LabVIEW Tutorial

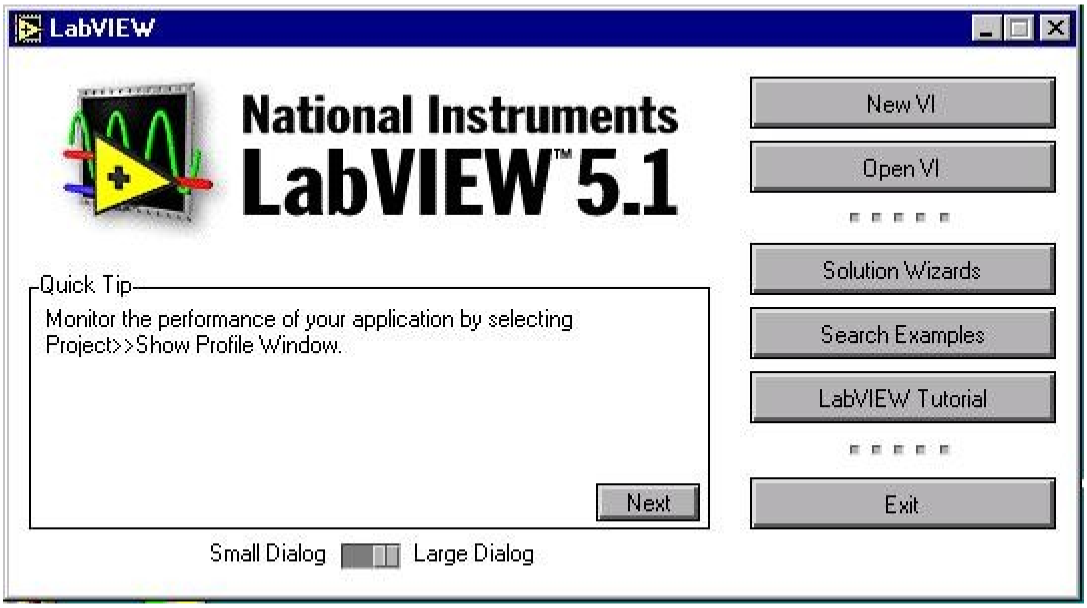
Note: This tutorial was created using LabVIEW version 5.1. Other versions may differ in what features and options are available.

Introduction

LabVIEW is a program used to automate testing and data gathering. It is basically a graphical programming language in which the user can set up the program to manipulate and store data. The rest of this tutorial is a basic introduction to LabVIEW and to the features available. This is meant only as an introduction and you are encouraged to explore other features of this powerful program independently.

Front Panel

When you start LabVIEW, you will see a screen with a few options on it.

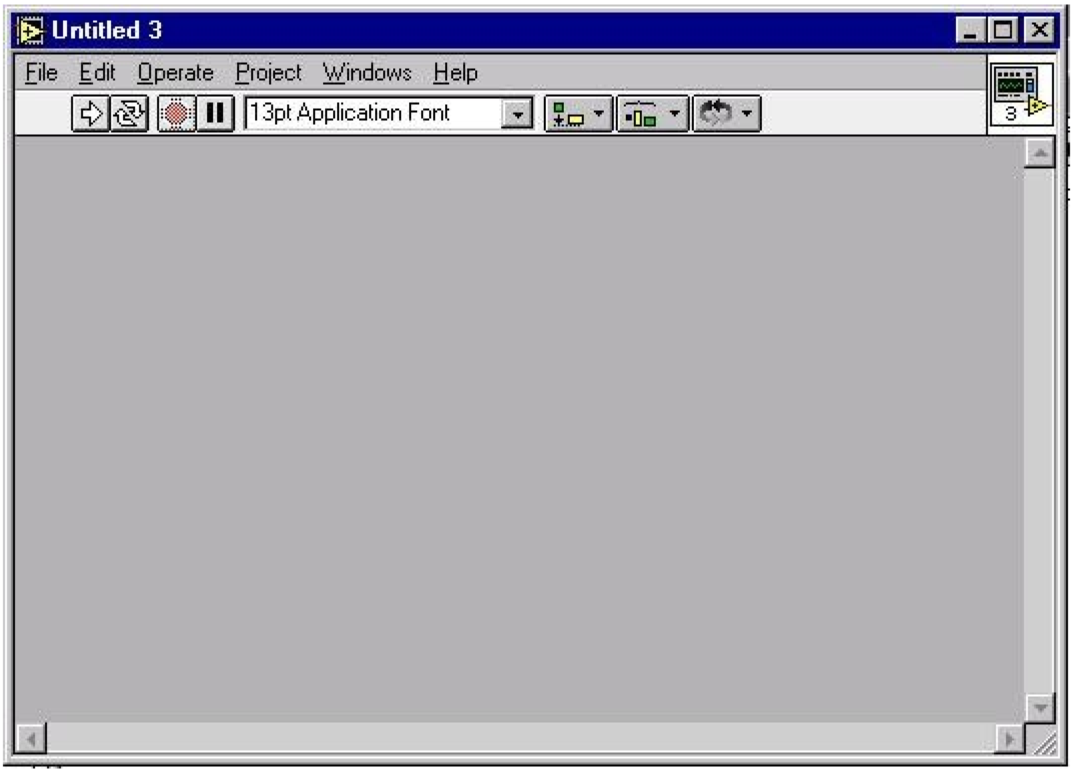


Quick Tip

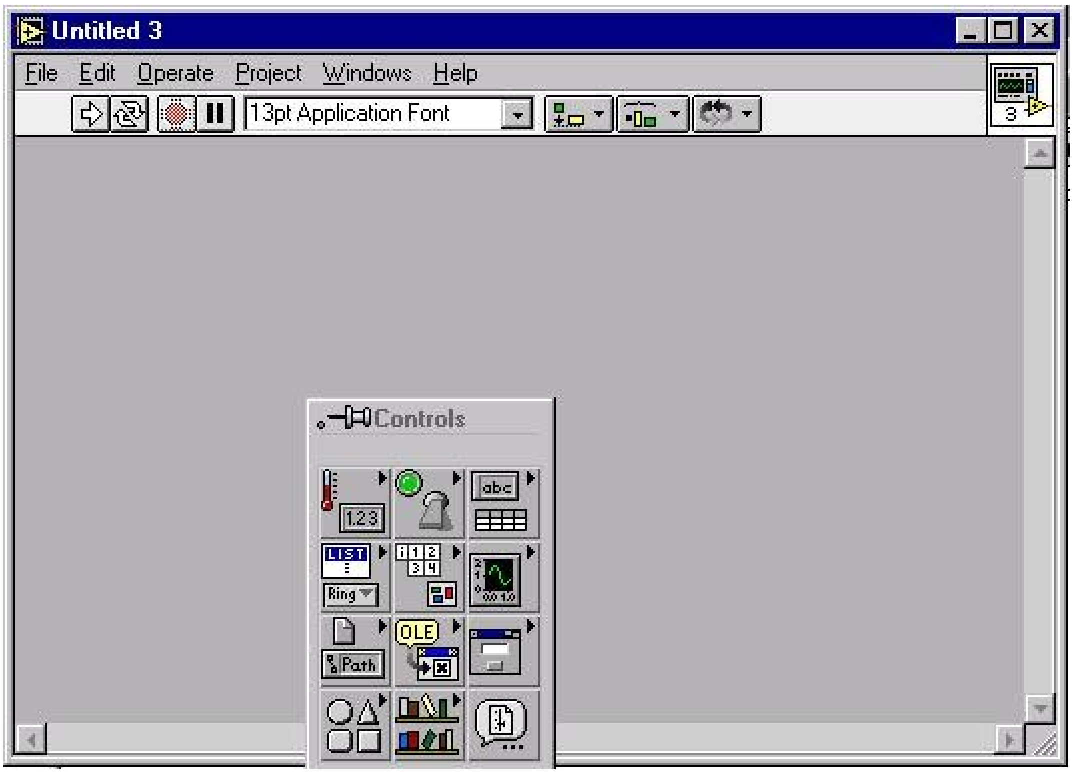
Monitor the performance of your application by selecting Project>>Show Profile Window.

Two of the options are "New VI" and "Open VI". VI's are the programs you create in LabVIEW. Later you will probably want to open a previously created VI, but for now select New VI.

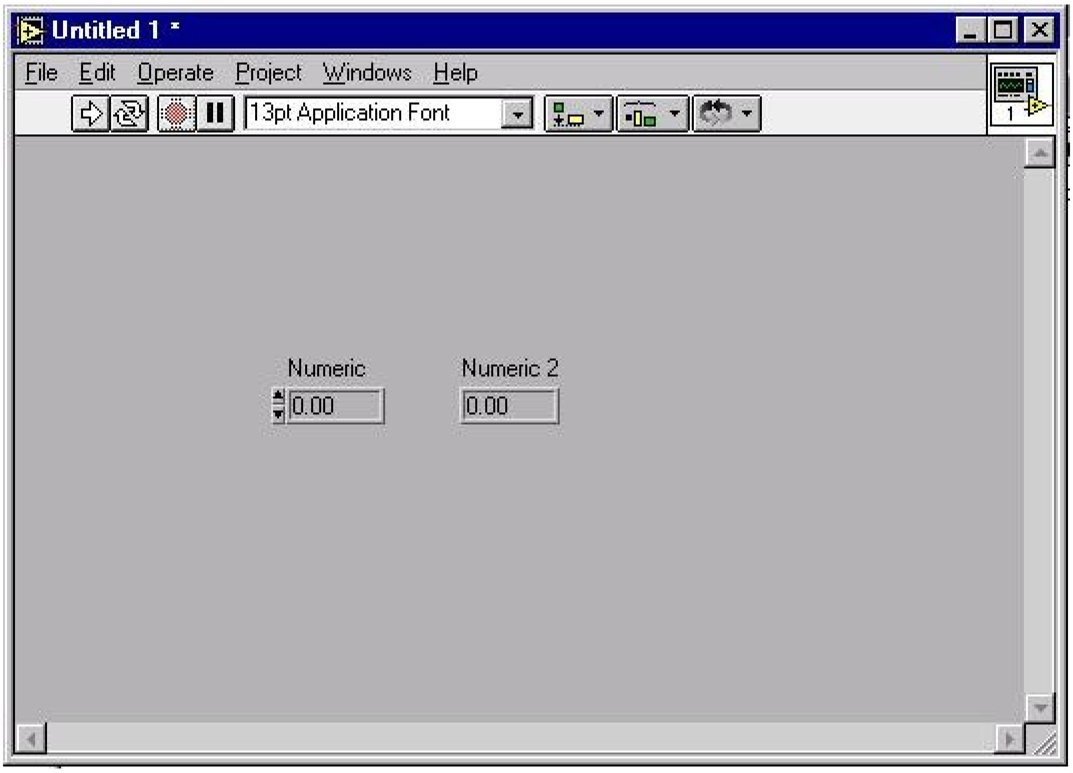
Two new screens will now pop up. One of them looks like this.



This is called the Front Panel. This is one of the places where the user will be able to input data to the program and view results. (Data I/O can also occur from files or through devices such as D/A converter boards.) When creating a program, you will first need to decide what sort of inputs and outputs will be available. Put the cursor over the gray area in the front panel and right click. A window called Controls should pop up.



In this window there are various menus with input and output controls on them. There are numeric, Boolean, string, array, graph, and other controls. For now go to the Numeric control menu and select a Digital Control. Place it anywhere on the control panel. Now do the same with a Digital Indicator.

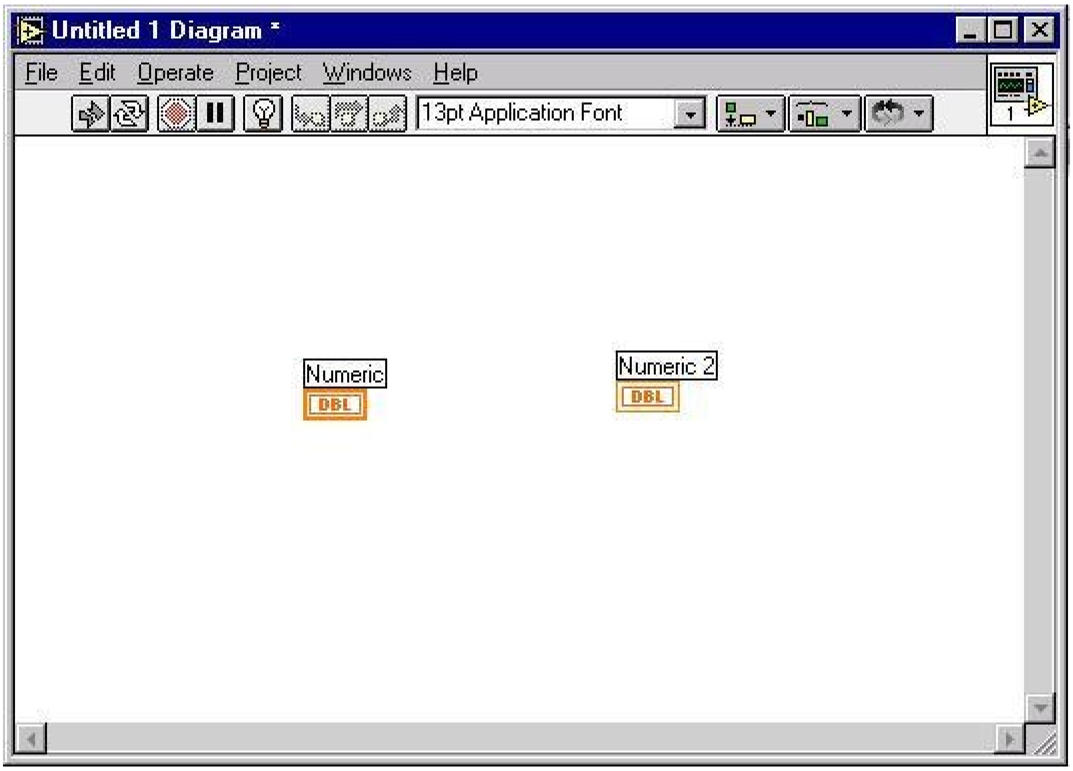


You will now be able to use both when you create your program. This bring us to the next window.

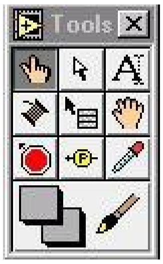
Block Diagram

The second window is the place where you create the underlying code for your program. You will create the program graphically using the inputs and outputs created in the Front Panel and objects from the Functions window.

At the moment, your Block Diagram should look like this.

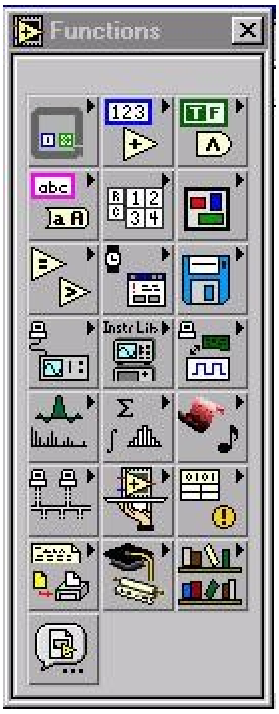


Let's take a closer look at the two menus available. First, the Tools menu is available when working on both the Front Panel and the Block Diagram.



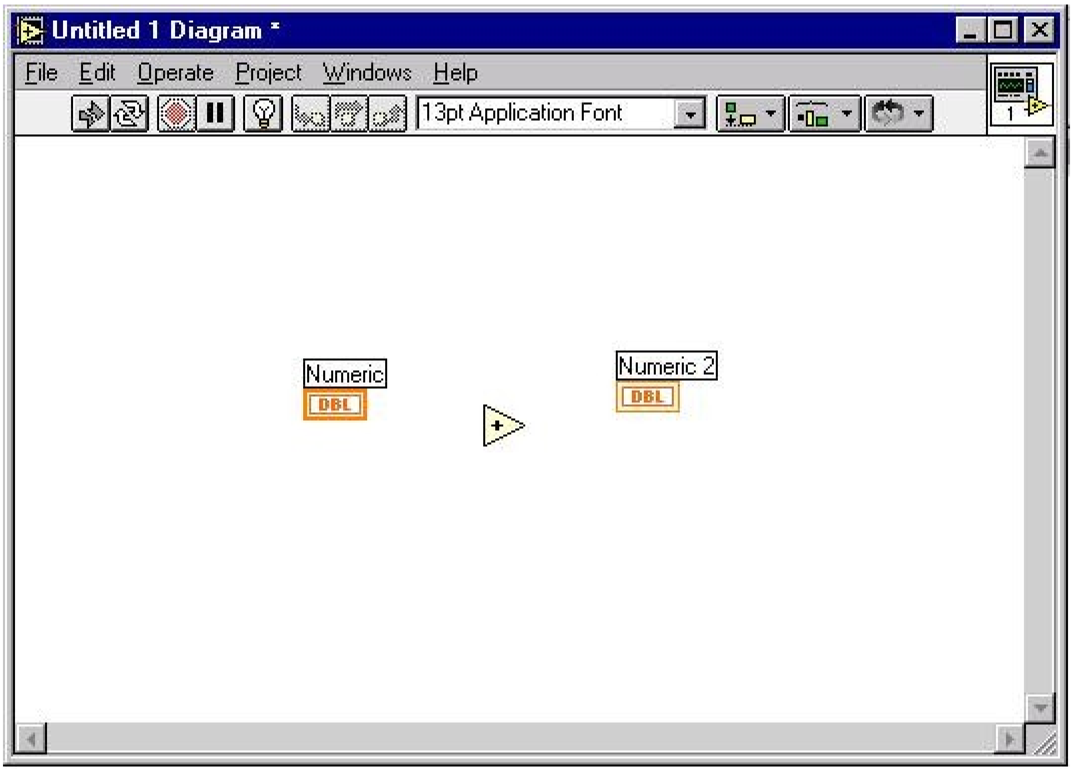
The Tools menu allows you to change the function of the cursor. For example, in one mode the cursor can be used to change values, and in another it can be used to move and size items. If you let the mouse cursor hover over a button on the Tools menu, a description of what that button does will appear.

Next is the Function menu. This is available only when working on the Block Diagram.



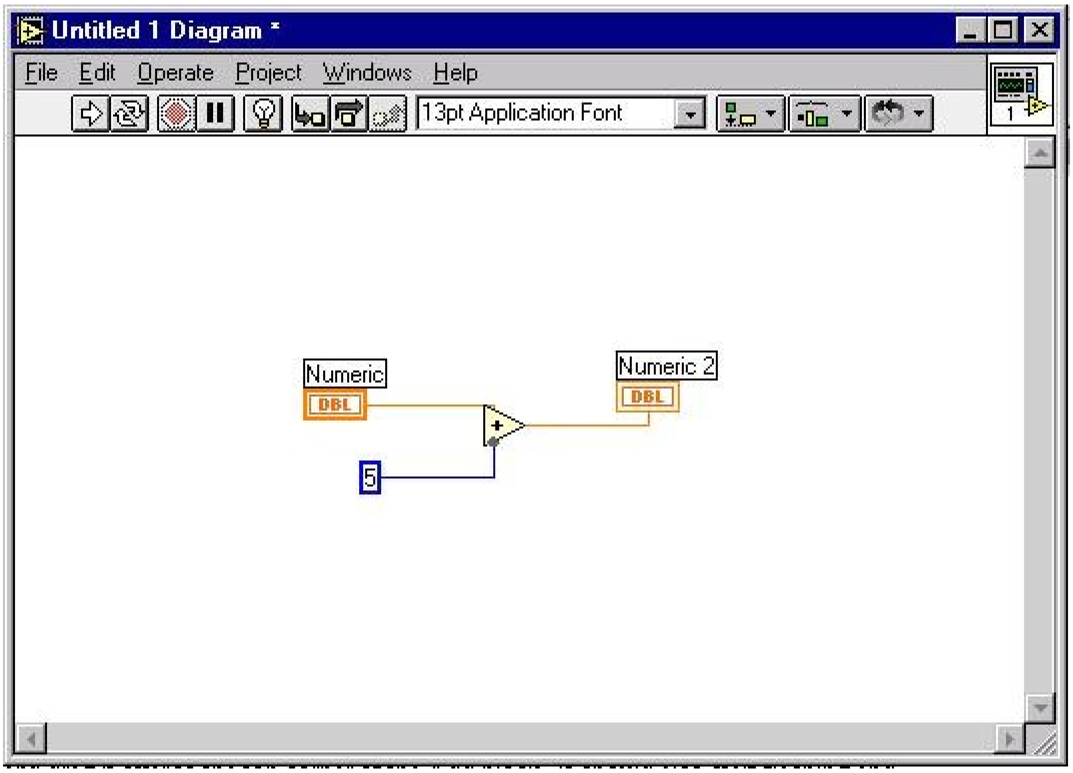
This window contains many menus and sometimes submenus for creating your program. There are a great many functions available and this tutorial will only touch on a few of them. For more information on what each function button does, refer to the Simple Help popup that can be enabled from the Help menu or the more complete online help also available in the Help menu. The complete documentation is also available online at http://www.natinst.com/manuals.

For now we will just create a very simple program. If you look at the Block Diagram window you will notice that there are two blocks already present, Numeric and Numeric2. These are representations of what is on the Front Panel. Remember that Numeric is a Digital Control while Numeric2 is a Digital Indicator. This means that Numeric will probably be an input value and Numeric2 will be an output value. The first thing we will try to do is add a constant to Numeric and display the result in Numeric2. Go to the Numeric menu on the Function window, and select the Add block. Place this block between Numeric and Numeric2 on the Block Diagram window. You might want to use the Position/Size/Select cursor (the arrow in the Tools menu) to move Numeric and Numeric2 farther apart.



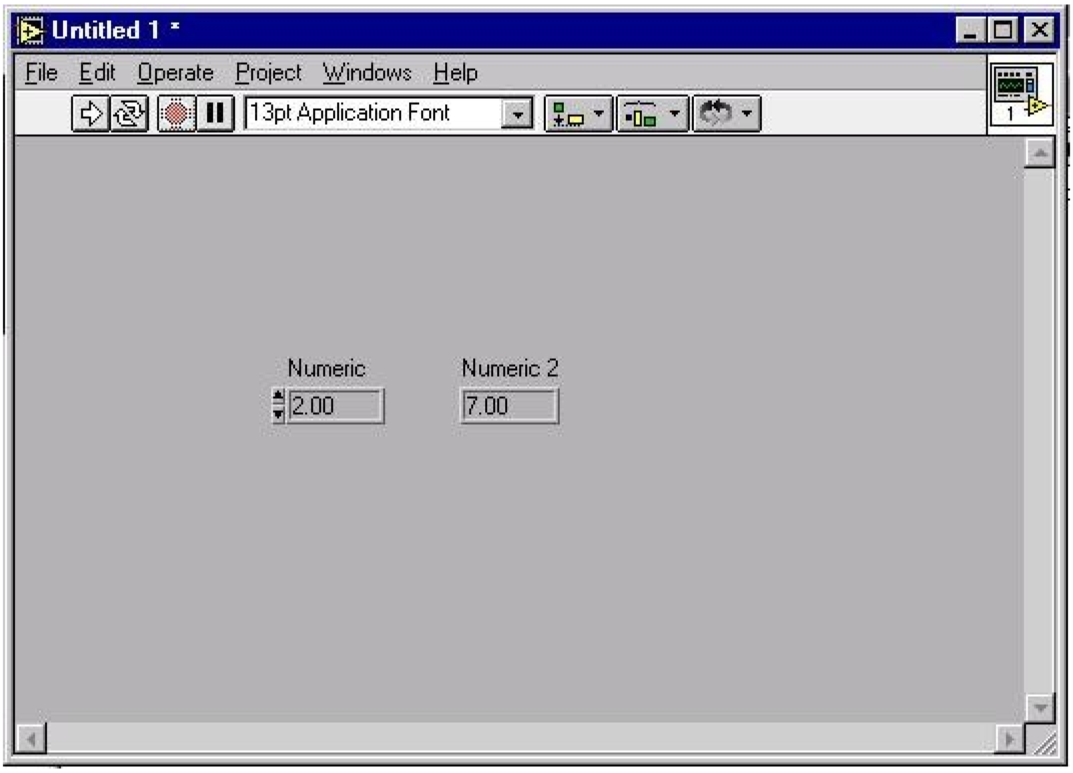
Now, go back to the Numeric menu and select Numeric Constant. Go to the Tools menu and click on the Operate Value button (the pointy finger). Now click on the constant and change the value from 0 to 5.

All the blocks are in place to input a number from Numeric and output that number plus five to Numeric2. Now we need to wire together the blocks. Select the wiring tool from the Tools menu. Move the wiring tool over Numeric in the Block Diagram window. The numeric block should start flashing. Click and hold down the mouse button and drag it over to the top corner of the Add block. It should also start flashing and an "x" should pop up on the screen. Release the mouse button and the wire will be connected. Do the same to connect the constant five to the bottom input of the Add block, y. Then connect the output, x + y, to Numeric2. When finished, the Block Diagram should look like this.



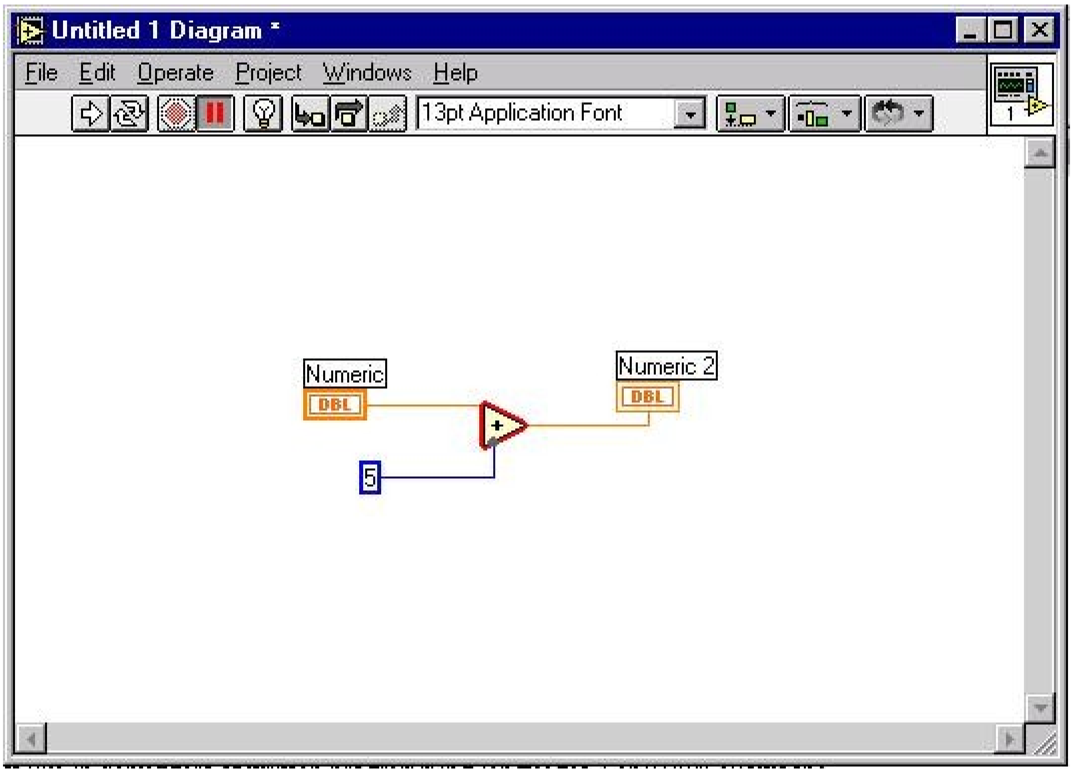
If you mistakenly left any wires unattached, they would appear as dotted block lines instead of solid colored ones. If there are any unconnected wires the simulation will not run. Use the position tool to select and delete unconnected wires and then use the wiring tool to reconnect them properly.

Now, go back to the Front Panel. Click on the Operate Value tool in the Tools menu and use it to set Numeric to 2. Then click on the arrow icon at the left of the icon bar on the top of the Front Panel. This will cause the program to run.

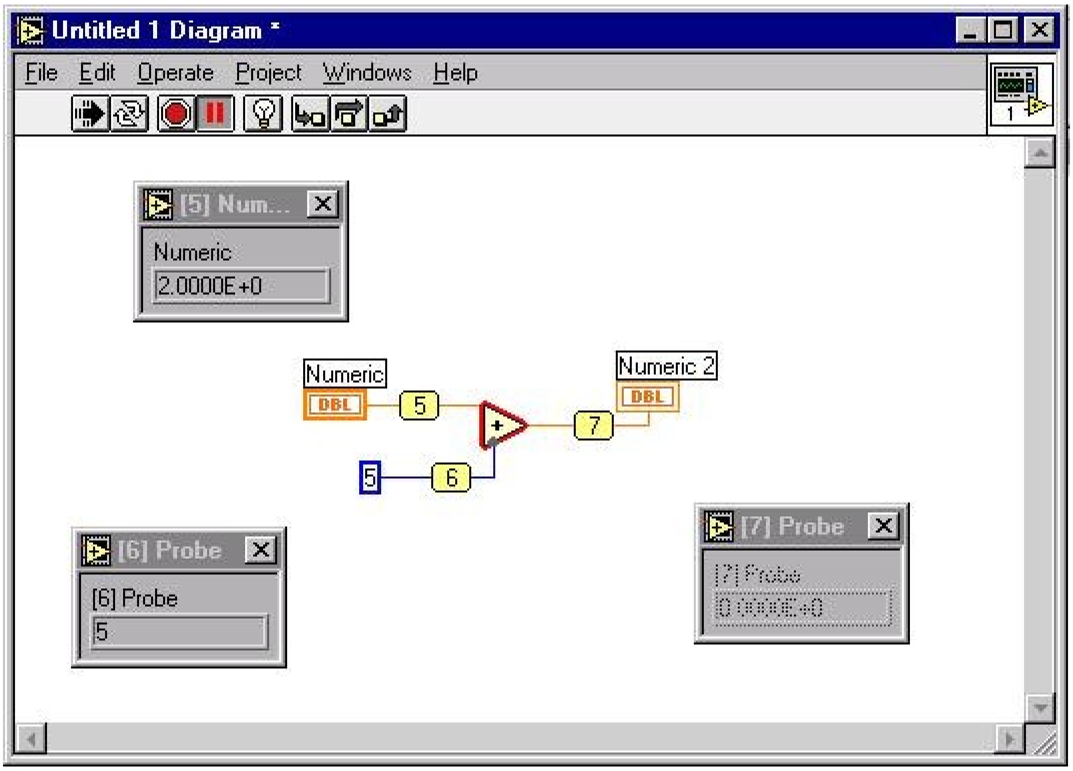


Now try clicking on the button next to the arrow. This will cause the program to run continuously. Try changing values while the program is running. The output value will change as you change the input.

Go back to the Block Diagram. As the programs you create get more complicated, it may be hard to figure out where errors are. Just like in debuggers of written programming languages, LabVIEW allows the programmer to set breakpoints and examine data values inside of the program. Click on the Set/Clear Breakpoint button (red dot) on the Tools menu and then set a breakpoint at the Add block by clicking on it. The block should now be outlined in red.



Now, click on the run button while still on the Block Diagram window. The program will start and then stop and the Add block will start flashing. Select the Probe tool from the Tools menu. Now click on the three different parts of the Add block. Three windows will pop up with the values of each part of the block.

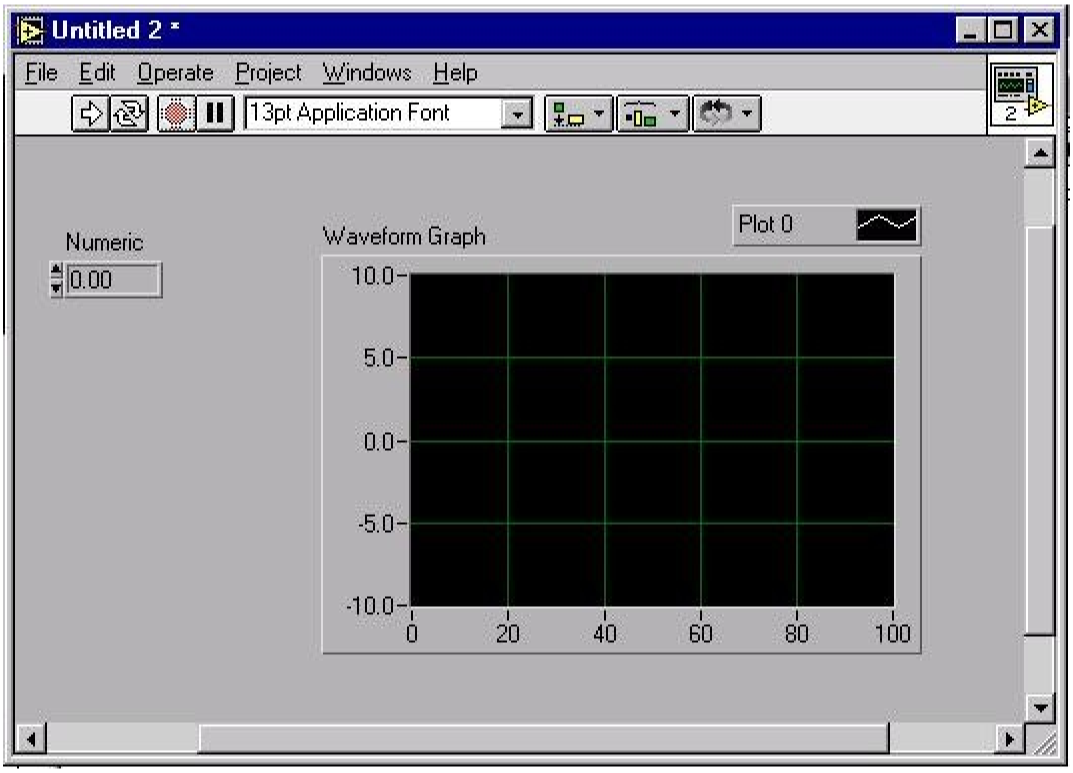


As you can see, x = 2 (or whatever value you currently have in the Digital Control) and y=5, but x + y is still undefined because the program has stopped at the Add block. Click on the run icon again to let the program finish. Now x + y has the appropriate value.

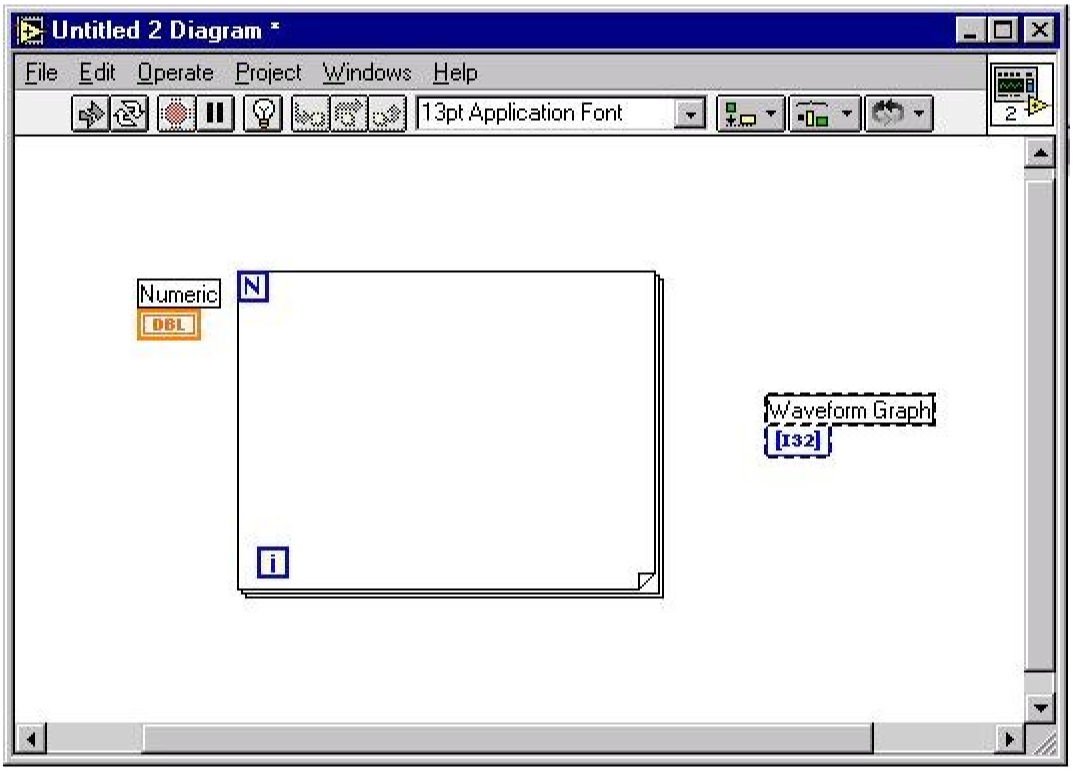
Other Features

We will now create a VI that will incorporate some of the other features of LabVIEW including structures, arrays, graphs, and file I/O. First, close the current VI and create a new one.

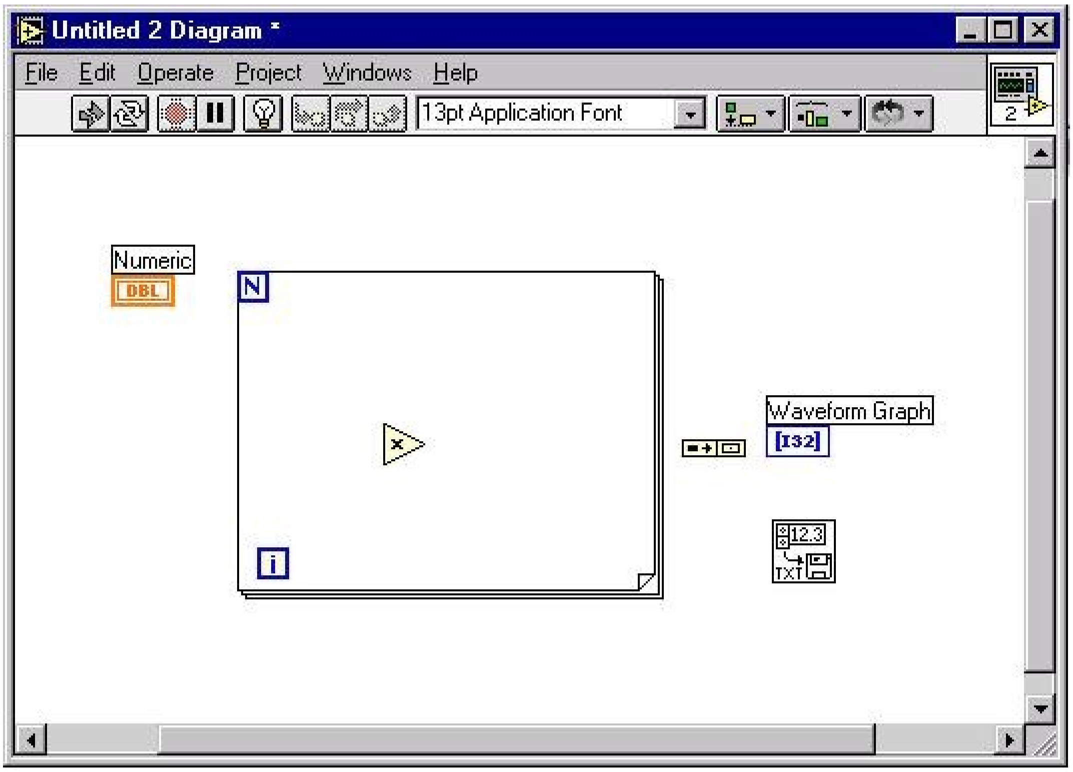
Now, go to the Front Panel and select a Digital Control. Then place a Waveform Graph on the Front Panel. The window should now look like this.



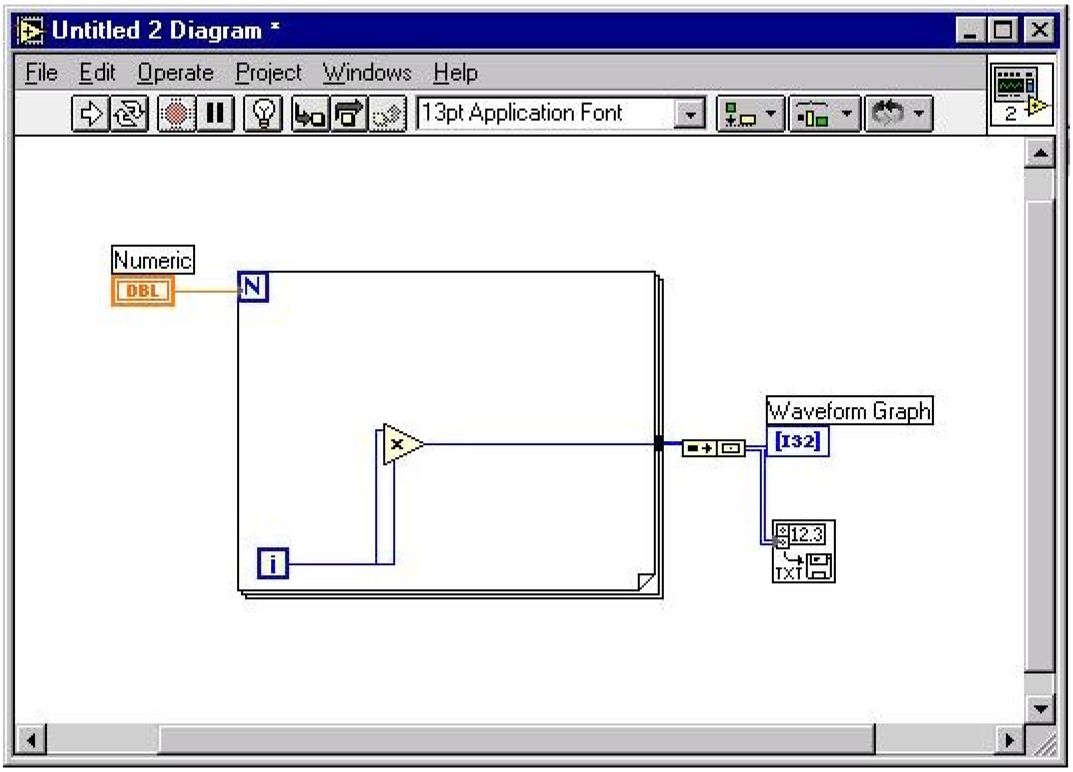
Go to the Block Diagram. There should be two icons, one for the Digital Control (Numeric) and one for the Waveform Graph. Now go to the Functions window, and select a For Loop from the Structures menu. Make the block about 1/4 of the screen or so and then arrange the block as shown below.



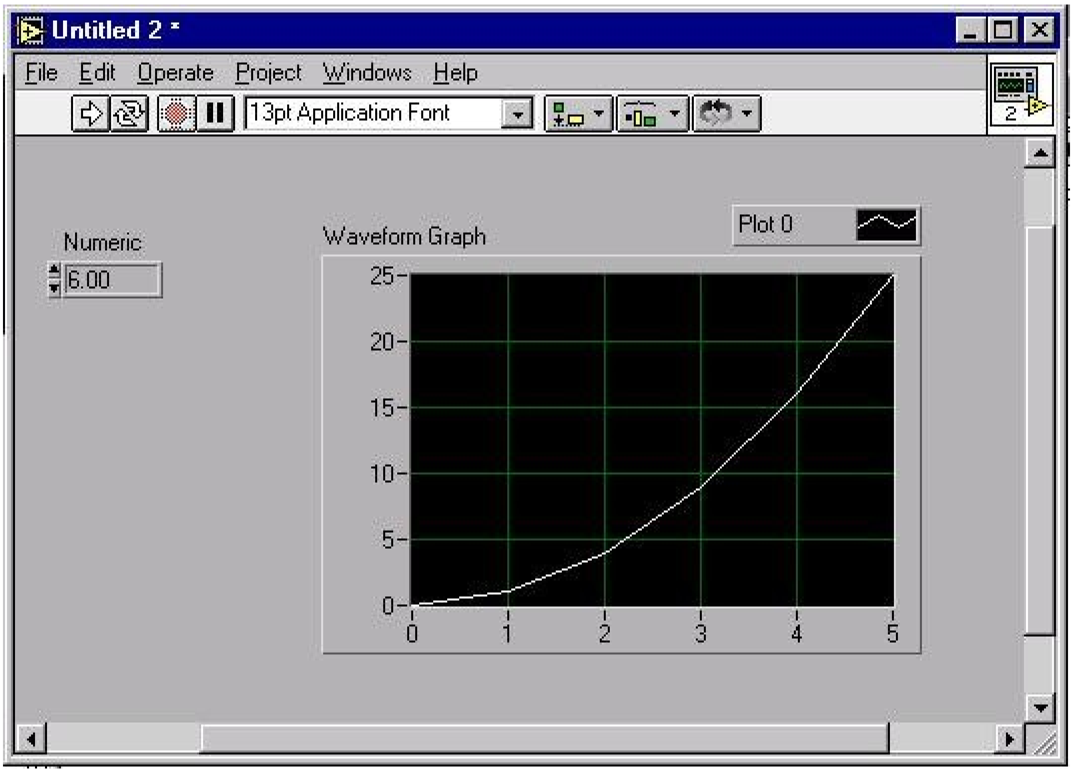
Now, for the rest of the blocks we will need a Multiply (under the Numeric menu), a Build Array (under the Array menu), and a Write to Spreadsheet block (under the File I/O menu). Place these blocks as show below.



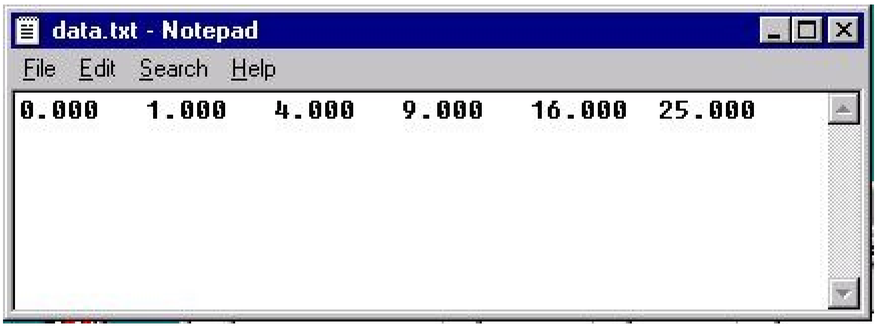
Now we need to wire the blocks together. Connect the Numeric block to the N on the For Loop. This means that the number inputted in the Digital Control will control how many times the For Loop runs. Connect the i in the For Loop to both inputs of the Multiply. The For Loop variable i keeps track of the iteration that the loop is on. It begins at zero so it will go through the range of 0 to N-1 where N is the number of iterations. Therefore, the output of this multiplier is i^2 and ranges from 0 to $(N-1)^2$. Next wire the output of the Multiplier to the input of the Build Array Block. This will take the different values for each iteration of the loop and place them in an array. You need to do this to sort the data so that it can be graphed or stored. Next connect the output of the Build Array to the Waveform Graph and the 2D Data of the Write to Spreadsheet block. The diagram should now look something like this.



The program is now set up to calculate the squares of the integers from zero to one less than the inputted number. It will then represent them in two forms, a text file and a graph. Put the number six in the Digital Control and run the program. The computer will prompt you for a file name and a place to save the file. Give it a name like demo.txt and save it.



As you can see, the waveform is graphed and the graph goes from zero to one to four and so on up to 25. The points are connected by default but you can change those options by clicking on the graph. Now open the file that you saved.



```
0.000    1.000    4.000    9.000    16.000   25.000
```

The results have been recorded into a text file. This is a very useful alternative to recording data by hand.

As you can see, LabVIEW is a very powerful tool which can simplify experiments by intelligently processing and presenting data. There are many more features than those described in this tutorial, but hopefully this will give you a background to build off of. This concludes the LabVIEW tutorial.