

# LabVIEW™ Core 1 Exercises

Course Software Version 2012  
August 2012 Edition  
Part Number 325291D-01

## Copyright

© 1993–2012 National Instruments. All rights reserved.

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

National Instruments respects the intellectual property of others, and we ask our users to do the same. NI software is protected by copyright and other intellectual property laws. Where NI software may be used to reproduce software or other materials belonging to others, you may use NI software only to reproduce materials that you may reproduce in accordance with the terms of any applicable license or other legal restriction.

## End-User License Agreements and Third-Party Legal Notices

You can find end-user license agreements (EULAs) and third-party legal notices in the following locations:

- Notices are located in the <National Instruments>\\_Legal Information and <National Instruments> directories.
- EULAs are located in the <National Instruments>\Shared\MDF\Legal\license directory.
- Review <National Instruments>\\_Legal Information.txt for more information on including legal information in installers built with NI products.

## Trademarks

LabVIEW, National Instruments, NI, ni.com, the National Instruments corporate logo, and the Eagle logo are trademarks of National Instruments Corporation. Refer to the *Trademark Information* at [ni.com/trademarks](http://ni.com/trademarks) for other National Instruments trademarks.

Other product and company names mentioned herein are trademarks or trade names of their respective companies.

## Patents

For patents covering National Instruments products/technology, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your media, or the *National Instruments Patent Notice* at [ni.com/patents](http://ni.com/patents).

Sample

Sample

**Worldwide Technical Support and Product Information**  
ni.com

**Worldwide Offices**

Visit [ni.com/niglobal](http://ni.com/niglobal) to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

**National Instruments Corporate Headquarters**

11500 North Mopac Expressway Austin, Texas 78759-3504 USA Tel: 512 683 0100

To comment on National Instruments documentation, refer to the National Instruments Web site at [ni.com/info](http://ni.com/info) and enter the Info Code feedback.

Sample

# Contents

---

## Student Guide

A. NI Certification.....	vii
B. Course Description.....	viii
C. What You Need to Get Started.....	ix
D. Installing the Course Software.....	x
E. Course Goals.....	xi
F. Course Conventions.....	xii

## Lesson 1

### Navigating LabVIEW

Exercise 1-1	Concept: Exploring a VI.....	1-3
Exercise 1-2	Concept: Locating Controls, Functions, and VIs.....	1-8
Exercise 1-3	Concept: Using Tool Selection.....	1-11
Exercise 1-4	Concept: Dataflow.....	1-19
Exercise 1-5	Simple AAP VI.....	1-24

## Lesson 2

### Troubleshooting and Debugging VIs

Exercise 2-1	Concept: Debugging.....	2-2
--------------	-------------------------	-----

## Lesson 3

### Implementing a VI

Exercise 3-1	Temperature Warnings VI.....	3-2
Exercise 3-2	Auto Match VI.....	3-11
Exercise 3-3	Concept: While Loops versus For Loops.....	3-21
Exercise 3-4	Temperature Monitor VI—Average Temperature.....	3-25
Exercise 3-5	Temperature Monitor VI—Plot Multiple Temperatures.....	3-29
Exercise 3-6	Temperature Warnings VI—With Error Handling.....	3-35

**Lesson 4****Developing Modular Applications**

Exercise 4-1	Temperature Warnings VI—As SubVI .....	4-3
--------------	--	-----

**Lesson 5****Creating and Leveraging Structures**

Exercise 5-1	Concept: Manipulating Arrays.....	5-3
Exercise 5-2	Concept: Clusters.....	5-15
Exercise 5-3	Concept: Type Definition .....	5-23

**Lesson 6****Managing File and Hardware Resources**

Exercise 6-1	Concept: Spreadsheet Example VI .....	6-3
Exercise 6-2	Temperature Monitor VI—Logging Data.....	6-7
Exercise 6-3	Concept: NI Measurement & Automation Explorer (MAX).....	6-13
Exercise 6-4	Using DAQmx .....	6-19
Exercise 6-5	Concept: GPIB Configuration with MAX.....	6-23
Exercise 6-6	Concept: NI Devsim VI .....	6-27

**Lesson 7****Using Sequential and State Machine Algorithms**

Exercise 7-1	State Machine VI .....	7-3
--------------	------------------------	-----

**Lesson 8****Solving Dataflow Challenges Using Variables**

Exercise 8-1	Weather Station UI VI with Local Variables.....	8-3
--------------	---	-----

---

# Creating and Leveraging Structures

Complete the exercises in this lesson to reinforce the concepts you learn in Lesson 5 of LabVIEW Core 1.

## Exercises

---

- Exercise 5-1    Concept: Manipulating Arrays
- Exercise 5-2    Concept: Clusters
- Exercise 5-3    Concept: Type Definition

Sample

Sample



## Exercise 5-1      Concept: Manipulating Arrays

### Goal

Manipulate arrays using various LabVIEW functions.

### Description

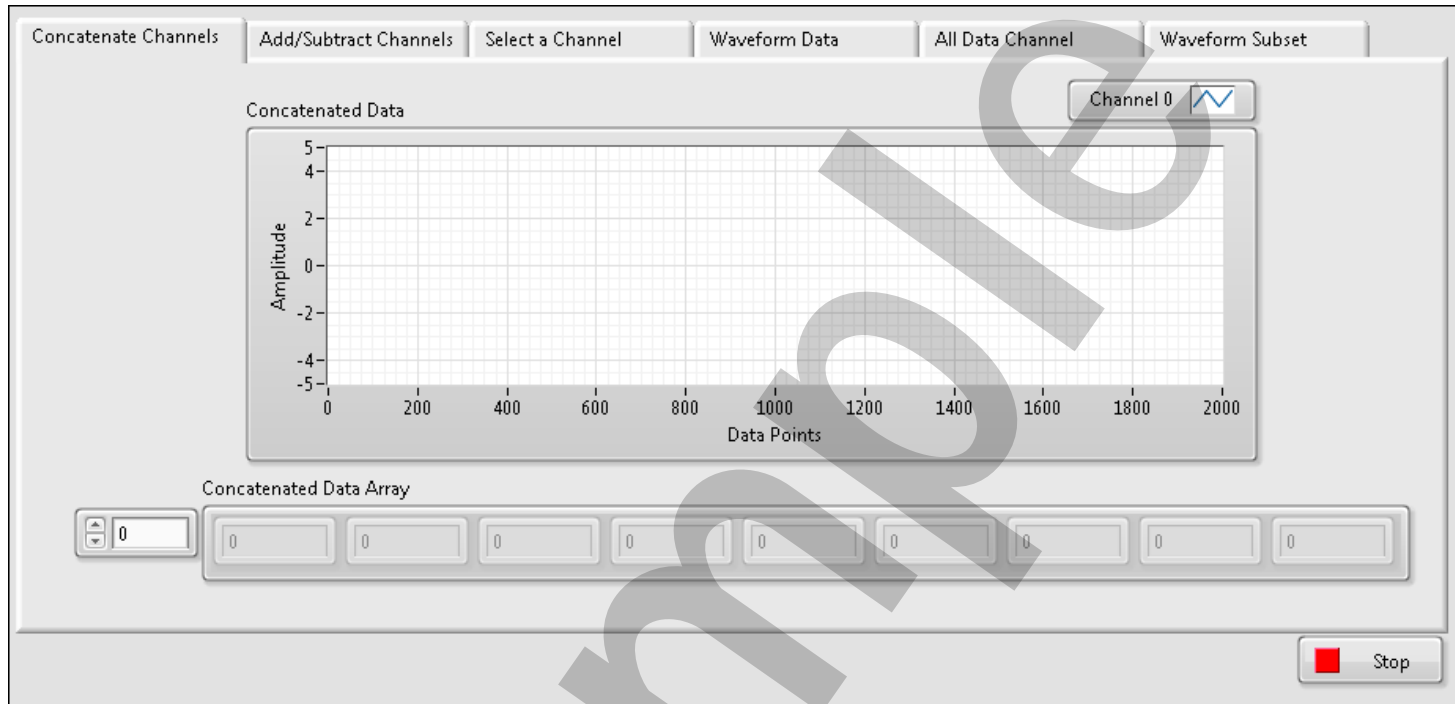
You are given a VI and asked to enhance it for a variety of purposes. The front panel of this VI is built. You complete the block diagram to practice several different techniques to manipulate arrays.

### Implementation

1. Open `Manipulating Arrays.lvproj` in the `<Exercises>\LabVIEW Core 1\Manipulating Arrays` directory.
2. Open **Array Manipulation** VI from the **Project Explorer** window. The front panel, shown in Figure 5-1, is already built for you.

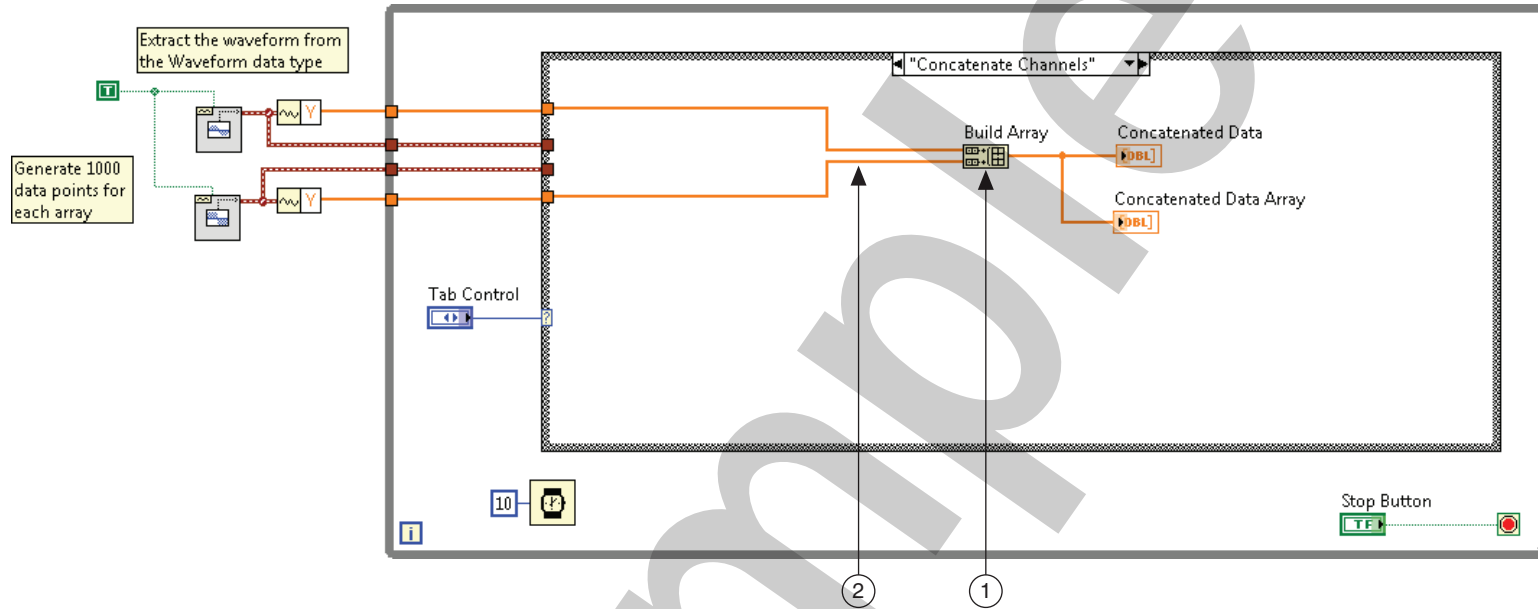
Sample

Figure 5-1. Array Manipulation VI Front Panel



3. Open the block diagram and complete each of the cases that correspond to the tabs on the front panel as shown in Figures 5-2 through 5-8.

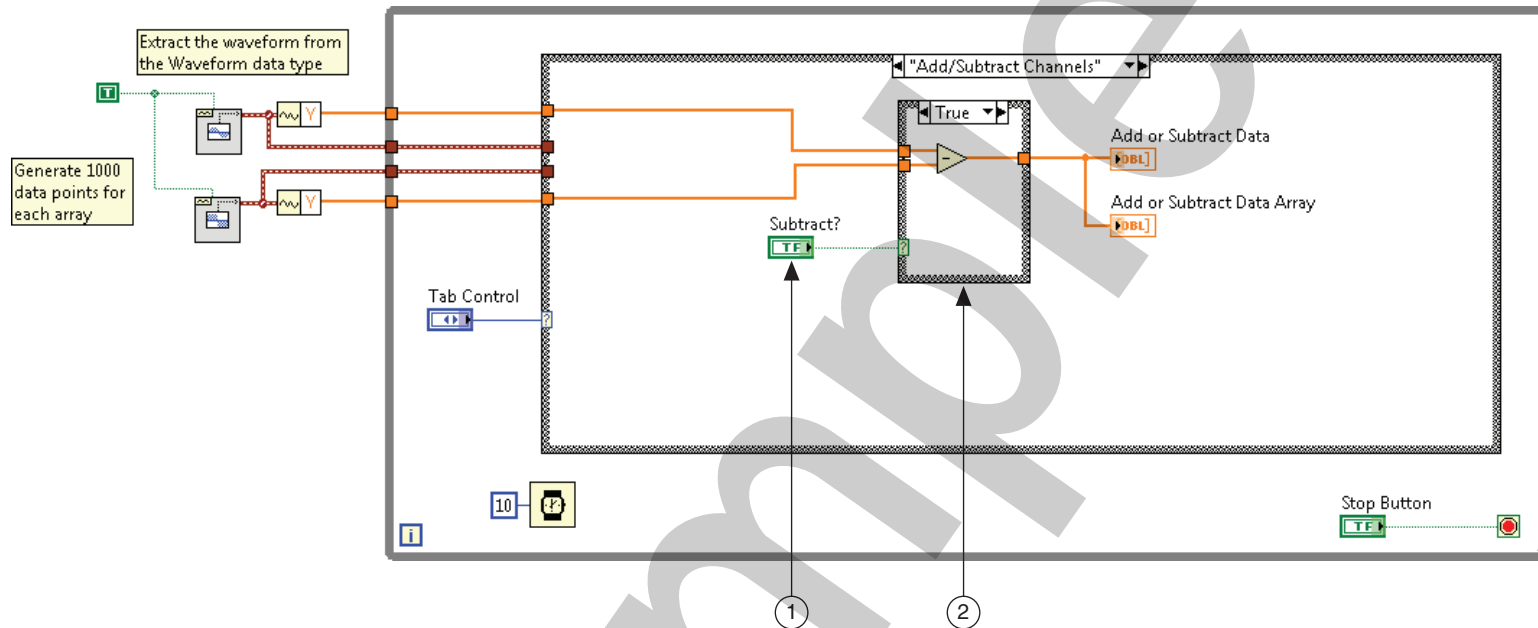
**Figure 5-2.** Array Manipulation VI—Concatenate Channels Case



- 1 **Build Array**—Expand this node to accept two inputs, and then right-click and select **Concatenate inputs** from the shortcut menu.
- 2 Wire the sine wave and square wave outputs to the Build Array function to create a 1D array with both waveforms.
4. Switch to the front panel and test the Concatenate Channels case.
  - On the front panel, click the **Concatenate Channels** tab.
  - Run the VI and notice that the sine wave is concatenated with a square wave.
5. Stop the VI.
6. Switch to the block diagram and select the Add/Subtract Channels case.

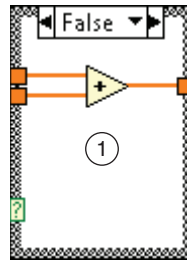
7. Complete the Add/Subtract Channels case as shown in Figure 5-3 and Figure 5-4.

**Figure 5-3.** Array Manipulation VI—Add/Subtract Channels True Case




- 1 **Subtract?**—Wire this to the case selector terminal so that the correct case executes when you click the Subtract? button on the front panel.
- 2 **Case Structure**—Place a Subtract function in the True case, so that the VI subtracts the elements of the array when the Subtract? button on the front panel is pressed.

**Figure 5-4.** Array Manipulation VI—Add/Subtract Channels False Case



1 When the value of the Subtract? Boolean control is False, the array elements are added.

 **Note** This case demonstrates polymorphic functionality by adding and subtracting elements of the array.

8. Switch to the front panel and test the Add/Subtract Channels case.

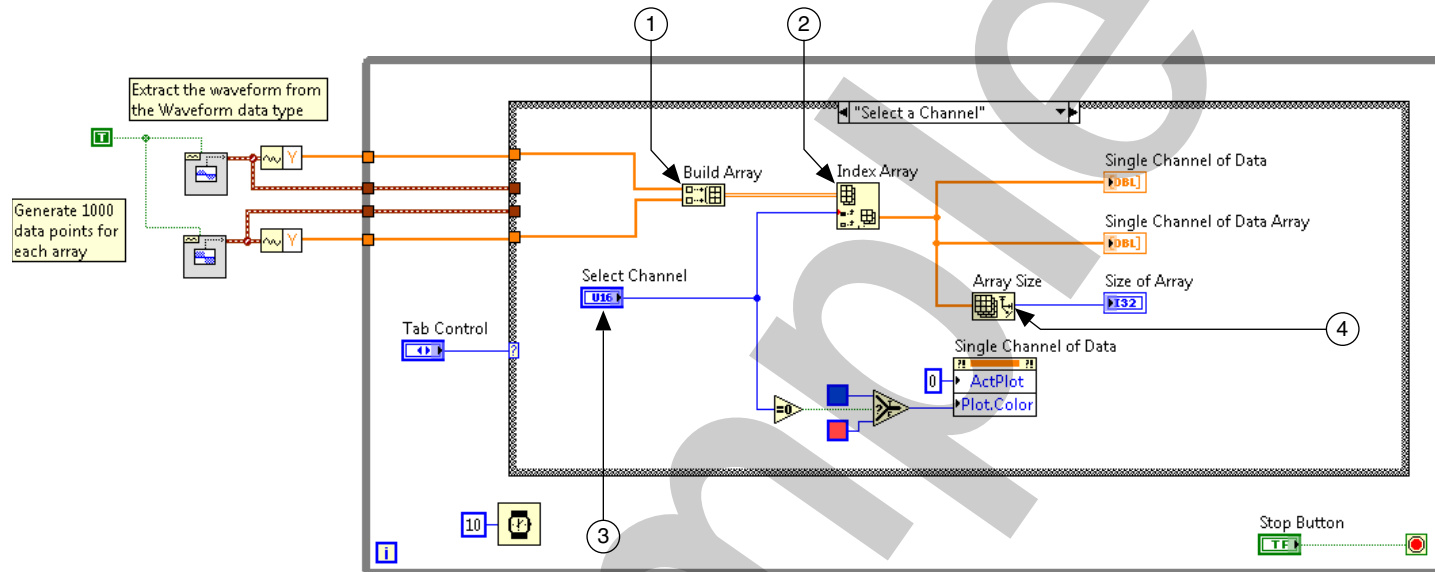
- On the front panel, click the **Add/Subtract Channels** tab.
- Run the VI.
- Click the **Subtract?** button and observe the behavior of subtracting the square wave from the sine wave.

9. Stop the VI.


10. Switch to the block diagram and select the Select a Channel case.

11. Complete the Select a Channel case as shown in Figure 5-5.

**Figure 5-5.** Array Manipulation VI—Select a Channel



- 1 **Build Array**—Combines the sine and square waves into one 2D array.
- 2 **Index Array**—Extracts row 0 or 1 from the 2D array. The output from this function is a 1D array and is the waveform you select with the **Select Channel** control. The waveform is displayed on the **Single Channel of Data Waveform Graph** and the **Single Channel of Data Array** indicator.
- 3 **Select Channel**—Wire to the **row** input of the Index Array function.
- 4 **Array Size**—Because you are using a 1D array, this function outputs a scalar value.

 **Note** The Select a Channel case uses a property node to change the color of the graph plot. You learn about Property Nodes *LabVIEW Core 2*.

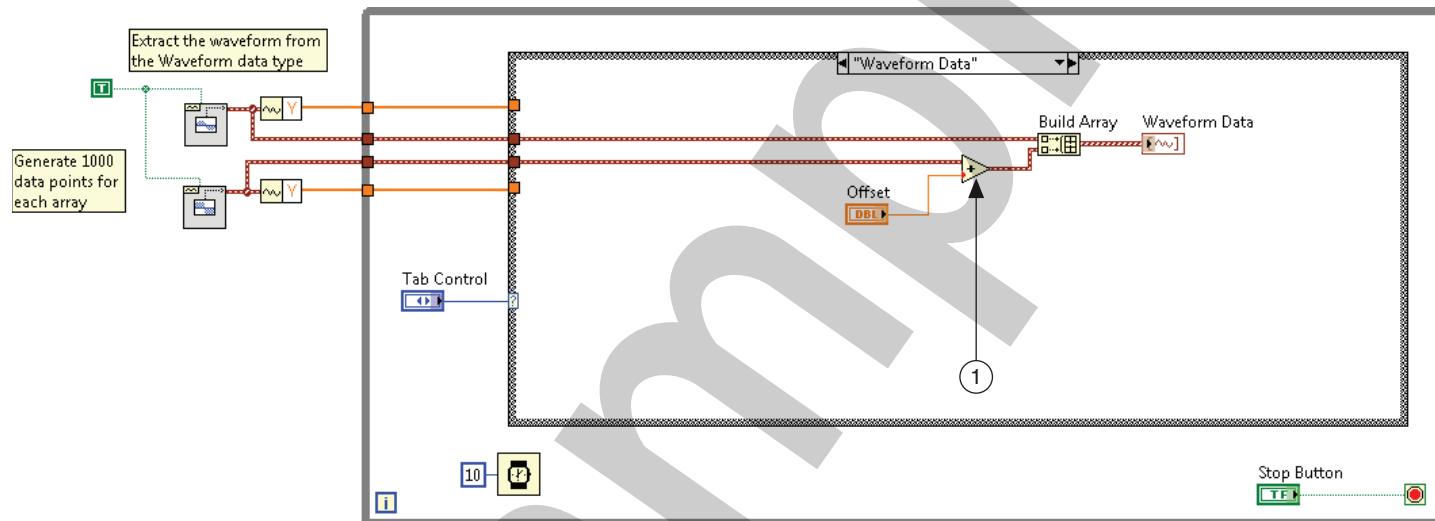
12. Switch to the front panel and test the Select a Channel case.

- On the front panel, click the **Select a Channel** tab.
- Run the VI.
- Switch between Channel 0 and Channel 1 and notice the different values shown in the **Single Channel of Data Array** indicator.

13. Stop the VI.
14. Switch to the block diagram and select the Waveform Data case.
15. Complete the Waveform Data case block diagram as shown in Figure 5-6.

The waveform datatype is a special kind of cluster that contains additional timing information about the waveform.

**Figure 5-6.** Array Manipulation VI—Waveform Data



- 1 **Add**—Uses the value from the **Offset** control to modify the value of the waveform in the waveform datatype. Notice the value from the **Offset** control must be coerced to be used with the waveform datatype.

**Note** Polymorphism is the ability of VIs and functions to automatically adapt to accept input data of different data types, including arrays, scalars, and waveforms. VIs and functions are polymorphic to varying degrees.

16. Switch to the front panel and test the Waveform Data case.
  - On the front panel, click the **Waveform Data** tab.
  - Run the VI.

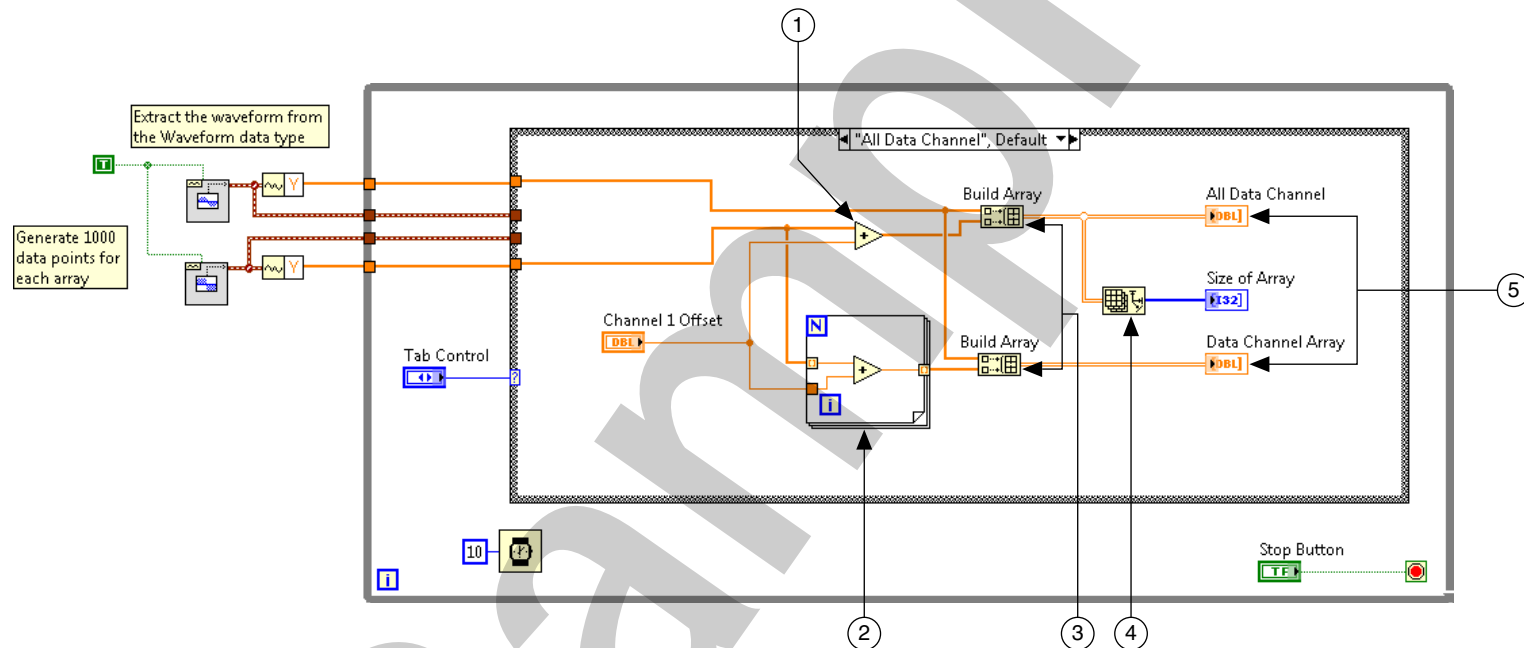
- Change the value of the **Offset** control and notice the square wave move on the **Waveform Data** chart.

17. Stop the VI.

18. Switch to the block diagram and select the All Data Channel case.

19. Complete the All Data Channel case as shown in Figure 5-7.

**Figure 5-7.** Array Manipulation VI—All Data



- 1 Add**—Modify the same data in one array by adding the value of the Channel 1 Offset to each element of the array.
- 2 For Loop**—Extracts each element of the array using auto indexing so that the Add function in the For Loop can add the scalar value.
- 3 Build Array**—Takes the two 1D arrays and builds a 2D array. Each 1D array becomes a row in the 2D array.
- 4 Array Size**—Outputs a 1D array where each element shows the size of each dimension. In this exercise, you have 2 elements of data for the number of rows and columns.
- 5 All Data Channel** and **Data Channel Array** indicators display the same data.





**Note** The polymorphic functionality of LabVIEW functions allows you to perform the same operation on each element without extracting the array elements, as you do with the two Add functions in the All Data Channel case.

20. Switch to the front panel and test the All Data Channel case.

- On the front panel, click the **All Data Channel** tab.
- Run the VI.
- Change the value of the **Channel 1 Offset** control and observe the behavior.

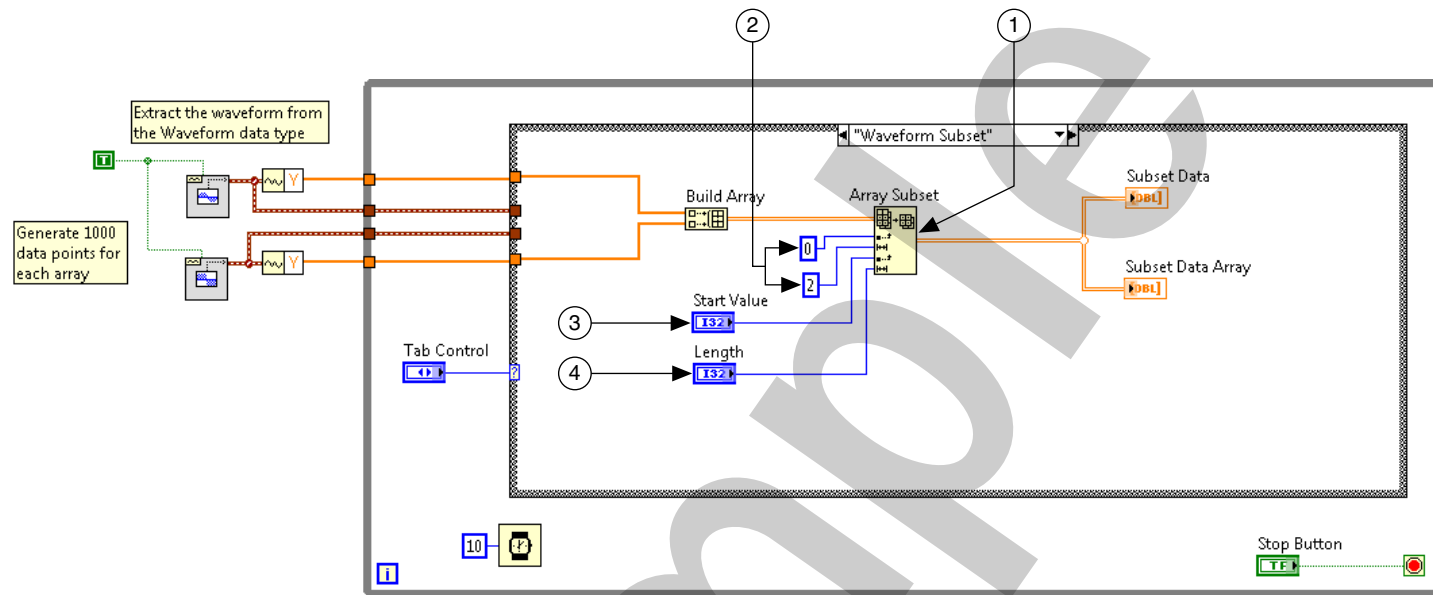
21. Stop the VI.

22. Switch to the block diagram and select the Waveform Subset case.

23. Complete the Waveform Subset case as shown in Figure 5-8.

Sample

Figure 5-8. Array Manipulation VI—Waveform Subset



- 1 **Array Subset**—Extracts a subset of an existing array. In this exercise, you use this function to zoom in on a subset of the waveform you generated.
  - 2 **Numeric Constant**—These constants specify that the function extract the first two rows starting at element 0.
  - 3 **Start Value**—Sets the start index. The default value is set to start at element 0.
  - 4 **Length**—Sets the number of elements to extract. The default value is set to output 1000 elements.
24. Switch to the front panel and test the Waveform Subset case.
    - On the front panel, click the **Waveform Subset** tab.
    - Run the VI.
    - Change value of the **Start Value** and **Length** sliders and notice that the **Subset Data** waveform graph x-axis starts at zero and finishes at the number of elements in the new array. The x-axis starts at zero because the VI creates a brand new array and the graph does not know where the data was located in the original array.
  25. Stop the VI.

## Using the NI Example Finder to Learn More about Arrays

Use the NI Example Finder to browse or search examples installed on your computer or on the NI Developer Zone at [ni.com/zone](http://ni.com/zone). Example VIs can show you how to use specific functions and programming concepts such as arrays and polymorphism.

Complete the following steps to use the NI Example finder to locate example VIs that demonstrate different ways to use the Array function.

1. Select **Help»Find Examples** to start the NI Example Finder.
2. Click the **Search** tab and enter the keyword `array`.
3. Click the **Search** button to find VIs using that keyword.
4. Click one of the example VIs in the search results list and read the description.
5. Double-click an example VI to open it.
6. Read through the comments on the front panel and block diagram to learn more about what this example VI demonstrates.
7. Run the example, examine the different cases, and click the **Stop** button to exit.
8. Close the VIs and the NI Example Finder when you are finished.

End of Exercise 5-1

Sample

## Exercise 5-2 Concept: Clusters

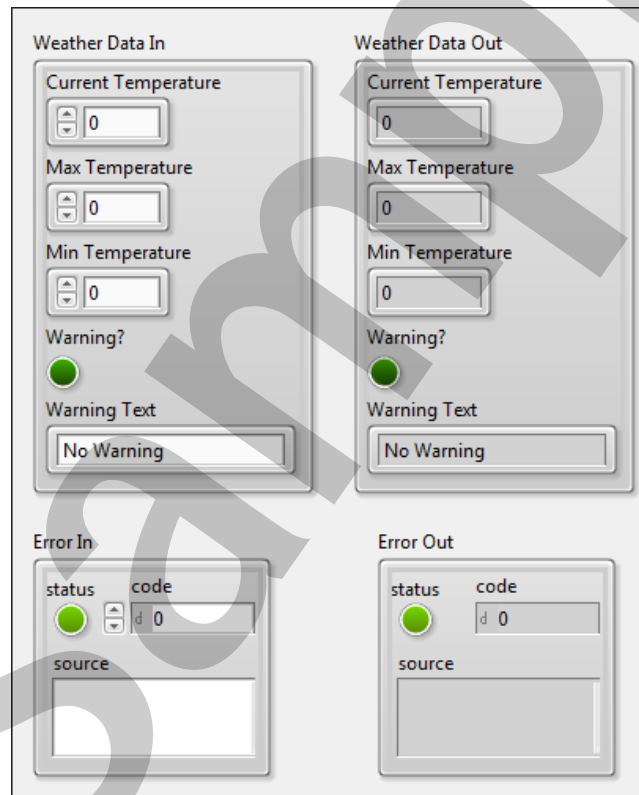
### Goal

Create a cluster datatype containing the data to be passed around an application and, in the process, create scalable, readable code.

### Description

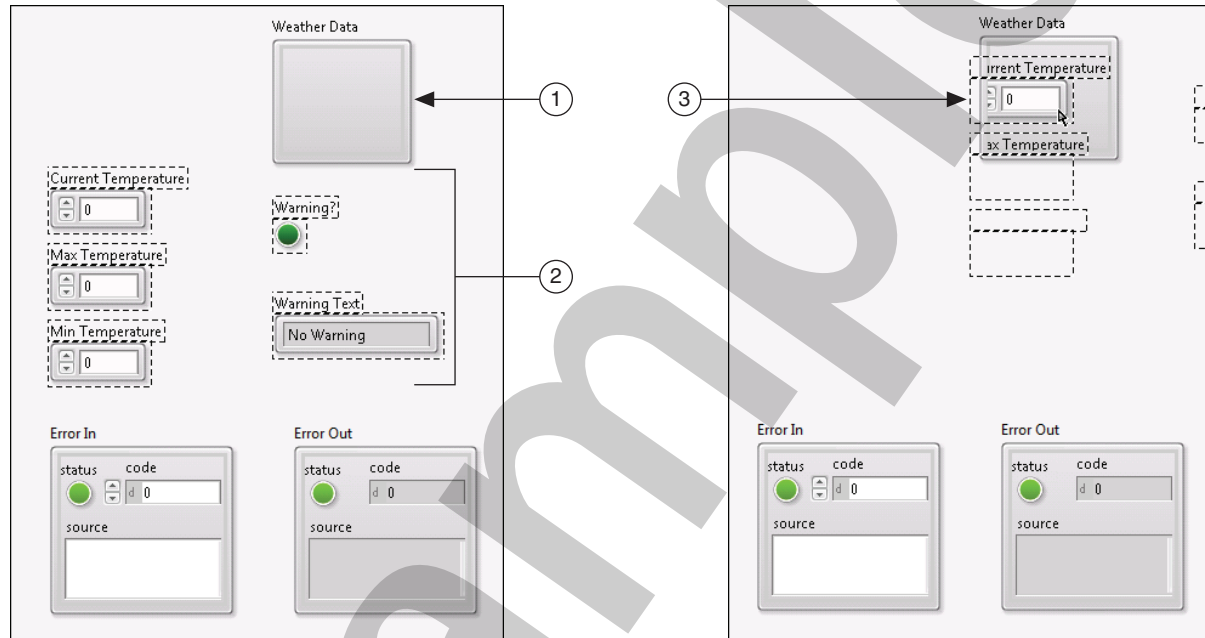
You create a cluster which contains the data used by the Temperature Warnings VI you created in Lesson 3. You modify the Temperature Warnings VI to receive and return data in the form of that same cluster as shown in Figure 5-9. The modified VI works in a more modular fashion with other subVIs in the overall application.

**Figure 5-9.** Temperature Warnings VI with Clusters Front Panel



1. Open `Weather Warnings.lvproj` in the `<Exercises>\LabVIEW Core 1\Weather Warnings` directory.
2. Open **Temperature Warnings VI** from the **Project Explorer** window.
3. Place existing controls and indicators in a cluster named `Weather Data` as shown in Figure 5-10.

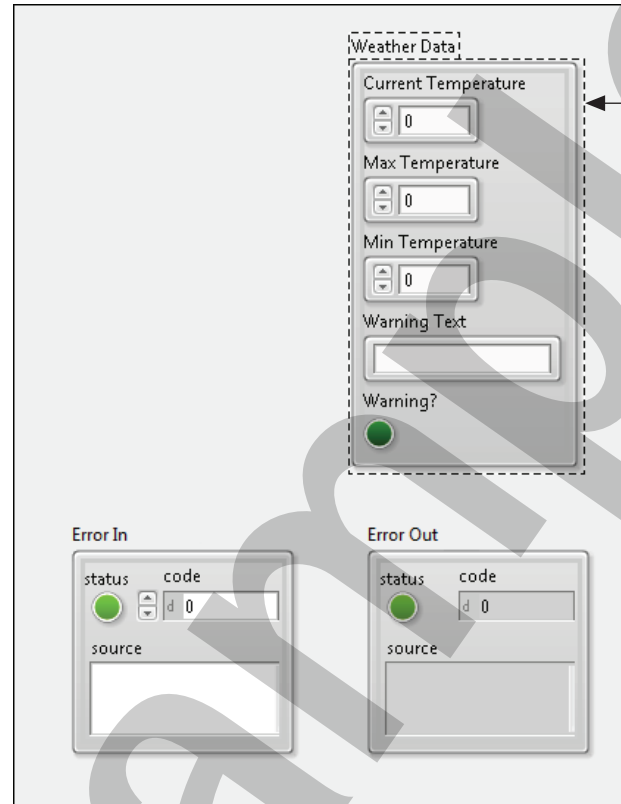
Figure 5-10. Create Cluster



- 1 **Cluster**—Use the Cluster control from the **Silver** palette and change the label to `Weather Data`.
- 2 Select controls and indicators to include in the cluster. `<Shift>`-click to select multiple objects.
- 3 Drag the controls and indicators into the **Weather Data** cluster.

4. Resize the cluster so that all the elements are visible and arranged vertically as shown in Figure 5-11.

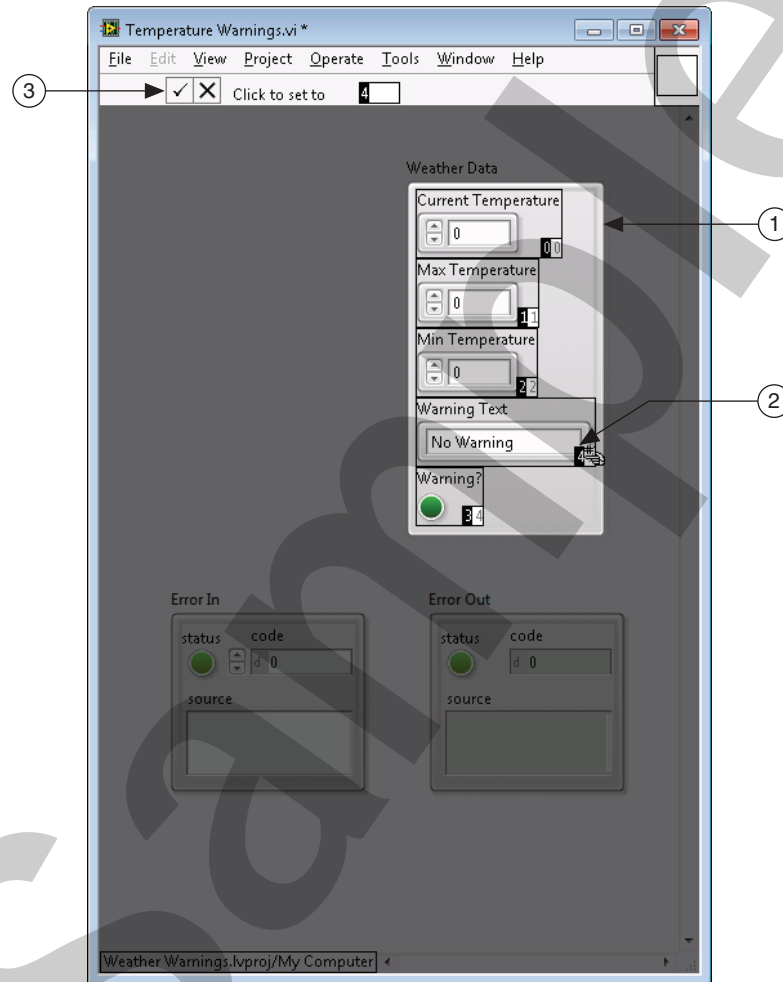
**Figure 5-11. Resize Cluster Control**



- 1 Autosize cluster—LabVIEW can rearrange and resize the cluster for you. Right-click the border of the **Weather Data** cluster and select **AutoSizing»Arrange Vertically**.

5. Reorder the items in the cluster as shown in Figure 5-12

**Figure 5-12. Reorder Cluster**

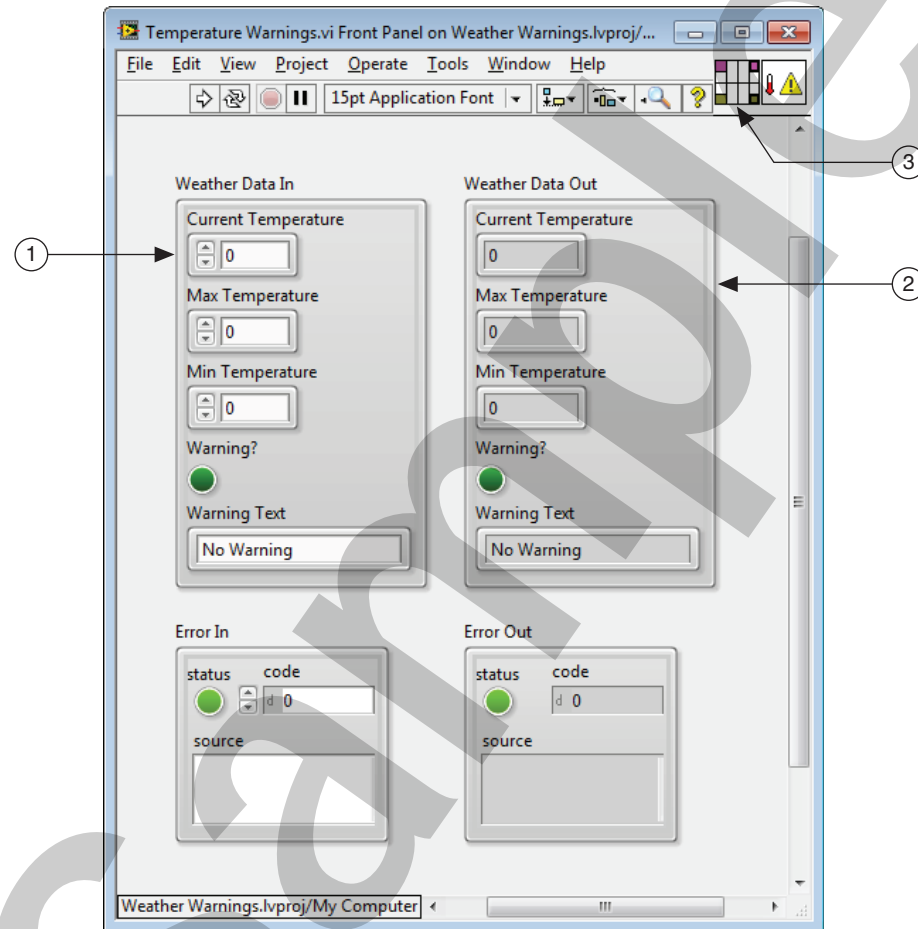


- 1 Right-click the edge of the cluster and select **Reorder Controls in Cluster**.
- 2 Click the black numbers to toggle the order of the items in the cluster.
- 3 Click the **Confirm** button to save the changes.



6. Modify the VI to receive and return cluster data.

**Figure 5-13.** Temperature Warnings—Weather Data In and Weather Data Out Clusters

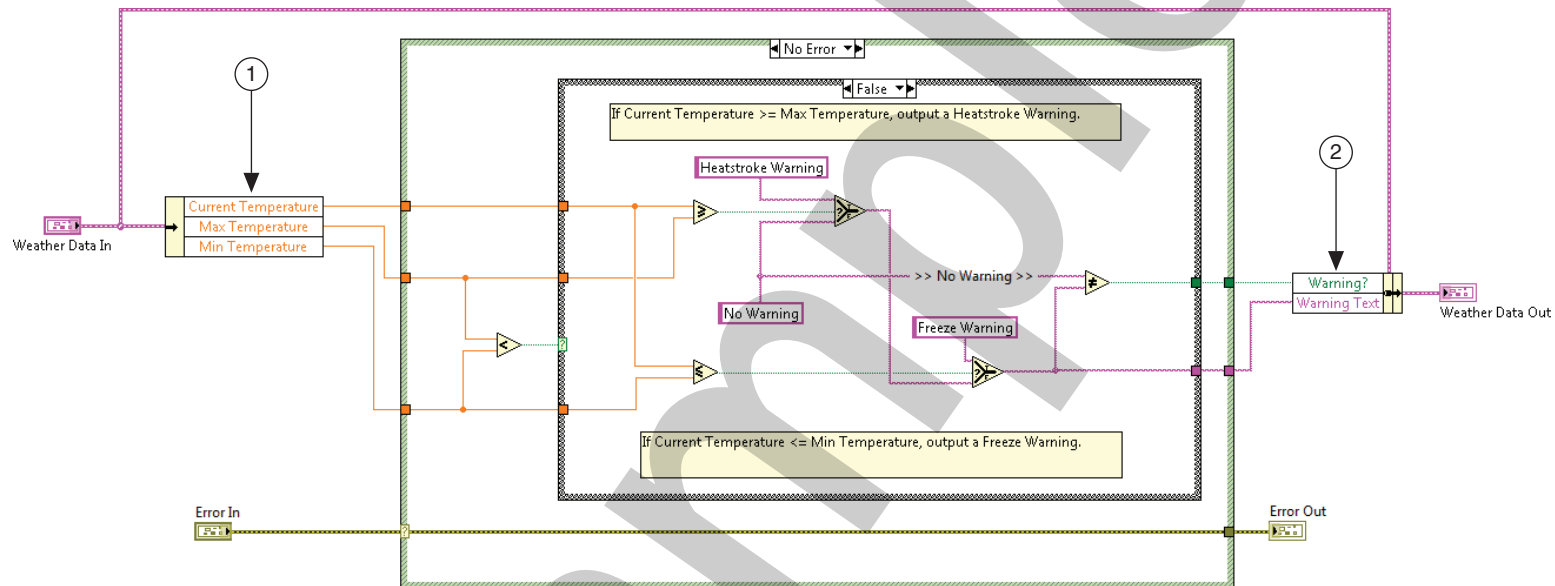


- 1 **Weather Data**—<Ctrl>-click the **Weather Data** cluster and drag it to create a copy. Rename the copy **Weather Data In**.
- 2 **Weather Data**—Right-click the original cluster and select **Change to Indicator**. Rename the indicator **Weather Data Out**.
- 3 **Wire connector pane**—Wire the **Weather Data In** control to the top left corner of the connector. Wire **Weather Data Out** to the top right corner.


7. Modify the block diagram as shown in Figure 5-14 to extract data from the input cluster.

- Move the **Weather Data In** cluster to the left, outside the Case structures.
- Move the **Weather Data Out** cluster to the right outside the Case structures.

**Figure 5-14.** Temperature Warnings with Clusters Block Diagram



- 1 **Unbundle By Name**—Wire the **Weather Data In** control and expand the Unbundle By Name function to display three items. Wire the outputs of the Unbundle By Name function to the broken wires in the order shown. Because you moved individual controls and indicators into a single cluster, you must use the Unbundle By Name function to wire the internal controls and indicators independently of each other.
- 2 **Bundle By Name**—Wire the **Weather Data In** cluster around the Case structure to the input cluster of the Bundle by Name function. Display two elements and use the Operating tool to select **Warning?** and **Warning Text** elements. Connect the broken wires to the Unbundle By Name inputs as shown.

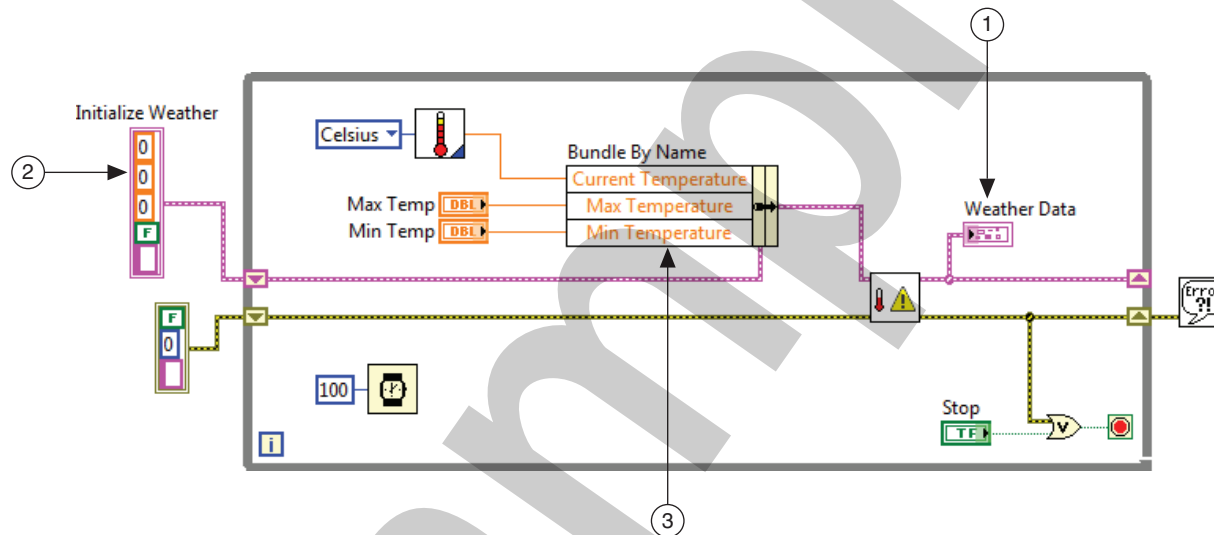
 **Note** If the order of the elements in the Unbundle By Name and the Bundle By Name functions is different than what you want, can use the Operating tool to change the order.

8. Save and close the Temperature Warnings VI.

## Test

1. Use the SubVI Tester VI to test the modified Temperature Warnings VI as a subVI.
  - ❑ On the block diagram, delete the **Thermometer**, **Warning?**, and **Warning Text** indicators and any broken wires. Press <Ctrl-B> to clean up all broken wires.
  - ❑ Modify the SubVI Tester VI to work with the Temperature Warnings VI that now uses clusters as shown in Figure 5-15.

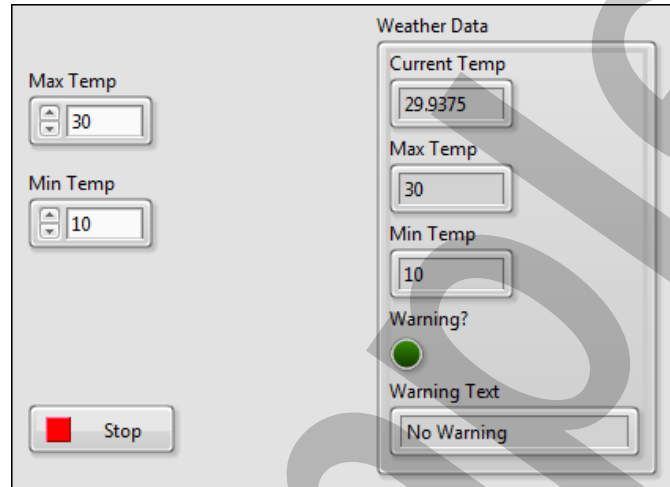
**Figure 5-15.** SubVI Tester VI Block Diagram Modified to Work with Modified Temperature Warnings VI



- 1 **Weather Data**—Right-click the **Weather Data Out** output of the Temperature Warnings VI and select **Create»Indicator**. Wire the output to the edge of the While Loop, right-click the tunnel, and then select **Replace with Shift Register** from the shortcut menu. The shift register stores the cluster data from one iteration to the next.
- 2 **Constant**—Right-click the left shift register and select **Create constant** to initialize weather data to values. Rename the constant *Initialize Weather*.
- 3 **Bundle By Name**—Wire the shift register from the **Initialize Weather** constant to the input cluster of the **Bundle By Name** function. Expand the **Bundle By Name** function to display three elements.

- Arrange the front panel as shown in Figure 5-16.

**Figure 5-16.** SubVI Tester VI Front Panel Modified with Clusters



- Enter values in the **Max Temperature** and **Min Temperature** controls.
- Run the VI and verify that the **Weather Data** indicator displays correct values.

2. Save and close the VI

End of Exercise 5-2

## Exercise 5-3      Concept: Type Definition

### Goal

To improve the scalability of your application by using type definitions made from custom cluster controls, indicators, and constants of a particular data type.

### Scenario

As a LabVIEW developer, you can encounter situations where you need to define your own custom data types in the form of clusters and enums. A challenge associated with using custom data types is that you may need to change them later in development. In addition, you may need to change them after they have already been used in VIs. For example, you create copies of a custom data type and use them as controls, indicators, or constants in one or more VIs. Then you realize that the custom data type needs to change. You need to add, remove, or change items in the cluster data type or the enum.

As a developer you must ask yourself the following questions:

- What should happen to the copies of the custom data types used in VIs that are already saved?
- Should the copies remain unchanged or should they update themselves to reflect changes to the original?

Usually, you want all the copies of the custom data type to update if you update the original custom data type. To achieve this you need copies of the custom data types to be tied to a type definition, which is defined as follows:

Type definition—A master copy of a custom data type that multiple VIs can use.

### Implementation

In this exercise, you modify the Temperature Warnings VI that you revised in Exercise 5-2 in such a way that the changes to the **Weather Data** custom data type propagate through the application.

When complete, the Weather Station application monitors temperature and wind information. This exercise modifies the Temperature Warnings VI. In the *Challenge* exercise, you modify the Windspeed Warnings VI.

1. Open `Weather Warnings.lvproj` in the `<Exercises>\LabVIEW Core 1\Weather Warnings` directory.
2. Open **Temperature Warnings** VI from the **Project Explorer** window.

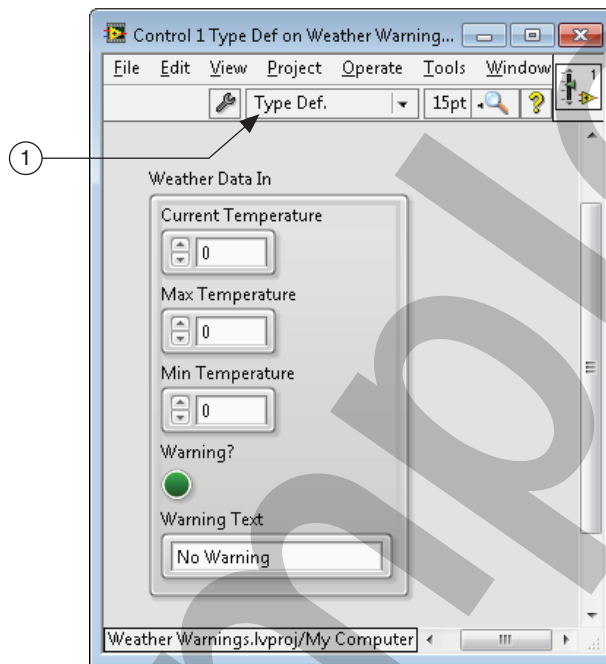
3. Experiment with changing an existing cluster.

- Place a **File Path Control (Silver)** in the **Weather Data In** cluster control.
- Notice that the Temperature Warnings VI is broken. This is because the **Weather Data In** and **Weather Data Out** clusters are no longer the same data type.
- Open the block diagram and notice the broken wire connected to the **Weather Data Out** terminal.
- Press <Ctrl-Z> to undo the addition of the File Path Control.

4. Make a type definition.

- Right-click the border of the **Weather Data In** control and select **Make Type Def**.
- On the block diagram the **Weather Data In** terminal now has a black triangle on the corner indicating that it is connected to a type definition.
- Right-click the border of the **Weather Data In** control and select **Open Type Def** to display the **Custom Control Editor** window as shown in Figure 5-17. The window looks like the front panel of a VI but it does not have a block diagram.

**Figure 5-17.** Custom Control Editor Window



1 The control type is Type Def, which maintains the link between this file and the custom control copies used in VIs.

- Save the custom control as `Weather Data.ctl` in the `<Exercises>\LabVIEW Core 1\Weather Warnings` directory and close the control editor window.
  - On the block diagram of the Temperature Warnings VI, notice the coercion dot on the **Weather Data Out** indicator terminal. This indicates that the indicator is not tied to the type definition.
5. Tie the **Weather Data Out** indicator to the type definition.
- Right-click the border of the **Weather Data Out** indicator on the front panel and select **Replace>Select a Control** from the shortcut menu.
  - Browse to and select the `Weather Data.ctl` file you just created.



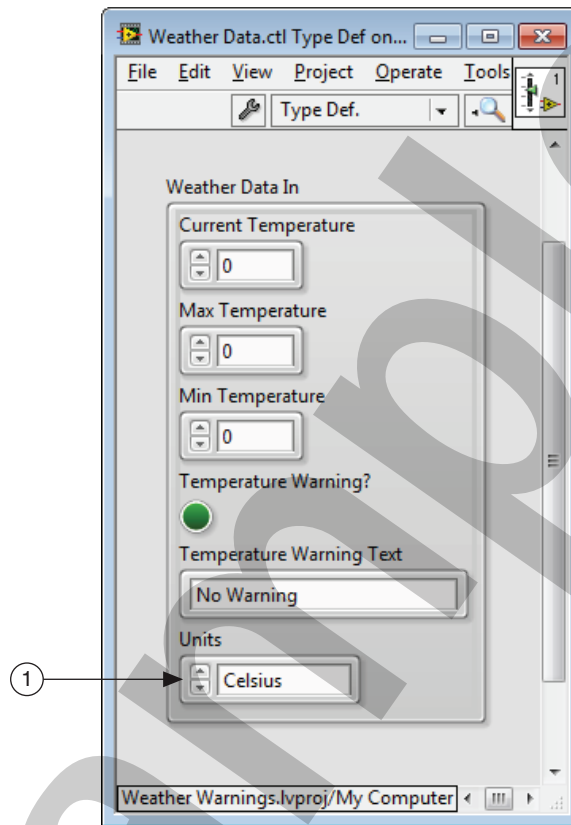
**Note** You can no longer add or remove elements to or from the cluster control and indicator on the front panel. You must open the type definition and add or remove the element from the control editor window.

- Save the Temperature Warnings VI.
6. Edit the Weather Data type definition to include unit information.
- Right-click the border of the **Weather Data In** control and select **Open Type Def** from the shortcut menu.
  - Modify the front panel as shown in Figure 5-18.

Sample



**Figure 5-18.** Weather Data Type Definition with Temperature Units

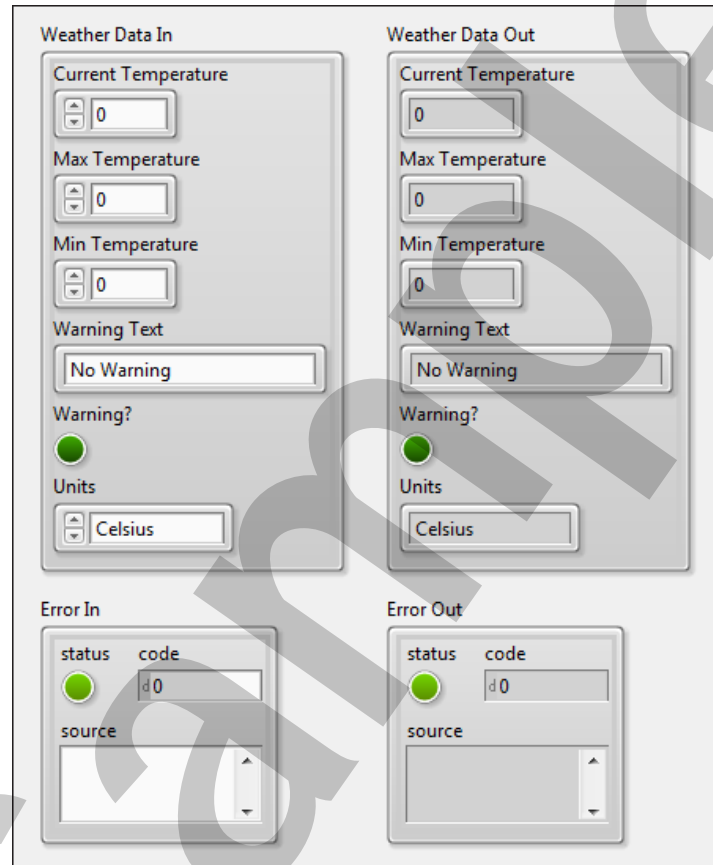


**1 Enum (Silver)**—Place an enum in the cluster and rename it `Units`. Right-click the enum and select **Edit items**. Create an item for `Celsius` and `Fahrenheit`.

- Save the Weather Data type definition and close the control editor window.

- Notice that the **Weather Data In** control and **Weather Data Out** indicator on the Temperature Warnings VI have been updated with the changes you made to the Weather Data type definition. Arrange the front panel of the VI as shown in Figure 5-19.

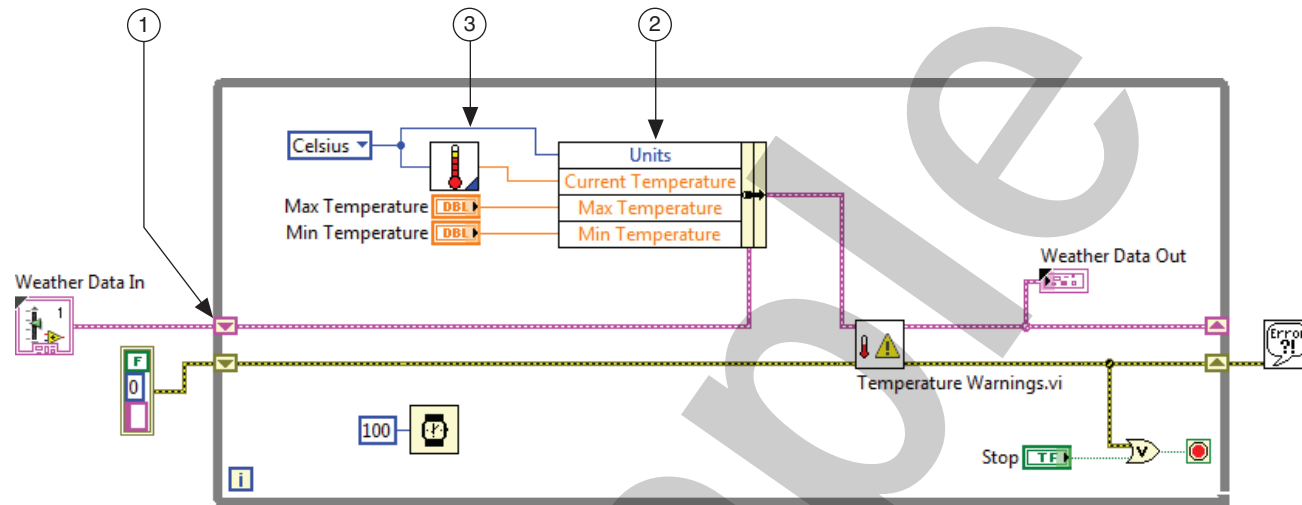
**Figure 5-19.** Temperature Warnings VI with Type Def Controls and Indicators



7. Save the Temperature Warnings VI.

## Test

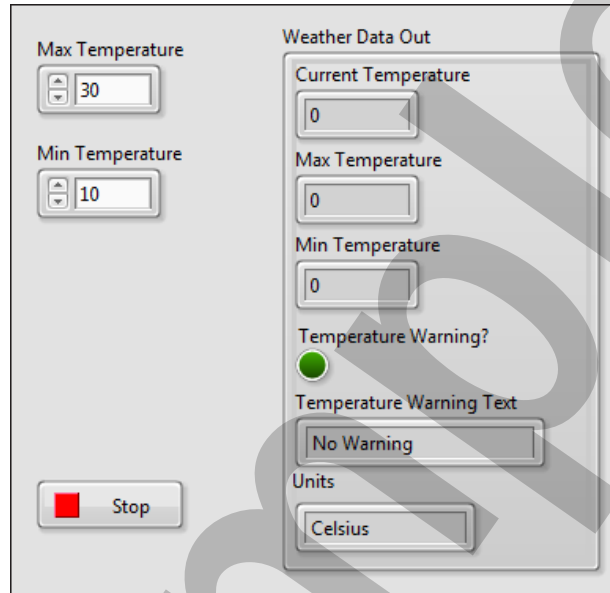
1. Use the SubVI Tester VI to test the modified Temperature Warnings VI as subVIs.
  - Open the SubVI Tester VI from the **Project Explorer** window.
  - On the front panel, right-click the **Weather Data Out** cluster and select **Replace»Select a Control** from the shortcut menu.
  - Navigate to the `Weather Data.ct1` file in the `<Exercises>\LabVIEW Core 1\Weather Warnings` directory and select it.
  - On the block diagram of SubVI Tester VI, delete the following items:
    - Initialize Weather constant wired to the left shift register
    - Any broken wires. Press `<Ctrl-B>` to clean up all broken wires.
  - Modify the SubVI Tester VI to work with the Temperature Warnings VI which now uses type definitions as shown in Figure 5-20.

**Figure 5-20.** SubVI Tester VI Block Diagram Modified to Test Temperature Warnings SubVI

- 1 Initialize shift register—Right-click the left shift register and select **Create»Constant** to initialize the shift register. Right-click the cluster and select **View Cluster as Icon**.
- 2 **Bundle By Name**—Expand the node to display four elements. Use the Operating tool to select **Units**.
- 3 Wire the **Units (0:Celsius)** constant to the **Units** element of the Bundle By Name function.

- Arrange the front panel as shown in Figure 5-16.

**Figure 5-21.** SubVI Tester VI Front Panel Modified with Type Def



- Enter values in the **Max Temperature** and **Min Temperature** controls.
2. Run the VI and verify that correct values are displayed in the **Weather Data Out** indicator.
  3. Save and close the VI.

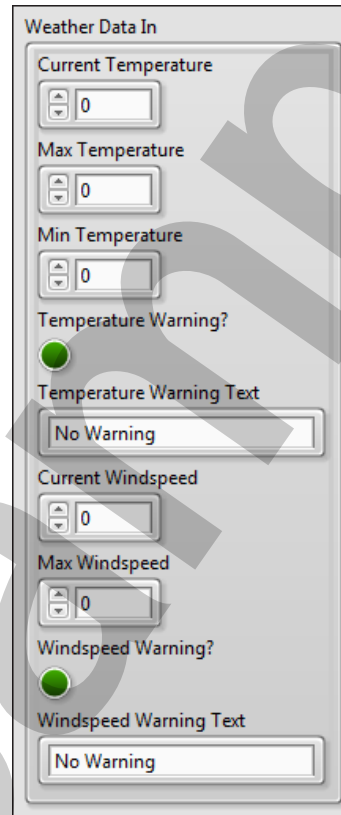
### Challenge

In this challenge exercise, you modify the Windspeed Warnings VI to augment the Weather Station application. You then modify test the Windspeed Warnings VI as a subVI.

1. Add the Windspeed Warning VI to the Weather Station project.
  - In the Project Explorer window, right click **My Computer** and select **Add>File** from the shortcut menu.
  - Navigate to <Exercises>\LabVIEW Core 1\Weather Warnings\Support VIs and select Windspeed Warnings.vi.

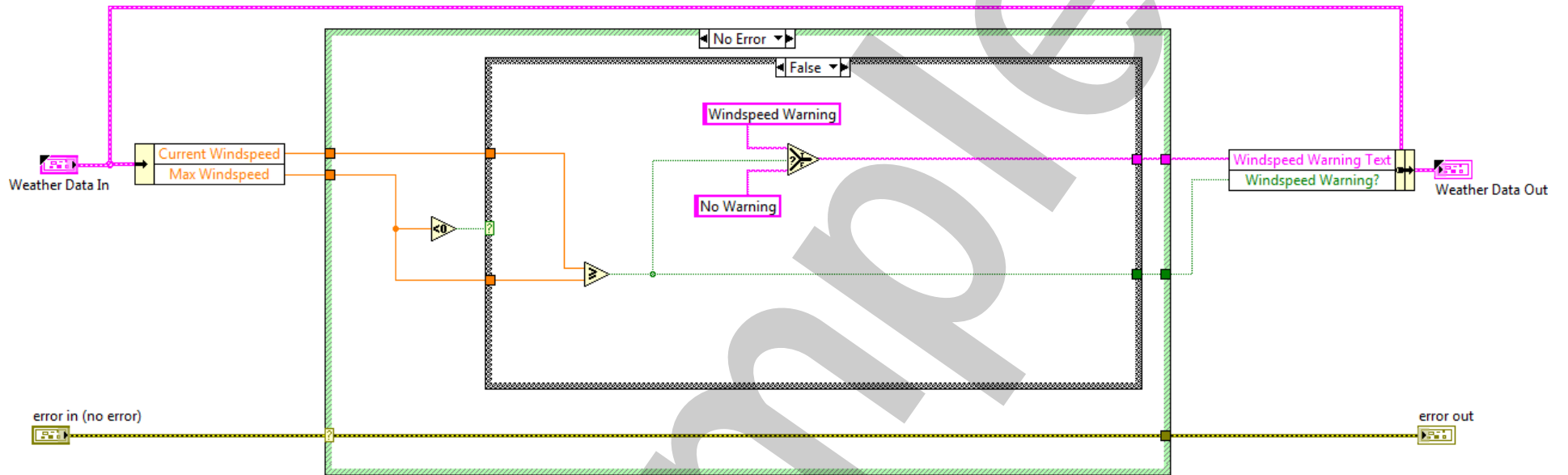
2. Open the Windspeed Warnings VI.
3. Copy the **Weather Data In** cluster from the Temperature Warnings VI to the Windspeed Warnings VI.
4. Right-click the **Weather Data In** cluster and select **Open Type Def** from the shortcut menu.
5. Modify the Weather Data type definition with windspeed controls as shown in Figure 5-22.

**Figure 5-22.** Windspeed Warnings VI Type Definition Controls and Indicators



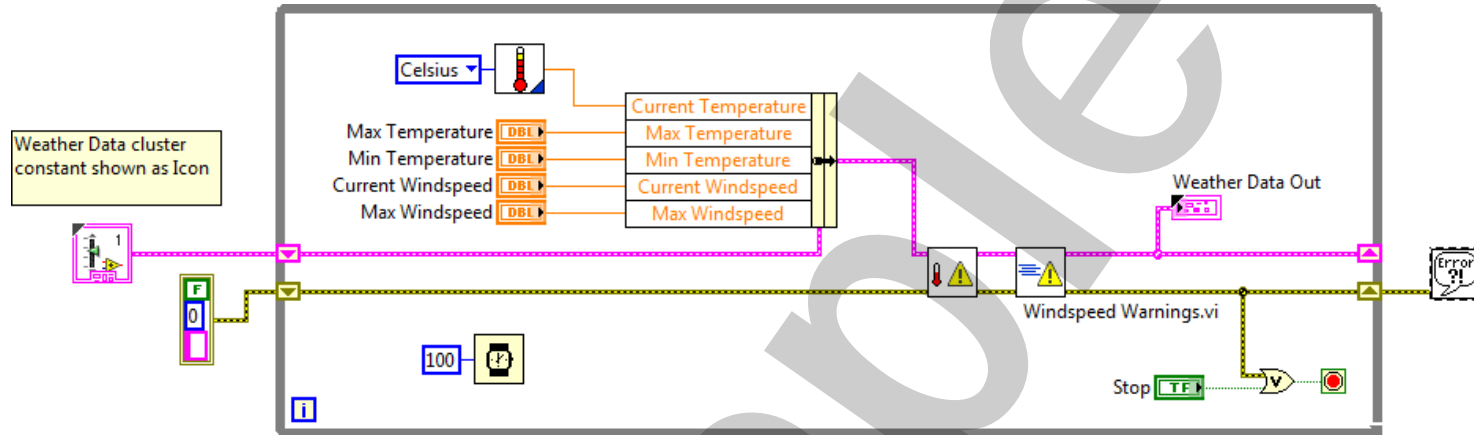
6. Modify the block diagram of the Windspeed Warnings VI to use the new Weather Data type definition instead of individual controls and indicators, as shown in Figure 5-23.

**Figure 5-23.** Windspeed Warnings VI Using Type Definitions



7. Modify the SubVI Tester VI to work with the Windspeed Warnings VI, as shown in Figure 5-24.

**Figure 5-24.** SubVI Tester VI Block Diagram Modified to Test Windspeed Warnings SubVI



8. Run the SubVI Tester VI and verify that correct values are displayed in the **Weather Data Out** indicator.
9. Save and close the VI and the project.

End of Exercise 5-3