CHAPTER 2

Structure of VI

2.1 Front Panel and Diagram Window

When you launch LabVIEW by double clicking the icon of LabVIEW, it will come up with two windows: one with a gray background which is called the front panel, and the other with a white background which is called the diagram window. Figure 2.1 and Figure 2.2 show them, respectively. LabVIEW is a G (Graphical)-programming language for virtual instrumentation (VI), and these two windows reflect the characteristics of G-language.

The front panel resembles the front panel of an instrument: it has gauges, switches, knobs, and buttons. All of the wires and each individual component of the instrument are hidden in the chassis of the instrument that corresponds to the diagram window in LabVIEW. The files in LabVIEW have **.vi** extensions under PC environment, but not necessarily under Macintosh environment. (There is no extension for files on Macintosh computers.) Since the diagram window corresponds to the wires and connections in real instruments, you should expect to perform some wiring tasks in the diagram window. Wiring technique is very important and is addressed separately in a later section.

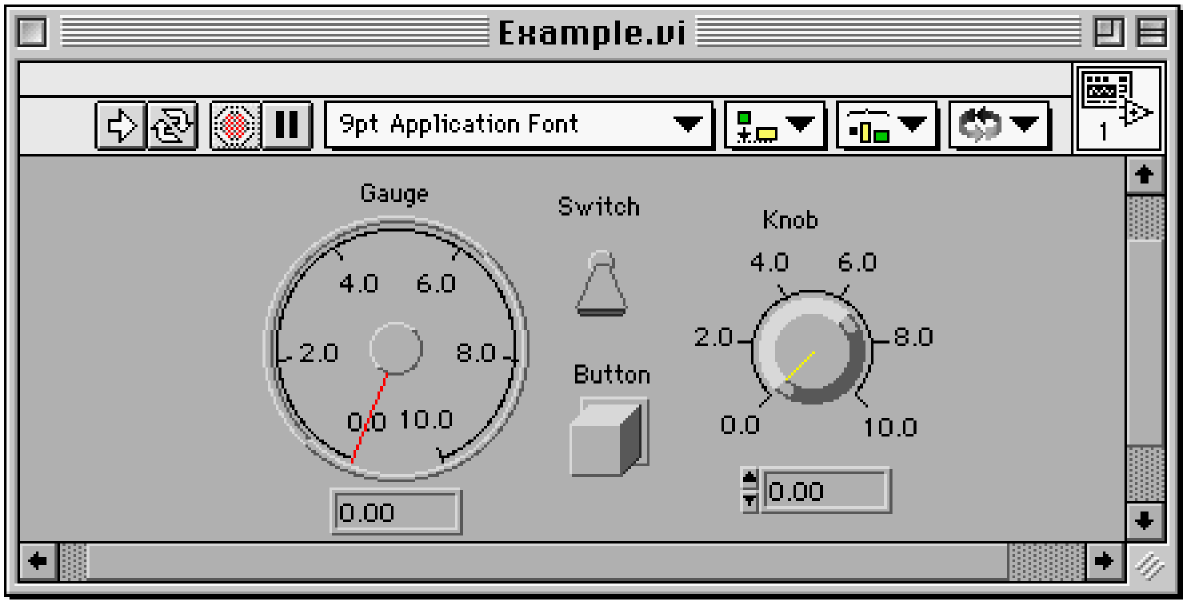


Figure 2.1 Front panel of a VI. All of the screen shots of VIs in this book are used with the permission of National Instruments.

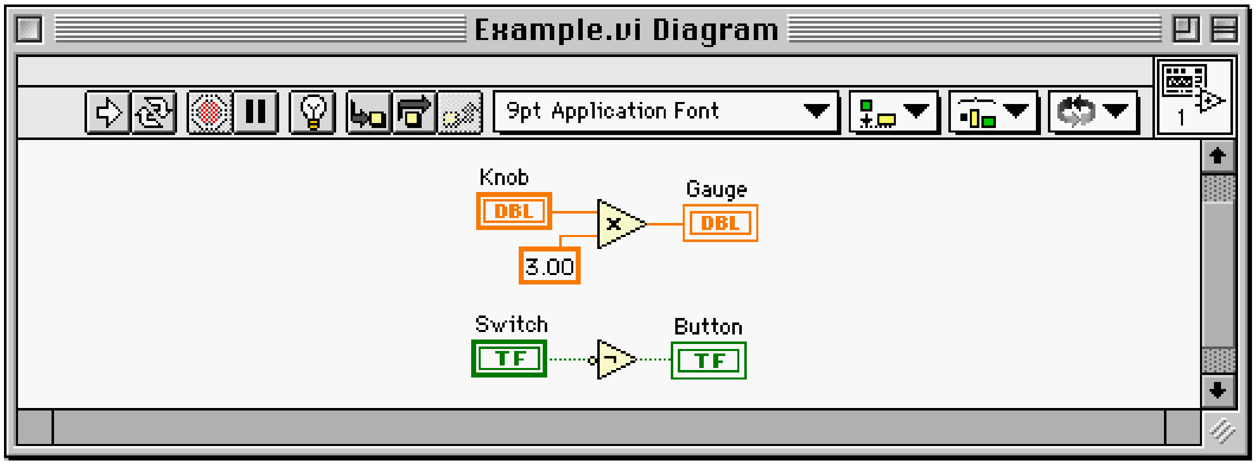


Figure 2.2 Diagram window of a VI

First, examine the buttons in those two windows.

Run When this button is pressed, LabVIEW will execute the VI that is currently selected.

Run Continuously This button will run the VI continuously until the button **Abort Execution** is pressed. It is strongly recommended to avoid the use of this button until you become familiar with LabVIEW.

Abort Execution This stops the execution of the VI.

Pause This will pause the execution. To resume it, press the button again.

Highlight Execution The bulb will be lit if you click this button. If it is on, LabVIEW will display the signal flow while running the VI. A ball running along the wire while showing the current value at each node of the wire indicates the signal flow. This is very useful when debugging a VI since you can view the value as well as the execution flow; however, the speed of execution will be much slower than normal execution.

Step Into This is a different type of **Run** button which provides debugging capability. If you click this button instead of the **Run** button, LabVIEW will step through each node of wires and each element. The difference from the **Highlight Execution** button is that **Step Into** will run one node and wait until you hit the button again, whereas **Highlight Execution** runs (slowly) without stopping.

Step Over While running a VI via **Step Into**, this button allows you to skip any node or element.

Step Out This exits the **Step Into** run mode.

9pt Application Font **Text Settings** This allows you to change the feature of fonts such as color, size, etc., with this menu ring.

Align Objects With this, you can vertically align objects in the windows. First, select the objects you want to align vertically, then select one of the modes from the pull-down menu.

Distribute Objects With this you can align objects horizontally. This is done by first selecting the objects and then choosing a mode from the pull-down menu.

Reorder With this feature, you can bring objects to the front, or to the back, or move them to the top or under the other objects. This is useful when multiple objects are overlapped and you are trying to change their visibility on the screen.

2.2 Objects in VI: Controls and Indicators

In the previous section, the word *Objects* was used in relation to VIs, and it will be defined in this section. As discussed earlier, LabVIEW launches with the front panel and the diagram window. Consider the spectrum analyzer as an example to explain objects in the front panel and the diagram windows. The analyzer may have switches to select different time bases or different sampling rates on the panel with a display of the spectrum. If you open the case, you will see wires, Fast Fourier Transform (FFT) chips, proper display units, and so on. In LabVIEW, these are referred to as *objects*, and they are the fundamental elements that constitute a VI.

In LabVIEW, there are two types of objects: controls and indicators. Controls are objects such as knobs, levers, or switches that allow users to enter or set values. Indicators are for display purposes only and include graphs, gauges, or meters. A simple VI that simulates a voltage source and a voltmeter is shown in Figure 2.3 and Figure 2.4. There are two objects in both the front panel and the diagram window: **Input Voltage** and **Voltmeter**. The front panel in Figure 2.3 shows you that the **Input Voltage** is a slide input switch, and the **Voltmeter** is a gauge that displays the input voltage. Figure 2.4 then shows the wire connection of these two objects. In this case, the **Input Voltage** is the control and the **Voltmeter** is the indicator since you can *set* values to the **Input Voltage** and *display* them with the **Voltmeter**. However, just looking at these two objects in the front panel will not tell you which is the control and which is the indicator. When trying to distinguish controls from indicators, remember the following statement:

> If you are in doubt about any object, right click on it to bring up the pop-up menu. On Macintosh platforms, right clicking the mouse corresponds to the Command-click.

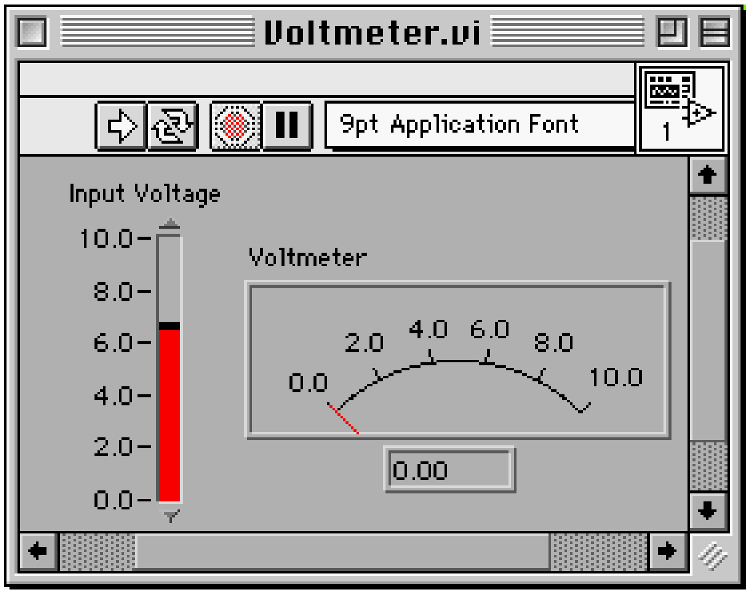


Figure 2.3 **Voltmeter.vi** (front panel)

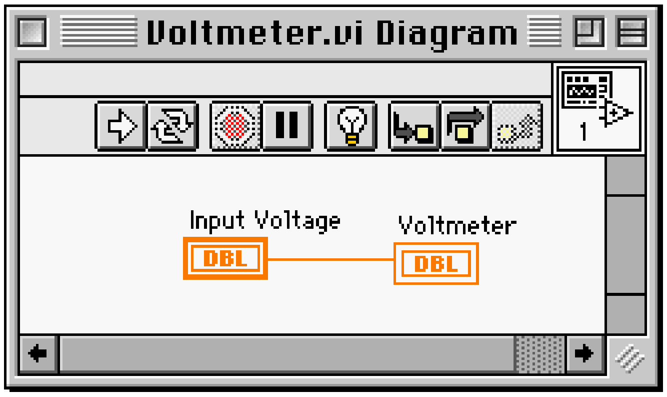


Figure 2.4 **Voltmeter.vi** (diagram window)

If you right click on either the **Input Voltage** or the **Voltmeter** in the front panel, the first option in the pop-up menu will be either **Change to Indicator** or **Change to Control**. If it has **Change to Indicator**, it means that the object is a control; otherwise, it is an indicator. However, in the diagram window in Figure 2.4, you will see an easier way of telling them apart. The **Input Voltage** and the **Voltmeter** have DBL in the box in the diagram window and they are connected through a wire. The letters DBL imply that the data type of those two objects is double precision, which you can change to any other types such as extended precision, single precision, or integers. (Can you guess how to change it?) The thickness of the box tells you the type of the object: thicker lines indicate controls and thinner lines, indicators. Now, you may ask the following question:

Why is it important to know the type of objects?

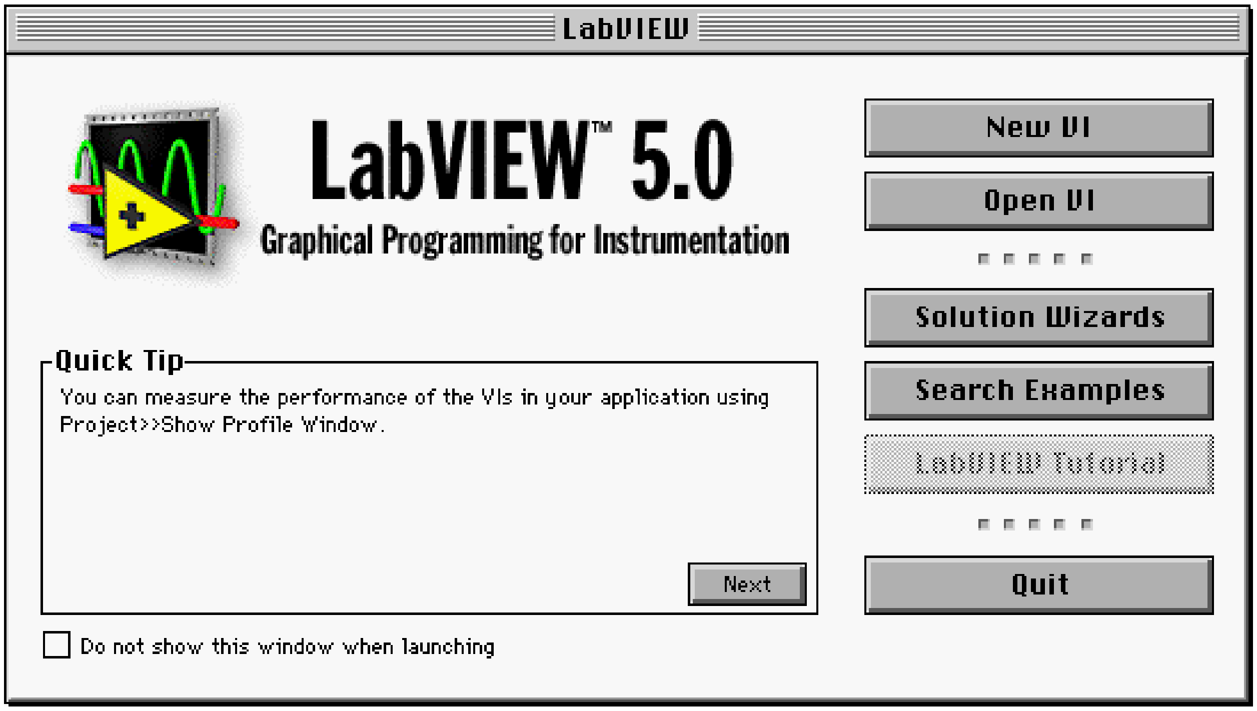
Recalling that VIs represent virtual instruments, what happens if you have two displays connected together? Assuming that displays can only receive data and *display* them on the screen, connecting two displays would be meaningless. The same logic applies to a situation where you have two knobs with which you set values. (Again, you are assuming that switches cannot receive data even though some real switches can receive data and display them.) If you try to wire two (or more) controls together, LabVIEW will return an error and will not execute the VI. This also applies to indicators. Surprisingly enough, one of the major errors that beginning LabVIEW programmers make results from this type of conflict!

One major error made by beginning LabVIEW programmers results from the following wiring conflict: controls or indicators are wired to their own kind.

In debugging with text based languages such as C or C++, simple mistakes such as spelling errors are usually only discovered after days of painful effort. Spelling errors are simple, but their simplicity causes programmers to be careless, and, therefore, make such mistakes. Should you have an error in LabVIEW, the **Run** button will show up as a Broken Run Arrow. If you click on the Broken Run Arrow, it will display a list of errors. To locate the origin of each one, double clicking on each error will bring you to the location. Make sure that you are not wiring controls to controls or indicators to indicators.

List Errors (Broken Run Arrow) This indicates that an error or errors exist in the VI. To find out what errors have been made, click the button and it will bring up a list of errors. If you double click on a specific error listing, LabVIEW will take you to the location where the error occurred.

2.2.1 Building an example



Quick Tip
You can measure the performance of the VIs in your application using Project>>Show Profile Window.

☐ Do not show this window when launching

Figure 2.5 Startup window of LabVIEW 5.0. (Macintosh platforms) This book uses the names and VI paths of version 5.0. However, all of the topics and examples in this text are carefully chosen to be *independent* of any specific version, either lower or higher than 5.0. The locations of the VIs or functions in your LabVIEW may be different from those in this book. If so, refer to the information about the differences between different versions, which is provided in your LabVIEW package.

In this section, you will build a simple example step by step. If you have not yet done so, invoke LabVIEW by double clicking its icon. It will come up with the startup window as shown in Figure 2.5. Note that LabVIEW tutorial is only available on PC platforms; therefore, it will be grayed out on Macintosh platforms as shown in Figure 2.5. Click on **New VI**, and this will bring up two windows as shown in Figure 2.1 and Figure 2.2. Go to the front panel by selecting the window with the gray background. Right click on any place in the gray window to bring up the **Controls** palette. You will see many icons (sub-palettes) that will be studied later. When the **Controls** palette comes up, fix it to the desktop by pointing the mouse arrow to the thumb pin located in the upper left corner of the palette. Once you fix it, point the mouse arrow to the first sub-palette **Numeric**. (The label will show up as you point the mouse to the sub-palettes.) Click the sub-palette to open it, and fix it using the thumb pin in the manner described earlier.

In the **Numeric** sub-palette, you will see many different types of slides, gauges, and knobs. Place a **Vertical Fill Slide** in the front panel by clicking it once and dragging it to the front panel. When you put it in the front panel, it will be ready to take the label from the user. (The label is already selected.) Type `Input Voltage` and hit return. To hide the digital display, right click on the **Vertical Fill Slide** and uncheck **Show** >> **Digital Display**. If you clicked outside the label without typing in `Input Voltage`, you will realize that the label has disappeared. You can show it by right clicking on it and selecting **Show** >> **Label**. Once it shows up, you can then enter the label. Similarly, place a **Meter** from the **Numeric** sub-palette next to the **Vertical Fill Slide** and label it **Voltmeter**.

Now, go to the diagram window by pressing **Ctrl-E**. This shortcut is very useful when navigating between the front panel and the diagram window. However, you can do the same thing by selecting either **Show Diagram** or **Show Panel** under the **Window** pull-down menu. The importance of this shortcut is further emphasized in a later section. Until then, try to get used to the shortcuts that will be introduced in this section without questions.

Use shortcuts! **Ctrl-E** toggles between the front panel and the diagram window.

When you get to the diagram window, you will see two rectangular boxes (objects) with DBL in them. DBL indicates that the items (in this case, the **Input Voltage** and the **Voltmeter**) are double precision data type. If the labels are not shown, right click on the objects and select **Show** >> **Label** to make them appear in the diagram window or in the front panel. If you followed the instructions correctly until now, your mouse pointer on the screen should be a Hand Tool. Regardless of the tool that is currently selected, go to the pull-down menu **Windows** and select **Show Tools Palette.** Now select the Wire Tool from the **Tools** palette, and connect the **Input Voltage** to the **Voltmeter** using the Wire Tool.

Operate Value (Hand Tool) **Connect Wire** (Wire Tool)

Note that wiring technique is another factor which causes most of the errors that the beginning LabVIEW programmers make; therefore, the correct wiring method will be discussed in the following section separately.

2.3 Wiring Technique

Beginning LabVIEW users demonstrate a tendency to try wiring items together by holding the mouse button after clicking the first item and releasing it on the second one. If you follow this method, LabVIEW is just waiting to cause you problems. Fortunately, the correct wiring technique is simpler than this instinctive process. The following subsection discusses the correct way of wiring objects in the diagram window.

2.3.1 Wiring steps

There are three steps in wiring: click the origin, move to the target, and click the target. Note that the first step says *click* on the origin, not *click and hold* the mouse button on the origin. Now, if you have not yet done so, select the wire tool from the **Tools** palette and refer to Figure 2.6. This illustrates the first step where you click the origin. When you place the wiring tool on the origin, **Input Voltage**, which is a control, the icon will start blinking to indicate that it has been selected. Once it starts blinking, click once and release the mouse button.

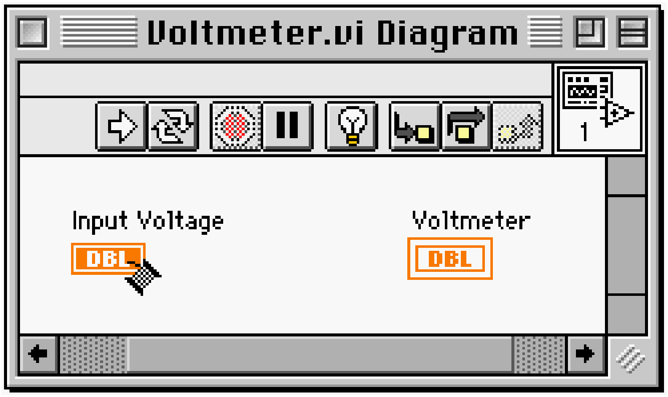


Figure 2.6 Wiring Step 1

After completing the first step, drag the mouse. You will now see a dotted line attached to the Wire Tool. If your target is located in a place where you cannot reach it with a straight wire, you can set the break point by clicking once at a point where you want to make a turn. For now, drag the Wire Tool onto the target **Voltmeter**. This second step is shown in Figure 2.7.

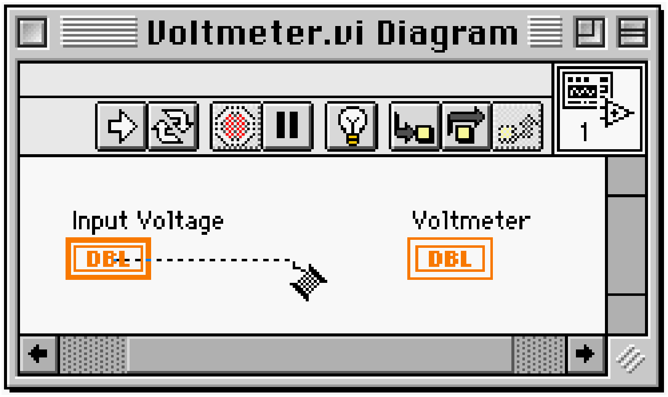


Figure 2.7 Wiring Step 2

The final step of completing the wiring is shown in Figure 2.8. When you place the Wire Tool on the target **Voltmeter**, it will start to blink indicating that the wiring tool is right on the target. Once you confirm the blinking, click the mouse button once and release it; this will complete the wiring.

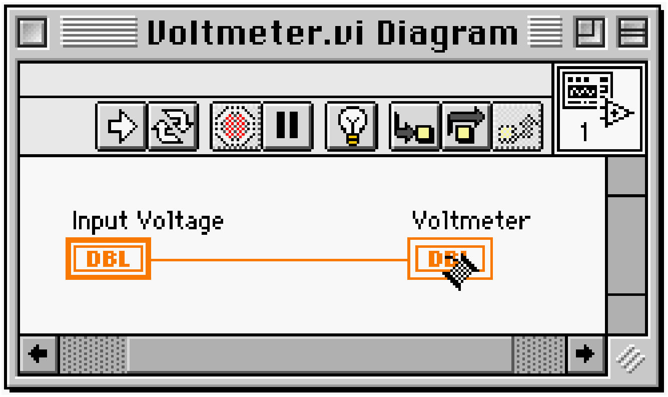


Figure 2.8 Wiring Step 3

Good wiring habits will help LabVIEW programmers write well-organized G programs and avoid a great deal of trouble. If nothing seems to be wrong with your VI, but it still gives you the broken run arrow, check if you wired controls to controls or indicators to indicators. Next, check for bad wires and make sure they are wired to the correct targets. These two trouble shooting methods should eliminate most of the errors typically made by beginning LabVIEW programmers.

> If nothing seems to be wrong with your VI but it returns a broken run arrow, first check for an object type conflict and then check for bad wires.

Once you finish wiring, go to the front panel and switch the Wire Tool to the Hand Tool. Now assign a value to the **Input Voltage** and click the **Run** button to see if the value is displayed on the gauge.

2.3.2 Shortcuts

In this section, you will be introduced to more useful shortcuts in LabVIEW and understand why they are important. Some shortcuts will seem inevitable to be used and, therefore, users are strongly recommended to use them. First, an explanation of **Ctrl-E** and its significance.

Assume that you have multiple VIs with their front panels and diagram windows both on screen. As a programmer, there may be situations where you assign similar names to VIs such as Test 1, Test 2, and so on. If you simply jump between the front panels and the diagram windows by mouse, you may end up modifying the wrong diagram window by mistake. Of course, you can use the **Windows** pull-down menu to jump back and forth, but this will slow down programming and even become annoying as you develop into an experienced user. However, if you use **Ctrl-E**, you will quickly jump to the correct diagram window or front panel and thereby enhance your programming efficiency. The following is a list of useful shortcuts in LabVIEW:

> **Ctrl-E**: Toggles between the front panel and the diagram window. (**Command-E** on Macintosh platforms.)
>
> **Ctrl-B**: This will remove all of the bad wires appearing as dotted lines. (**Command-B** on Macintosh platforms.)
>
> **Ctrl-H**: While pointing at an object, use this shortcut to bring up the online help for the object. (**Command-H** on Macintosh platforms.)

Ctrl-Shift-right click: This will pop up the **Tools** palette. (**Command-Shift**-click on Macintosh platforms.)

Tab key: This will shift between Hand Tool, Arrow Tool, Text Editor, and Wire Tool. See section 2.4 for more details about those tools.

Space bar: This will toggle between Arrow Tool and Wire Tool. See section 2.4 for more details about those tools.

Ctrl-Z: Undo shortcut. You can change the maximum undo steps per VI under **Edit** >> **Preferences** >> **Block Diagram**. LabVIEW allows a maximum of 99 steps of undo. This is a new feature in LabVIEW 5.0. (**Command-Z** on Macintosh platforms.)

Ctrl-drag: This creates a duplicate of objects. Using the Arrow Tool (**Position/Size/Select** in the **Tools** palette), select an object or multiple objects in the front panel or in the diagram window and **Ctrl-**drag the selected object(s). This will create a duplicate. (**Option**-drag on Macintosh platforms.)

To find out which objects in the diagram window correspond to those on the front panel, right click on the object and select **Find Terminal**. If you want to find a corresponding object in the front panel, double click on the object in the diagram window, and LabVIEW will take you to its counterpart in the front panel.

2.3.3 Creating and deleting objects

To create or place an object either in the front panel or in the diagram window from either the **Controls** or the **Functions** panel, first right click in the place where you want to place the object (the front panel or the diagram window). This will pop up either the **Controls** or the **Functions** panel. Then, select an object and drop it in the proper window.

To delete an object, you must go to the place where the object was originally created. Once you are in the correct place, select an object or multiple objects using the Arrow Tool and hit the **Delete** key. This will remove the selected object(s). For example, if you created a control or an indicator in the *front* panel, you must go to the *front* panel to delete it; it cannot be deleted from the diagram window. Similarly, if you placed a VI such as **Add** from **Functions** >> **Numeric**, you must go to the *diagram* window to delete it. To select more than one object using