

LabVIEW Environment

LabVIEW programs are called virtual instruments (VIs). Each VI has two windows: a front panel that contains the user interface, and a block diagram with the graphical code. This chapter discusses how to operate within these two windows; as with each chapter in this book, it is best to read the chapter through and then to perform the activity that follows. During the activity you will be able to try out the features described in the chapter.

FRONT PANEL

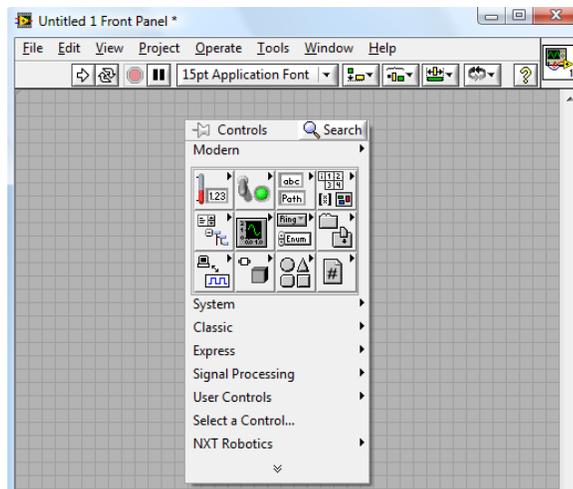
When you open a new or existing VI, the front panel window of the VI appears and functions as the graphical user interface (GUI) of a VI. Controls and indicators are placed on the front panel, and are the interactive input and output terminals respectively. They are automatically given corresponding terminals on the block diagram. The front panel controls and indicators are collectively referred to as objects. Refer to the “Block Diagram” section below for more information on block diagram terminals.

LabVIEW programmers design the front panel as they develop the VI. Users interact with the front panel when the program is running. Users can control the program, change inputs, and see data updated in real time. It is important for a VI to have an intuitive and easy-to-read front panel. The front panel is essentially the gateway for all user input and output of a VI; therefore, it is essential that the programmer has a good grasp of how to effectively design a front panel.

The front-panel window contains a toolbar across the top and a Controls palette that you can access by right-clicking anywhere on the front panel. You can recognize the front panel by its grey background and grid.

Front Panel Controls Palette

Use the Controls palette to place controls and indicators on the front panel. The Controls palette is available only on the front panel. To view the palette, choose Controls Palette from the View menu. You can also display the Controls palette by right-clicking an open area on the front panel. Tack down the Controls palette by clicking the pushpin on the upper-left corner of the palette. This will ensure that the Controls palette is always visible on the front panel.

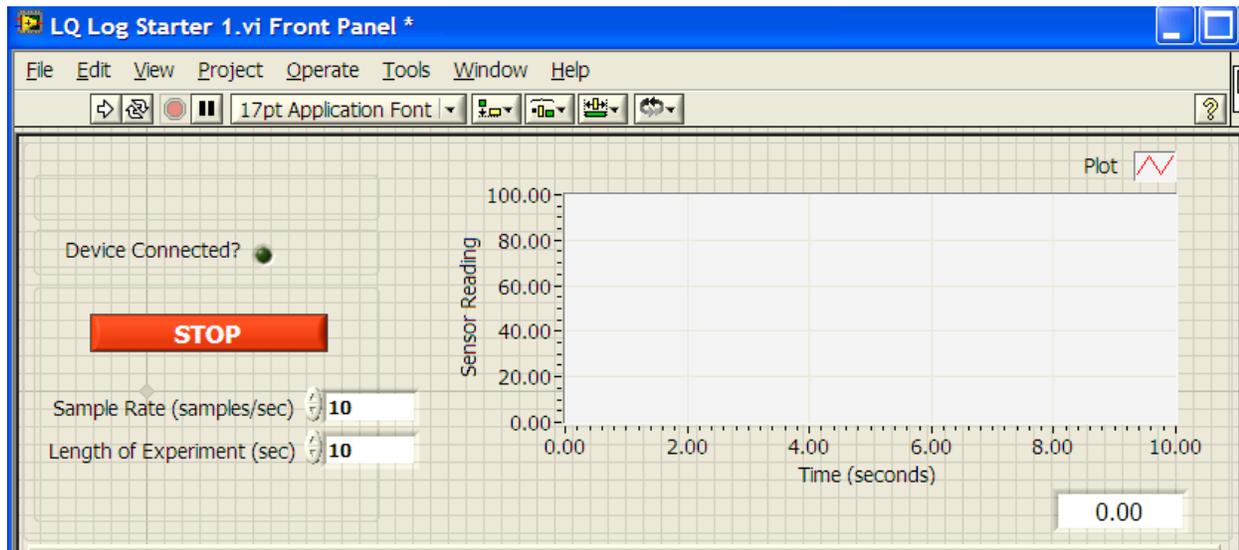


Chapter 2

After opening the Controls palette, use it to place controls and indicators on the front panel by dragging and dropping the icons from the palette into the front panel workspace.

Tip: Using the pushpin to pin the Controls palette to the front panel provides a View button to change what controls and indicators are visible on the Controls palette, and a Search button if you have trouble finding a specific control or indicator. You can also have multiple palettes open to have your most commonly used subpalettes available simultaneously.

Controls (knobs, push buttons, dials, and other input devices) are the interactive input terminals, while indicators (graphs, LEDs, and other displays) are the interactive output terminals of the VI. Controls simulate instrument input devices and supply data to the block diagram of the VI. Indicators simulate instrument output devices and display data the block diagram acquires or generates.



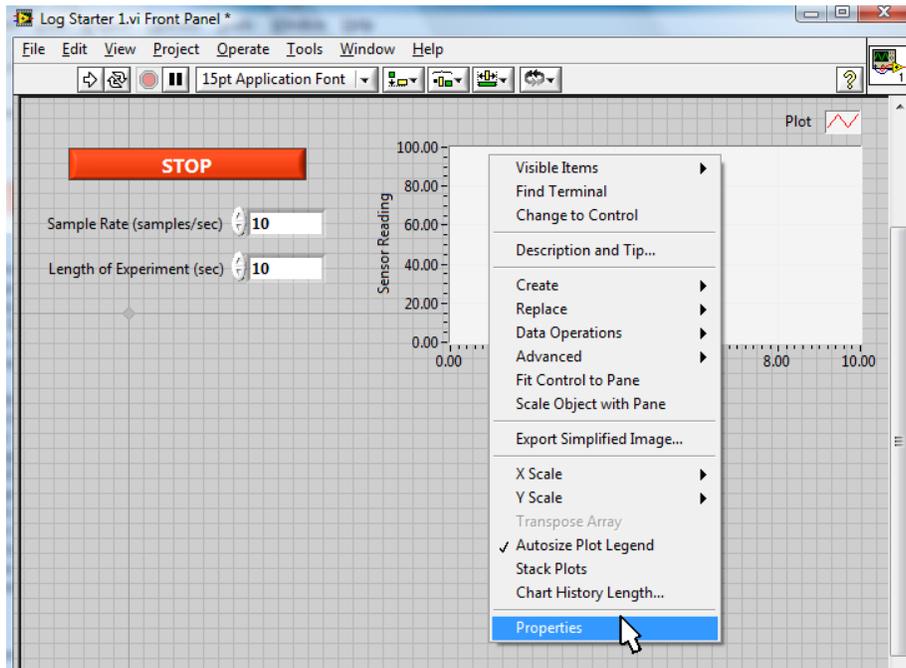
The figure above has three controls labeled STOP, Sample Rate, and Length of Experiment. The user can click the STOP button to change this control's value from True to False. The Sample Rate and Length of Experiment controls can be adjusted by clicking the up/down arrows, or by typing new values in the text box.

There are two indicators: a waveform chart and an LED labeled Device Connected. The user can see the value generated by the VI (in the block diagram) on the chart indicator. In addition, the user will have feedback from the LED on whether the SensorDAQ or LabQuest interface is connected (LED will turn bright green) or not connected (LED remains dark green).

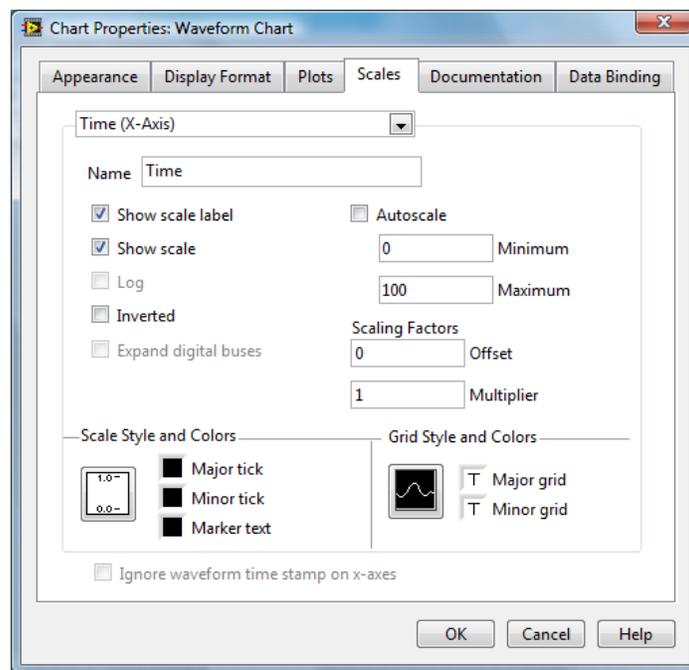
Front Panel Control Properties

All LabVIEW objects (e.g., controls and indicators) have associated shortcut menus and property dialog boxes. As you create a VI, use the shortcut menu items and/or the properties dialog box to change the appearance and/or behavior of front panel and block diagram objects. To access the

shortcut menu, right-click the object you want to modify. To access the Properties dialog box, select Properties from the shortcut menu.



For example, in the figure above, the user might modify some chart properties (such as the chart's grid styles and axis labels) by right-clicking the chart, and selecting Properties from the shortcut menu.



After clicking on Properties, the Properties dialog box opens and provides a way to modify the chart. The tab shown in the figure above provides scale properties that can easily be modified.

Front Panel Window Toolbar

Each window has a toolbar associated with it. Use the front panel window toolbar buttons to run and edit the VI. The following toolbar appears on the front panel window.



 Click the Run button to run your VI. You can run a VI if the Run button appears as a solid white arrow, shown at left.

 The Run button appears broken when the VI you are creating or editing contains errors. If the Run button still appears broken after you finish wiring the block diagram, the VI is broken and cannot run. Click this button to display the Error List window, which lists all errors and warnings.

 Click Run Continuously to run the VI until you abort or pause execution. You can also click the button again to disable continuous running. Be aware that this is not a good method for running a VI continuously; you should instead create code that provides this feature.

 While the VI runs, the Abort Execution button appears. Click this button to stop the VI immediately if there is no other way to stop the VI. If more than one running top-level VI uses the VI, the button is dimmed.

Caution: Avoid using the Abort Execution button. It stops the VI immediately before it finishes the current iteration. Aborting a VI that uses external resources, such as external hardware, might leave the resources in an unknown state by not resetting or releasing them properly. Design VIs with a STOP button to avoid this problem.

 Click Pause to pause a running VI. When you click the Pause button, LabVIEW highlights on the block diagram the location where you paused execution, and the Pause button appears red. Click the Pause button again to continue running the VI. Again, be aware that this might leave resources in a troubled state if you have external hardware.

 Select the Text Settings pull-down menu to change the font settings for the selected portions of your VI, including size, style, and color.

 Click the Align Objects pull-down menu to align objects along axes, including vertical and horizontal edges and centers.

 Click the Distribute Objects pull-down menu to space multiple front panel objects evenly.

 Click the Resize Objects pull-down menu to resize multiple front panel objects to the same size.

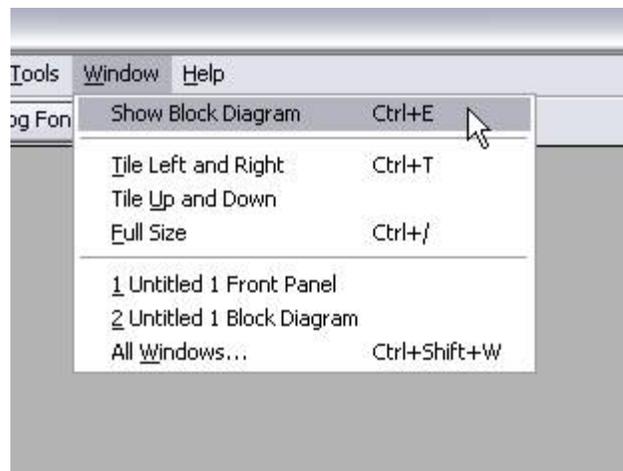
 Click the Reorder pull-down menu when your objects overlap each other and you want to define which one is in front or back of another. Select one of the objects with the Positioning tool and then select from Move Forward, Move Backward, Move To Front, and Move To Back.

 Click the Show Context Help Window button to toggle the display of the context help window. Context Help displays relevant information of an object when you hover your mouse over the object.

 The Enter Text button only appears (on the far left side of the toolbar) when the user has typed on the front panel. Pressing the Enter Text button finalizes the text or value changes. The Enter Text button disappears when you click it. Instead of pressing the Enter Text button, the user may also press the <Enter> key or click the front panel or block diagram workspace.

BLOCK DIAGRAM

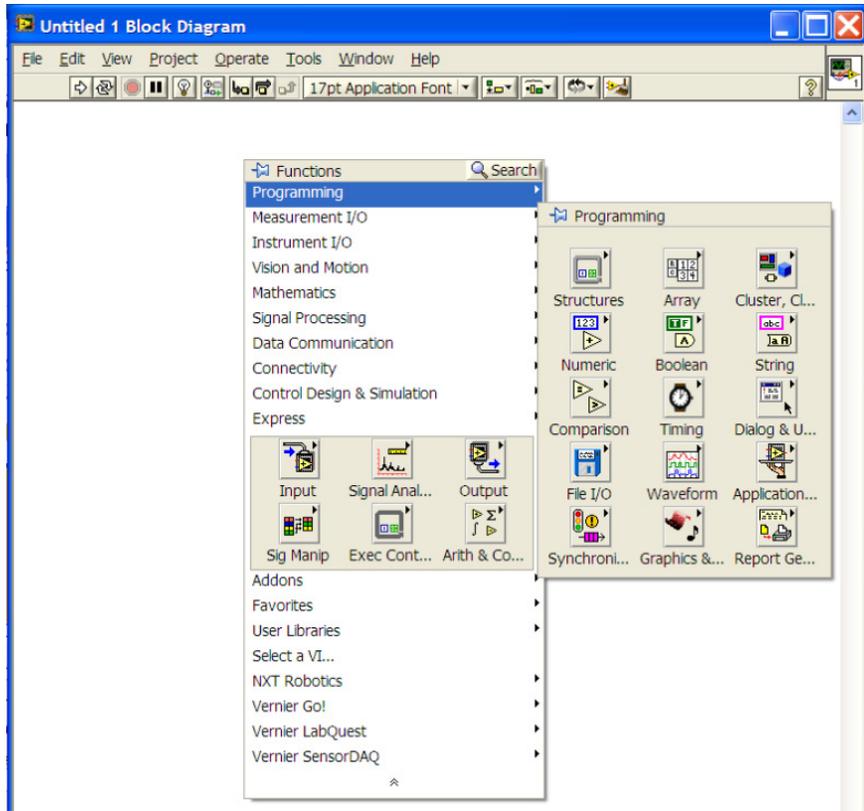
When you create or open a new VI, the front panel opens automatically. To bring up the block diagram, choose Show Block Diagram from the Window menu. Additionally, you can toggle between the block diagram and the front panel by pressing <Ctrl-E>.



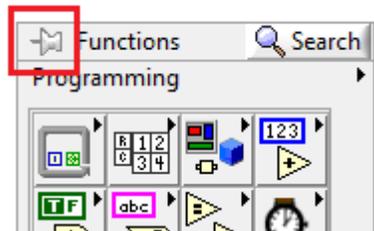
In LabVIEW, you build a user interface with controls and indicators on the front panel. The programmer must then add code using graphical representations of functions to control these front panel objects. The block diagram contains this code. In some ways, the block diagram resembles a flowchart. The concept of the block diagram is to separate the graphical source code from the user interface in a logical and simple manner. You can recognize the block diagram by its white background.

Block Diagram Functions Palette

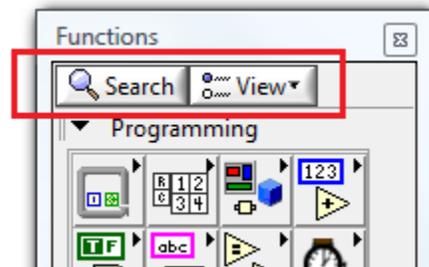
Block diagram objects include terminals (linked to front panel objects), functions, constants, structures, subVIs, and wires that transfer data among other block diagram objects. To place objects on the block diagram, simply drag and drop them from the Functions palette. The Functions palette automatically appears when you right-click anywhere on the block diagram workspace or choose Functions Palette from the View menu. It contains functions, constants, structures, and some subVIs.



The Functions palette appears when you right-click, and will automatically disappear when you left-click on something new. If you prefer to keep the Functions palette visible at all times, there is a pushpin on the top-left corner of the palette that can be clicked.



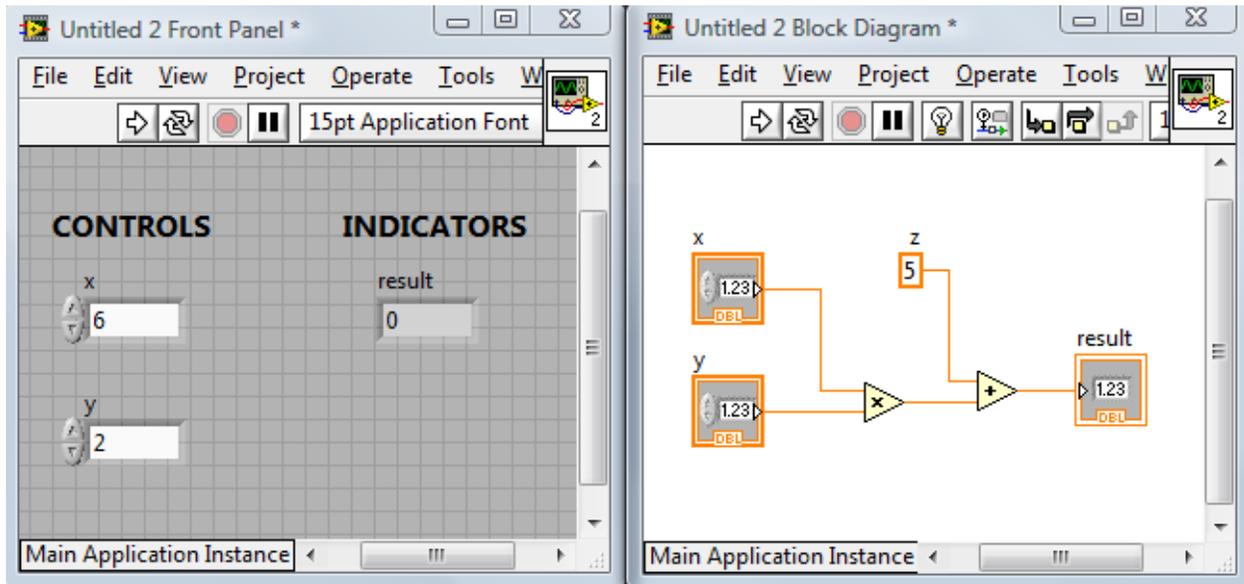
Clicking the pushpin will tack down the palette menu. Tacking the palette menu also provides a View button to change the subpalettes that are visible on the Functions palette, and a Search button if you have trouble finding a function.



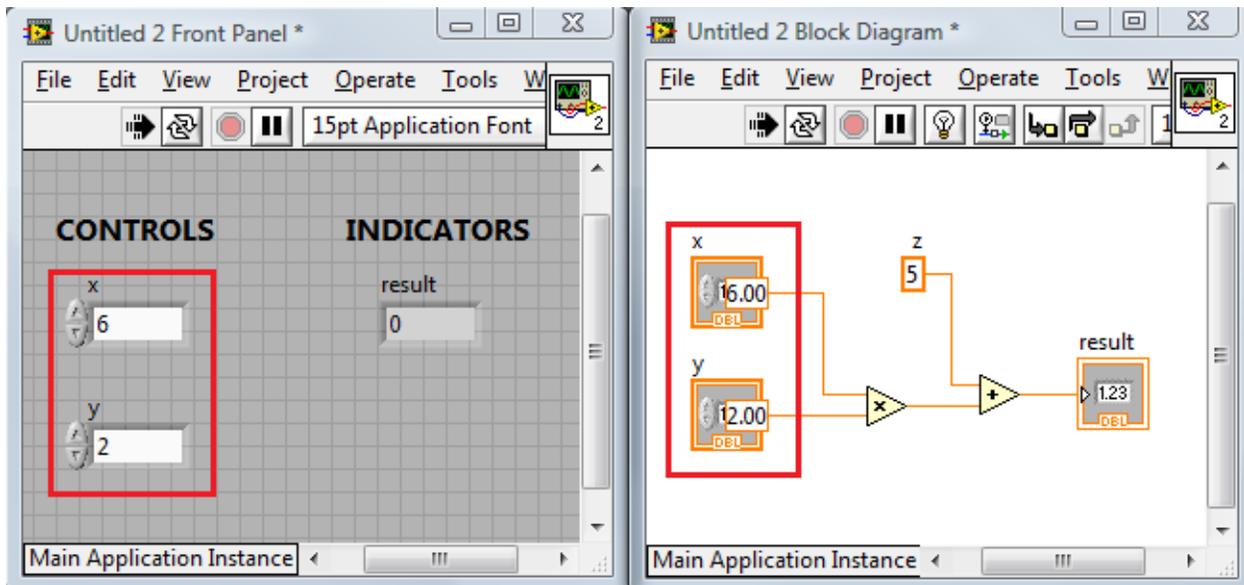
Tip: It can be very helpful to only make a select number of subpalettes visible. Consider only showing the Programming, Express, and Vernier palettes when you are first using the program.

Block Diagram Controls, Indicators and Constants

Front panel objects appear as terminals on the block diagram, and although they appear on separate windows, the two are linked. Block diagram control terminals receive their values from the front panel and pass these values into the block diagram logic. Block diagram indicator terminals receive their values from the block diagram logic and pass their values to the front panel. Constants do not appear on the front panel window. A constant only exists on the block diagram and simply passes a value within the block diagram logic.



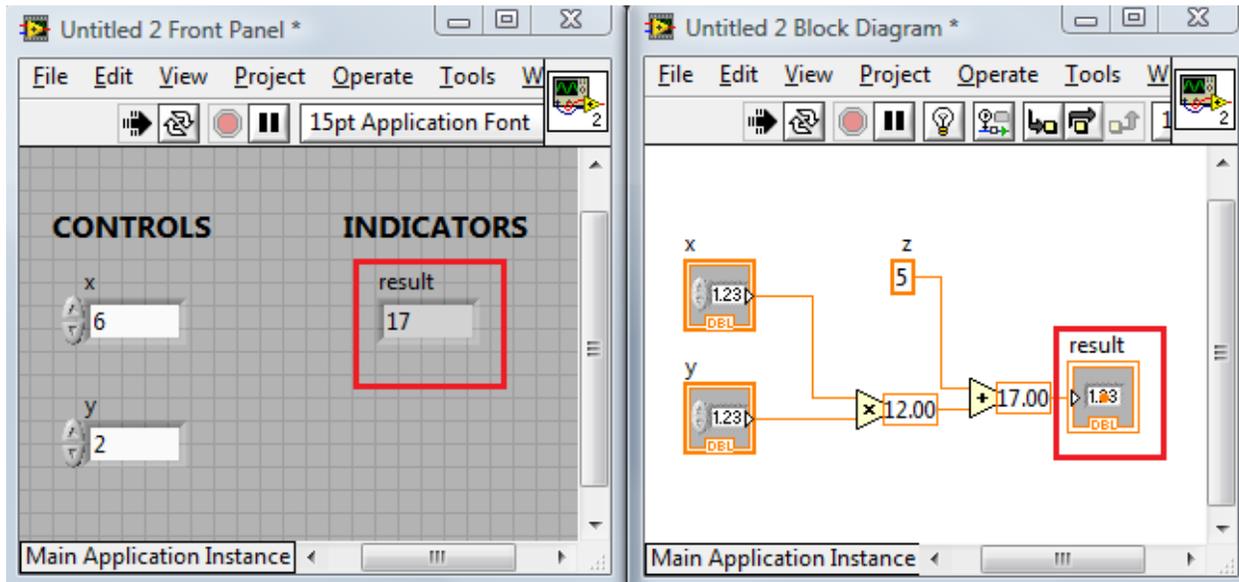
In the figure above, the front panel and block diagram are both shown prior to running this VI. When this VI is run, the data values of the front panel controls, x and y, enter the block diagram through their respective control terminals, x and y.



The data then flow from the terminals to the multiply function. When the multiply function completes the calculation, the new data value flows through the wire to the addition function,

Chapter 2

where the constant z , with a value of 5, is added. The new data value leaves the addition function and flows to the indicator terminal called result.



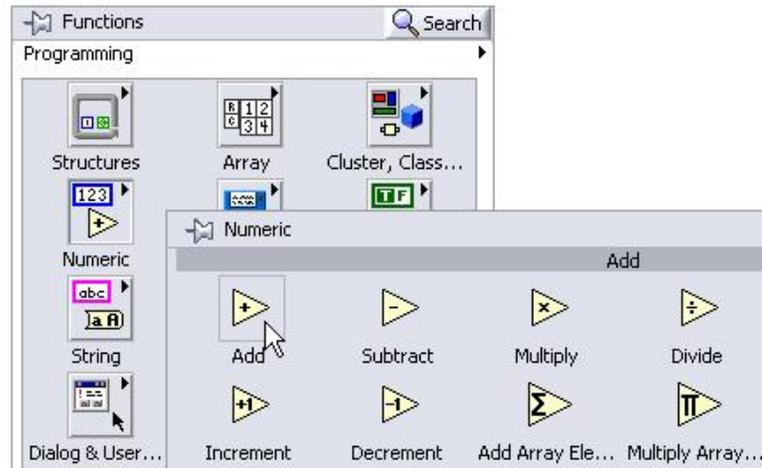
At this point, the front panel indicator linked to the result terminal is updated with the value calculated by the block diagram logic.

Notice that the x and y block diagram terminals look different from the result terminal. There are two distinguishing characteristics between a control and an indicator on the block diagram. The first is an arrow on the terminal that indicates the direction of data flow. The controls have arrows showing the data leaving the terminal, whereas the indicator has an arrow showing the data entering the terminal. The second distinguishing characteristic is the border around the terminal. Controls have a thick border and indicators have a thin border.



Block Diagram Functions

Functions are the fundamental operating elements of LabVIEW. Functions have input and output terminals for passing data in and out. You can tell if a block diagram object is a function by the pale yellow background on its icon. The Functions palette has functions arranged in groups based on the type of function they perform; for example, look in the Numeric subpalette for functions that perform numeric operations.



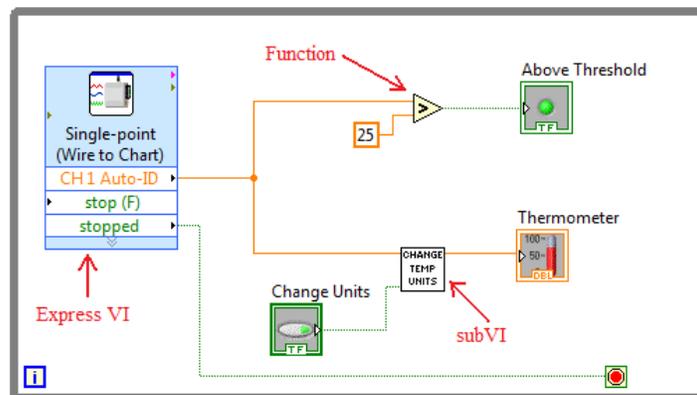
There are many different types of functions. Remember that a function has a pale yellow background like the functions shown below.



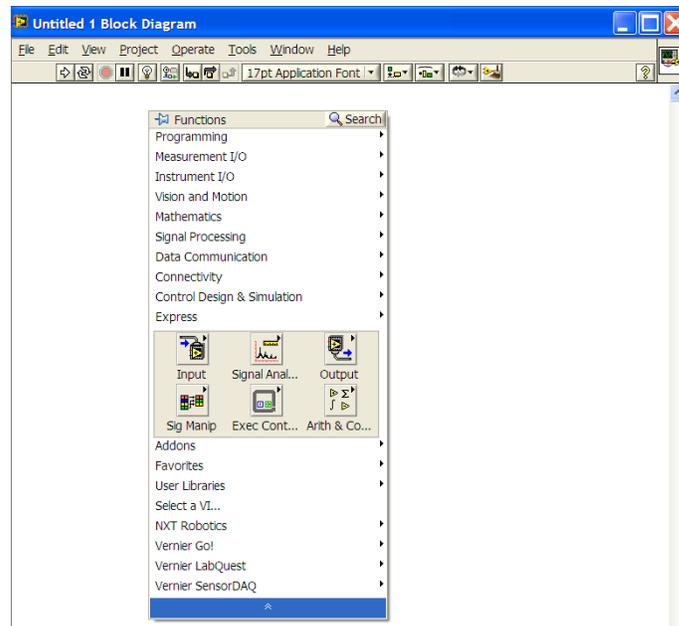
Block Diagram SubVIs and Express VIs

SubVIs are VIs that you create to use inside another VI, or that you access on the Functions palette. Any VI has the potential to be used as a subVI. When you double-click a subVI that is on the block diagram, its front panel window appears and you can access its block diagram. The icon for a subVI can be customized, but in most cases subVIs appear as square blocks. SubVIs are discussed in detail in Chapter 7.

Express VIs appear on the block diagram as expandable nodes with icons surrounded by a blue field. The primary benefit of Express VIs is their interactive configurability through a dialog box that is similar to a wizard or setup assistant. This dialog box appears when you place the Express VI on the block diagram. To re-open the dialog box, simply double-click the Express VI or right-click the Express VI and select Properties from the shortcut menu. You can also configure the Express VI by wiring values to terminals of the Express VI on the block diagram.

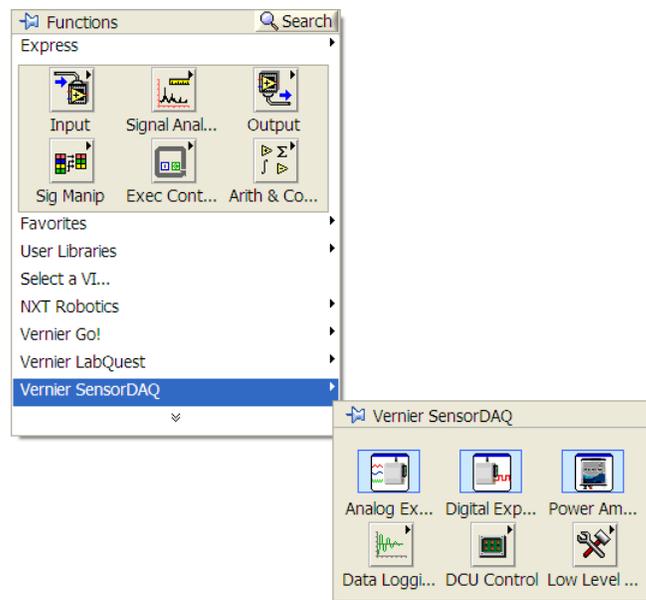


Special palettes for data collection with Vernier interfaces are available. They include standard and Express VIs. They are listed at the bottom of the list when you display the Functions palette.



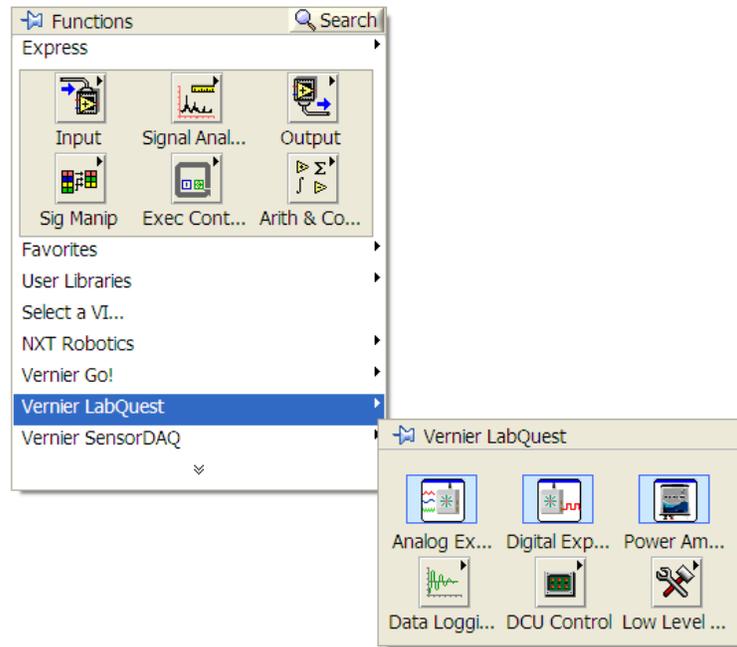
Using SensorDAQ

The SensorDAQ palette contains Express VIs and subVIs. When you are building a custom VI that communicates with the SensorDAQ, you will use these tools.



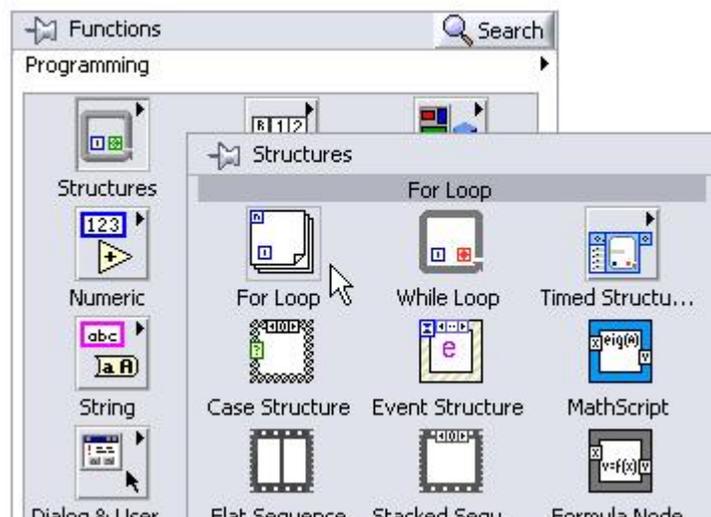
Using a LabQuest Interface

The LabQuest palette contains Express VIs and subVIs for data collection using any of the Vernier LabQuest interfaces.

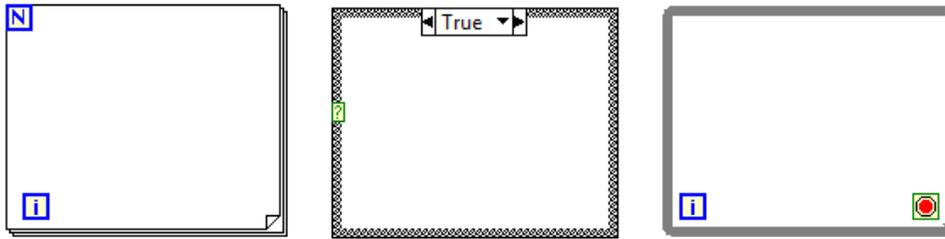


Block Diagram Structures

Structures, which include For Loops, Case Structures, and While Loops, are used for process control. The While Loop is examined in more detail in Chapter 4. The Case Structure is examined in more detail in Chapter 6. You can open the Structures subpalette from the Functions palette under Programming.



Below are a For Loop, Case Structure, and While Loop, as they appear on the block diagram.



Block Diagram Window Toolbar

When you open a VI, the following toolbar appears on the block diagram. Many of these buttons are the exact same as those found on the front panel window toolbar, and were covered on pages 2–4 and 2-5. The buttons not covered in that section, which only appear on the block diagram toolbar, are described below.



 Click the Highlight Execution button to see the flow of data through the block diagram. Click the button again to disable execution highlighting.

 Click Retain Wire Values button to save the wire values at each point in the flow of execution so that when you place a probe on a wire (the wire can be probed by right-clicking on the wire and selecting Probe from the shortcut menu), you can immediately obtain the most recent value of the data that passed through the wire.

 Click the Step Into button to single-step into a loop, subVI, and so on. Single-stepping through a VI steps through the VI node to node. Each node blinks to denote when it is ready to execute. By stepping into the node, you are ready to single-step inside the node.

 Click the Step Over button to step over a loop, subVI, and so on. By stepping over the node, you execute the node without single-stepping through the node.

 Click the Step Out button to step out of a loop, subVI, and so on. By stepping out of a node, you complete single-stepping through the node and go to the next node.

 Click the Clean Up Diagram button to re-route all existing wires and reorder objects on the block diagram for a cleaner, more readable block diagram.

TOOLS PALETTE

Working in LabVIEW requires the use of different tools. You will notice that as you move your cursor over LabVIEW front panel and block diagram objects, the cursor automatically changes icons. These different cursor icons represent LabVIEW tools. For example, if the cursor is near a block diagram terminal, the cursor icon will automatically change to the wiring tool. If the cursor is near a front-panel numeric control, the cursor icon will automatically change to the operating tool.

LabVIEW is configured to automatically select the tool based on the cursor location. In some cases, you will need to disable automatic tool selection and manually select the tool. To do this, you will need access to the LabVIEW tools. Select View ► Tools Palette to display the Tools palette.



Select your tool (such as the coloring tool) by clicking on the Tools palette button that represents the desired tool. When you click on a tool button, the automatic tool selection becomes disabled; therefore, when you are finished using the selected tool you must click the Automatic Tool Selection button to enable this feature again.



If automatic tool selection is enabled and you move the cursor over objects on the front panel or block diagram, LabVIEW automatically selects the corresponding tool from the Tools palette. Toggle automatic tool selection by clicking the Automatic Tool Selection button in the Tools palette.



Use the Operating tool to change the values of a control or to select the text within a control.



Use the Positioning tool to select, move, or resize objects. The Positioning tool changes shape when it moves over a corner of a resizable object.



Use the Labeling tool to edit text and create free labels. The Labeling tool changes to a cursor when you create free labels.



Use the Wiring tool to wire objects together on the block diagram.



Use the Shortcut Menu tool to open the shortcut menu of an object.



Use the Scrolling tool to scroll the window without using the scroll bars.



Use the Breakpoint tool to pause execution at a specified location of the block diagram.



Use the Probe tool on a wire to check the value of the wire as the program runs.



Use the Color Copy tool to copy the color into the Coloring Tool palette.



Use the Coloring tool to set the foreground and background color of front panel objects, front panel panes, and block diagram workspaces.