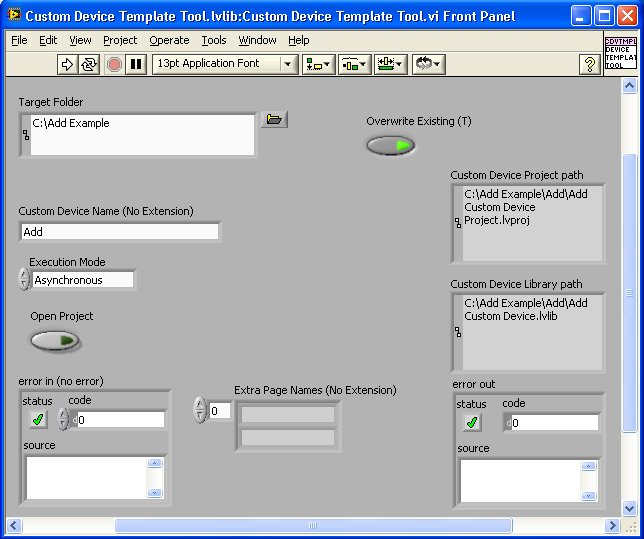
Example Walkthrough

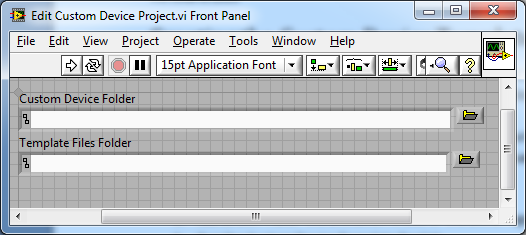
If you have not read the readme.docx, please do so first.

# Creating the Custom Device Template

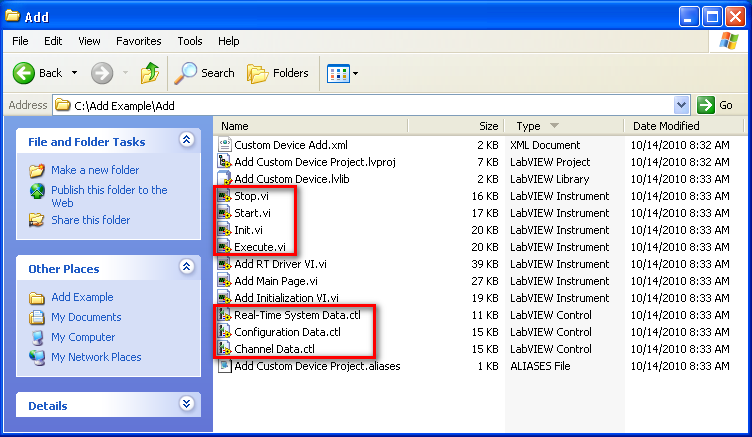
1. Open the Custom Device Template Tool VI, located in <Program Files>\National Instruments\LabVIEW 2010\vi.lib\NI VeriStand\Custom Device Tools\Custom Device Template Tool.
2. Click the **Browse** button next to the Target Folder input. Choose a folder in which to save the Custom Device. In this example, we’ll use C:\Add Example.
3. Type *Add* as the Custom Device Name.
4. Deselect **Open Project**.
5. **NOTE:** Don’t alter the Execution Mode or the Extra Page Names input.
6. Run the Custom Device Template Tool VI.



1. When the VI finishes executing, close the VI.
2. Open the Edit Custom Device Project VI inside the Edit Project Folder, which you downloaded from this article.
3. Click **Browse** next to the Custom Device Folder input. Choose the Custom Device folder you created with the Custom Device Template Tool VI.
4. Click **Browse** next to the Template Files Folder input. Choose the Template Files folder you downloaded from this article.



1. Run the Edit Custom Device Project VI.
2. When the VI finishes executing, close the VI.
3. Navigate to the Custom Device Folder you specified, in this example C:\Add Example\Add.
4. There will be 14 files in this folder. **We will be editing the Init, Start, Execute, and Stop VIs, and the Real-Time System Data, Configuration Data, and Channel Data controls.**

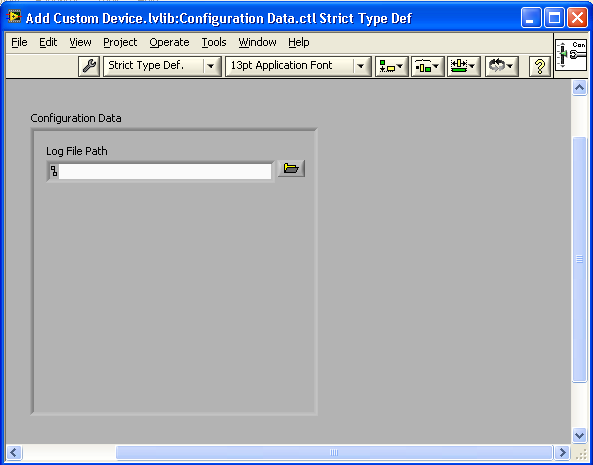


# Setting up the Strict Type Defined Controls

## Configuration Data

This cluster contains the data that you want to the user to configure from the NI VeriStand System Explorer. It will be displayed in the System Explorer for the user to configure. Items in this cluster could include run time settings like file paths, hardware names, and so on.

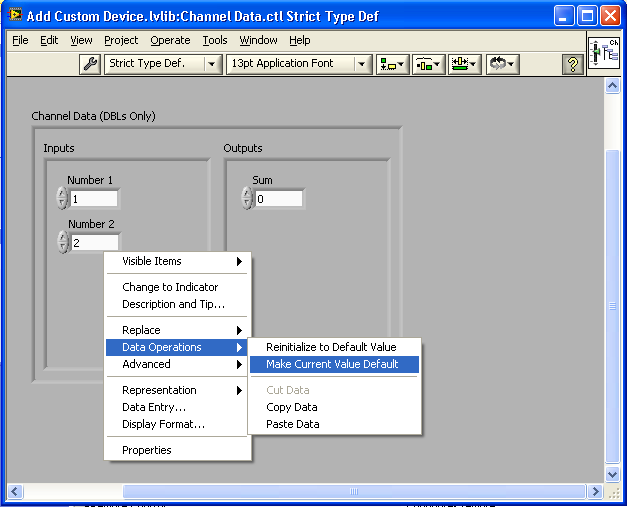
1. Open Configuration Data.ctl.
2. Add a File Path control to the cluster. Name this *Log File Path*.
3. Save the control, and close it.



## Channel Data

This cluster contains the data that the NI VeriStand engine must access as channels. It contains a cluster of Inputs and a cluster of Outputs. The controls in this cluster are added as channels in the System Explorer under the custom device. They represent the inputs and outputs of the custom device at run-time.

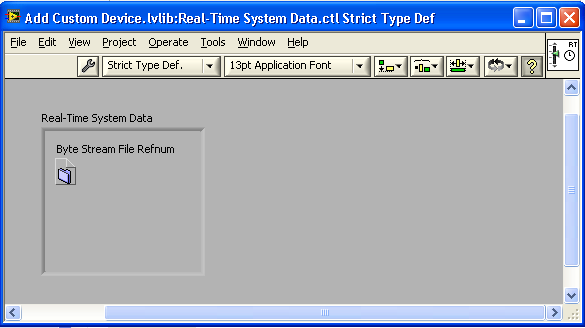
1. Open the Channel Data.ctl.
2. Add two Numeric Controls to the Inputs cluster. Name these *Number 1* and *Number 2*.
3. The default values of the channels are set in the Channel Data control. Set the default values of *Number 1* and *Number 2* to 1 and 2.
4. Add one Numeric Control to the Outputs cluster. Name this *Sum*.
5. **NOTE:** All of the data contained in the Channel Data control **must be Double data**.
6. Save the control, and close it.



## Real-Time System Data

This cluster contains data or references that are used during the execution of the VeriStand engine. Place any references that you need to use at run-time, as such file references or DAQmx tasks.

1. Open the Real-Time System Data.ctl.
2. Add Byte Stream File Refnum, from the Refnum palette, to the cluster.
3. Save the control, and close it.

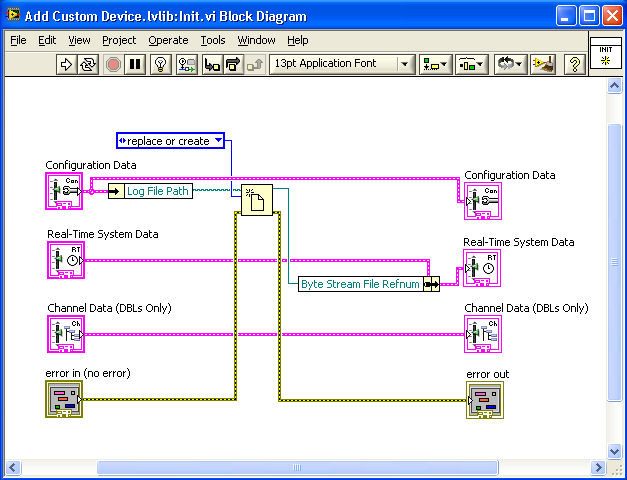


# Setting up the Custom Device VIs

## Init VI

Once the custom device is deployed and running on the VeriStand engine, this is the first VI to execute. It should initialize values, set up tasks, or open references.

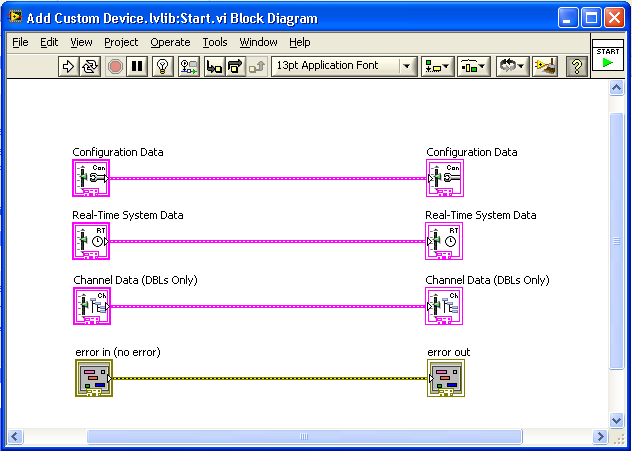
1. Open the Init VI. **NOTE:** Do not edit the connector pane of this VI.
2. Press CTRL+E to switch to the block diagram.
3. Add an Unbundle By Name operation to the block diagram.
4. Wire the Configuration Data cluster to the Unbundle By Name.
5. Add an Open/Create/Replace File VI.
6. Wire the Log File Path to the *file path* input Open/Create/Replace File VI.
7. Create a Constant for the *operation* input, and select **replace or create**.
8. Add a Bundle By Name operation.
9. Wire the Real-Time System Data cluster to the *input cluster* input of the Bundle By Name.
10. Wire the *refnum out* output of the Open/Create/Replace File VI to the Byte Stream File Refnum input of the Bundle By Name.
11. Wire the *output cluster* output to the Real-Time System Data indicator.
12. Pass the Error Cluster through the Open/Create/Replace File VI.
13. Pass the Configuration Data and Channel Data through the VI.
14. Save the VI, and close it.



## Start VI

This VI executes before the looping structure of the custom device. It should start any tasks that must begin before the loop.

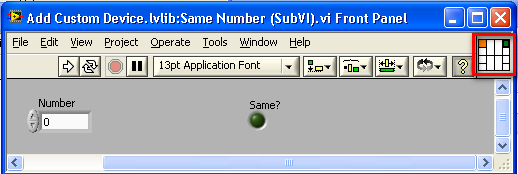
1. We do not need to add anything to this VI in this example; however, we do need to pass the data through the VI.
2. Open the Start VI.
3. Press CTRL+E to switch to the block diagram.
4. Pass the Configuration Data, Real-Time System Data, Channel Data, and Error Cluster through the VI.
5. Save the VI, and close it.



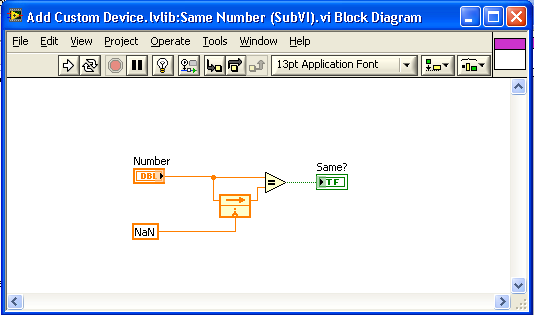
## Sub VIs

If you will be using sub-VIs in your custom device, the sub-VIs need to be added to the Custom Device library created for your custom device. Adding the sub-VIs to this library includes the VI in the libraries namespace, allowing VIs of the same name to be used across multiple custom devices.

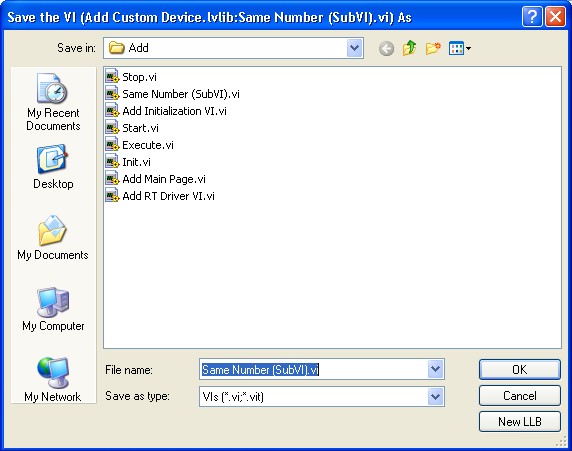
1. Open the Add Custom Device Project.
2. Right-click Add Custom Device.lvlib and select **New>>VI**.
3. Add a Numeric Control to the front panel, and name it *Number*.
4. Add a Round LED Boolean Indicator to the front panel, and name it *Same?*.
5. Right-click on the VI Icon and select **Show Connector**.
6. Connect the terminals as show below.



1. Press CTRL+E to switch to the block diagram.
2. Add an Equal? function, a NaN constant, and Feedback Node.
3. Change the direction of the Feedback Node by right-clicking it and selecting **Change Direction**.
4. Wire the block diagram as shown below.
5. Press CTRL+I to open VI properties. Select Execution from the drop down box. Check Reentrant execution. Close VI Properties.



1. Select **File>>Save As**, and save the VI as *Same Number (SubVI)* in the Add folder.

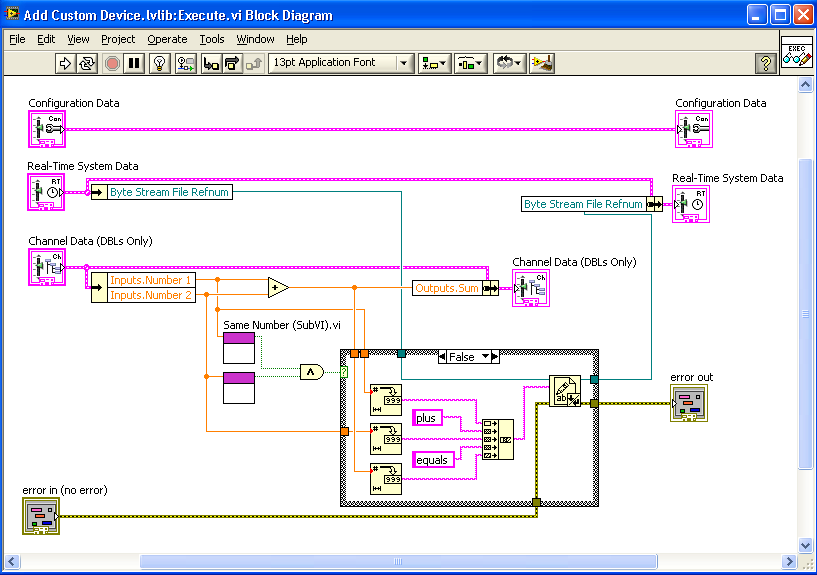


1. Close the Same Number (SubVI) VI.
2. Save and close the Add Custom Device Project.
3. If prompted to save any components of the project, select **Save**.

## Execute VI

This VI executes within the looping structure of the custom device. Each iteration, the latest value of the custom device input channels are provided to this subVI in the channel data cluster. The Execute VI can write these input values to hardware, log them to file, or perform other operations on the data. After the Execute VI finishes, the output values in the channel data cluster are used to update the output channels of the custom device.

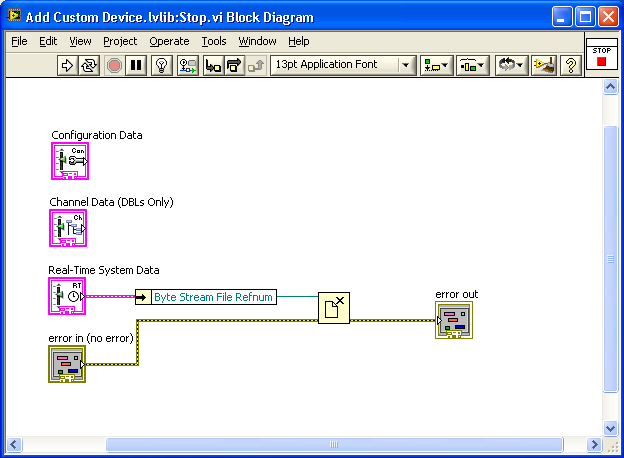
1. Open the Execute VI.
2. Press CTRL+E to switch to the block diagram.
3. Add an Unbundle By Name operation to the block diagram.
4. Wire the Real-Time System Data cluster to the Unbundle By Name.
5. Add another Unbundle By Name operation to the block diagram.
6. Wire the Channel Data cluster to the second Unbundle By Name.
7. Expand the second Unbundle By Name to create two outputs.
8. Select **Inputs>>Number 1** and **Inputs>>Number 2** as the outputs of the Unbundle By Name.
9. Wire Number 1 and Number 2 to an Add function.
10. Add a Bundle By Name to the block diagram, and connect the Channel Data cluster to the *input cluster*.
11. Select **Outputs>>Sum**.
12. Wire the sum from the Add function to the input of the Bundle By Name.
13. Wire the *output cluster* to the Channel Data indicator.
14. Add two instances of the Same Number (SubVI) to the block diagram.
15. Wire Number 1 to the *Number* input of one subVI, and wire Number 2 to the other subVI.
16. Wire the *Same?* outputs of the subVIs to an And function.
17. Add a Case Structure to the block diagram.
18. Wire the output of the And function to the selector terminal of the Case Structure.
19. Wire Number 1, Number 2, Sum, Byte Stream File Refnum, and the error cluster to the Case Structure.
20. Add three Number to Decimal String functions to the False case.
21. Add a Concatenate Strings function with five inputs to the False case.
22. Add a Write to Text file function to the False Case.
23. Complete the wiring as shown below, passing the *refnum out* and *error out* through the Case Structure.
24. Switch to the True case.
25. Wire the Byte Stream File Refnum and error cluster through the case.
26. Add a Bundle By Name to the block diagram, and connect the Real-Time System Data to the *input cluster*.
27. Wire the *refnum out* from the Case Structure to the Byte Stream File Refnum input of the Bundle By Name.
28. Wire the *output cluster* to the Real-Time System Data indicator.
29. Pass the Configuration Data through the VI.
30. Save the VI, and close it.



## Stop VI

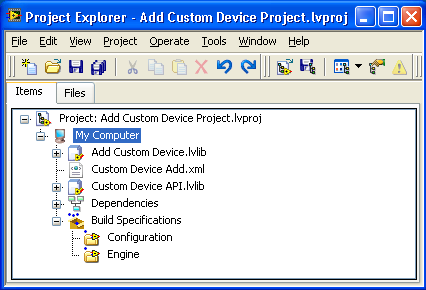
This VI executes when the custom device is stopped. It should close references, resolve errors, and perform shut-down tasks.

1. Open the Stop VI.
2. Press CTRL+E to switch to the block diagram.
3. Add an Unbundle By Name operation to the block diagram.
4. Wire the Real-Time System Data cluster to the Unbundle By Name.
5. Add a Close File function to the block diagram.
6. Wire the Byte Stream File Refnum and the error cluster to the Close File function.
7. Wire the *error out* from the Close File function to the error out indicator.



# Building the Custom Device

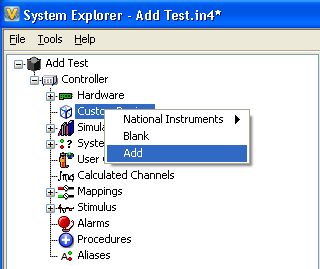
1. Open the Add Custom Device project explorer.



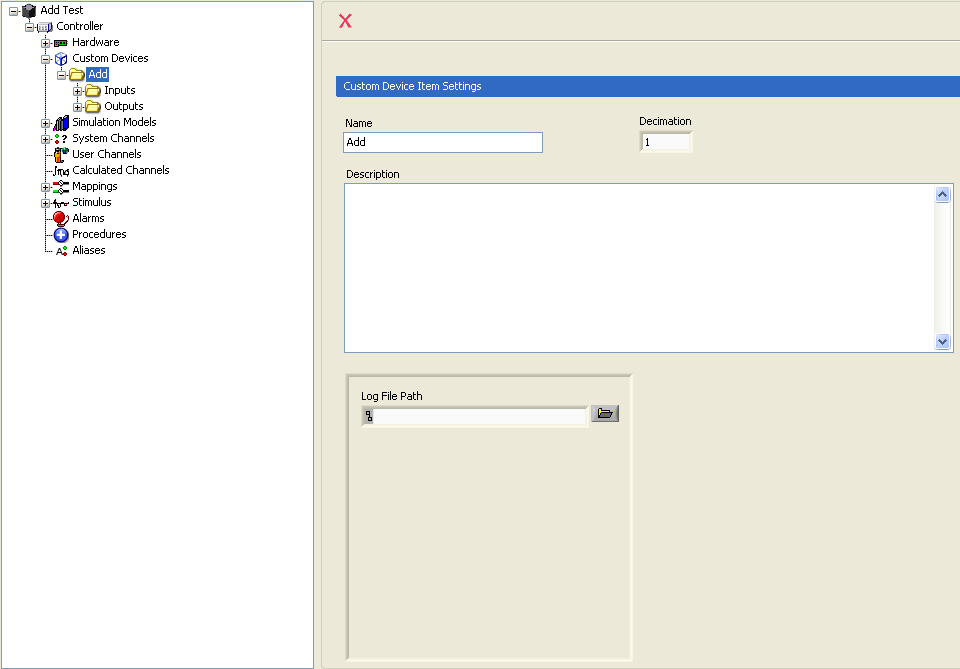
1. Right-click on the **Build Specification** section and select **Build All**.
2. When the build is complete, select **Done** to close the Build status window.

# Using the Custom Device

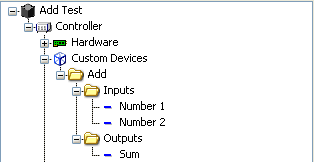
1. Open NI VeriStand 2010.
2. On the Getting Started screen, select **New NI VeriStand Project**.
3. Change the Project Name to *Add Test*.
4. Select **OK** to close the window.
5. From the project explorer, launch the System Explorer by expanding the system definition file section and double clicking the system definition file.
6. In the System Explorer, right-click the **Custom Devices** item and select the **Add** custom device.



1. Click on the Add folder now under Custom Devices, and you should see the main page to the right. It includes Name and Description fields for the custom device, a Decimation value, and the Configuration Data cluster you created.



1. The Decimation value defines the rate at which the custom device executes, as a decimation of the rate of the Primary Control Loop. Set this value to 2 to execute at half the speed of the Primary Control Loop.
2. In the Log File Path input, type C:\Add Example\Log.txt.
3. Expand the Inputs and Outputs folder under the Add custom device, and you will see the inputs and outputs you created in the Channel Data cluster.



1. Save and exit the System Explorer.
2. Select the **Operate** and then **Run**.
3. Drop two numeric controls and map them to the custom device inputs.
4. Drop one numeric indicator and map it to the custom device outputs.
5. Change the numeric control values and notice the indicator reflects the sum of the inputs.
6. Open the log file and view the log you are creating.