

Exercise 5-1: Inject Errors

Goal

Test how an application handles an error by injecting an error.

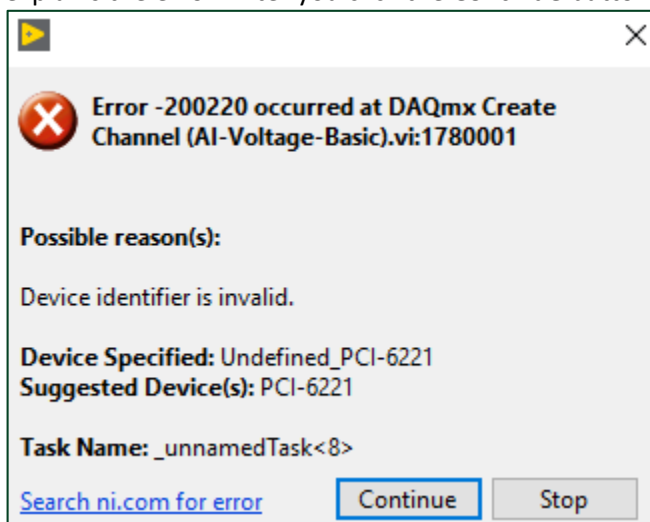
Hardware Setup:

(Hardware) In the exercises where we work with Analog Input/Output channels, we use PCI-6221/USB-6212 multifunction I/O device paired with the BNC-2120 shielded connector block. Analog Input 2 should be connected to the Sine/Triangle BNC connector. Analog Input 3 should be connected to the TTL Square Wave BNC connector. The Sine/Triangle waveform switch should be set to Sine.

Instructions

1. Open C:\Exercises\LabVIEW Core 2\Inject Error\Inject Error.lvproj.
2. From the **Project Explorer** window, open the Event-Driven State Machine - Inject Error VI.
3. Run the VI.
4. Double-click the **text** in the AI Voltage Channel control and set it to Undefined_PCI-6221/ai0:8.
5. Click the **Acquire** button. This will cause an error in the Acquire case of the state machine because Undefined_PCI-6221/ai0:8 does not exist in your system.
6. How does the application respond to this error?

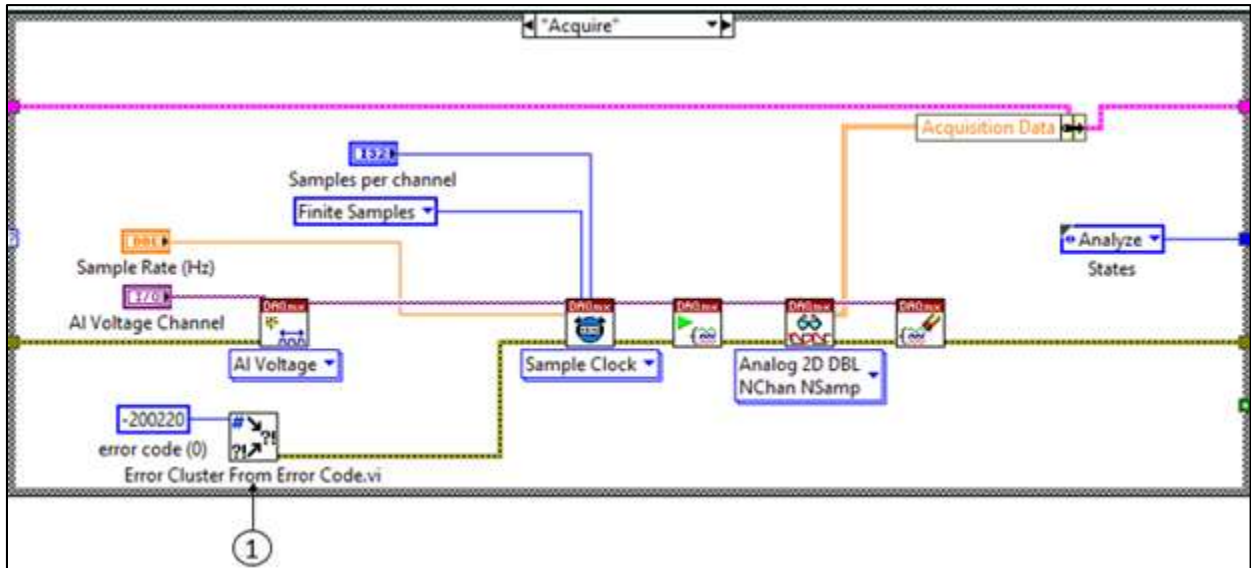
7. Notice that the application responded to the error by displaying the following error dialog, which explains the error. After you click the **Continue** button, the application stops executing and exits.



Simulate an error occurring by injecting the corresponding error cluster in the application code

If you want to test how your application handles a specific error occurring at a specific location, you can inject the corresponding error cluster at that location in your application.

1. On the block diagram, go to the Acquire case of the state diagram.



1. **Error Cluster From Error Code VI** – Right-click the **error code input** and select **Create» Constant**. Set the constant to -200220. Use the Context Help to learn the details of this VI.
2. Run the VI.
3. Click the **Acquire** button. This will cause Error -200220 in the Acquire case of the state machine because of your modifications to the code.
4. Using the Error Cluster From Error Code VI to inject an error is another way of testing how your application responds to a specific error.

On the Job

What errors do you need to inject into your application to test how your application handles errors?

End of Exercise 5-1