Updating the Print Console With Error Handling

Goal

Update a VI to report errors and include a shutdown state.

Scenario

You begin with a Print Console VI that simulates the control of print job submissions and monitors the status of the print jobs being processed. Each time you click the Queue Event button, you send a print job to the printer. You monitor the number of print jobs in the queue using a vertical bar. You use LEDs and a numeric indicator to monitor the process of the print jobs.

The Print Console VI does not report errors. You must update the VI to report potential errors and warnings.

You must also complete a shutdown procedure after the last print job.

Design

To implement error reporting, you must do the following.

❑ Use shift registers to monitor and track errors and warnings for all loop iterations.

❑ Use the Merge Errors function to obtain error information from both the producer and consumer loops.

❑ Use the Simple Error Handler VI to display both errors and warnings.

❑ Use an error handling VI that causes the state machine to execute the shutdown case when an error occurs or when the producer loop sends a shutdown message.

Implementation

The files that you need to complete this exercise are here:
`<NI eLearning>\LV Core 2\Handling Errors\Exercise.`

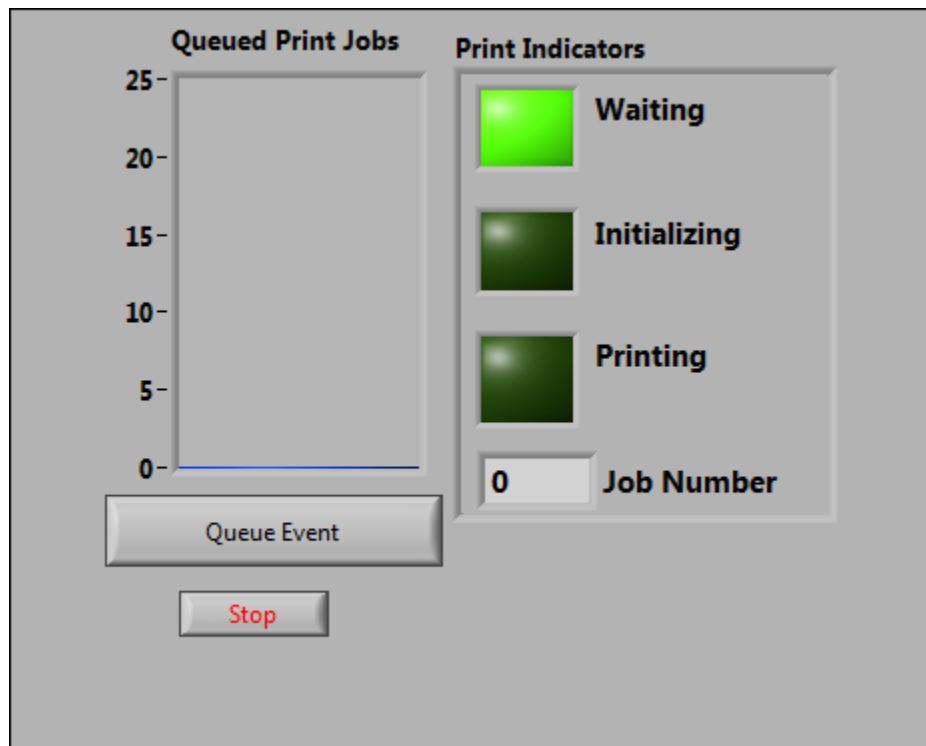1. Open `Print Console with Error Handling.lvproj` in the `<Exercise>` directory.

Figure 1. Front Panel of Print Console

2. Open and run the Print Console VI. The front panel of this VI is shown in Figure 1.

   ❑ Each time you click the Queue Event button, you add a print job to the queue. The Queued Print Jobs bar displays the backlog of print jobs.

   ❑ When there are no print jobs in the queue, the Waiting LED is turned on. When each print job is processed the Job Number indicator is updated to show the number of print jobs that have been processed. The Initializing LED is turned on during print initialization and the Printing LED is turned on during the print operation.

   ❑ Click the **Stop** button.

3. Examine the block diagram.

4. Notice that none of the errors are reported to the user.

5. Notice that the consumer loop stops only when there is an error. This occurs when the producer loop stops and the Release Queue function destroys the queue reference. Without a valid queue reference, the Dequeue Element function in the consumer loop returns error code **1122** with an explanation of **Refnum became invalid while node waited for it**.

Error Handling in Consumer Loop

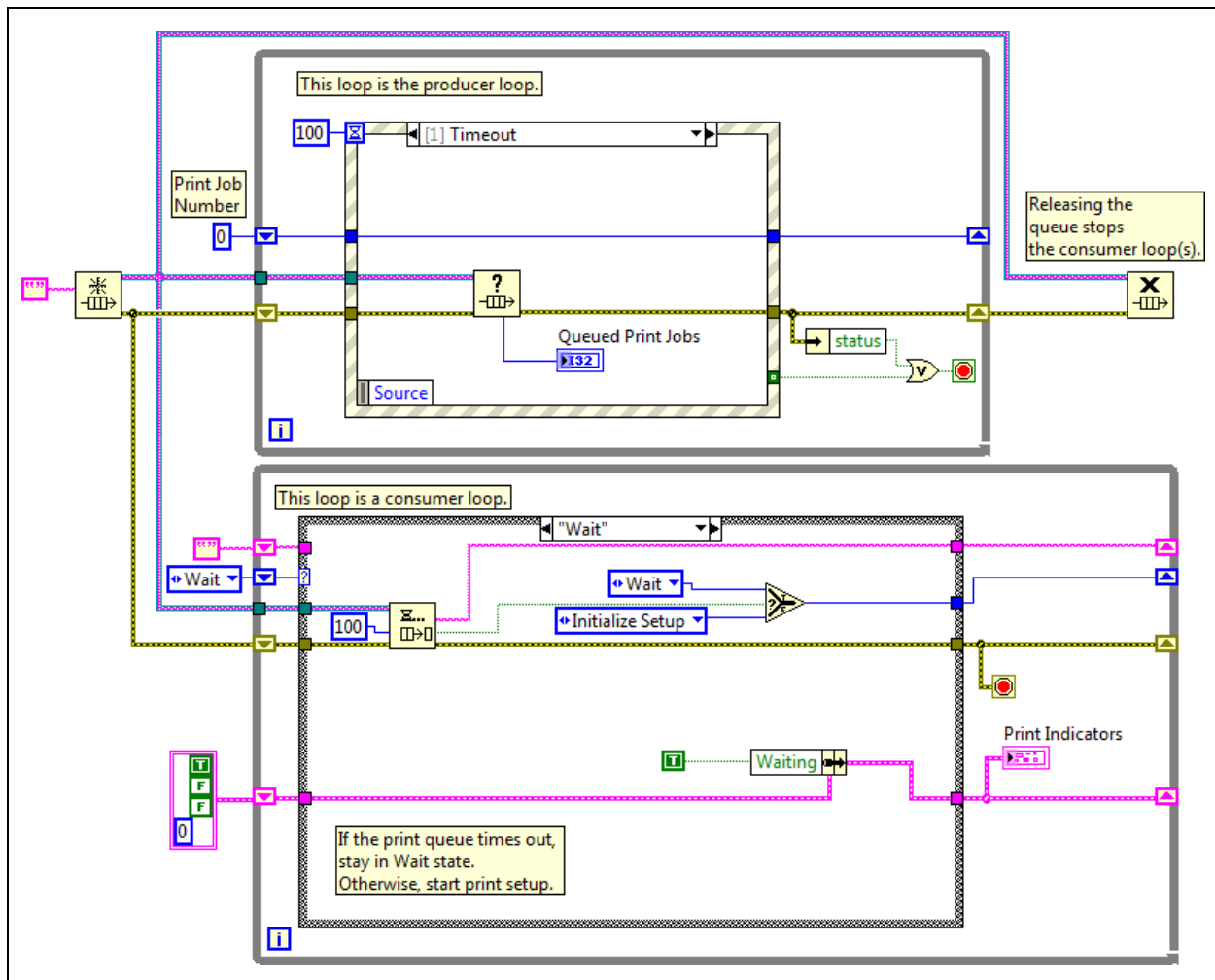1. Modify the block diagram as shown in Figure 2 to report errors.

Figure 2. Block Diagram with Error Handling in Consumer Loop

❑ Wire the error cluster of the Obtain Queue function to the consumer While Loop.

   ❑ Replace the error tunnel on the While Loop with a shift register.

   ❑ Wire the error cluster from the left shift register to the **error in** input of the Dequeue Element function.

   ❑ Wire the right-side error tunnel to the right shift register.

2. In the Initialize Setup case, wire the left-side error cluster tunnel to the **error in** terminal of the Scan from String function.

3. For all the remaining cases, wire the left-side error cluster tunnel to the right-side error cluster tunnel.

Merge and Report Errors

1. Add a Merge Errors function.

2. Add a Simple Error Handler VI.

   ❑ Right-click on the type of dialog terminal of the **Simple Error Handler VI** and select **Create»Constant**.

   ❑ Select **OK Message + warnings**.

Note   By default, the Simple Error Handler VI only displays a dialog when an error occurs. To report both warnings and errors, change the dialog selection to OK Message + warnings.

3. Complete wiring the Merge Errors function and the Simple Error Handler VI as shown in Figure 3.
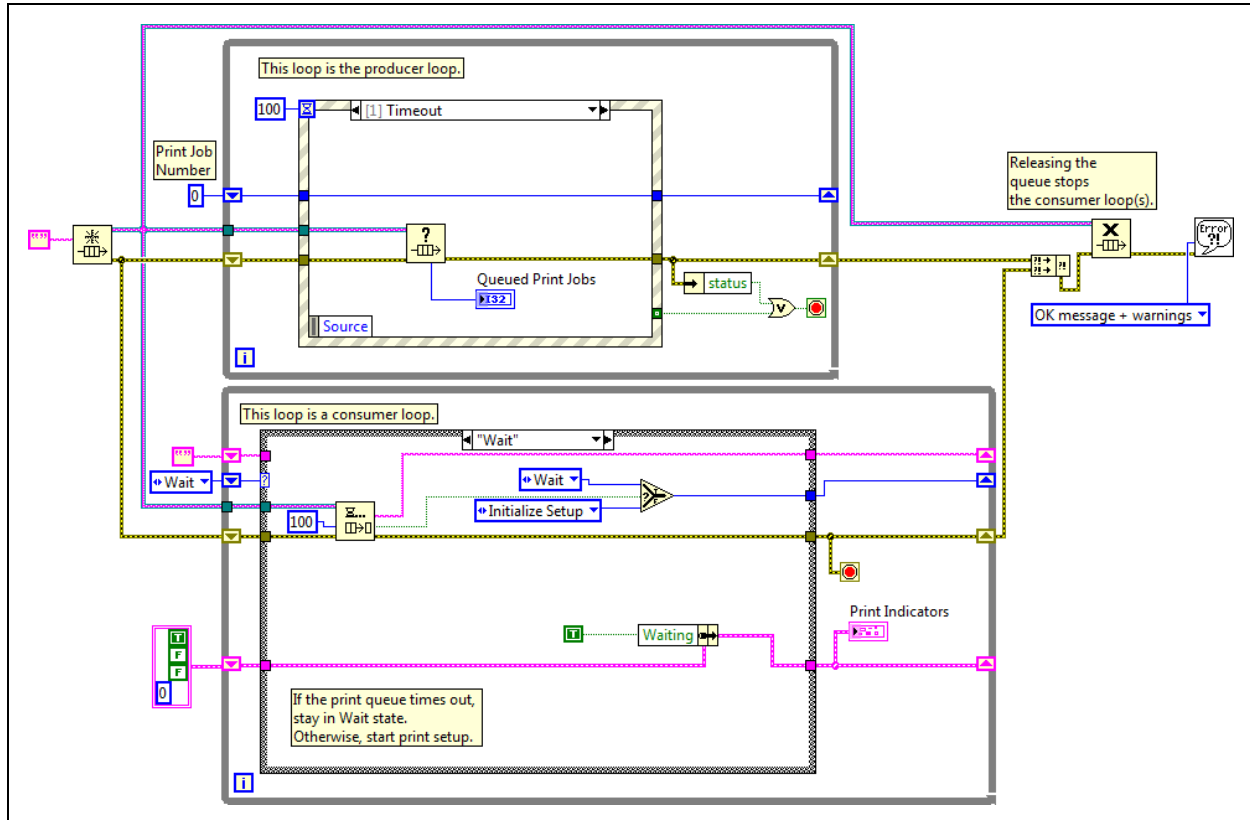

Figure 3. Block Diagram with Merge Errors Function and Simple Error Handler VI

4. Run the VI.

   ❑ Notice that the VI does not stop when you click the **Stop** button. You must stop the VI using the Abort button.

5. Examine the block diagram.

   ❑ Notice that the Merge Errors function cannot execute until both the producer and consumer loops terminate. Therefore, the Release Queue function cannot destroy the queue reference that the consumer loop relies on to stop the loop. Moving the Release Queue function before the Merge Errors function will cause the consumer loop to stop. However, relying on an error to stop the loop will cause the Simple Error Handler VI to report an error every time the VI runs. You will fix this when you modify the Shutdown state in the consumer loop.

Add an Error Handler VI

1. Modify the diagram as shown in Figure 4 so that the consumer loop executes the Shutdown case when the producer loop stops or when there is an error in the consumer loop.
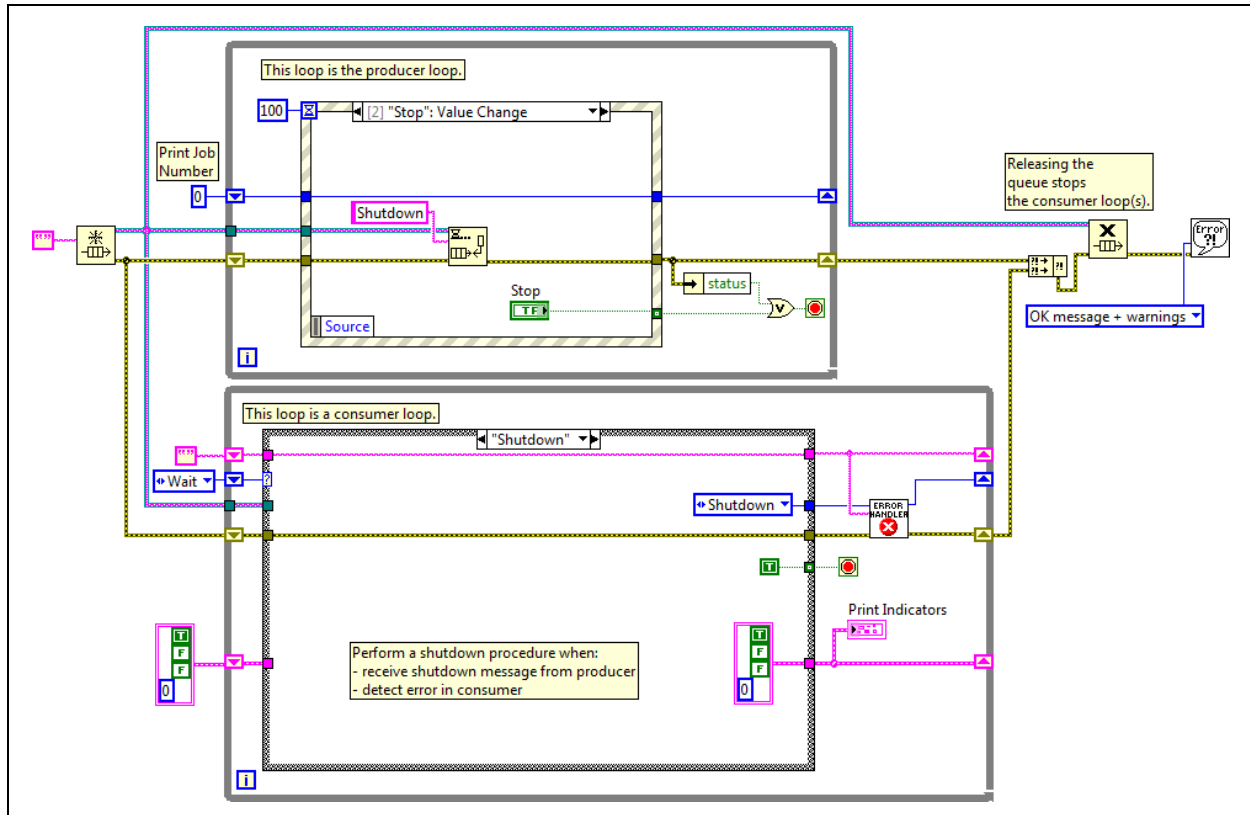

Figure 4. Block Diagram with Printer Error Handler VI

❑ Modify the producer loop to include a shutdown element at the front of the queue when the user clicks on the Stop button.

– In the Stop event case, add an **Enqueue Element At Opposite End** function.

– Right-click the **element** input of the Enqueue Element At Opposite End function and select **Create»Constant** from the shortcut menu.

– Type Shutdown in the string constant.

✎ **Note** The text in this string constant is case sensitive.

– Wire the Stop event case as shown in Figure 4.

❑ Modify the consumer loop to stop after executing the Shutdown case.

– Delete the error wire from the right-side error tunnel to the While Loop conditional terminal.

– Add a True constant to the Shutdown case diagram.

– Wire the True Constant to the While Loop conditional terminal.

– Right-click on the Boolean tunnel and select **Use Default if Unwired**.

❑ Modify the consumer loop to execute the Shutdown case when it receives the Shutdown element in the queue.

– Drag the **Printer Error Handler VI** from the subvis folder of the Print Console with Error Handling LabVIEW project and place it in the consumer loop.

– Wire the consumer loop as shown in Figure 4.

2. Double-click the **Printer Error Handler VI** to open the VI.

3. Examine the block diagram of the Printer Error Handler VI.

❑ Notice that the Next State is the same as the State In, except when the error in cluster contains an error or the Queue Data value is equal to Shutdown. If there is an error or a Shutdown match, the next state is Shutdown.

4. Close the Printer Error Handler VI.

5. Switch to the front panel of the Print Console VI.

6. Save and run the Print Console VI.

❑ Notice that the VI stops when you click the Stop button and no error is reported.

❑ Although the VI appears to work as you expect, you should force errors and warnings to test your error handling.

Test (Optional)

Time permitting, try inserting errors and warnings in your code and running the VI to test the error handling. You can force errors by breaking one of the error wires and replacing it with an Error Code Constant that has the status and code values set for an error or warning.

Do not save your VIs with forced errors.

End of Exercise

Notes