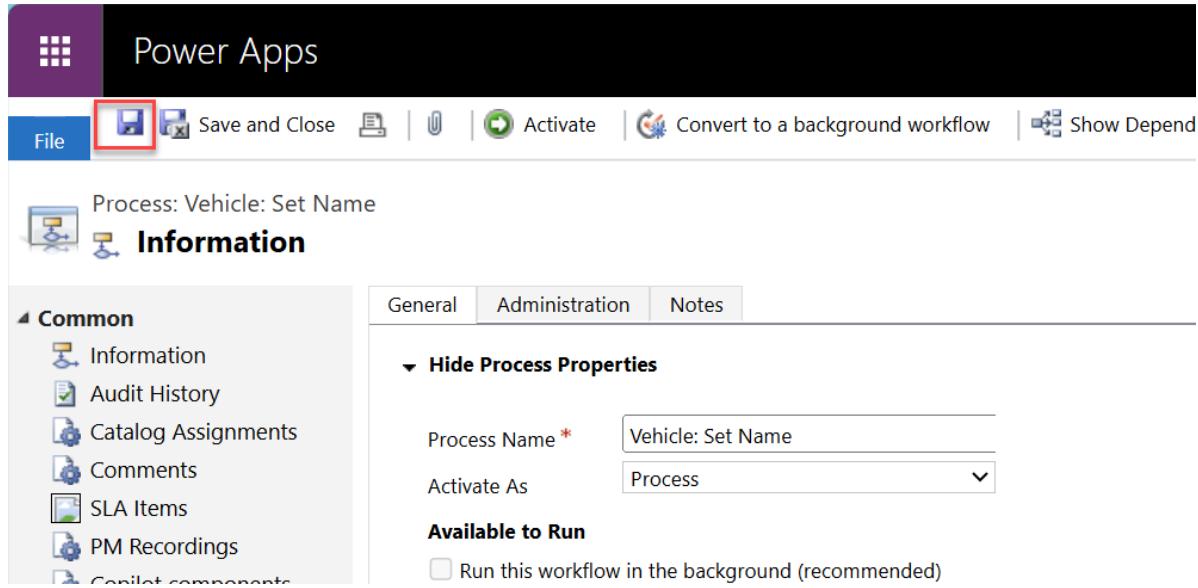


Process: Vehicle: Set Name

Update Vehicle

General

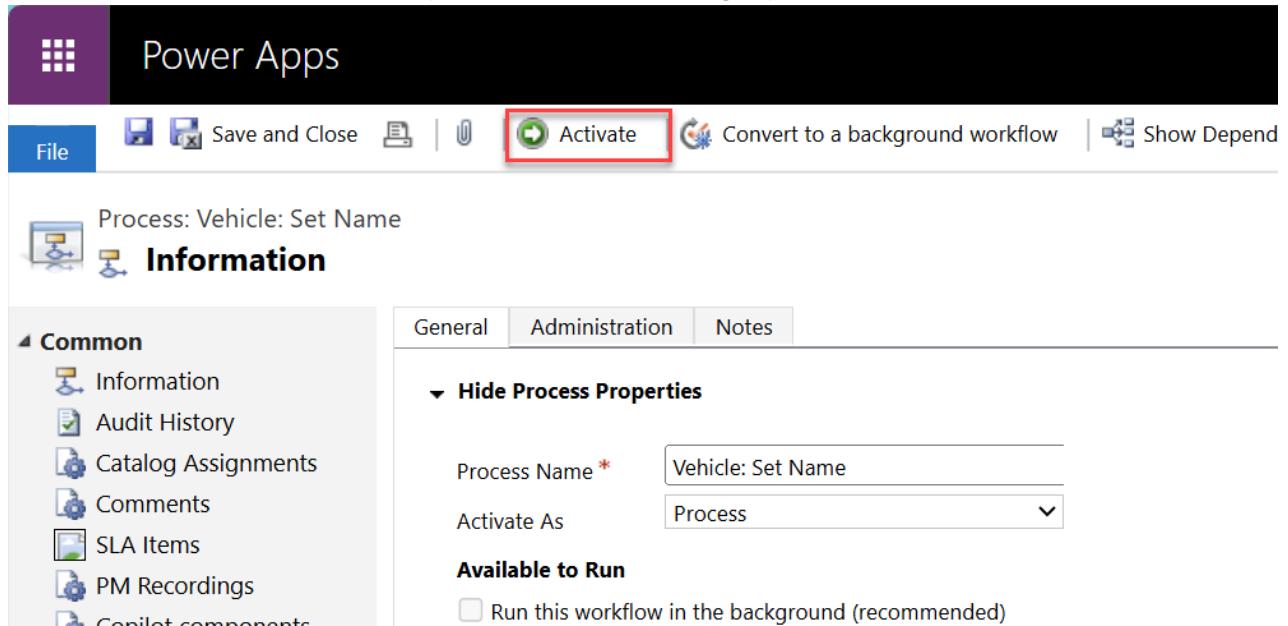
62. **Save** the workflow.



The screenshot shows the Power Apps interface with the title 'Power Apps' at the top. In the top navigation bar, there are several icons: 'File' (highlighted with a red box), 'Save and Close', 'Activate' (highlighted with a red box), 'Convert to a background workflow', and 'Show Depend'. Below the navigation bar, the title 'Process: Vehicle: Set Name' is displayed, followed by the word 'Information'. On the left, there is a sidebar with a 'Common' section containing links for 'Information', 'Audit History', 'Catalog Assignments', 'Comments', 'SLA Items', 'PM Recordings', and 'Copilot components'. The main content area has tabs for 'General', 'Administration', and 'Notes', with 'General' selected. Under the 'General' tab, there is a section titled 'Hide Process Properties' with fields for 'Process Name' (set to 'Vehicle: Set Name') and 'Activate As' (set to 'Process'). There is also a checkbox for 'Available to Run' and another for 'Run this workflow in the background (recommended)'.

63. **Activate** the workflow

Note: Similar to business rules, if you want to make changes you must deactivate the workflow first.



This screenshot is identical to the previous one, showing the 'Information' page for the 'Vehicle: Set Name' workflow. The 'Activate' button in the top navigation bar is again highlighted with a red box. The rest of the interface, including the sidebar, tabs, and process properties, remains the same.

64. Open the **Vehicle Inspection Management** app.

65. Expand Customers in the navigation and select **Vehicles**

66. Click **+New** to open the form for a new record.

The screenshot shows the Power Apps interface for 'Vehicle Inspection Management'. On the left, there's a navigation bar with items like Home, Recent, Pinned, Inspections, Customers (with an upward arrow), Contacts, Vehicles (which is highlighted with a red box), and Inspections Standards. The main area displays a list titled 'Active Vehicles' with columns for Name, Year, and Make. The list includes several vehicles from different years and manufacturers. At the top right, there are buttons for Show Chart, New (which is highlighted with a red box), Delete, Refresh, and Visualize this view.

Name	Year	Make
2004 Audi Q5 (Sidney Higa)	2004	Audi
2005 Hyundai Elantra (Paul Cannon)	2005	Hyundai
2006 Nissan Maxima (Paul Cannon)	2006	Nissan
2008 Mercedes-Benz E-Class (Patrick Sa...)	2008	Mercedes-Ber
2009 BMW X5 (Susan Burk)	2009	BMW
2010 Chevrolet Malibu (Susanna Stubbe...)	2010	Chevrolet
2012 Toyota Corolla (Thomas Andersen)	2012	Toyota

67. Enter the following details:

- a. VIN number **2G1FG3D33C9148061**
- b. Make: **Audi**
- c. Model: **A4**
- d. Year: **1994**
- e. Contact: **Jim Glynn**

68. Click **Save**

The screenshot shows the 'New Vehicle' form. At the top, there are buttons for Back, Forward, Save (which is highlighted with a red box), Save & Close, New, and Flow. Below that, the title is 'New Vehicle - Unsaved'. There are tabs for General, Inspections, and Admin, with General selected. Under the General tab, there's a section labeled 'DETAILS' with a 'License Plate #' field containing '---'. The rest of the form is mostly blank.

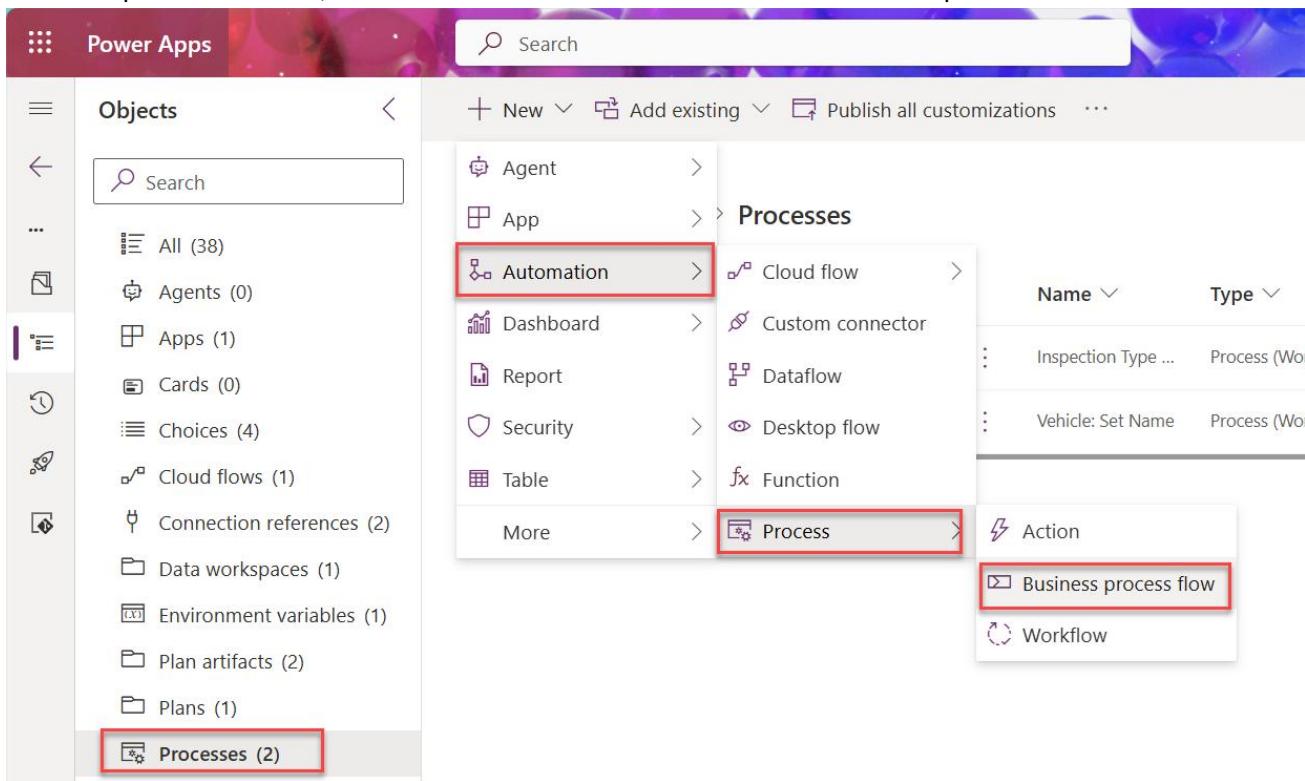
69. The record name updates in real-time and is reflected upon automatic refresh of the form.

The screenshot shows a Power Apps application window titled "Power Apps | Vehicle Inspection Management". The left sidebar has a "Vehicles" section selected. The main area displays a vehicle record for a "2014 Audi A4 (Jim Glynn) - Saved vehicle". The "General" tab is selected. The "DETAILS" section contains fields for "License Plate #" (G7H 8J9) and "VIN" (1J4GZ58S4VC030485). The "VEHICLE INFORMATION" section contains fields for "Make" (Audi), "Model" (A4), "Year" (2014), and "Trim". The "ADDITIONAL INFO" section contains fields for "Contact", "Email", and "Business". The top navigation bar includes buttons for Save, Save & Close, New, Deactivate, Delete, Refresh, and a refresh icon.

70. The name correctly matches the **YEAR MAKE MODEL (CONTACT FULL NAME)** pattern we configured.

Create a Business Process Flow

71. In the maker portal, navigate to solutions and open the **Vehicle Inspections** solution.
72. On the top command bar, select +New > Automation > Process > Business process flow



73. Enter the following information:
 - Display name: **Vehicle Inspection Process**
 - Name: **vehicleinspectionprocess** (default)
 - Table: **Inspection**

New business process flow X

Use business process flows to define a set of steps for people to follow to take them to a desired outcome.

Display name *

Vehicle Inspection Process

Name *

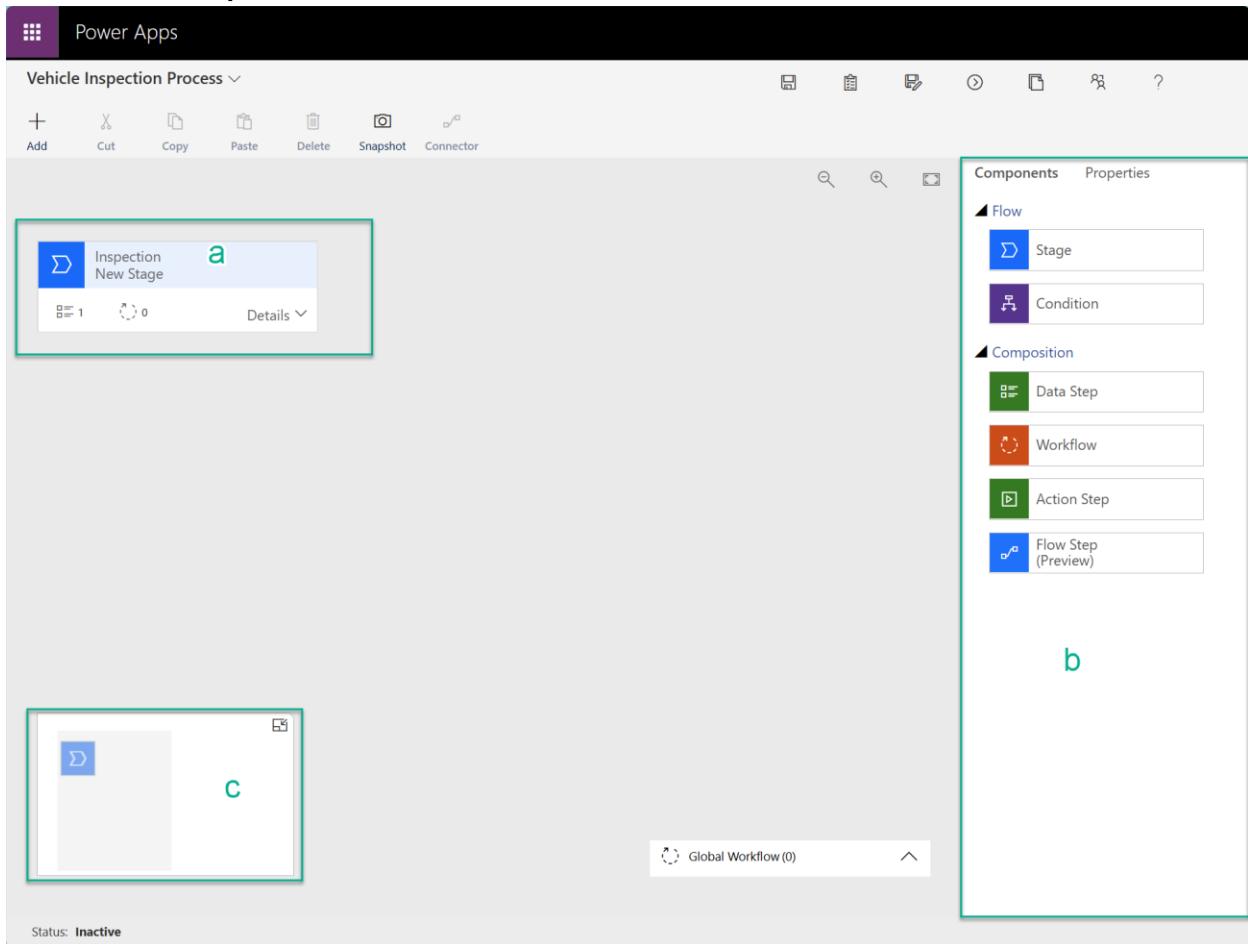
ppcc_ vehicleinspectionprocess

Table *

Inspection

74. The business process flow (BPF) designer will open in a new browser tab.
75. Similar to the business rules designer, you see:

- Main canvas where components are placed
- Side panel with **Components** and **Properties** of a selected component.
- A **minimap** in the bottom left



76. Select the existing stage component on the canvas to view its properties. Here we can set the stage Display Name and other properties.

- Display Name: **Intake**
- Category: **leave blank**
- Entity: **Inspection** (first stage is fixed to the table selected in the previous creation dialog)
- Click **Apply** to save the step change

Stage

Display Name

Intake

Category

Entity

Business Rules

[Business rules for this stage's entity](#)

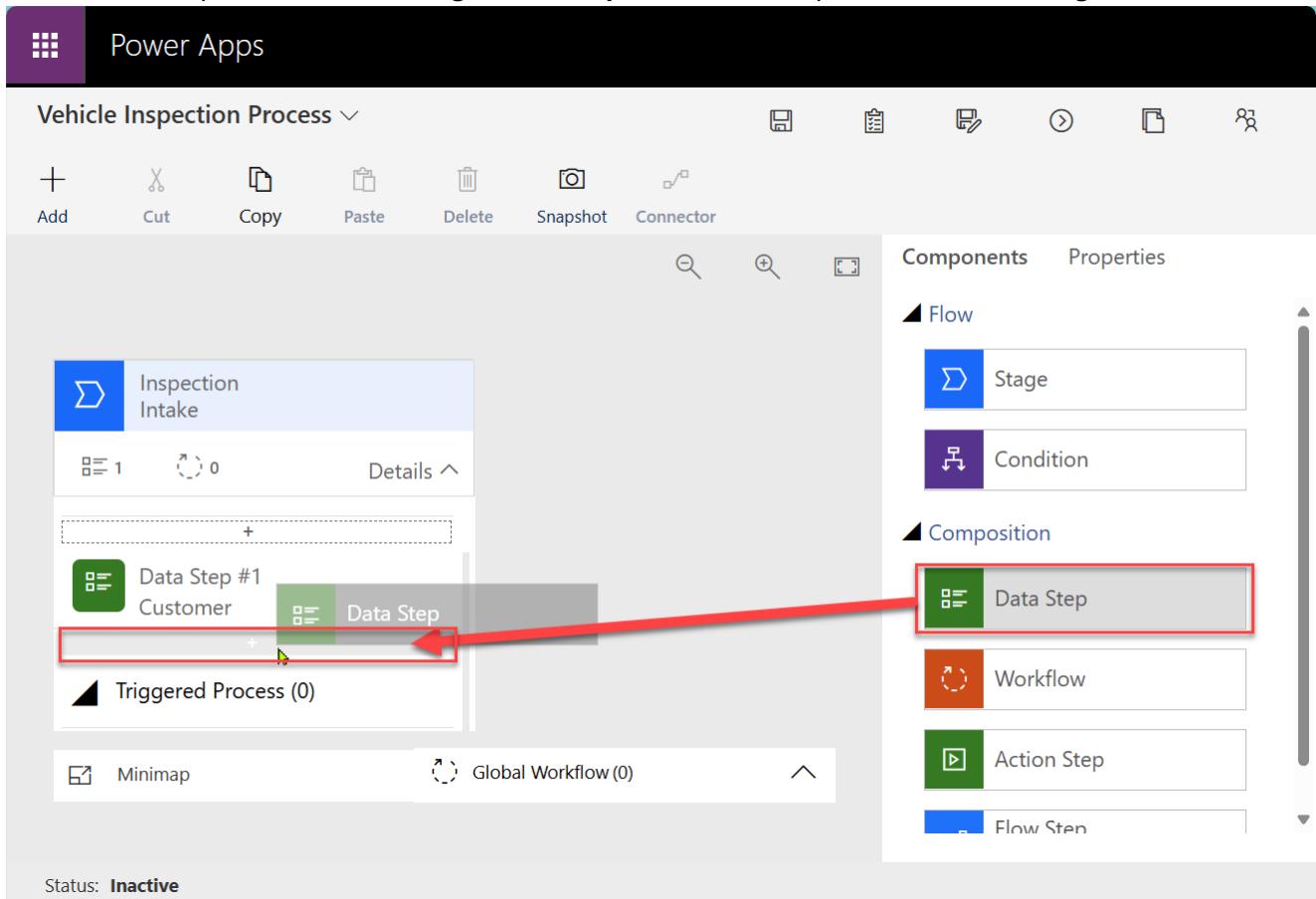
! Apply Discard

77. Expand the Details of the Intake stage. Select **Data Step #1** to configure its Properties:

- Step Name: **Customer**
- Data Field: **Customer**
- Required: **Check**
- Sequence: 1
- Click **Apply**

The screenshot shows the Power Apps canvas interface for the "Vehicle Inspection Process". The "Intake" stage is currently selected. In the "Data Step" section of the properties pane, the "Step Name" is set to "Customer" and the "Data Field" is also set to "Customer". The "Required" checkbox is checked. The "Sequence" is set to 1. At the bottom right of the properties pane, the "Apply" button is highlighted with a red box. A tooltip below the "Apply" button states: "The Apply button is available. Changes can be applied".

78. Select the Components tab and drag a **Data Step** under Data Step #1 of the Intake stage.



79. Enter Data Step #2 properties:

- Step Name: **Vehicle**
- Data Field: **Vehicle**
- Required: **Check**
- Sequence: 2
- Click **Apply**

80. Select the Components tab and drag a **Data Step** under Data Step #2 of the Intake stage.

81. Enter Data Step #3 properties:

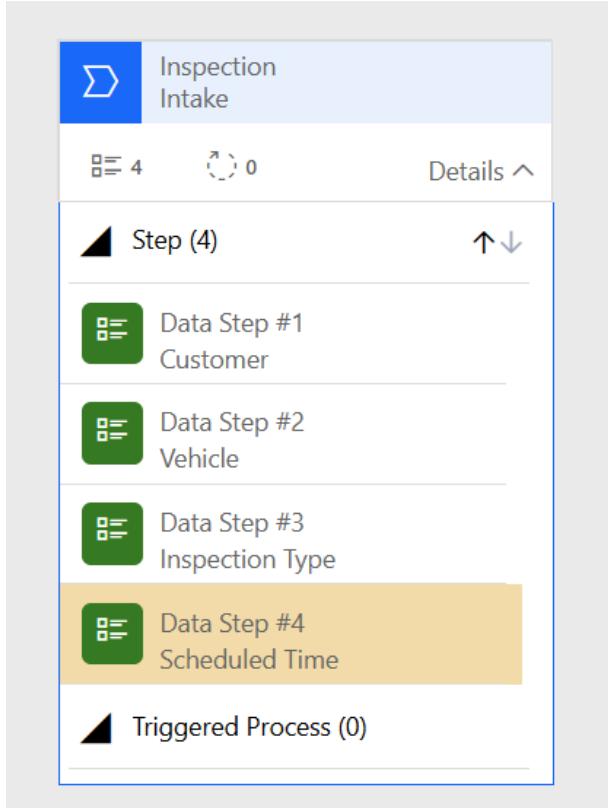
- Step Name: **Inspection Type**
- Data Field: **Inspection Type**
- Required: **Check**
- Sequence: 3
- Click **Apply**

82. Select the Components tab and drag a **Data Step** under Data Step #3 of the Intake stage.

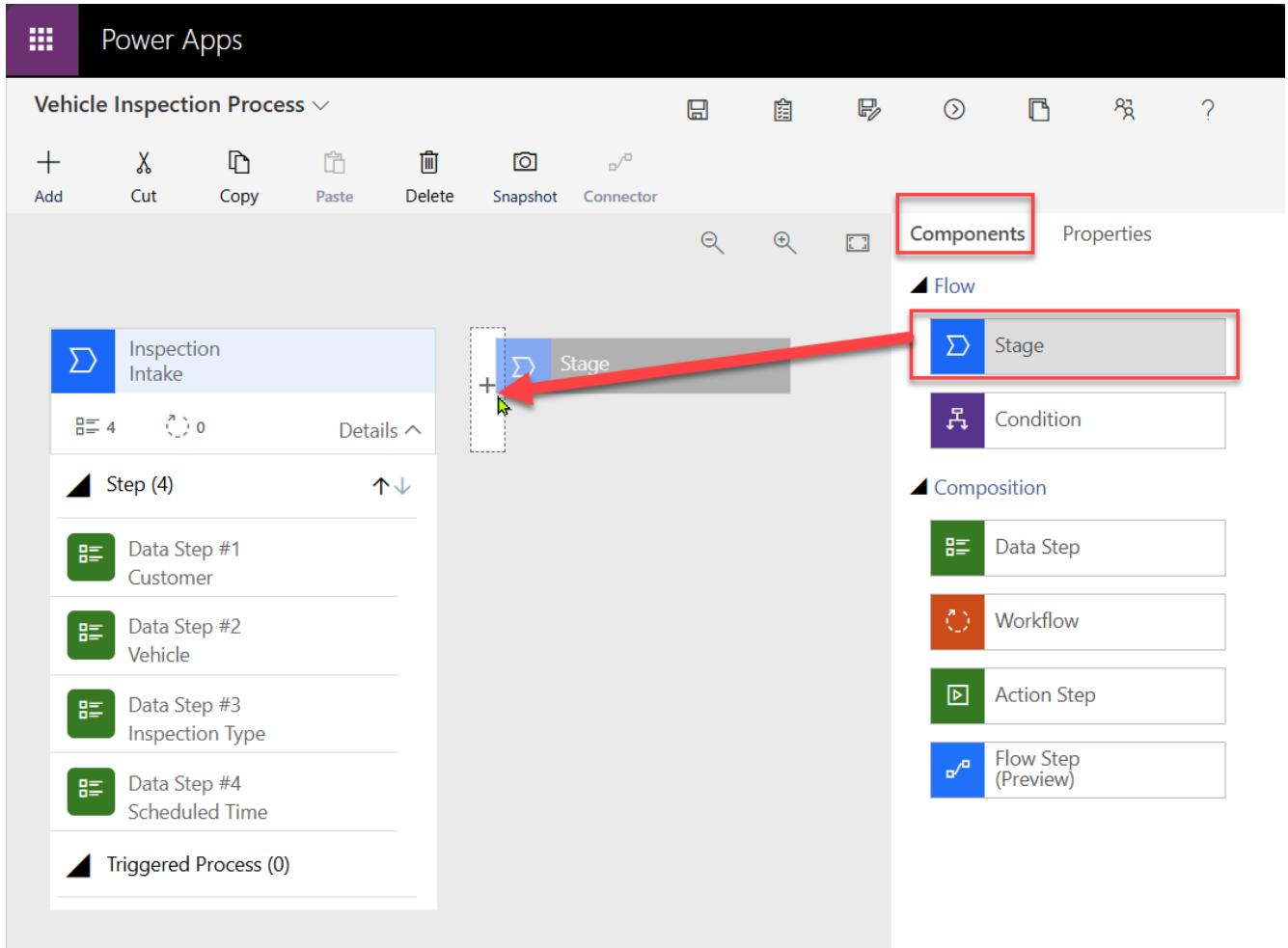
83. Enter Data Step #3 properties:

- Step Name: **Scheduled Time**
- Data Field: **Scheduled Time**
- Required: **Check**
- Sequence: 4
- Click **Apply**

84. You should now have 4 data steps configured for the Intake stage.



85. Select the **Components** tab and drag a new **Stage** next to the Inspection stage.

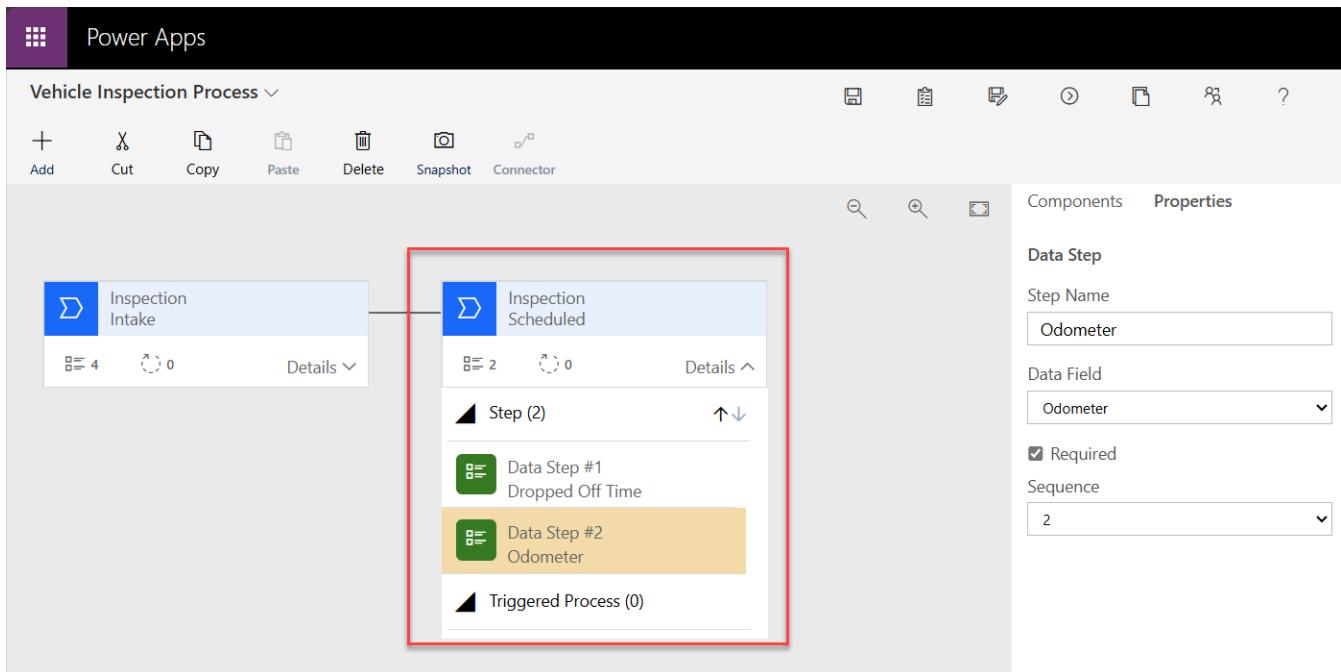


86. Enter the following details:

- Display Name: **Scheduled**
- Entity: **Inspection**

87. Add the following **required** data steps:

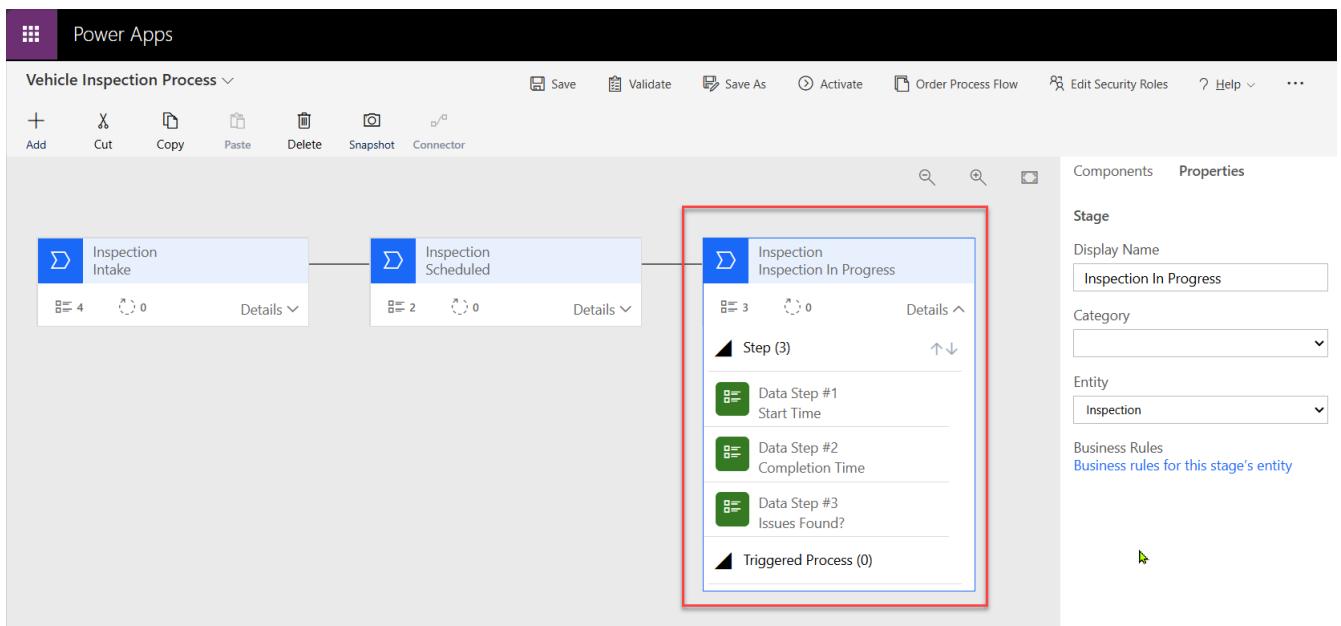
- Dropped Off Time
- Odometer



88. Add another stage called **Inspection In Progress**. Set the entity to **Inspection**.

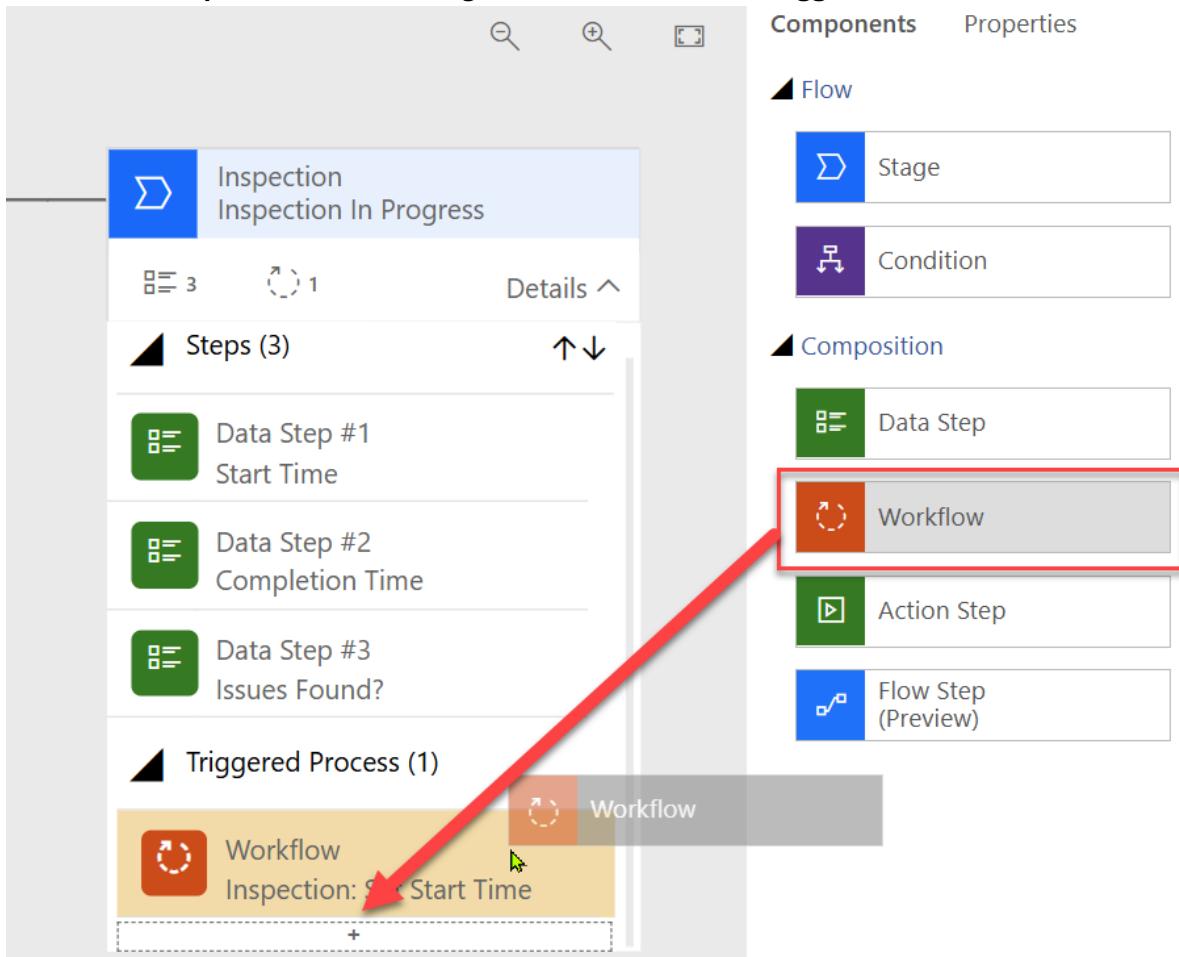
89. Add the following **required** data steps:

- Start Time
- Completion Time
- Issues Found?



Next we will configure the Inspection In Progress stage to trigger a workflow to set the Start Time to the current time when the stage is entered.

90. Select the **Components** tab and drag a **Workflow** under the **Triggered Processes** section of the stage.



91. Select the Workflow to set its properties.

- Trigger: **Stage Entry**.
- Workflows: Click the **magnifying glass icon** and select the **Inspection: Set Start Time workflow**

Workflow

Display Name: Inspection: Set Start Time

Entity: Inspection

Trigger: Stage Entry

Workflows: + New

Inspection: Set Start Time (Selected)

Inspection: Set Start Time (Search Result)

Look Up More Records

1 result

Workflow: Inspection: Set Start Time

92. Add a final stage called **Inspection Completed**. Set the entity to **Inspection**.

93. Add the following **required** data steps:

- Pick Up Time

Data Step

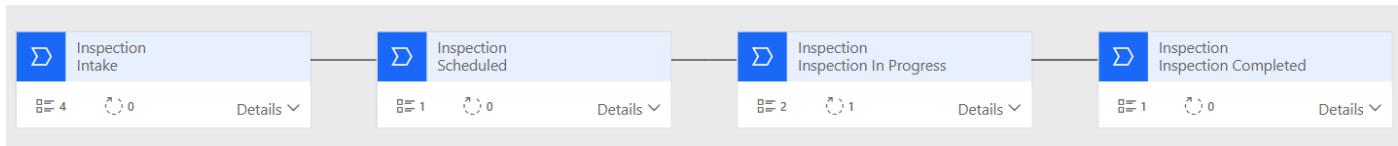
Step Name: Pick Up Time

Data Field: Pick Up Time

Required:

Sequence: 1

94. The business process flow (BPF) should now look like below:



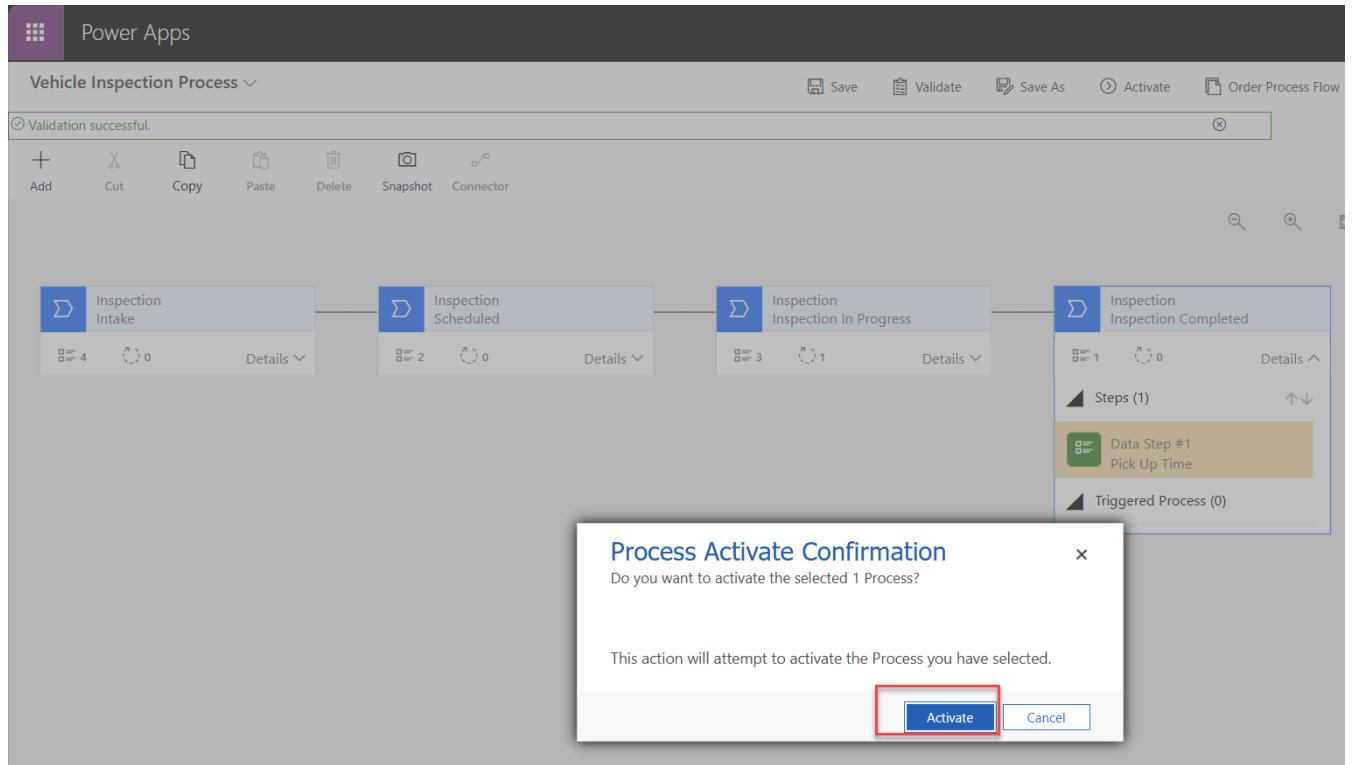
95. **Validate** the BPF. You must fix any errors before you can save.

96. **Save** the BPF.

97. **Activate** the BPF

98. The Process Activate Confirmation dialog should appear, click **Activate**.

Note: Sometimes the designer is finnicky and you may need to click Activate again for the dialog to appear

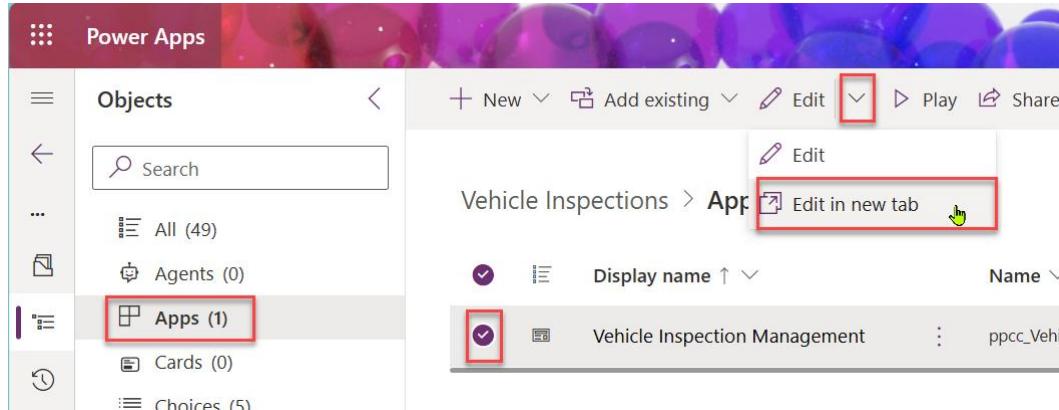


99.

100. Next we need to add the BPF to the model-driven app.

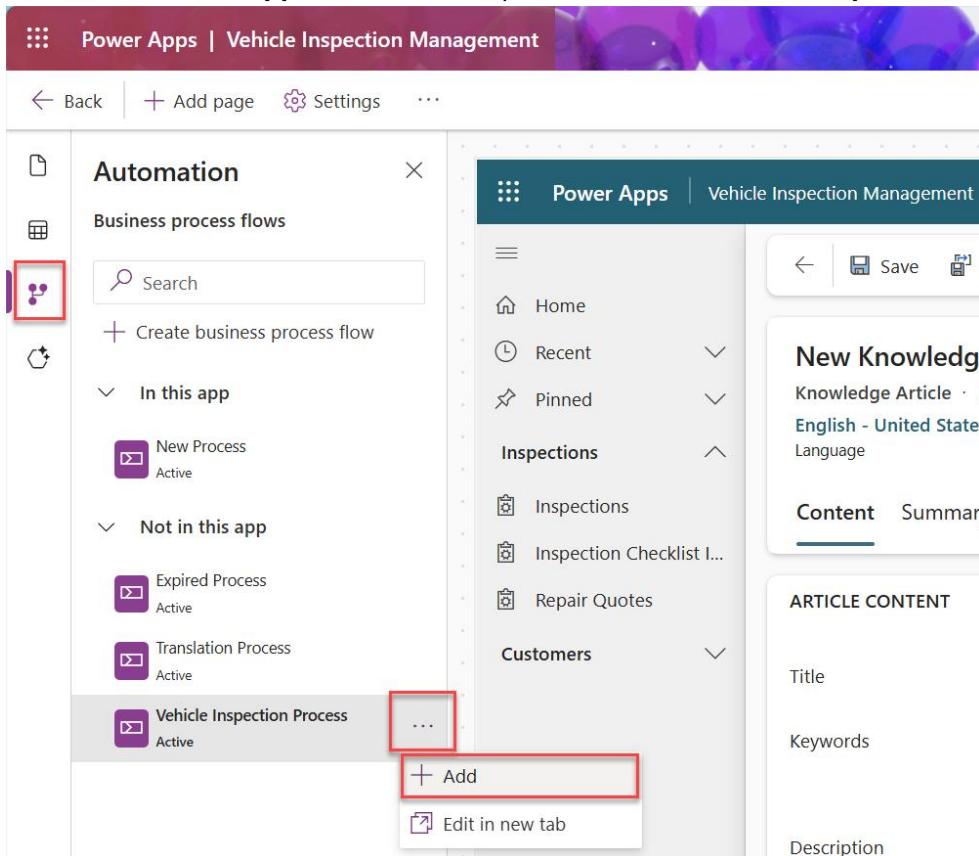
101. In the maker portal. Navigate to **Apps** under the solution object tree.

102. Select the **Vehicle Inspection Management** app to **Edit it in new tab**.

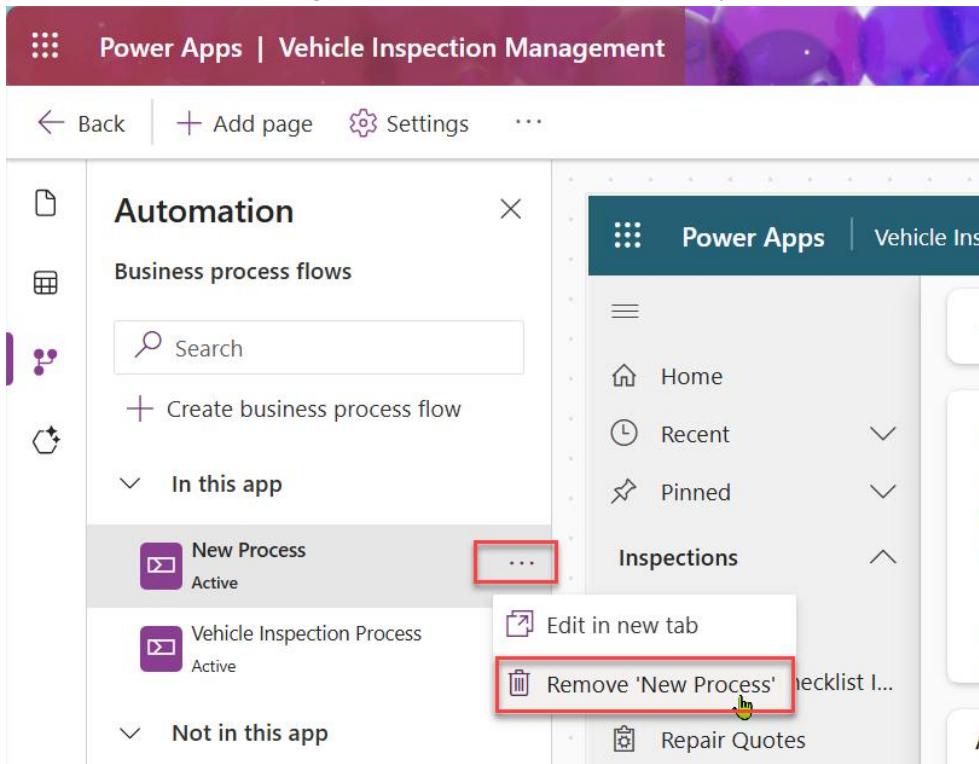


103. On the left hand navigation, select the **Automation** icon.

104. Under **Not in this app**, select the 3 ellipses next to the **Vehicle Inspection Process** and click **+Add**



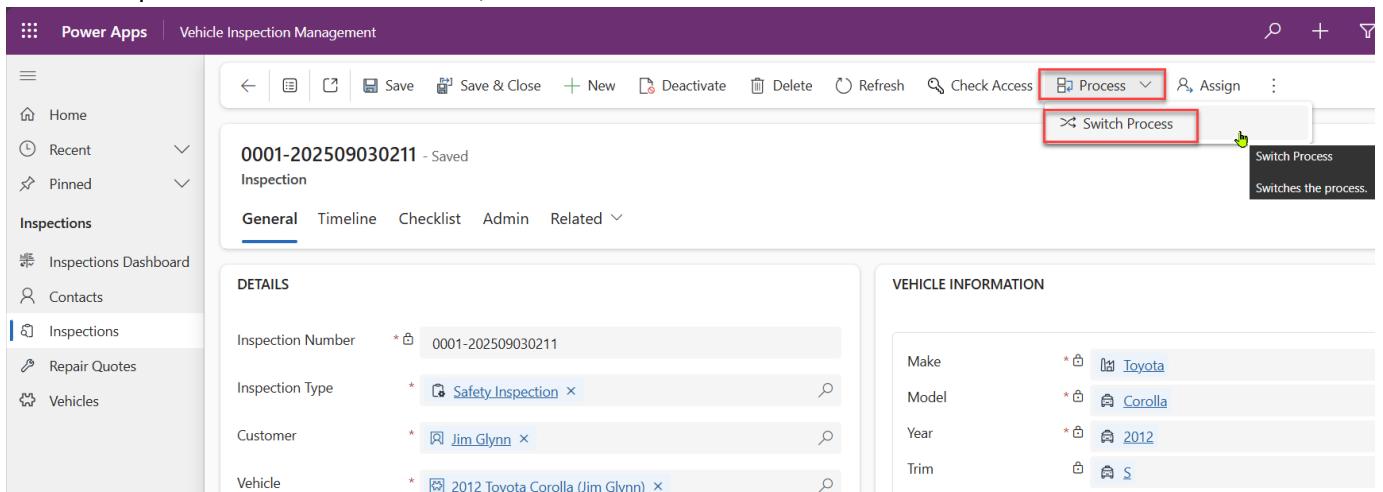
105. Under In this app, select the 3 ellipses next to the **New Process** and remove it (this is an out of box process for the Knowledge Article table that we don't use).



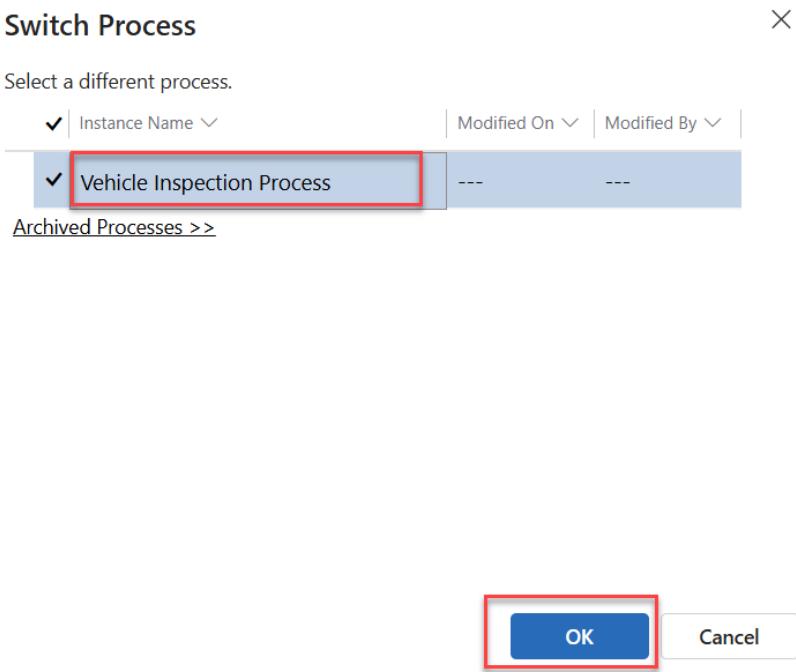
106. Click the **Save and Publish** button.

107. Click the **Play** button to open the app.

108. Navigate to **Inspections**, and open an existing record.
109. Business Process Flows are not automatically added to existing records. We must switch the process.
110. On the top of the form command bar, select **Process > Switch Process**



111. Select the **Vehicle Inspection Process**, then click **OK**.



112. The business process flow now appears below the record title

0001-202509030211 - Saved
Inspection

Vehicle Inspection Process
Active for less than one minute

Intake (< 1 Min) Scheduled Inspection In Progress Inspection Completed

General Timeline Checklist Admin Related Form assist

DETAILS

Inspection Number: 0001-202509030211

Inspection Type: Safety Inspection

Customer: Jim Glynn

Vehicle: 2012 Toyota Corolla (Jim Glynn)

VEHICLE INFORMATION

Make: Toyota

Model: Corolla

Year: 2012

Trim: S

Color:

113. Select the **Intake** stage to view the details. Enter in required data steps, then click **Next Stage**.

0001-202509030211 - Saved
Inspection

Vehicle Inspection Process
Active for 5 minutes

Intake (5 Min) Scheduled

General Timeline Checklist Admin

DETAILS

Inspection Number: 0001-202509030211

Inspection Type: Safety Inspection

Customer: Jim Glynn

Vehicle: 2012 Toyota Corolla (Jim Glynn)

Scheduled Time: 10/27/2025, 8:00 AM

VEHICL

Customer: Jim Glynn

Vehicle: 2012 Toy...

Inspection Type: Safety Ins...

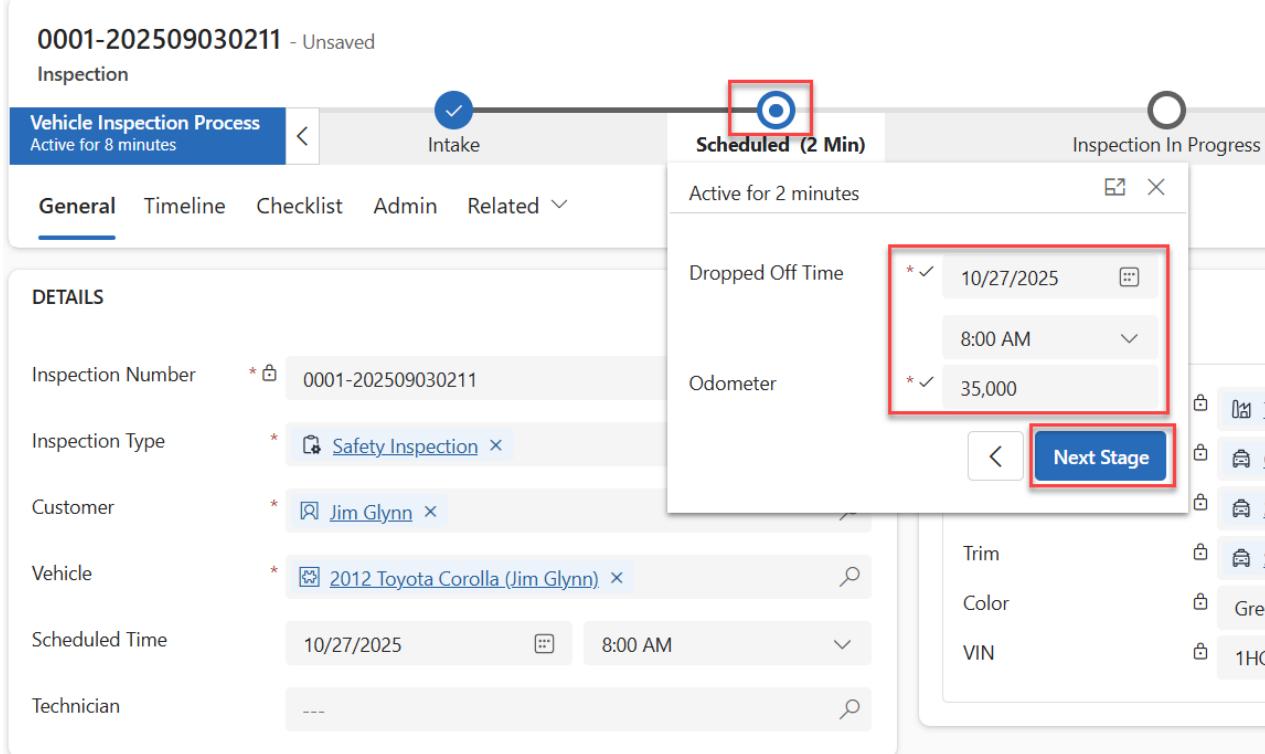
Scheduled Time: 10/27/2025, 8:00 AM

Next Stage

114. Select the **Scheduled** stage and enter the following:

- Dropped Off Time:
 - 10/27/2025
 - 8:00 AM
- Odometer: 35000

115. Select **Next Stage**.



116. You can move back and forth between stages if required.

We'll now use branching to add Recommend Repair and Repairing Vehicle stages, then add conditions so the stages only show when criteria are met:

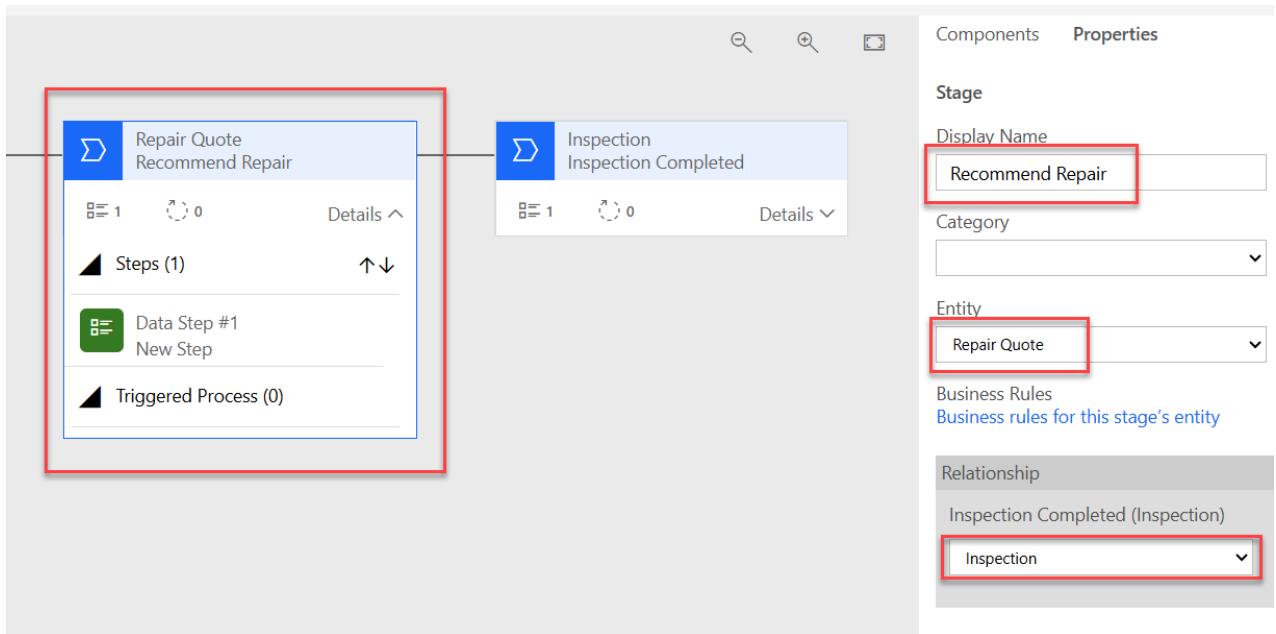
- **Recommend Repair** stage shows if the **Issues Found** column is set to **Yes**
- **Repairing Vehicle** stage shows if the **Customer Decision** column from the related **Repair Quote** record is set to **Approved**.

117. Go back to the business process flow designer.

118. Add a stage called **Recommend Repair** before the Inspection Completed stage.

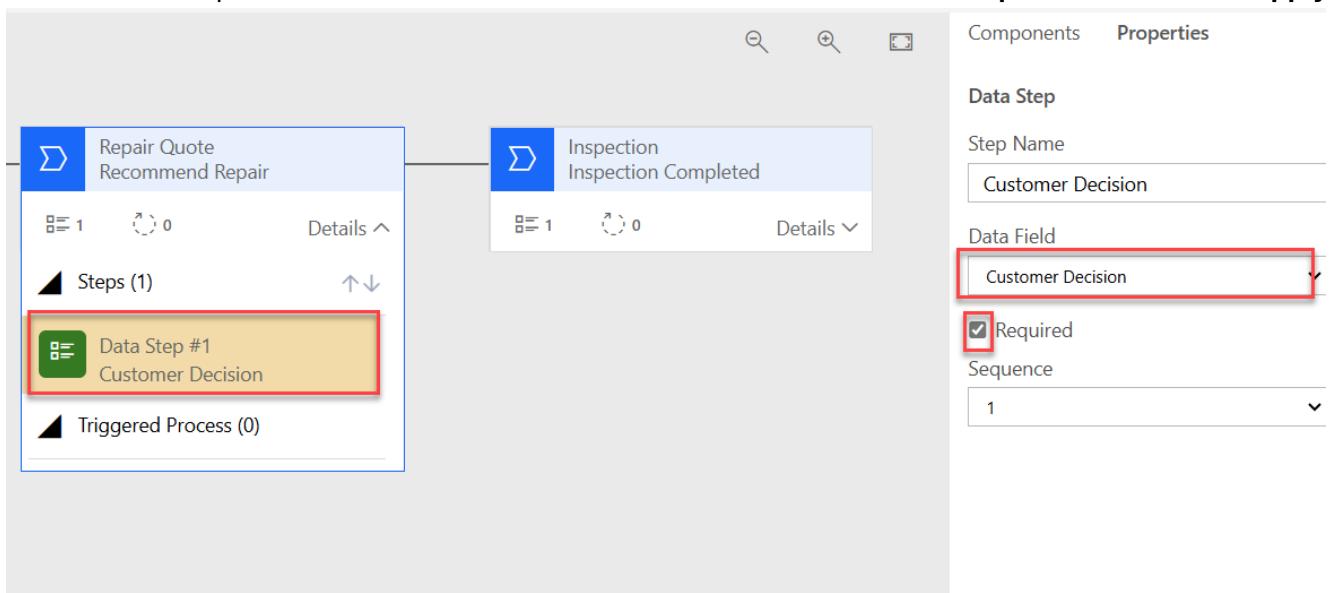
119. Set the entity to **Repair Quote**.

120. Select the relationship dropdown and choose **Inspection**.



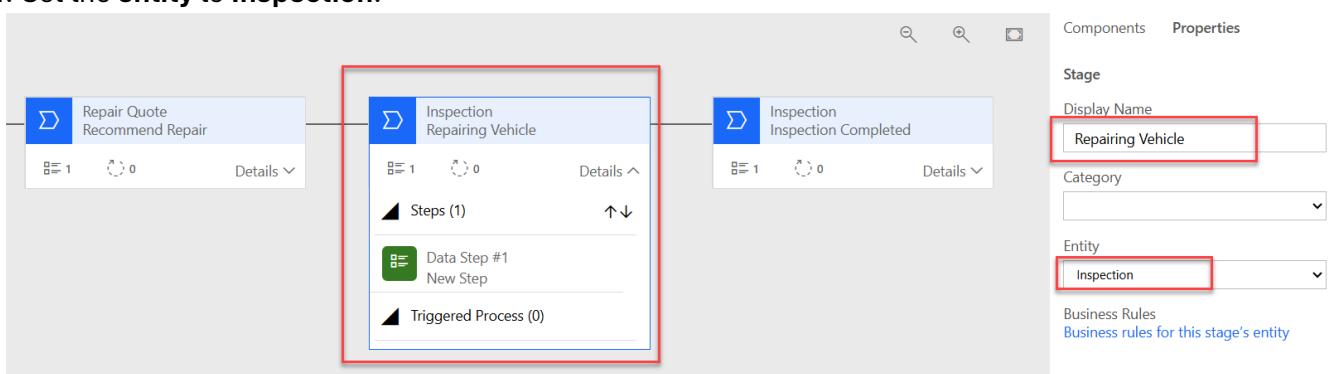
121. Click **Apply**.

122. Select Data Step #1 and choose the **Customer Decision** column. Tick the **Required** box and click **Apply**.

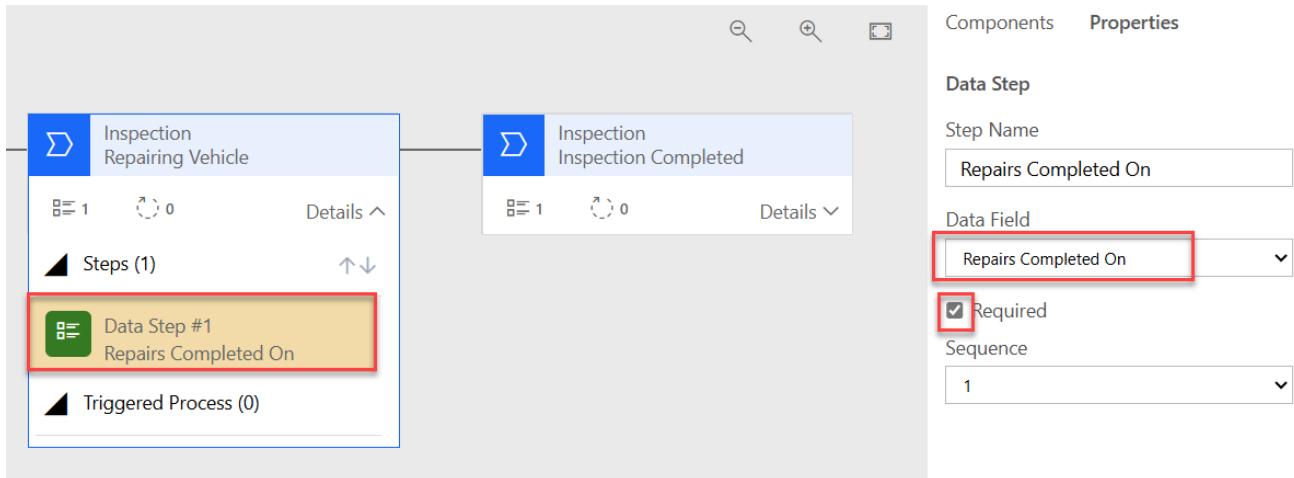


123. Add another stage called **Repairing Vehicle** before the Inspection Completed stage

124. Set the **entity** to **Inspection**.



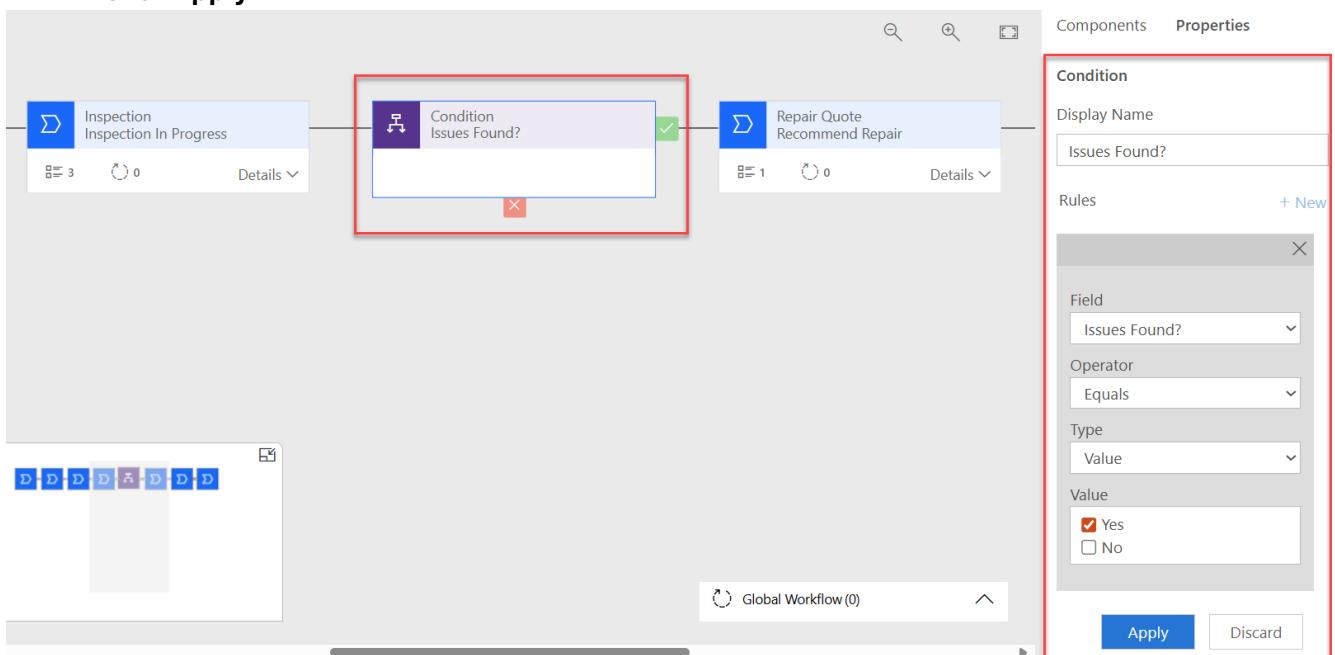
125. Select Data Step #1 and choose the **Repairs Completed On** column. Tick the **Required** box and click **Apply**.



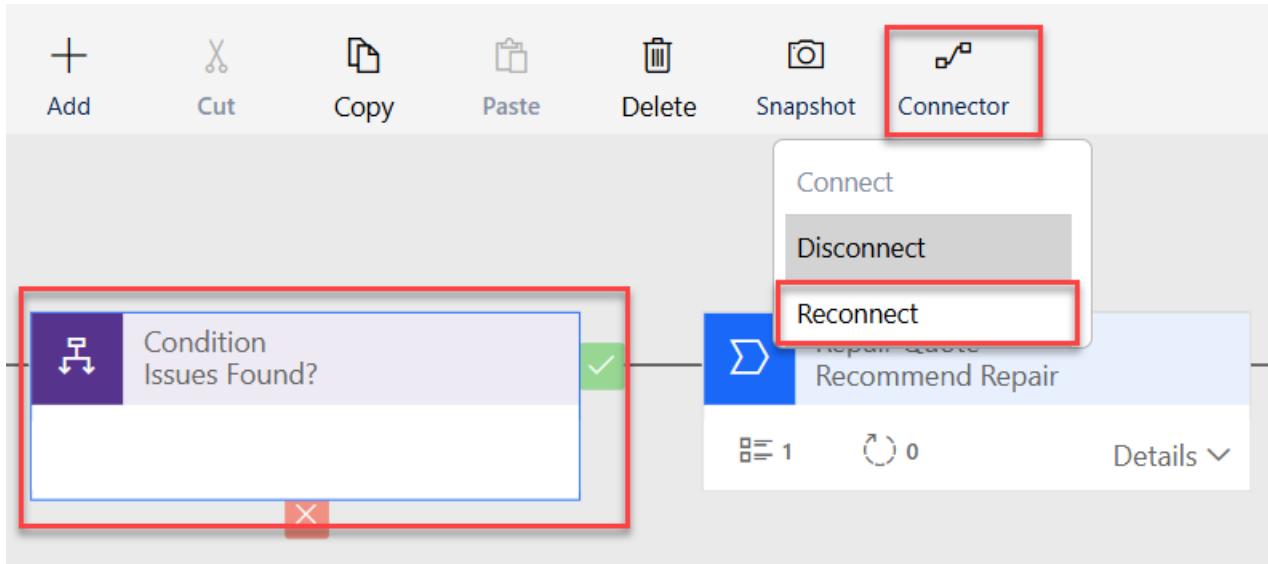
126. Select the **Components** tab in the side panel and drag a **condition** before the **Recommend Repair** stage.

127. Configure the condition:

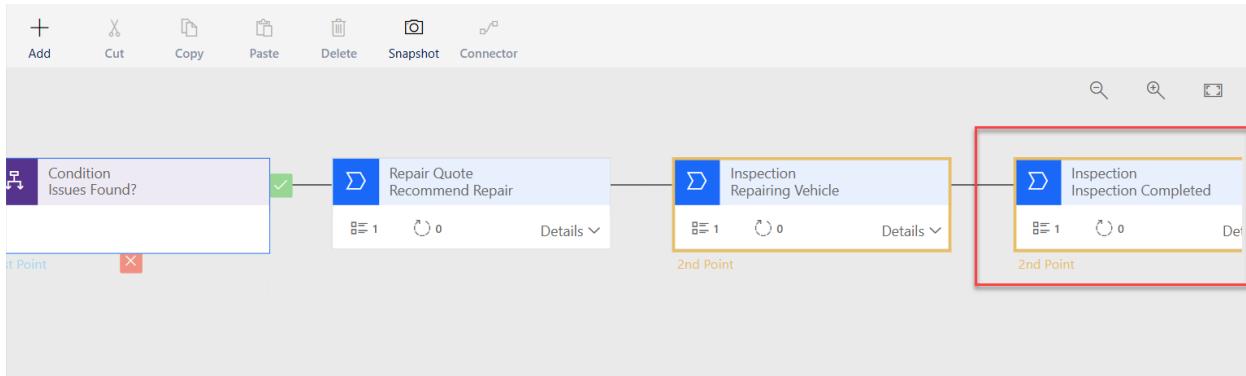
- Display Name: **Issues Found?**
- Rule:
 - i. Field: **Issues Found?**
 - ii. Operator: **Equals**
 - iii. Type: **Value**
 - iv. Value: **Yes**
- Click **Apply**



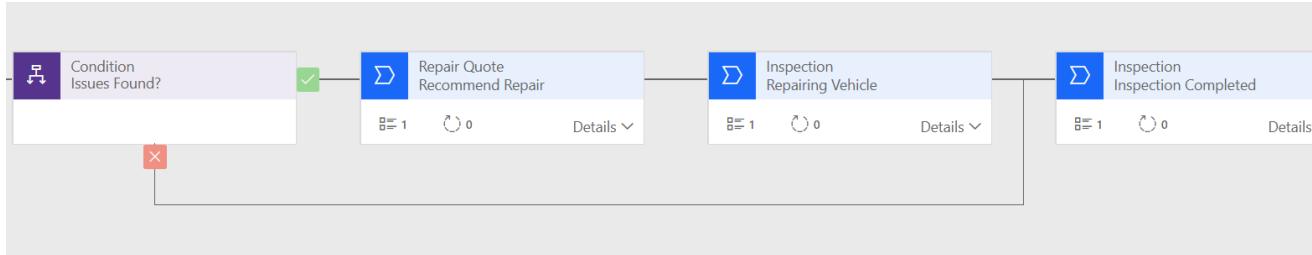
128. With the Condition still selected, above the canvas click **Connector > Reconnect**



129. The orange highlighted stages annotated with “2nd Point” are available to connect to. Select the **Inspection Completed** stage.

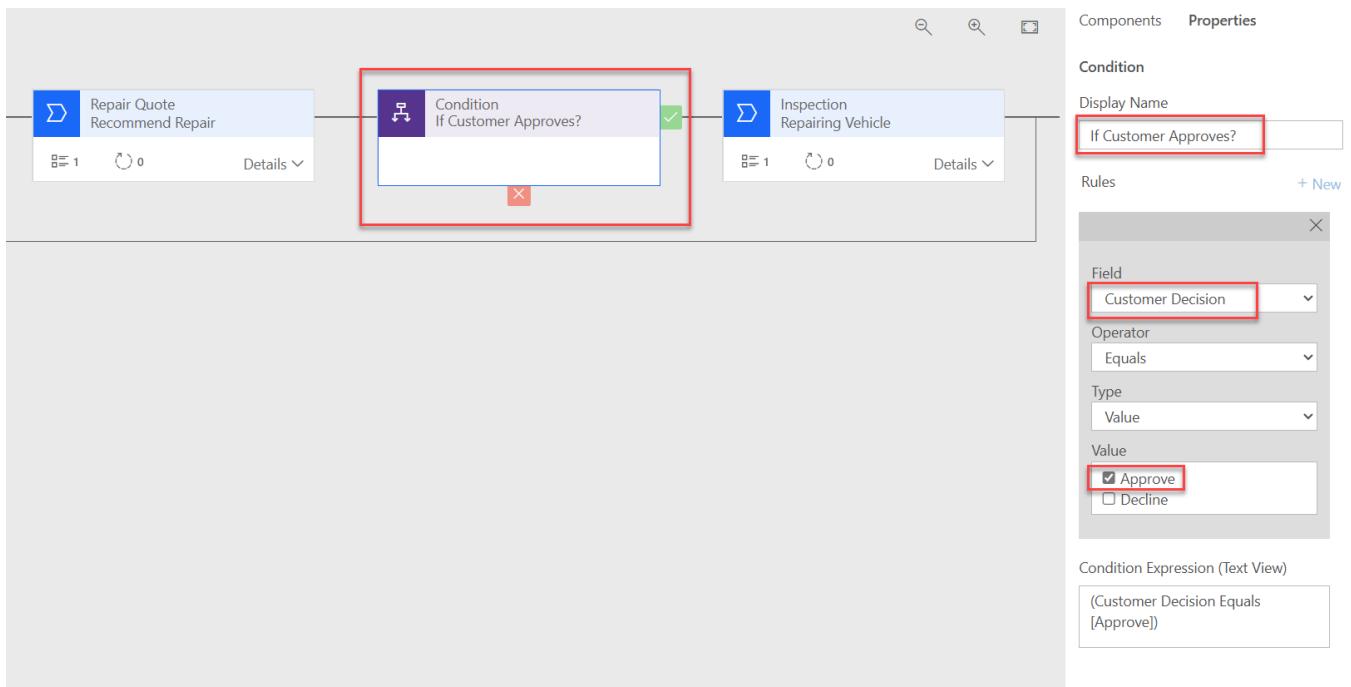


130. You will now see the connector line from the false condition to the Inspection Completed stage, bypassing the Recommend Repair and Repairing Vehicle stages.

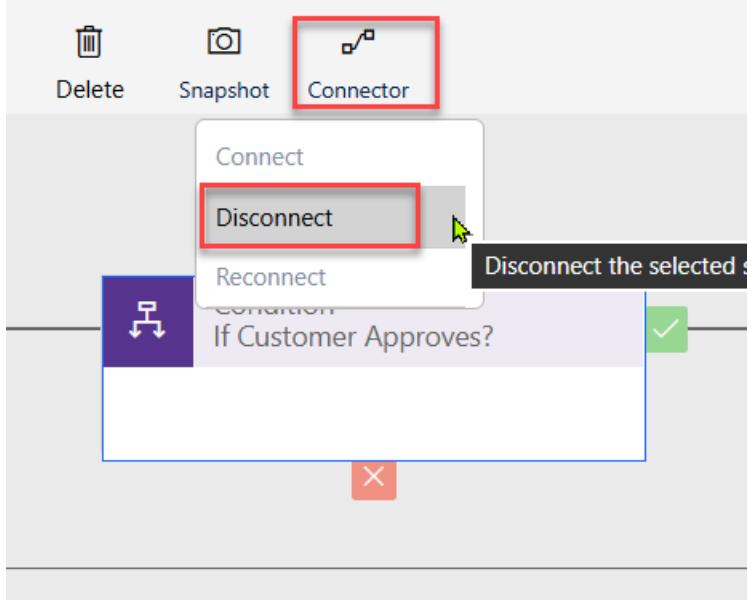


131. Select the **Components** tab and add one more **condition**, this time between the Recommend Repair and Repairing Vehicle stages.

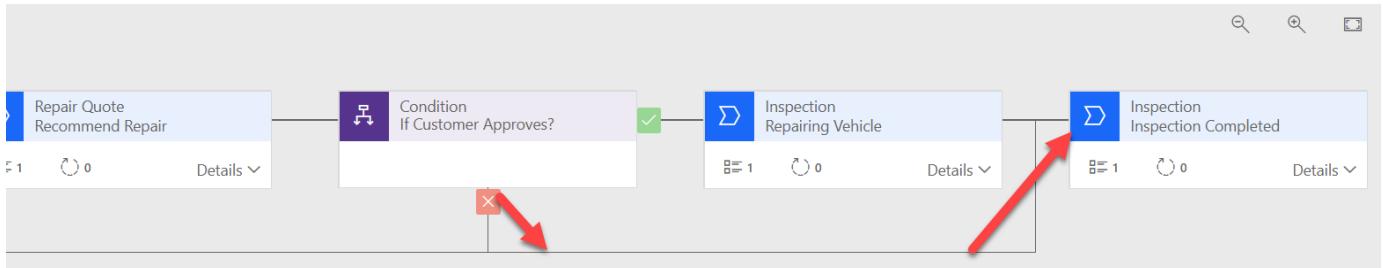
- Display Name: **If Customer Approves**
- Field: **Customer Decision**
- Operator: **Equals**
- Type: **Value**
- Value: **Approve**
- Click **Apply**



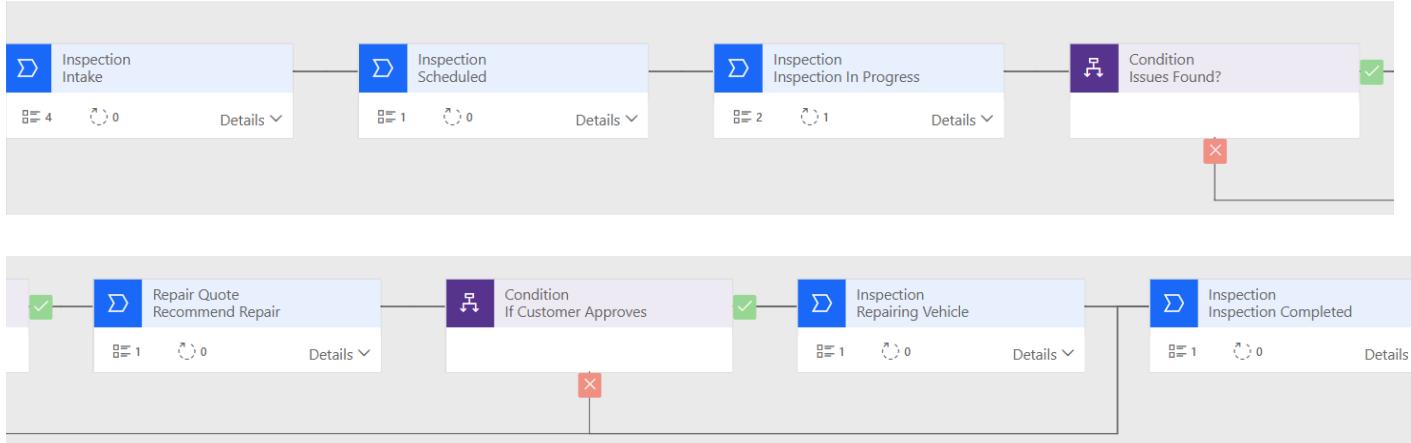
132. With the Condition still selected, above the canvas click **Connector > Disconnect**.



133. The false branch will now connect to the Inspection Completed stage.



134. The completed business process flow (BPF) should look like below:



135. **Validate** the BPF.

136. **Update** the BPF.

137. Switch back to the **Vehicle Inspection Management** app.

138. Select the Inspection In Progress stage on the Inspection record and set the Completion Time:

a. Date: **10/27/2025**

b. Time: **12:00 PM**

139. You will notice the Issues Found? field is locked for editing.

The screenshot shows the 'Inspection In Progress (2 Hrs)' stage of a vehicle inspection. The 'Completion Time' field is set to '10/27/2025 12:00 PM'. A yellow arrow points to the 'Issues Found?' field, which is labeled 'Read-only'.

In the next section, we will use a Power Automate workflow to update the Issues Found column so that we can test the rest of the business process flow.

Use Power Automate Flows with Dataverse

We will use Power Automate to trigger a workflow when an **Inspection Checklist Item** record is created or updated and where the Condition is set to “Requires Immediate Attention”.

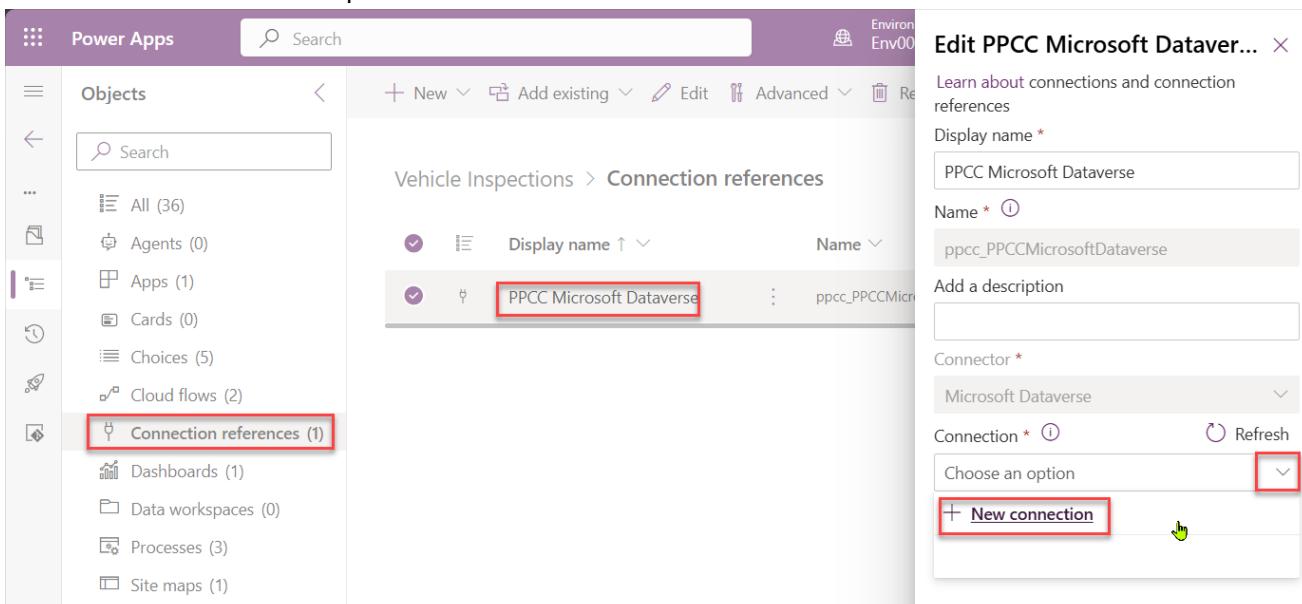
It will perform the following actions:

- Retrieve the related Inspection
- Retrieve Repair Quotes (if any) related to the Inspection.
- If both of the following criteria are true:
 - **Issues Found** column on the Inspection is currently set to **No**
 - If there are no child Repair Quotes associated with the Inspection
- Then:
 - Create a Repair Quote record and associate it with the Inspection
 - Set the Issues Found column on the inspection to **Yes**

The workflow will satisfy a condition for the business process flow to branch to the Recommend Repair stage.

A connection must be created for any connection references used by flows.

140. Navigate to **Solutions** and open the **Vehicle Inspections** solution.
141. Select **Connection references** under the object tree.
142. Click the PPCC Microsoft Dataverse to open the properties panel of the connection reference.
143. Select the Connection dropdown and click **+New connection**



144. A new browser tab is opened to the Connections area of the maker portal.

145. Enter **Dataverse** in the top right **search bar**.

146. Select the **Microsoft Dataverse** connector with the green icon from the list.

The screenshot shows the 'Connections > New connection' page. At the top right, there is a search bar with the text 'Dataverse' and a red box highlighting it. Below the search bar are buttons for 'Back', 'New connections experience (preview)', 'Refresh', and a close button. The main area is a table with columns: Name, Publisher, Type, and Actions. There are two entries for 'Microsoft Dataverse': one with a grey camera icon and another with a green person icon. The entry with the green person icon is highlighted with a red box. The 'Actions' column contains a '+' sign for both rows.

147. Select **Oauth** for Authentication Type, then **Create**.

The screenshot shows the 'Microsoft Dataverse' connector creation page. At the top center is the Microsoft logo. Below it is the title 'Microsoft Dataverse'. A subtext states 'Provides access to Microsoft Dataverse actions and triggers for Power Platform environments.' Underneath, there is a dropdown menu labeled 'Authentication Type' with 'Oauth' selected, which is highlighted with a red box. At the bottom right are 'Create' and 'Cancel' buttons, with 'Create' also highlighted with a red box.

148. Select your assigned user login which should already be signed in.

149. Click **Allow access** in the confirmation required pop up.

150. You should see the newly created connection in the list.

The screenshot shows the 'Connections' page. At the top left is a 'Connections' button. Below it is a table with columns: Name, Last modified, and Status. There is one entry: 'User001@ppcc25workshop06.onmicrosoft.com' (Microsoft Dataverse), last modified '0 seconds ago', and status 'Connected' with a green checkmark. A red box highlights the 'Connected' status.

151. Return to the browser tab with the Connection References panel.

152. Click refresh and select the created connection from the dropdown.

Edit PPCC Microsoft Dataver... X

Learn about connections and connection references

Display name *

PPCC Microsoft Dataverse

Name * (i)

ppcc_PPCCMicrosoftDataverse

Add a description

Connector *

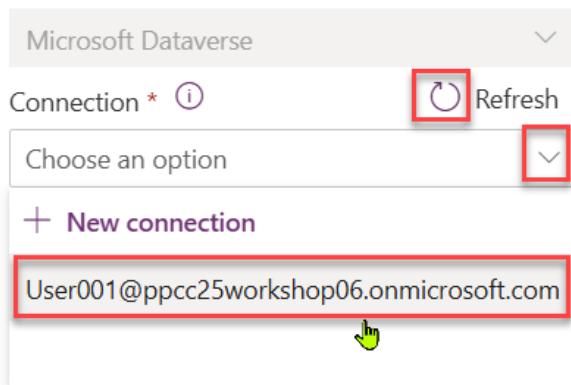
Microsoft Dataverse

Connection * (i) ↻ Refresh

Choose an option ▼

+ New connection

User001@ppcc25workshop06.onmicrosoft.com User001@ppcc25workshop06.onmicrosoft.com

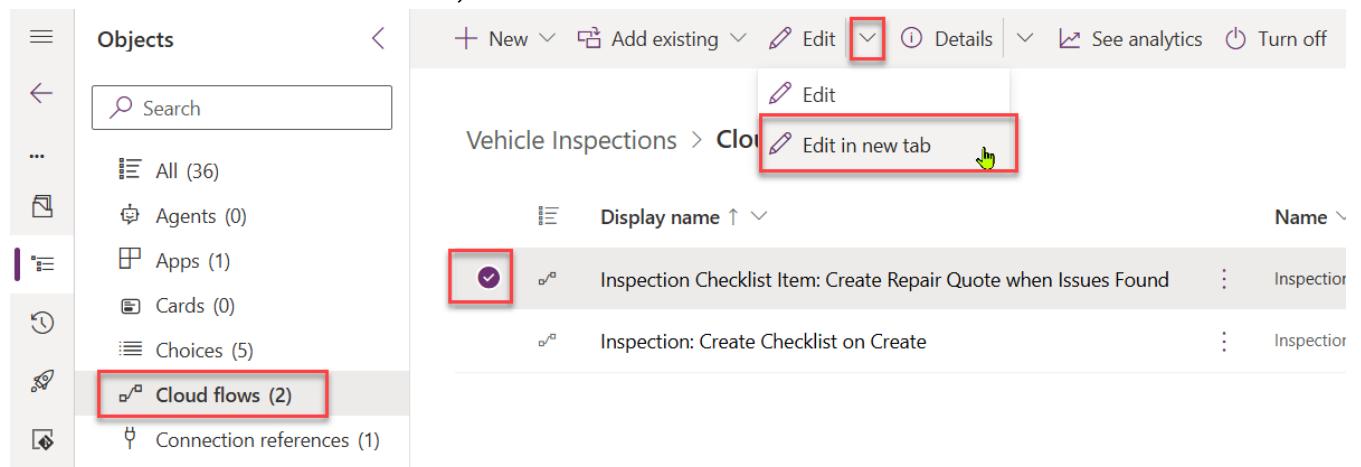


We will inspect a workflow that has been created and explore the Dataverse trigger and actions used.

153. Select **Cloud flows** under the solution object tree.

154. Select the row for **Inspection Checklist Item: Create Repair Quote when Issues Found**

155. Select the down arrow next to Edit, then select **Edit in new tab**



Display name	Trigger	Action
Inspection Checklist Item: Create Repair Quote when Issues Found	Vehicle Inspections > Close inspection	Inspection: Create Repair Quote
Inspection: Create Checklist on Create	Vehicle Inspections > Create inspection	Inspection: Create Checklist

156. The flow uses the Dataverse trigger **When a row is added, modified or deleted (Microsoft Dataverse)**

Back Inspection Checklist Item: Create Repair Quote when Issues Found • Published

Inspection-Checklist-Item-Added-Or-Modified

Parameters Settings Code view About

Change type * Added or Modified

Table name * Inspection Checklist Items

Scope * Organization

Select columns ppcc_condition

Filter rows ppcc_condition eq 423040002

Advanced parameters Showing 0 of 2 [Show all](#) [Clear all](#)

Connected to PPCC Microsoft Dataverse. [Change connection reference](#)

157. Select the trigger to view its properties:

- Change type: **Added or Modified**
- Table name: **Inspection Checklist Items**
- Scope: **Organization** (triggers for all records of that table regardless of the record owner)
- Select columns: **ppcc_condition** (requires a comma selected list of column logical names)
To retrieve the column logical name in the maker portal: **Tables > Inspection Checklist Item >**

Columns > Condition (select 3 ellipses) > Advanced > Tools > Copy logical name

The screenshot shows the 'Condition' column properties in the Power Apps maker portal. The 'Advanced' button in the context menu is highlighted with a red box. Other options in the menu include 'Edit', 'Remove', 'Show dependencies', 'See solution layers', 'Managed properties', 'Remove active customizations', 'Add to solution', 'Tools', 'Copy schema name', 'Copy logical name', 'API link to column definition', and 'API link to column data'. The 'Copy logical name' option is also highlighted with a red box.

Column Name	Type	Description	Unique Identifier	Single line of text	Yes	Yes
Checklist Item	ppcc_Checklistitem...	Lookup	No	Yes	Yes	
Completed On	ppcc_Completed...	Date and time	No	Yes	Yes	
Condition	ppcc_Condition	Choice	No	Yes	Yes	
Created By	ppcc_CreatedBy	Lookup	No	Yes	Yes	
Created By (Delegate)	ppcc_CreatedByDelegat...	Edit	Yes	Yes		
Created On	ppcc_CreatedOn	Show dependencies	Yes	Yes		
Import Sequence Nu...	ImportSequenceNu...	See solution layers	Yes	Yes		
Inspection	ppcc_InspectionId	Managed properties	Yes	Yes		
Inspection Area	ppcc_InspectionArea	Remove active customizations	Yes	Yes		
Inspection Checklist It...	ppcc_InspectionChecklis...	Add to solution	Yes	Yes		
Item Name	ppcc_ItemName	Tools	Yes	Yes		

- Filter rows: **ppcc_condition eq 423040002** (Condition = “Requires Immediate Attention” option)
Note: For choice columns, the integer value must be specified.

You can retrieve from the maker portal by navigating to **Choices > Checklist Item Conditions > Copy the Value for Requires Immediate Attention**

The screenshot shows the 'Choices' screen for 'Vehicle Inspections'. The 'Checklist Item Condition' choice is highlighted with a red box. The 'Display name' field is set to 'Checklist Item Condition'. The 'Choices' table shows three entries: 'Satisfactory' (Value: 423,040,000), 'May Require Further Attention' (Value: 423,040,001), and 'Requires Immediate Attention' (Value: 423,040,002). The 'Requires Immediate Attention' entry is also highlighted with a red box.

Name	Label *	Value *
ppcc_Satisfactory	Satisfactory	423,040,000
ppcc_MayRequireFurtherAttention	May Require Further Attention	423,040,001
ppcc_RequiresImmediateAttention	Requires Immediate Attention	423,040,002

158. The **Get-Inspection** step uses the **Get a Row** action.

Get-Inspection

Parameters Settings Code view Testing About

Table name *

Inspections

Row ID *

Inspection (Value) X

Advanced parameters

Showing 4 of 4

Return Full Metadata

The header parameter to customer to opt-in returning full odata metadata in respo... X

Select columns

ppcc_inspectionid,ppcc_inspectionnumber,ppcc_issuesfound X

Expand Query

ppcc_repairquote_InspectionId_ppcc_inspection(\$select=ppcc_name) X

Partition Id

An option to specify the partitionId while retrieving data for NoSQL tables X

Connected to PPCC Microsoft Dataverse. [Change connection reference](#)

- Table name: **Inspections**
- Row ID: the Unique Identifier (GUID) of the row to select.
Inspection (value) is selected from the trigger step columns.
- Columns: The logical names of the columns we need to use (Inspection Id, Inspection Number and Issues Found?):
ppcc_inspectionid,ppcc_inspectionnumber,ppcc_issuesfound
- Expand Query: Retrieve a related lookup record (N:1 relationship), or child records (1:N relationship) using **OData \$expand** query syntax.

When retrieving **child** records, you need to specify the **relationship name**, then in parenthesis a \$select= with comma separated column logical names:

ppcc_repairquote_InspectionId_ppcc_inspection(\$select=ppcc_name)

The screenshot shows the Power Apps 'Relationships' screen. On the left, there's a navigation pane with 'Objects' selected. Under 'Repair Quote', 'Relationships' is highlighted with a red box, and 'Inspection' is also highlighted with a red box. The main area displays the relationship configuration:

- Current (Many)**: Repair Quote
- Related (One)**: Inspection
- Lookup column display name**: Inspection
- Lookup column name**: ppcc_inspectionId
- Lookup column requirement**: Business Required
- Searchable**: Checked
- Relationship name**: ppcc_repairquote_InspectionId_ppcc_inspection
- Lookup column description**: (empty)

159. The **If-No-Existing-Quotes-And-No-Issues-Found** step uses a **Condition** action is used with the following conditions:

The screenshot shows the Power Automate designer with the following steps:

- Inspection-Checklist-Item-Added-Or-Modified**
- Get-Inspection**
- If-No-Existing-Quotes-And-No-Issues-Found** (highlighted with a red box)

The condition expression for the third step is:

```
length(body('Get-Inspection')?['ppcc_repairquote_InspectionId_ppcc_inspection'])
```

The condition expression parameters are:

- AND**
- Left**: Click to view the expression (fx) `length(...)`
- Operator**: **is equal to**
- Right**: **0**
- Left**: Click to view the expression (fx) `body/ppcc_issues... X`
- Operator**: **is equal to**
- Right**: `false`

- **Left**: Click to view the expression (fx) `length(body('Get-Inspection')?['ppcc_repairquote_InspectionId_ppcc_inspection'])`
- **Operator**: **is equal to**
- **Right**: **0**

160. Another row into the condition step:

- **Left**: **Issues Found?** column from the **Get-Inspection** step.
- **Operator**: **is equal to**
- **Right**: Click to view the expression (fx) `false`

161. In the **True** branch, the Create-Repair-Quote uses the Dataverse **Add a new row** action:

The screenshot shows the Power Automate Designer interface. On the left, under 'Parameters', the 'Table name' field is set to 'Repair Quotes' and the 'Inspection (Inspections)' field is set to 'ppcc_inspections(Inspection)'. A red box highlights these two fields. Below this, there are several other parameter fields: 'Name' (set to 'Quote for Inspection: Inspection Num...'), 'Advanced parameters' (showing 15 of 15), and various deprecated metadata fields like 'Stage Id' and 'Traversed Path'. On the right, the workflow logic is displayed. It starts with an 'If-Item-Added-Or-Modified' condition (triggered by 'Inspection-Checklist-Item-Added-Or-Modified'). This leads to a 'Get-Inspection' action, followed by a decision block 'If-No-Existing-Quotes-And-No-Issues-Found'. The 'True' branch of this decision leads to a 'Create-Repair-Quote' action (highlighted with a red box), which then flows into an 'Update-Inspection' action. The 'False' branch leads to '0 Actions'. The entire workflow is contained within a single step.

- Table name: **Repair Quotes**

162. The **Inspection (Inspections)** column is a **lookup**, which must be set using the OData format:

<TableSetName><open parenthesis><UniquelIdentifier of record><close parenthesis>

e.g. **ppcc_inspections(4cd38ecc-da88-f011-b4cc-000d3af37bb2)**

163. You can retrieve the **set name** for the **Inspection** table via the maker portal as shown below

The screenshot shows the Power Apps Objects screen. On the left, there's a sidebar with various icons. The main area shows a list of objects: Contact, Inspection (which is selected and highlighted with a red box), Columns, Relationships, Keys, Forms, Views, Charts, and Dashboards. To the right, under 'Vehicle Inspections > Tables > Inspection', are 'Table properties' and a context menu. The context menu has several options: Properties (highlighted with a red box), Copy set name (also highlighted with a red box and has a green mouse cursor over it), Copy schema name, Copy logical name, API link to table definition, and API link to table data.

164. Add the **open/close parenthesis**

165. Click in between the parenthesis to bring up the output selector and search for Inspection from the **Inspection (value)** from the trigger

The screenshot shows the Power Automate 'Create-Repair-Quote' flow editor. In the main area, there's a 'Parameters' tab with fields for 'Table name *' (set to 'Repair Quotes') and 'Inspection (Inspections) *' (set to 'ppcc_inspections(Inspection)'). A red arrow points from this 'Inspection' field to the 'Get-Inspection' step in the right pane. The 'Get-Inspection' step has a 'Search' bar and a list of parameters: 'Inspection Number', 'OData Id', 'OData row id', 'Inspection' (with a tooltip 'Unique identifier for entity instances'), and 'Issues Found?'. The 'Issues Found?' field is described as 'Set to Yes if issues found.'

166. For the **Name** column, enter the text “**Quote for Inspection:**“ and select the **Inspection number** from the previous Get a row step.

The screenshot shows the 'Create-Repair-Quote' flow in Power Automate. In the 'Parameters' section, there are fields for 'Table name *' (set to 'Repair Quotes') and 'Name *' (set to 'Quote for Inspection: body/ppcc_insp...'). Below these, an 'Advanced parameters' section shows 'Showing 0 of 15' items. To the right, the 'Get-Inspection' step is expanded, showing its configuration. The 'Inspection Number' field is highlighted with a red box and a yellow cursor icon. Other visible fields include 'OData Id' and 'Issues Found?'. At the bottom, there are buttons for 'When a row is added, modified or deleted' and 'See more (48)'.

Quote for Inspection: @{{outputs('Get-Inspection')?['body/ppcc_inspectionnumber']}}

167. The **Update Inspection** step uses the Dataverse **Update a row** action:

- Table name: **Inspections**

The screenshot shows the 'Update-Inspection' workflow configuration on the left and its runtime view on the right.

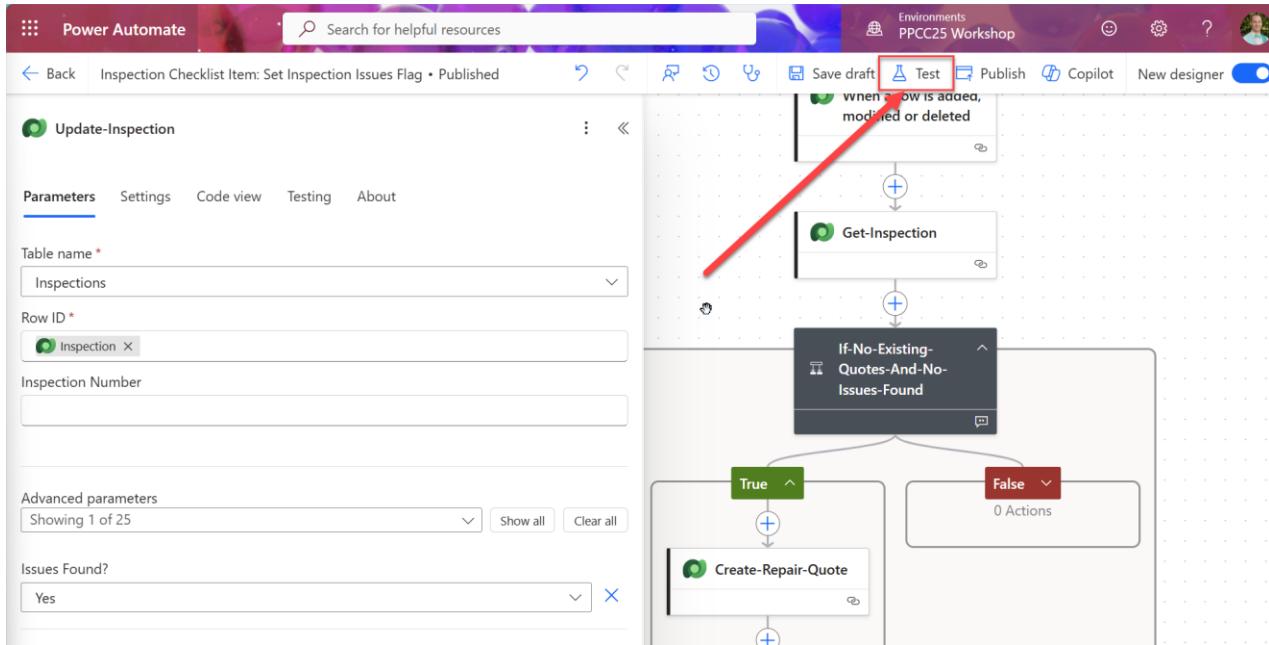
Workflow Configuration (Left):

- Parameters:**
 - Table name *: Inspections
 - Row ID *: Inspection
 - Customer (Accounts)
 - Customer (Contacts)
 - Inspection Number
 - Type of Inspection (Inspection Types)
 - Vehicle (Vehicles)
 - Advanced parameters: Showing 1 of 21
 - Issues Found?: Yes
- Issues Found? (dropdown):** Yes
- Runtime View (Right):** Shows the workflow steps in sequence:
 - If-No-Quote-Issues
 - True (Decision step)
 - Create-Repair-Quote
 - Update-Inspection (highlighted with a red box)

168. For **Row ID**, the **Inspection** (unique identifier) column from the **Get-Inspection** step is used.
169. **Issues Found?** is set to **Yes**

Now we will Test and validate that the workflow is doing what we expect.

170. Click **Test** in the top designer command bar corner.



171. Select **Manually**, then **Test**.

172. You will now need to **create or update** an **Inspection Checklist Item** record with the **Condition** column set to *Requires Immediate Attention*.

173. Open the **Vehicle Inspections Management** model-driven app.

174. Navigate to **Inspections** and open an existing record.

175. Click the **Related** tab, then select **Repair Quotes**.

The screenshot shows the 'Vehicle Inspections Management' model-driven app. At the top, there's a toolbar with buttons for Save, Save & Close, New, Deactivate, Delete, Refresh, Check Access, and Process. Below the toolbar, the record ID '0001-202509030211' is displayed, followed by the word 'Saved' and the word 'Inspection'. A progress bar at the top indicates the process is active for 13 hours. Below the progress bar, there are tabs for General, Timeline, Checklist, Admin, and Related. The 'Related' tab is highlighted with a red box. A dropdown menu is open under the 'Related' tab, showing options: 'Audit History', 'Repair Quotes' (which is highlighted with a red box), 'Related - Process Sessions', and 'Background Processes'. To the right of the dropdown, there are sections for 'DETAILS' and 'VEHICLE INFORMATION'. The 'DETAILS' section includes fields for Inspection Number, Inspection Type, Customer, and Vehicle. The 'VEHICLE INFORMATION' section includes fields for Make, Model, Year, and Trim.

176. There should be no associated records.

General Timeline Checklist Admin **Repair Quotes** Related ▾

Show Chart + New Repair Quote Refresh Flow Run Report Excel

Repair Quote Associated View ▾

Name ↑ ▾

We didn't find anything to show here

177. Click the **Checklist** tab.

178. In the checklist editable sub-grid, click into the **Condition** cell of one of the rows, and change the value to **Requires Immediate Attention**

General Timeline **Checklist** Admin Repair Quotes Related Form

Checklist

Odometer: 35,000 Odometer Unit: Mile (m)

Checklist Count: 3 Issues Found?: No

Last updated: 10/13/2025 11:55 PM

<input type="checkbox"/> Name* ↑ ▾	Area Name* ↑ ▾	Condition ▾	Completed On ▾
<input checked="" type="checkbox"/> Anchored properly & 1 mat only in driver footwell	Safety	<input type="text"/> Enter or pick date	
<input type="checkbox"/> Battery Condition	Safety	Satisfactory	
<input type="checkbox"/> Genuine Floor Mats Installed	Safety	May Require Further Attention	
		Requires Immediate Attention	

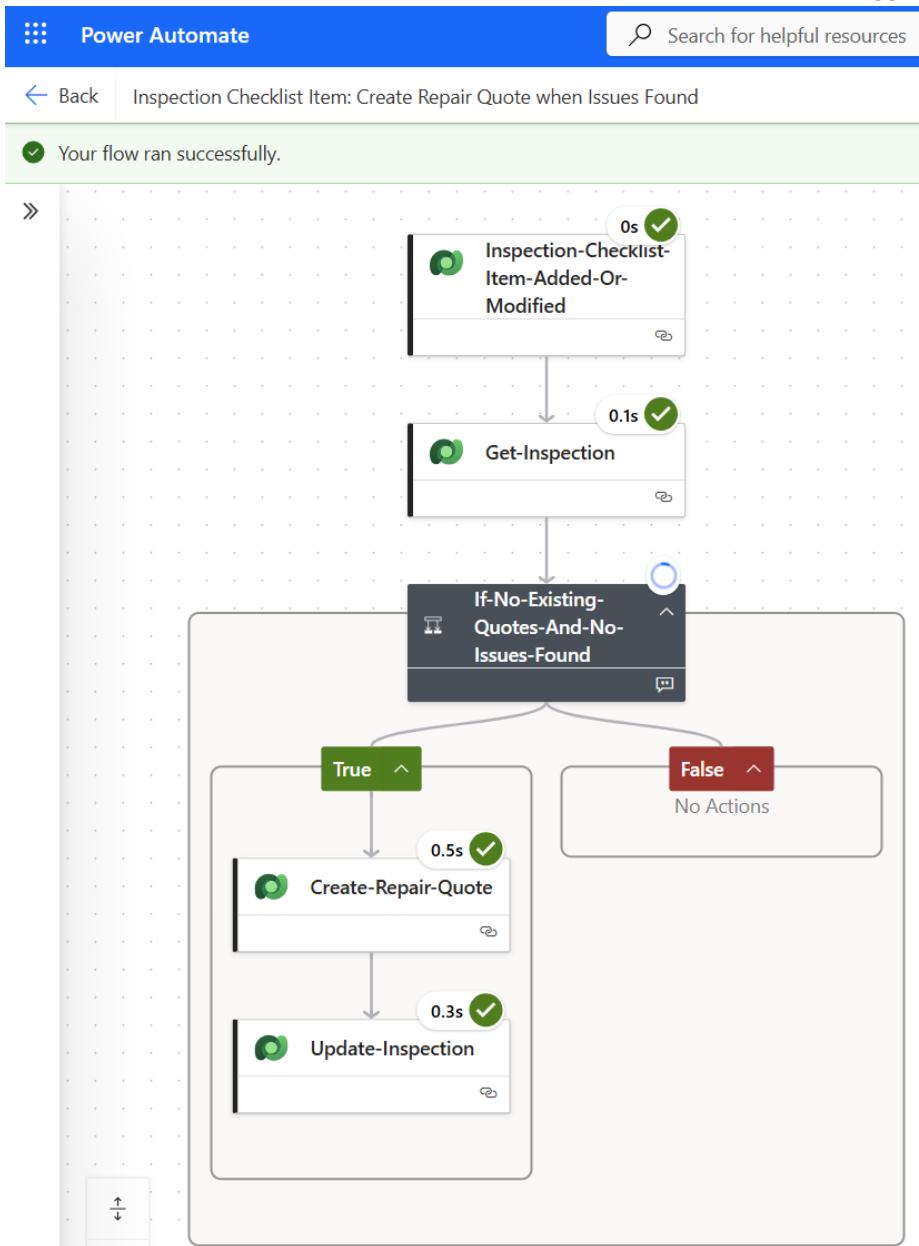
Rows: 3 Selected: 1

179. The record should update as soon as you select the choice.

Name* ↑	Area Name* ↑ ↓	Condition	Completed On
<input checked="" type="checkbox"/> Anchored properly & 1 mat only in driver footwell	Safety	Requires Imm...	
<input type="checkbox"/> Battery Condition	Safety		
<input type="checkbox"/> Genuine Floor Mats Installed	Safety		

Rows: 3 Selected: 1

180. Go back to the Power Automate workflow which should now have triggered and ran successfully.



181. Switch back to the **Vehicle Inspections Management** app.

182. Click **Refresh** at the top of the form.

183. Click the **Related** tab, then select **Repair Quotes**.

184. You should see the Repair Quote record in the associated view.

The screenshot shows the 'Repair Quotes' tab selected in the top navigation bar. Below it is a toolbar with 'Show Chart', 'New Repair Quote', 'Refresh', 'Flow', and 'Run R'. The main area is titled 'Repair Quote Associated View' and contains a list of repair quotes. The first item in the list has a red box around its quote number: 'Quote for Inspection: 0001-202509030211'.

185. Select the **Checklist** tab.

186. The **Issues Found?** column should have changed to Yes.

187. The **Recommended Repair** stage now appears as the next possible stage.

The screenshot shows the 'Vehicle Inspection Process' screen. At the top, there's a process flow diagram with stages: Intake, Scheduled, Inspection In Progress (13 Hrs), Recommend Repair, and Inspection Completed. The 'Recommend Repair' stage is highlighted with a red arrow pointing to it from the text above. Below the process flow is a 'Checklist' section. In the checklist table, the 'Issues Found?' column has a red box around the 'Yes' checkbox. The table also includes columns for Odometer, Odometer Unit, and Mile (m).

We will now test the rest of the Business Process Flow is working.

188. Select the **Inspection In Progress** stage and click **Next Stage**.

189. A Select Repair Quote dialog will appear. Select the record that was just created automatically.

The screenshot shows a software interface for vehicle inspection. At the top, there are tabs: 'Scheduled' (with a checkmark), 'Inspection In Progress (13 Hrs)' (with a blue circle icon), 'Recommend Repair' (with a lock icon), and a right-pointing arrow. Below these are sections for 'Odometer' and 'Issues Found?'. A modal dialog box titled 'Select Repair Quote' is open, containing a list of repair quotes. One quote is selected and highlighted with a red box: 'Quote for Inspection: 0001-202509030211'. The 'Next Stage' button at the bottom of the dialog is also highlighted with a red box.

190. The form for the Repair Quote loads since it was set as the entity (table) for the Recommend Repair stage.

The screenshot shows a detailed view of the 'Repair Quote' entity for the inspection quote. The top bar shows the quote number 'Quote for Inspection: 0001-202509030211 - Saved' and the owner 'User 001'. Below the bar, a navigation bar shows stages: 'Vehicle Inspection Process' (Active for 13 hours), 'Intake', 'Scheduled', 'Inspection In Progress', 'Recommend Repair (< 1 Min)' (highlighted with a blue circle), and 'Inspection Completed'. The main area shows fields: 'Name' (Quote for Inspection: 0001-202509030211), 'Total Estimated Costs' (---), 'Customer Decision' (dropdown menu showing 'Approve'), and 'Inspection' (dropdown menu showing '0001-202509030211').

191. Enter **500** for **Total Estimated Costs**.

192. Set the **Customer Decision** to **Approve**.

193. The **Repairing Vehicle** stage now becomes available.

The screenshot shows a process flow with three stages: "Inspection In Progress" (checkmark icon), "Recommend Repair (1 Min)" (blue circle icon), and "Repairing Vehicle" (grey circle icon). A red arrow points from the "Repairing Vehicle" stage towards the "Repairing Vehicle" stage in the next screenshot. Below the stages, there are two form fields: "Total Estimated Costs" (\$500.00) and "Customer Decision" (button labeled "Approve").

194. Click the Recommend Repair stage and click **Next Stage**.

The screenshot shows the "Recommend Repair (2 Min)" stage (blue circle icon) selected. A modal dialog is open, showing the "Active for 2 minutes" status and a "Customer Decision" field with a dropdown menu containing "Approve". A red box highlights the "Next Stage" button at the bottom right of the modal.

195. The form for the Inspection loads since it was set as the entity (table) for the Repairing Vehicle stage.

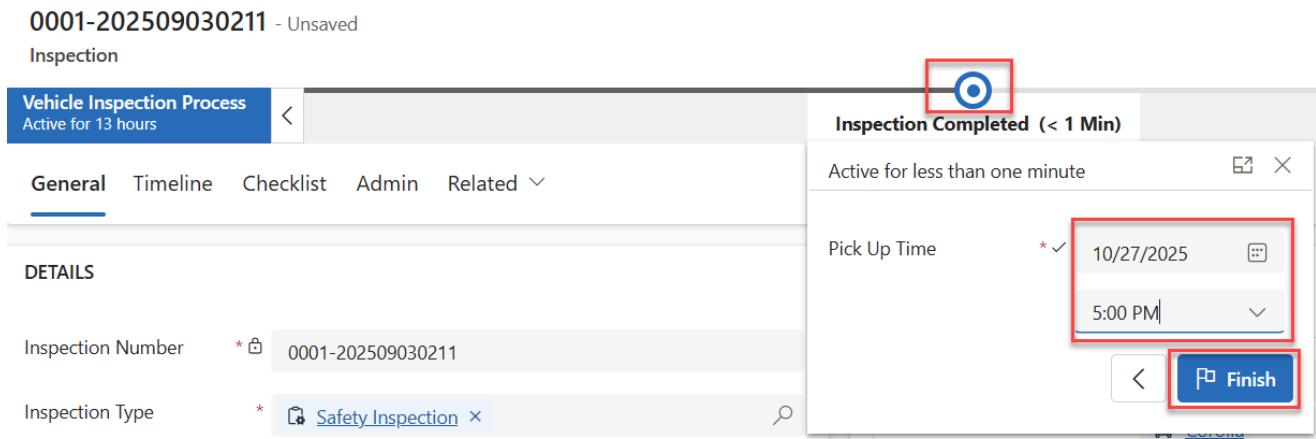
196. Click the **Repairing Vehicle** stage and set Repairs Completed On to the current date and time.

197. Click **Next Stage**.

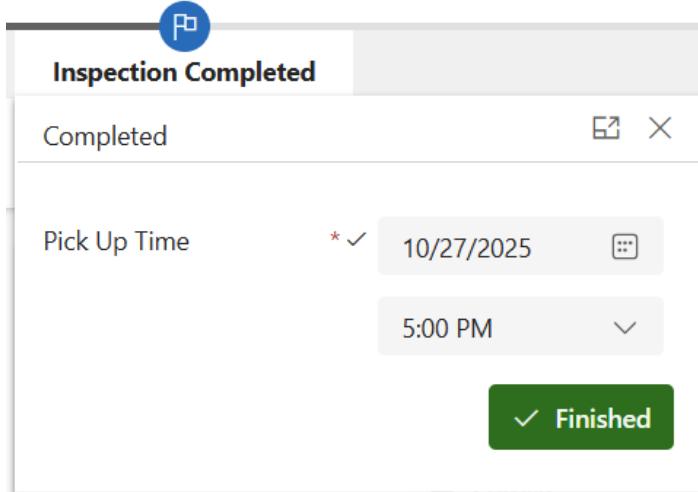
The screenshot shows the "Repairing Vehicle (< 1 Min)" stage (blue circle icon) selected. The "Repairs Completed On" field is highlighted with a red box, showing the value "10/27/2025 4:30 PM". A red box also highlights the "Next Stage" button at the bottom right of the stage area.

198. Click the **Inspection Completed** stage and set Vehicle Picked Up On to the current date and time +30m.

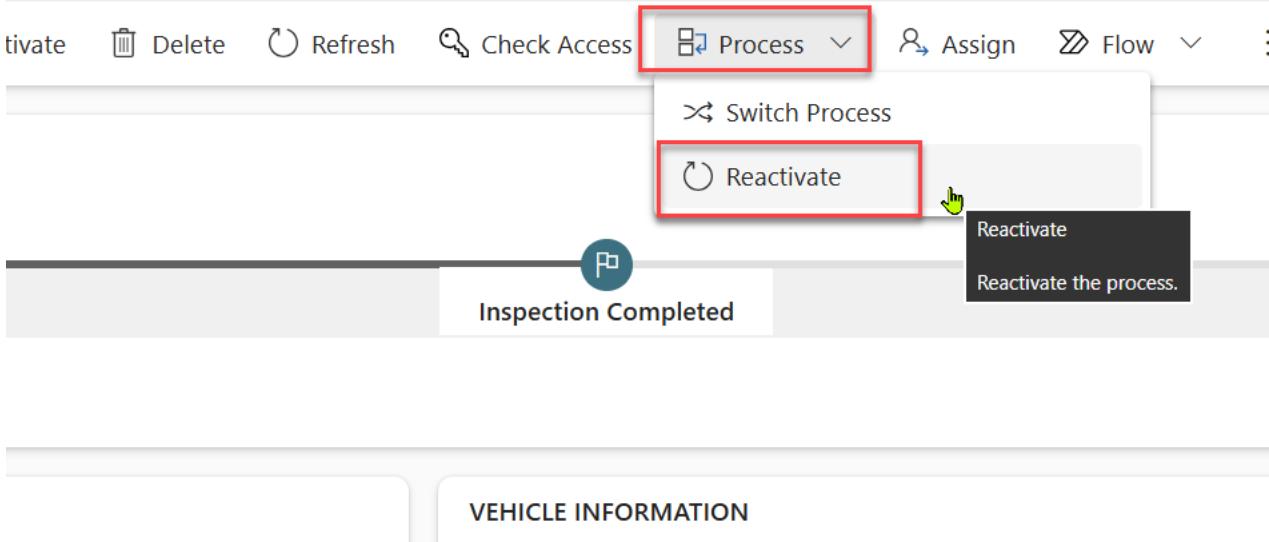
199. Click **Finish**.



200. The Process should now show as **Completed / Finished** and you will no longer be able to move between stages.



201. If the user needs to reactivate the process for the record to make changes or go to a previous stage, they can select **Process > Reactivate**.



Inspection Completed (< 1 Min)

Active for less than one minute



Pick Up Time

* ✓

10/27/2025



5:00 PM

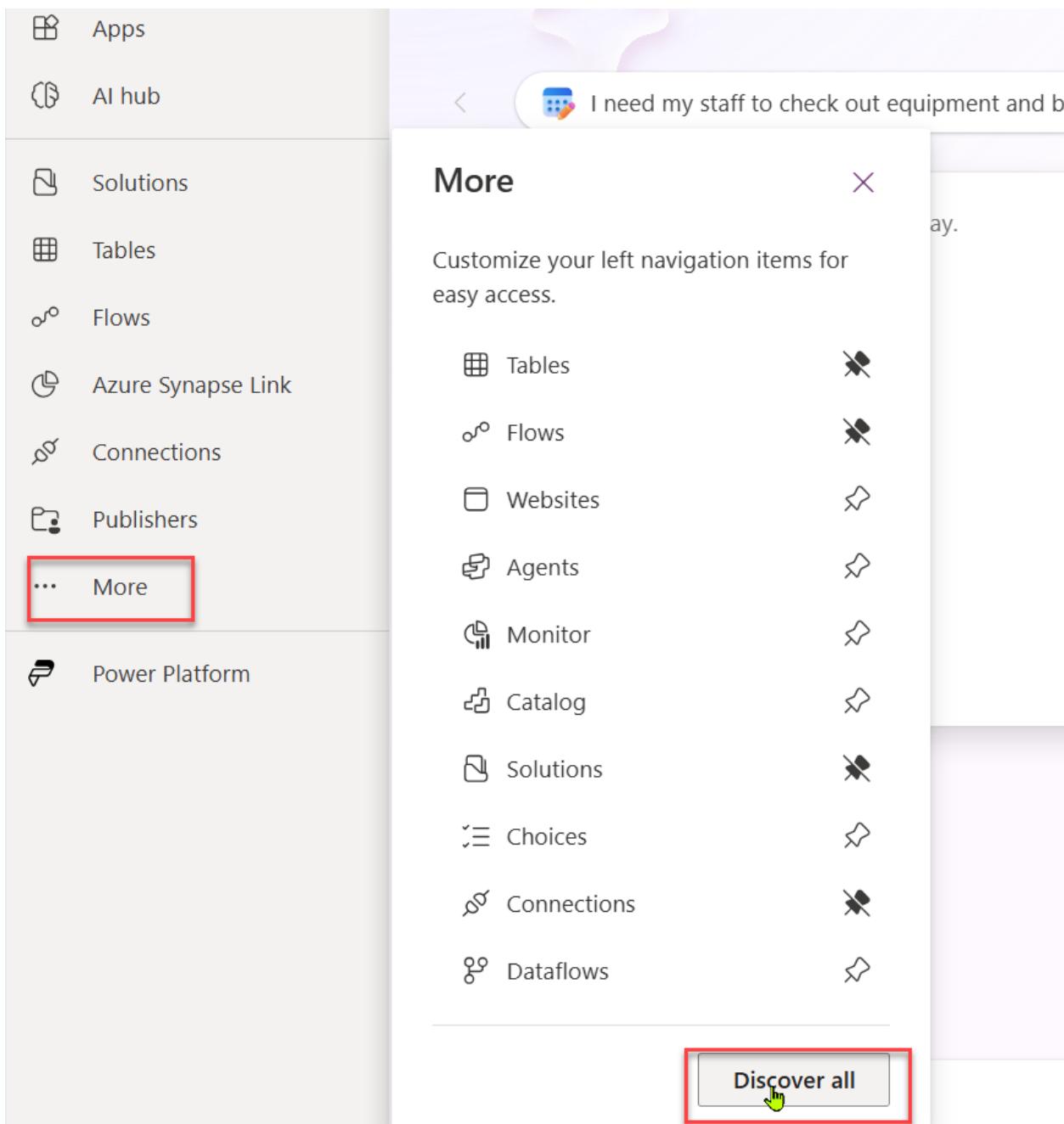


Finish

Create a Dataverse (Power Fx) Function

You will create a Dataverse Function to create a Repair Quote record using Power Fx.

202. In the Power Apps maker portal, navigate to the home page. In the left navigation menu, select **More > Discover all**.



203. Scroll to the bottom of the page and select **Functions**.

The screenshot shows the Microsoft Power Platform Admin Center interface. It is organized into three main columns:

- Data Management** (left column):
 - [Dataflows](#): See all dataflows created with Power Query.
 - [Azure Synapse Link](#): Continuously export Dataverse tables and get insights.
 - [Gateways](#): Manage gateway connections and permissions.
 - [Retention Policies](#): Organize your data for maximum operational efficiency.
- App enhancements** (middle column):
 - [Catalog](#): List all catalog in this environment.
 - [Component libraries](#): Save app components so you can reuse them in future apps.
 - [Wrap projects](#): Package your app for distribution in app stores like the Apple App Store and Google Play Store.
- App Management** (right column):
 - [Monitor](#): Keep an eye on the apps you've made, and take action to improve their performance, health, and security quickly.
 - [Publishers](#): Indicate who developed a solution and any components created inside it.

In the center, there is a section titled "Functions (Preview)" with a sub-section titled "Functions". A red box highlights the "Functions" button. Below it, a callout box provides information about Power Fx functions.

Functions (Preview)

fx Functions

With Power Fx functions, you can create reusable custom logic in Dataverse. Functions can be called from apps, flows, and other items, ensuring consistent logic across the environment.

204. Here you will see all Functions that exist in the environment (if any).

205. Select **+New function**

The screenshot shows the Power Apps portal interface. The left sidebar includes options like Home, Create, Learn, Plans, Apps, AI hub, Tables, Flows, Solutions, Functions (highlighted with a red box), More, and Power Platform.

The main area displays the "Functions (Preview)" section. At the top right of this section, a "New function" button is highlighted with a red box. Below it, a "Create a function" callout box provides information about Power Fx functions.

Create a function

With Power Fx functions, you can create reusable custom logic in Dataverse. Functions can be called from apps, flows, and other items, ensuring consistent logic across the environment.

The main table area shows a single row with a purple circular icon containing a lightning bolt and the text "You don't have any functions".

Name	Description	Created	Owner
You don't have any functions			

206. A side panel will open where you will author the Function.

- Display Name: **Create Repair Quote for Inspection**
- Description **Creates a Repair Quote record and associates it to the specified Inspection**
- Input Parameters:
 - i. Name: **InspectionNumber**
 - ii. Data Type: **String**
- Output Parameters:
 - i. Name: **RepairQuoteNumber**
 - ii. Data Type: **String**
- Table References:
 - i. **Inspections**
 - ii. **Repair Quotes**
- Formula: Enter the below formula block

```
With(
{
    wthInspectionId: Coalesce(
        LookUp(
            Inspections,
            'Inspection Number' = InspectionNumber
        ),
        Blank()
    ),
    If (
        IsBlank(wthInspectionId),
        Error(
            {
                Kind: ErrorKind.Validation,
                Message: $"Inspection {InspectionNumber} not found"
            }
        ),
        {
            RepairQuoteNumber:
            Text(
                Collect(
                    'Repair Quotes',
                    {
                        Name:
                        $"Quote for Inspection: {wthInspectionId.'Inspection
Number'}",
                        Inspection: wthInspectionId
                    }
                ).'Quote Number'
            )
        }
    )
}
```

207. Review the information looks correct

Note: you can't edit the Display Name or Description after creating

New function X

Use Power Fx functions to create reusable custom logic. Functions can be called from apps, flows, and other items. [Learn more](#)

Display name *

Create Repair Quote for Inspection

Description *

Creates a Repair Quote record and associates it to the specified Inspection

Parameters

Name * Data type *

InspectionNumber String Delete

+ New input parameter

Name * Data type *

RepairQuoteNumber String Delete

+ New output result

Table references ⓘ

Inspections, Repair Quotes Delete

Formula * ⓘ

```
With(
    {
        wthInspectionId: Coalesce(
            LookUp(
                Inspections,
                'Inspection Number' = InspectionNumber
            ),
            Blank()
        ),
        If (
            IsBlank(wthInspectionId),
            Error(
                {
                    Kind: ErrorKind.Validation,
                    Message: $"Inspection {InspectionNumber} not found"
                }
            )
        )
    }
)
```

Advanced options ⌂

Save Cancel

208. Click **Save**.

209. Once the function is created, select the row in the list and click **Test** at the top of the command bar.

Power Apps

Search

Environment
PPCC25 Workshop

New function Edit Delete Copy code snippet Test

Functions (Preview)

Create a function

With Power Fx functions, you can create reusable custom logic in Dataverse. Functions can be called from apps, flows, and other items, ensuring consistent logic across the environment.

Name	Description	Created
Create Repair Quote for Inspection	Creates a Repair Quote record and ...	10/3/2025

210. Enter in an Inspection Number for an existing record (open the **Vehicle Inspection Management** model-driven app, navigate to Inspections, open any Inspection record and copy the Inspection Number.)

Power Apps | Vehicle Inspection Management

Home Recent Pinned Inspections

Vehicle Inspection Process Active for 30 days Intake (30 D)

General Timeline Checklist Admin Related

DETAILS

Inspection Number * 0015-202509030331

211. Click **Play**.

212. You should see a **Success** status and the **Response** in JSON format.

The screenshot shows the Power Apps Studio interface. In the center, there's a 'Create Repair Quote for Inspection' function page. On the left, a sidebar has icons for Back, Functions (Preview), Input parameters, and Play. The main area shows an 'Input parameters' section with 'InspectionNumber *' set to '0015-202509030331'. Below it is a 'Play' button in a red-bordered box. Under 'Response', it says 'Success' with a green checkmark. A green box highlights the JSON response: `{"@odata.context":"https://ppcc-mda.crm3.dynamics.com/api/data/v9.0/$metadata#Microsoft.Dynamics.CRM.ppcc_CreateRepairQuoteforInspectionResponse","RepairQuoteNumber":"QUO-000001-P2J7"}`.

213. Add a random character onto the end of the InspectionNumber and press Play again.

214. You should see an **Error** status and the custom error message that was set in the formula.

This screenshot shows the same Power Apps Studio interface as the previous one, but with an error. The 'InspectionNumber' field now contains '0015-202509030331X'. A red arrow points to this field. The 'Play' button is again in a red-bordered box. The 'Response' status is now 'Error' with a red exclamation mark. A red box highlights the error message in the JSON response: `{"error":{"code": "0x80040265", "message": "Plugin Create Repair Quote for Inspection failed with: CustomAPI parameter '$' failed with error Error: inspection 0015-202509030331X not found"}}`.

215. Navigate to **Solutions** and open the **Vehicle Inspections** solution.

216. You will notice the following components have been added to the solution:

- **Custom API** – Functions are created as Dataverse Custom APIs under the hood that can be invoked by other Power Platform components or programmatically.
- **Custom API Request Parameter** – Created for every **input parameter** defined in the function and associated with the Custom API
- **Custom API Response Parameter** - Created for every **output parameter** defined in the function and associated with the Custom API
- **Function** – The definition of your function (e.g. Display Name) and associated with the Custom API
- **FxExpression** – The Power Fx formula for the Function and is associated with the Custom API

The screenshot shows the Power Apps environment with the 'Objects' list open. The 'Custom API (1)' item under 'Custom API Request P...' is highlighted with a red box. The list also includes 'Function (1)' and 'FxExpression (1)'.

Display name ↑	Name ↓	Type ↴
Create Repair Quote for Inspection	CreateRepairQu...	FxExpression
Create Repair Quote for Inspection	CreateRepairQu...	Custom API
Create Repair Quote for Inspection	CreateRepairQu...	Function
InspectionNumber	InspectionNumber	Custom API Request Parameter
RepairQuoteNumber	RepairQuoteNu...	Custom API Response Property

217. You can select the Function here to edit it as well

The screenshot shows the 'Edit function' dialog for 'Create Repair Quote for Inspection'. The formula is:

```

With(
    {
        wthInspectionId: Coalesce(
            LookUp(
                ppcc_inspection,
                ppcc_inspectionnumber = InspectionNumber
            ),
            Blank()
        ),
        If(
            IsBlank(wthInspectionId),
            Error(
                {
                    Kind: ErrorKind.Validation,
                    Message: $"Inspection {InspectionNumber} not
                }
            )
        )
    }
)

```

218. You may also create a new Function from the solution

Note: There is no test button from the solution explorer at the time of writing.

Execute a Dataverse Function from Power Automate

Dataverse Functions can be used from a number of places in the Power Platform:

- Canvas Apps or Custom Page
- Power Automate Cloud Flows
- Copilot Studio via Generative Actions
- Model-Driven Apps using code-first (JavaScript web resources, C# Plugins)
- Programatically via the Dataverse Web API Actions or the .NET SDK

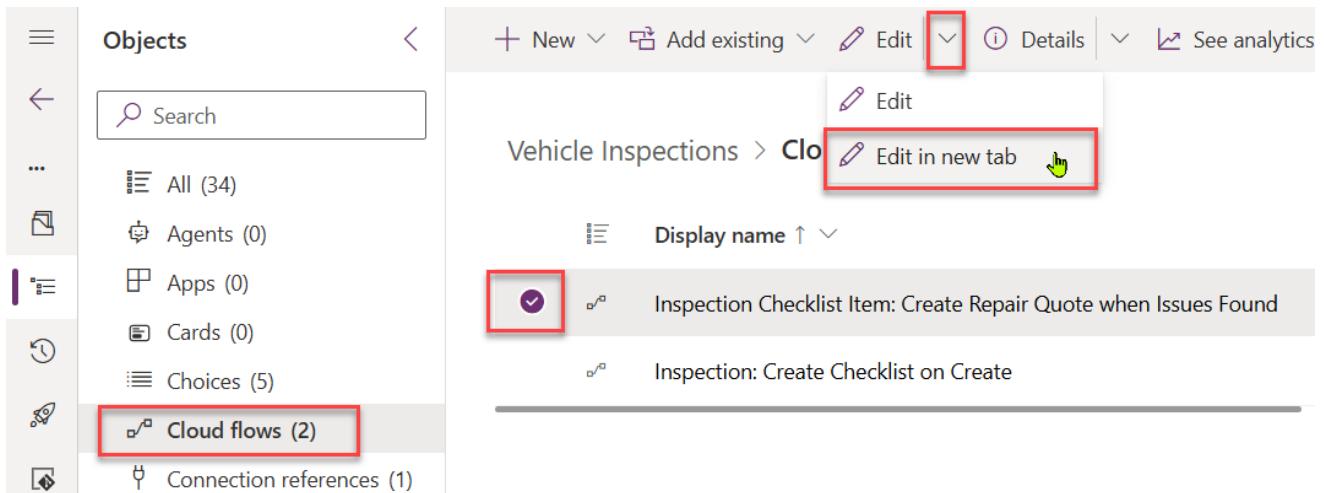
You will modify the **Inspection Checklist Item: Create Repair Quote when Issues Found** flow to invoke the Dataverse function.

219. Navigate to **Solutions** and open the **Vehicle Inspections** solution

220. Select **Cloud flows** under the solution object tree.

221. **Select the row for Inspection Checklist Item: Create Repair Quote when Issues Found**

222. Select the **down arrow** next to Edit, then select **Edit in new tab**



223. In the flow designer, expand the condition, then add a new action above the Create-Repair-Quote step.

224. Select Microsoft Dataverse > Perform an unbound action

Add an action > Microsoft Dataverse X

Microsoft Dataverse ★

Add a new row	Add a new row to selected environm...
Delete a row	Delete a row from selected environm...
Download a file or an image	Download a file or an image from sel...
Get a row by ID	Get a row by ID from selected enviro...
List rows	List rows from selected environment
Perform a background operation (pr...	Perform a bound action
Perform a bound action in selected e...	Perform an unbound action (highlighted)
Perform an unbound action in select...	Relate rows
Relate rows in selected environment	Search rows (preview)

225. Under Action Name, copy/paste in **ppcc_CreateRepairQuoteforInspection**.

226. Then select **ppcc_CreateRepairQuoteforInspection** as it becomes the only search result.

Perform an unbound action ⋮ ⌂

Parameters [Settings](#) [Code view](#) [Testing](#) [About](#)

Action Name *

ppcc_CreateRepairQuoteforInspection ▼

ppcc_CreateRepairQuoteforInspection (highlighted)

[Enter custom value](#)

227. Under Advanced parameters, click the **Show All** button.

Parameters Settings Code view Testing About

Action Name *

ppcc_CreateRepairQuoteforInspection

Advanced parameters

Showing 0 of 1

Show all (highlighted) Clear all

228. Click inside the **Inspection Number** from the **Get-Inspection** step.

Advanced parameters

Showing 1 of 1

Show all Clear all

Item/InspectionNumber

Press '/' to insert dynamic value or expression

Perform an unbound action

Parameters Settings Code view Testing About

Action Name *

ppcc_CreateRepairQuoteforInspection

Advanced parameters

Showing 1 of 1

Show all Clear all

Item/InspectionNumber

Inspection Num... (highlighted)

Connected to PPCC Microsoft Dataverse. [Change connection reference](#)

When a row is added, modified or deleted

Search

Parameters

Sharepoint Site Environment Variable - InspectionsStandard

SLA Web Client Deprecation Acknowledge (msdyn_SLAWeb)

Should the Peek Button Be Showed (msdyn_ShouldShowPeek)

LOB Guid List for user expansion (msdyn_lineofbusinessfilter)

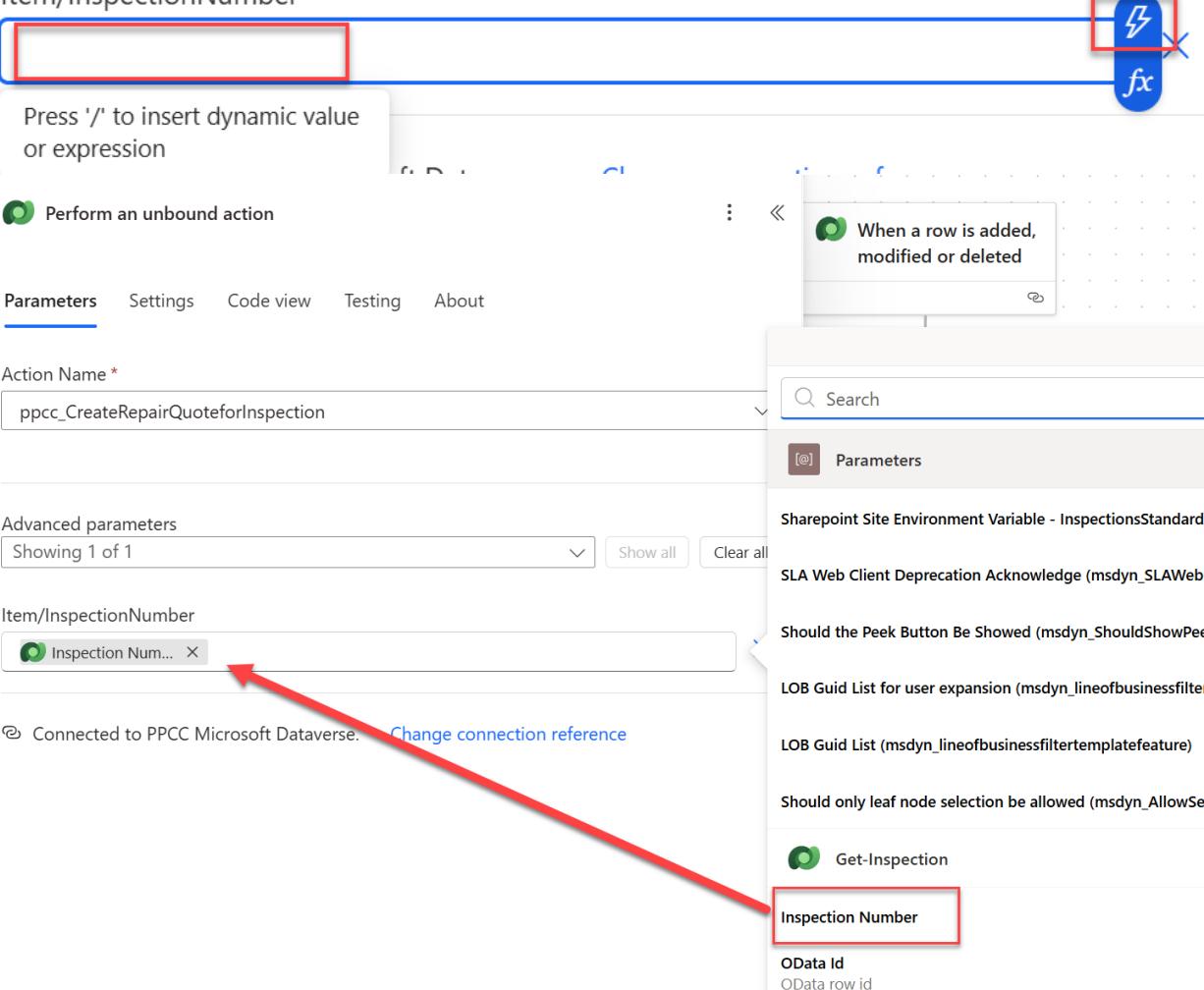
LOB Guid List (msdyn_lineofbusinessfiltertemplatefeature)

Should only leaf node selection be allowed (msdyn_AllowSelectionOnly)

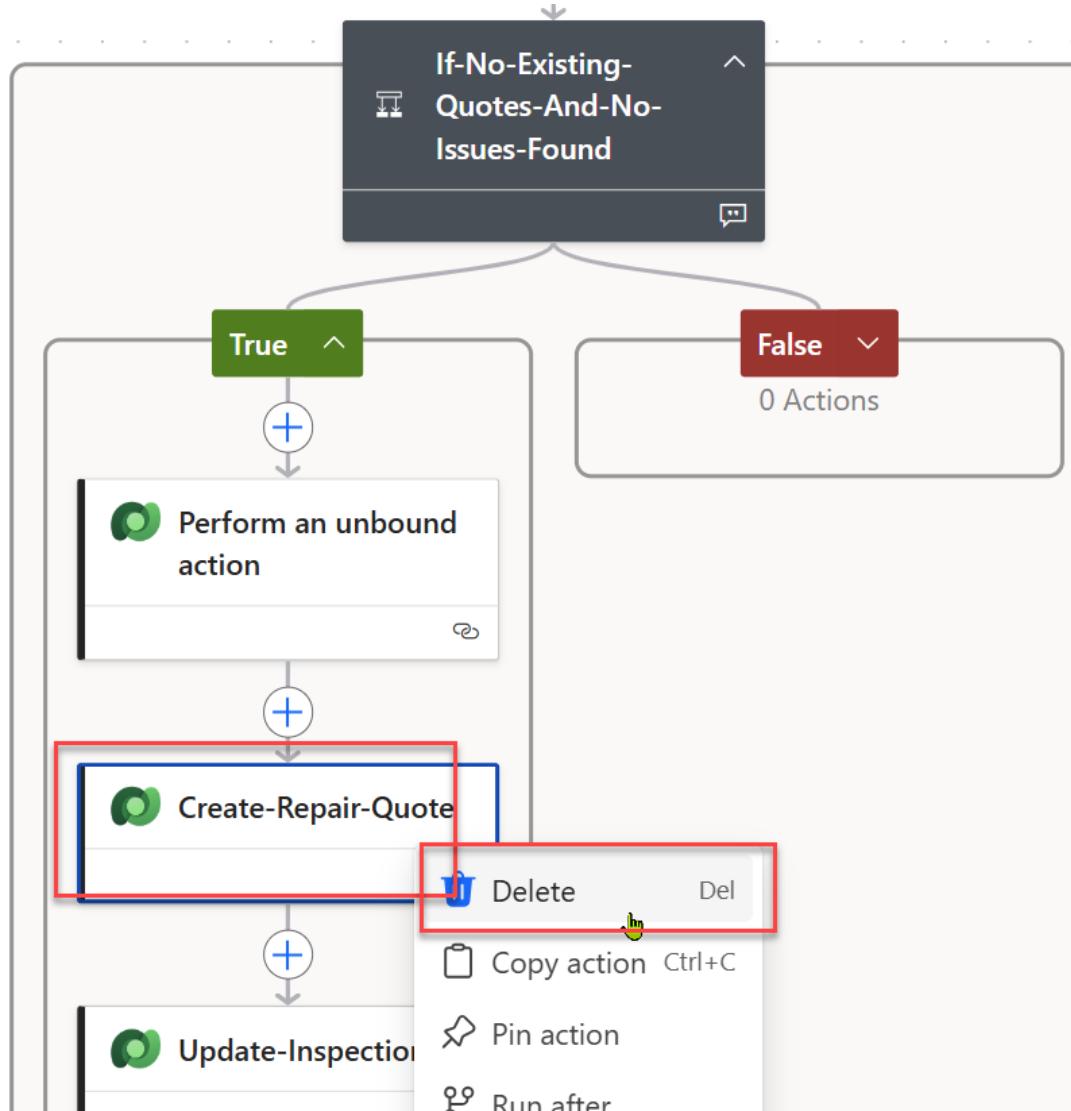
Get-Inspection

Inspection Number (highlighted)

OData Id
OData row id



229. Right click on the **Create-Repair-Quote** action and click **Delete** and confirm.



230. Click **Test > Manually > Publish & Test**

231. Open the **Vehicle Inspections Management** model-driven app.

232. Navigate to **Inspections** and open an existing record with at least 1 checklist item.

233. Click the **Checklist** tab.

234. In the checklist editable sub-grid, click into the Condition cell of one of the rows, and change the value to

Requires Immediate Attention

The screenshot shows a checklist record with the following details:

Name*	Area Name*	Condition	Completed On
Anchored properly & 1 mat only in driver footwell	Safety	Satisfactory	
Battery Condition	Safety	May Require Further Attention	
Genuine Floor Mats Installed	Safety	Requires Immediate Attention	

Below the grid, there is a message: "Rows: 3 Selected: 1".

235. The record should update as soon as you select the choice.

236. Go back to the Power Automate workflow and confirm it ran successfully.