Exercise 3-4 Create a Histogram Application

Goal

Modify the producer/consumer template to create a histogram from acquired data.

Scenario

You want to create an application which does the following:

- Simulates acquisition of a waveform.
- Simulates processing of the waveform which includes generating a histogram.
- Saves a snapshot of a histogram.

You can modify the producer/consumer template to handle those three tasks as well as errors and UI events from the producer/consumer template itself.

Design

After copying the template, you update the producer loop to generate waveform data and you update the consumer loop to display a histogram and take a snapshot of the histogram when the user specifies.

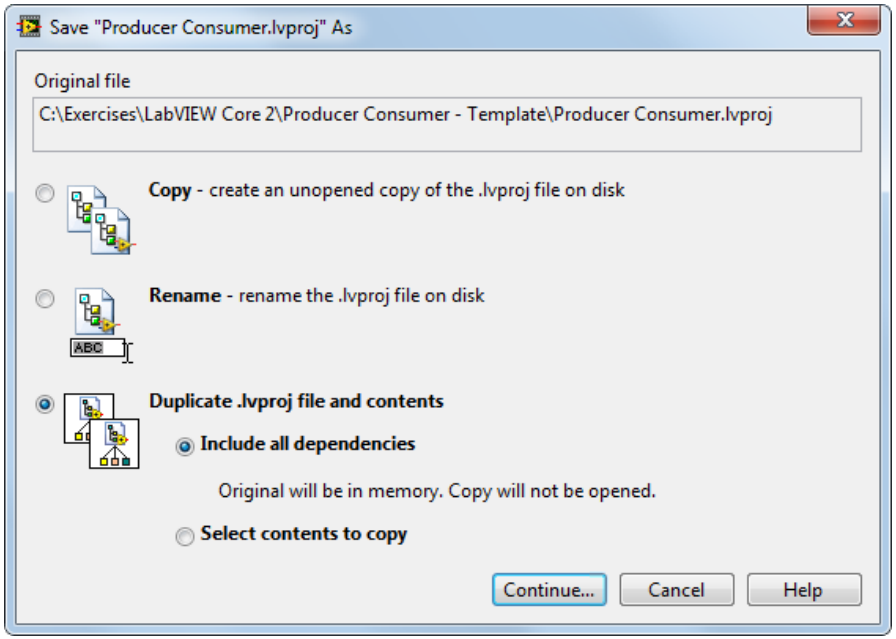
Implementation

1. Move and rename the Producer Consumer project and files.

   ☐ Open the `Producer Consumer.lvproj` located in the `<Exercises>\LabVIEW Core 2\Producer Consumer - Template` directory.

   ☐ Select **File»Save As** and set the save as options as shown in Figure 3-17, and then click the **Continue** button.

Figure 3-17. Save As Options



   ☐ Enter `Histogram` as the name of the project, and save the project to the `<Exercises>\LabVIEW Core 2\Histogram` directory.

2. Close the **Producer Consumer** - **Template** Project Explorer window.

3. Open Histogram.lvproj and rename the project VIs in LabVIEW so that LabVIEW can update all links and instances of the VIs.

   ☐ Right-click **Main.vi** in the **Project Explorer** window and select **Rename**.

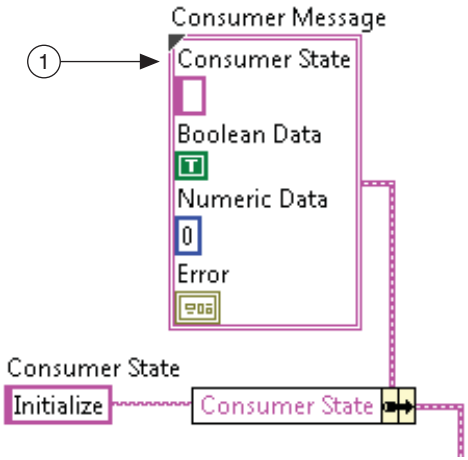
   ☐ Rename the VI as `Histogram Main.vi` and click **OK**.

4. Add the `Shared` folder to the project as an auto-populating folder. The Shared folder contains the Generate Data VI and the Running Histogram VI that you use later.

5. Open the block diagram of the Histogram Main VI.

6. Update the **Consumer Message** type definition, shown in Figure 3-18 to handle waveform data.

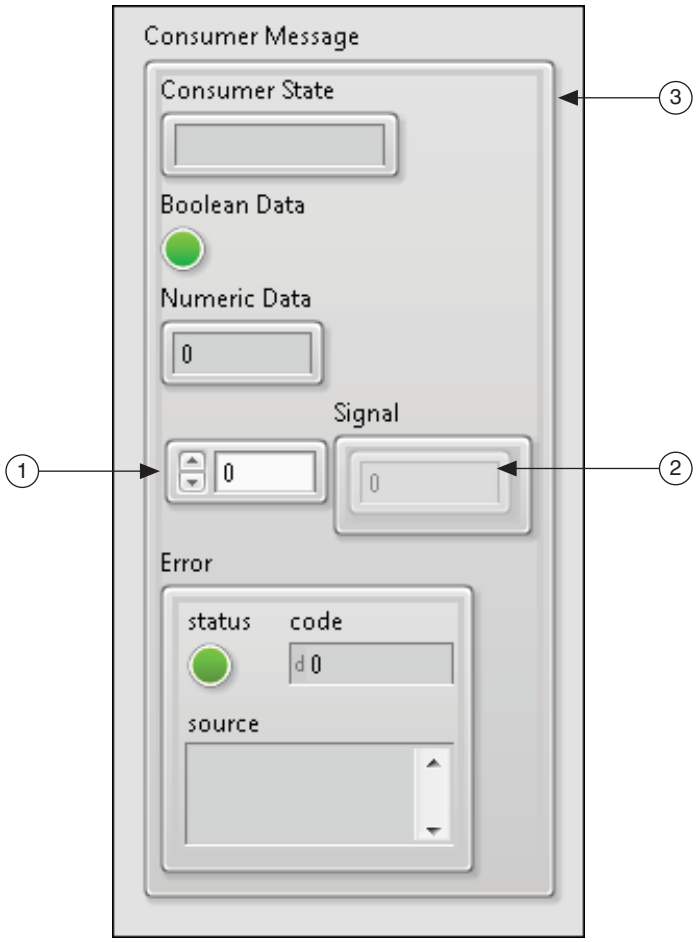
Figure 3-18. Consumer Message Type Definition Terminal on Histogram Main VI Block Diagram



1 Consumer Message type definition—Right-click the Consumer Message type definition located to the left of the producer loop on the Histogram Main VI block diagram and select **Open Type Def**.

a. Modify the Consumer Message type definition as shown in Figure 3-19.

Figure 3-19. Consumer Message Type Definition – Consumer Message.ctl

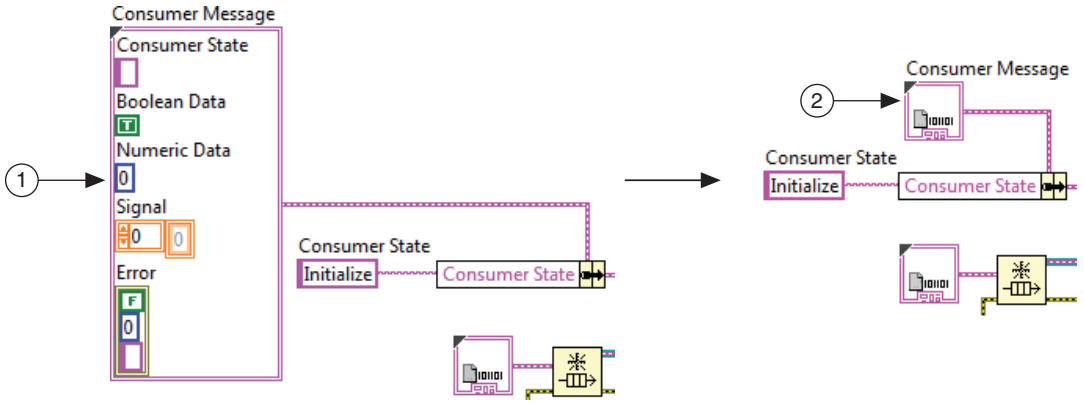


1 Array—Add an array to the type definition so it can handle waveform data. Rename the Array `Signal`.
2 Numeric Indicator—Add a numeric indicator to the array.
3 Right-click the cluster border and select **Reorder Controls In Cluster** and arrange them so that the Signal control is directly below the **Numeric Data** control.

b. Apply changes, save, and close the type definition.

7. Display the type definition as an icon on the block diagram as shown in Figure 3-20.

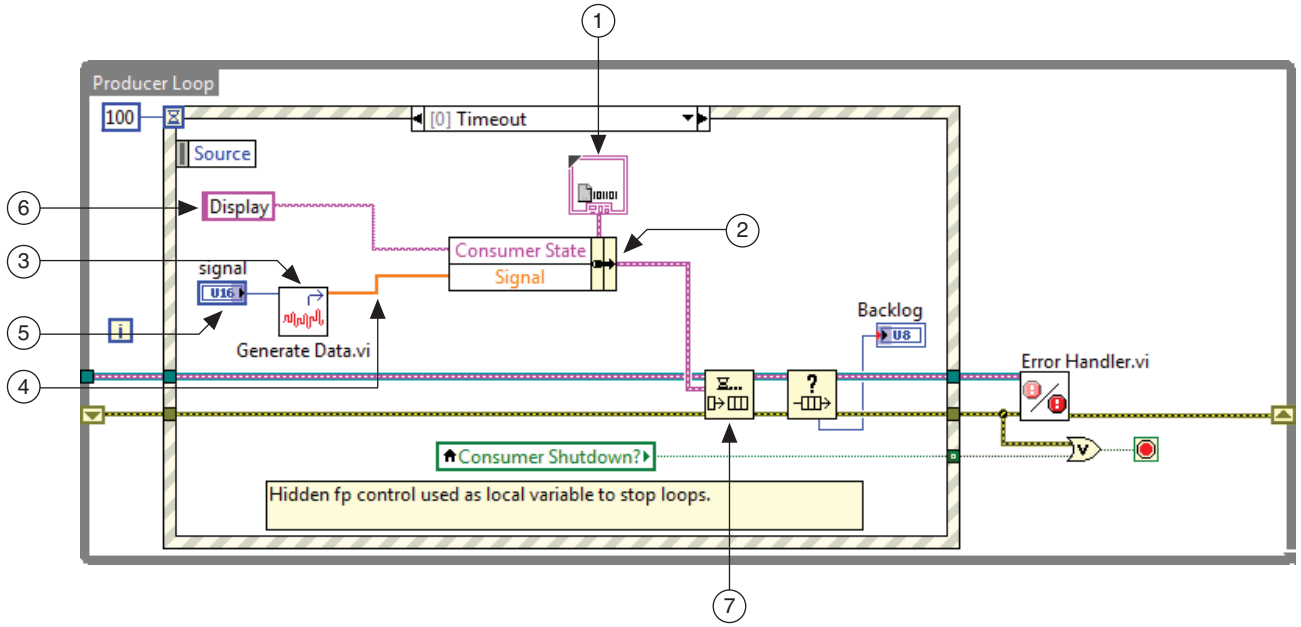
Figure 3-20. Viewing a Type Definition as an Icon



1 Right-click the Consumer Message type definition and select **AutoSizing»Arrange Vertically** from the shortcut menu.
2 Right-click the Consumer Message type definition and select **View Cluster as Icon** to save space on the block diagram.

8. Send signal data through the Consumer Message type definition. Complete the Timeout event in the producer loop as shown in Figure 3-21.

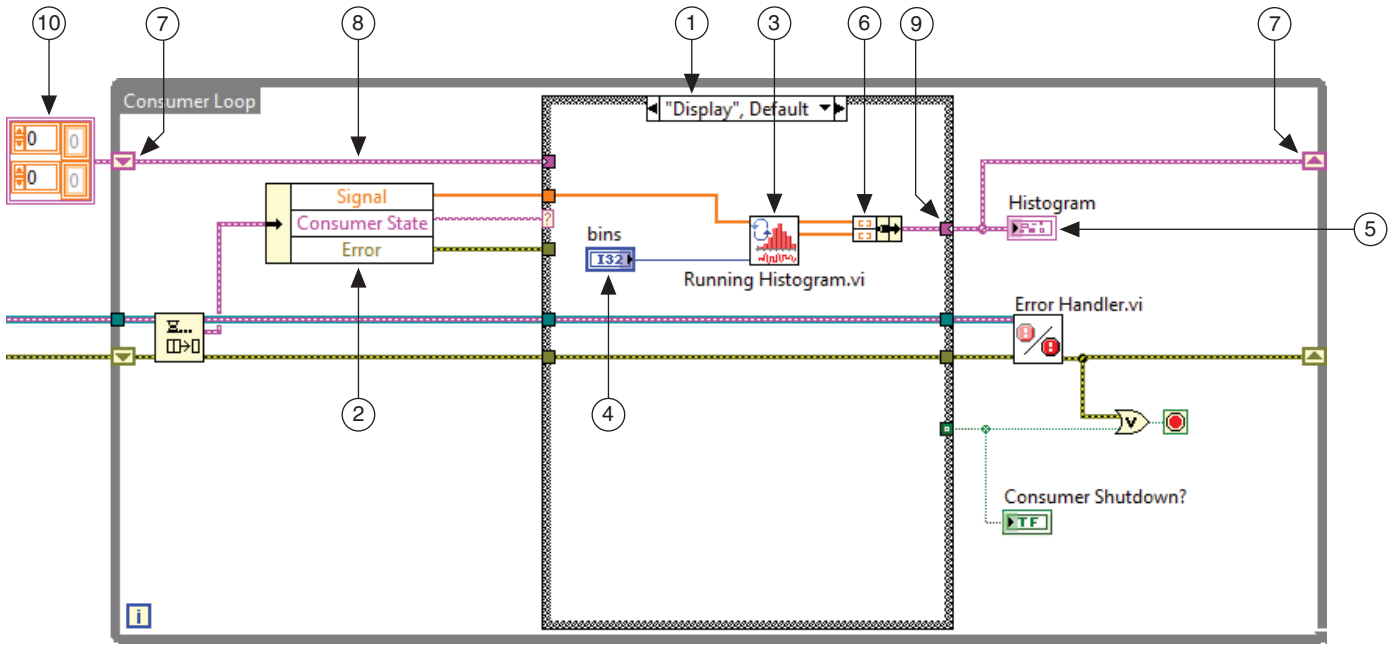
Figure 3-21. Updating the Producer Loop Timeout Event



1 Consumer Message type definition—Copy the Consumer Message type definition and paste it inside the Timeout event case.
2 Bundle By Name function—Wire the Consumer Message typedef to the **input cluster** input.
   ☐ Expand the node to display two elements.
   ☐ Select **Consumer State** and **Signal**.
3 Generate Data VI—Drag the Generate Data VI from the **Shared** folder in the **Project Explorer** window into the Timeout event case.
4 Wire the **Y** output of the Generate Data VI to the **Signal** input of the Bundle By Name function.
5 Create a control for the **signal** input of the Generate Data VI.
6 Create a constant for the **Consumer State** input.
7 Enqueue Element—Right-click the queue wire and select **Insert»Queue Operations Palette»Enqueue Element.**
   ☐ Wire the error wire through the Enqueue Element function to the Get Queue Status function. It will appear wired, but when you insert the node, the error wire is behind the Enqueue Element function.

9. Create the Display case in the consumer loop as shown in Figure 3-22.

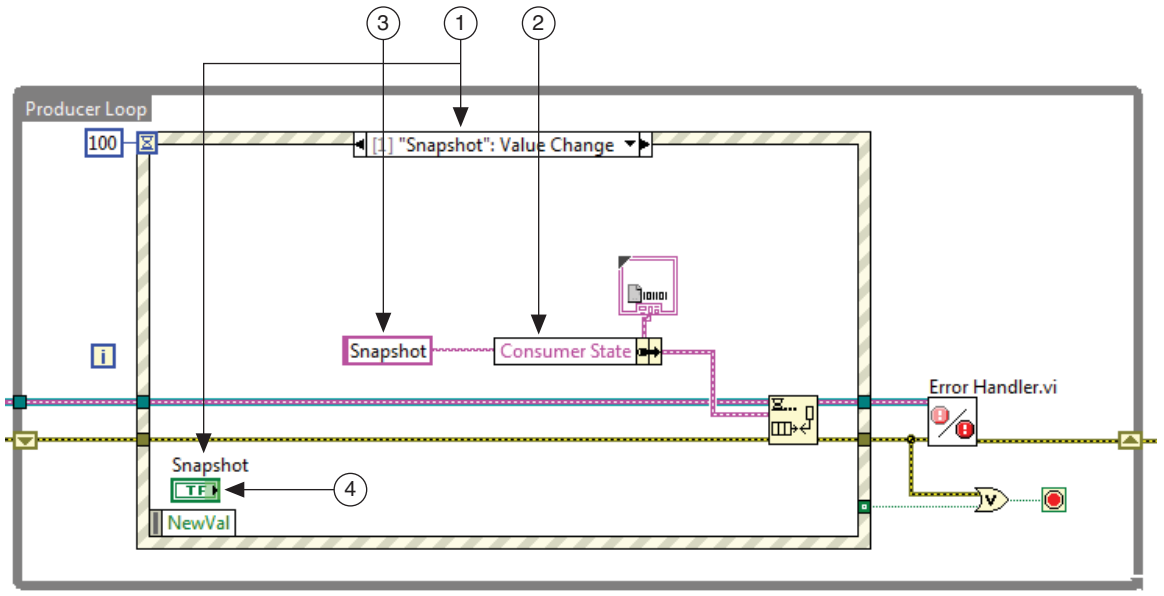
Figure 3-22. Updating the Consumer Loop Display Case



1 Open to the Default case of the Case structure and rename the case to `"Display", Default`.
2 Unbundle By Name function—Change the **Numeric Data** element to **Signal** and remove the **Boolean Data** wire and element.
3 Running Histogram VI—Drag the Running Histogram VI from the **Shared** folder in the **Project Explorer** window.
4 Numeric control—Create a control for the **bins** input and label the control `Bins`.
5 XY Graph (Silver)—On the front panel, place an XY Graph (Silver) and rename it `Histogram`.
6 Bundle function—Wire the **histogram** and **x axis** outputs from the Running Histogram VI to the Bundle function.
7 Replace the right Histogram tunnel with a shift register and complete the shift register.
8 Wire the left shift register to the Case structure.
9 Right-click the Histogram output tunnel and select **Linked Input tunnel»Create & Wire Unwired Cases** and then click the Histogram input tunnel on the left.
10 Right-click the left shift register and create a constant.

10. Create a Snapshot event in the producer loop by changing the “High Priority Message”: Value Change event, as shown in Figure 3-23.

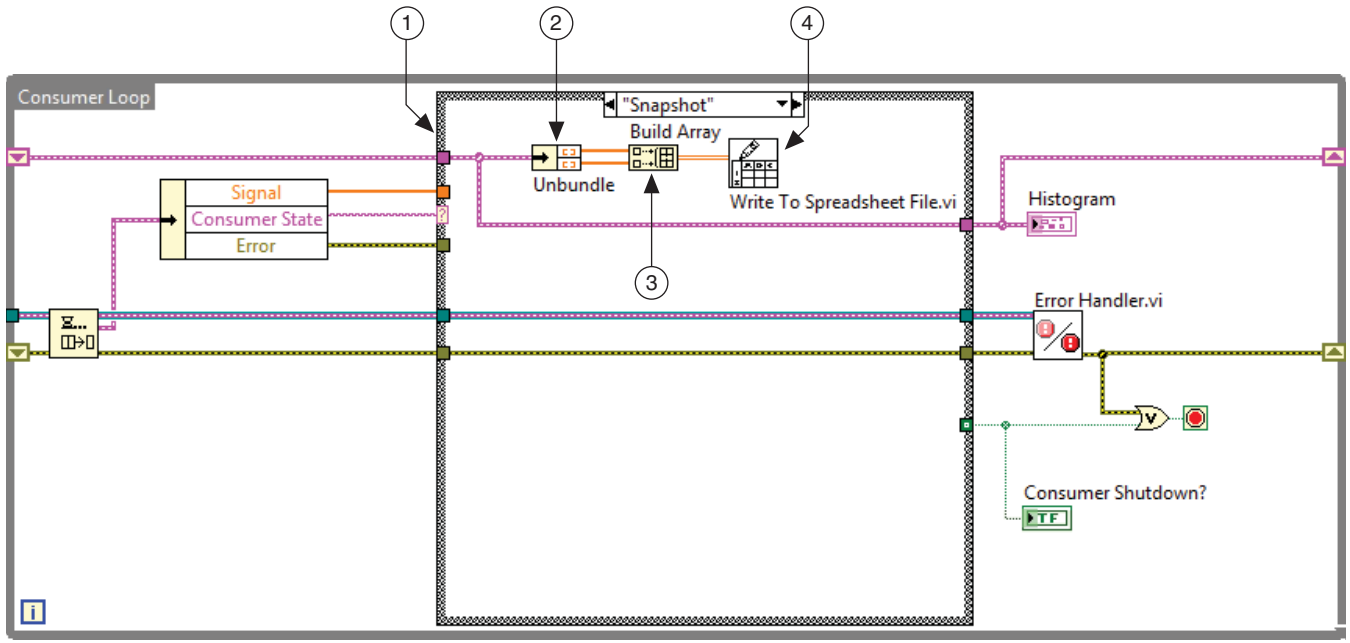
Figure 3-23. Updating the Producer Loop “Snapshot”: Value Change Event



1 Change the event name—Change the label of the **High Priority** button to `Snapshot`. Changing the name of the button changes the event name.
2 Bundle By Name function—Delete the values wired to the **Boolean Data** and **Numeric Data** inputs of the Bundle By Name function and hide the terminals.
3 Change the value of the **Consumer State** string constant to `Snapshot`.
4 Double-click the **Snapshot** control to locate the button on the front panel. Change the Boolean text displayed on the button to `Snapshot`.

11. Add the Snapshot case to the consumer loop as shown in Figure 3-24.

Figure 3-24. Updating the Consumer Loop Snapshot Event

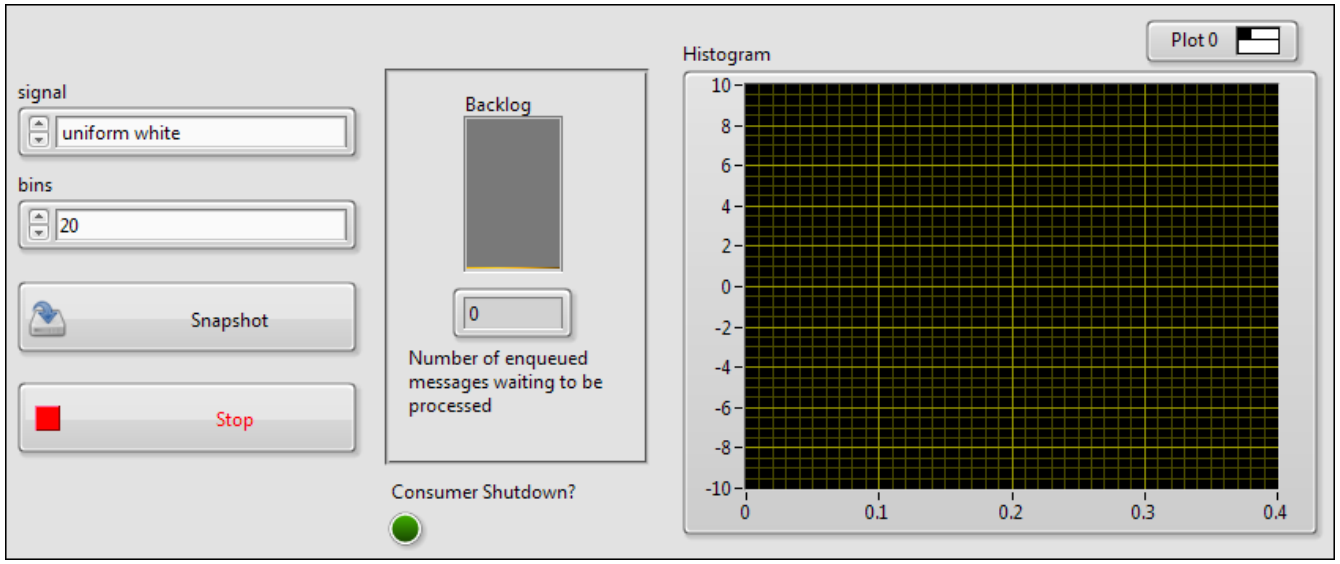


1 Duplicate the Initialize case—Right-click the case structure and select **Duplicate Case**. Rename the duplicate case `Snapshot`.
2 Unbundle function—After you wire the input, the Unbundle function contains two 1D arrays.
3 Build Array function—Wire both **1D Array** outputs to the Build Array function.
4 Write to Spreadsheet File VI—Wire the **appended array** output of the Build Array function to the **2D data** input.

12. Delete the Normal Priority Message event from the Event structure in the Producer Loop. LabVIEW deletes the corresponding button from the front panel.

13. Cleanup the front panel of the VI as shown in Figure 3-25.

Figure 3-25. Cleaning Up the Front Panel of the Histogram Main VI



Test

1. Run the VI.

2. To create the look of a histogram in the chart, click the plot legend and select a horizontal bar plot type from the bottom row. You may also want to remove the line interpolation by clicking the plot legend and selecting **Interpolation** from the shortcut menu.

3. Notice how changing the **Signal** and **Bins** values changes the look of the histogram.

4. Click the **Snapshot** button. A file dialog box displays so you can save the histogram file.

   ☐ Choose a name for the new file, including `.txt`.

   ☐ While the dialog box is open, the **Backlog** indicator rises.

   ☐ Click the **OK** button to save the file.

   ☐ The **Backlog** indicator should quickly decrease.

5. Click the **Stop** button to stop the VI.
6. Open the saved text file and review the contents to see the bins and values of the histogram.
7. Save and close the Histogram project.

End of Exercise 3-4