Update
for
Advanced LabVIEW Labs

Changes Needed*
when using
LabVIEW version 6.1 (v6.1)

*Book originally written in LabVIEW version 5.0 (v5)

by
John Essick
Reed College
10/25/2002
Chapter 1

Menu Items:

In v6.1, **Tools** and **Browse** replace **Project** from v5. **Windows** from v5 is renamed **Window**.

Functions Palette:

The Functions Palette for v6.1 is shown below. If you simply click on a subpalette button such as **Structures**, the subpalette appears and the Functions Palette disappears as shown below. To return to the Functions Palette, click on the arrow in the top left corner of the **Structures** subpalette.
To navigate through the Function Palette as described in the book, in v6.1 click the right mouse button ("right-click") in the Windows environment or press the Command button while clicking the mouse button ("command-click") for the Mac OS. This technique is demonstrated for the Structures subpalette below.

While Loop Features:
In v5, the While Loop's conditional terminal could be left unconnected ("unwired") to anything and it would then take on a default value of FALSE. In v6.1, the conditional terminal can no longer be left unwired, but must always be connected to a Boolean input. Also in v6.1, by default, the While Loop iterates until the conditional terminal receives a FALSE as described in the book. This behavior is called Continue If True. However, this behavior can be changed to Stop If True (i.e., iterate until the conditional terminal receives a TRUE) by selecting this option from the menu that appears when "popping up" on the conditional terminal. To access this pop-up menu, right-click (Windows) or command-click (Mac) on the conditional terminal.

Help Window
In v6.1, select Show Context Help in Help menu to activate the Help Window.

Boolean Constant:
In v5, there was only one Boolean Constant and you made it TRUE or FALSE using the Operating Tool. In v6.1, Functions>>Boolean contains a True Constant and a False Constant, which are the same Boolean Constant from v5, but already pre-set to
TRUE and FALSE Boolean values, respectively. Once placed on the block diagram, either of these icons can be toggled between the TRUE and FALSE setting using the Operating Tool.

For the sine-wave exercise in the book, select a True Constant from Functions>>Boolean as shown below.

Online Reference:
In v6.1, clicking the Question Mark at the bottom of the Help Window or selecting VI, Function, & How-To Help... in the Help pull-down menu will activate the Online Reference. In the Windows environment, a LabVIEW Help window will appear that appears as shown below.
On a Mac, you may find that the default web browser will be launched on your computer and the browser will be directed to display an appropriate HTML file in the LabVIEW folder that contains the help information. You can navigate to other useful help information using the displayed links. By clicking the **Question Mark** at the bottom of the **Sine & Cosine Help Window** on a Mac, the following information is displayed by Internet Explorer.

**Waveform Chart:**

In v6.1, the Waveform Chart displays only its **Label** and **Plot Legend** by default. The Label comes pre-loaded with the text “Waveform Chart” so there is no need to type this text in using the keyboard. Additionally, the axes can be labeled with text. By default, the x- and y-axis is labeled **Amplitude** and **Time**, respectively.
In v5, the **Waveform Chart** displayed the *Palette* by default, which could be used to control numerical labeling and scaling of axes. In v6.1, the *Scale Legend* and the *Graph Palette* control these features of the **Waveform Chart**. By default, the *Scale Legend* and *Graph Palette* are hidden.

**Waveform Chart's Pop-Up Menu:**
Pop-up on the **Waveform Chart**, then toggle the *Label* on and off by selecting *Label* from the **Visible Items** palette.

Note that in v6.1 **Change to Control** is no longer the first selection in the Waveform Chart's pop-up menu. This improvement will avoid a common accidental error in v5 programming.

In v6.1, the default data range for axes labeling on the Waveform Chart is $-10$ to $+10$ and $0$ to $100$ on the y- and x-axis, respectively.
**Autoscaling:**

In v6.1, select **Autoscale Y** from the **Advanced** palette of the Chart's pop-up menu.

In v5, autoscaling was activated in the Palette. In v6.1, this feature is controlled from the **Scale Legend**. In the Chart's pop-up menu, select **Scale Legend** from the **Visible Items** palette. Once visible, the Scale Legend allows you configure many scaling properties such as text and numerical labeling. To activate autoscaling, use the Operating Tool to click on the y-axis Lock icon. The lock will close and a green light will light on the neighboring Y-Axis icon indicating that autoscaling is activated on that axis.

**Front Panel Switch:**

Choose **Push Button** from **Controls>>Boolean**. You may wish to hide its Label using its pop-up menu.

**Improving Sine-Wave Resolution:**

Pop-up on the Waveform Chart's **Plot Legend** to access described menus.
Autocreation Feature:

In v6.1, to autocreate a **Numeric Constant** for Wait (ms), pop-up on its **millisecond to wait** input and select **Constant** from the **Create** palette.
Chapter 2

p. 47  

New VI:  
In v6.1, select **New VI** under the **File** menu to create a new program.

p. 47  

Waveform Graph:  
In v6.1, the Waveform Graph's *Label* is pre-loaded with the text *Waveform Graph*. By default, only the *Label* and *Plot Legend* are visible. In the Waveform Graph's pop-up menu select **Visible Items>>Scale Legend** so that the *Scale Legend* is available for easy activation of the autoscaling feature as shown below.

p. 48  

Hiding Labels:  
In v6.1, owned labels are toggled between visible and invisible by popping up on the label's owning icon and selecting **Visible Items>>Label**.
p. 54  Tunnels:
In v5, tunnels were black rectangles. In v6.1, a tunnel indicates what type of data it contains. In the case below, the tunnel is an orange set of brackets signifying an array of floating-point numbers (brackets indicate an array and orange signifies floating-point number).

![Diagram](image1.png)

p. 61-62  Resizing Bundle:
To resize the Bundle icon in v6.1, move the Positioning Tool over the Bundle so that resizing circles appear at the top and bottom of the icon.

![Diagram](image2.png)
Then move the cursor over one of the circles to change the cursor to a resizing handle and drag the border of the node vertically or horizontally to add terminals to the node.
Chapter 3

p. 92-93  Scale Legend:

In v6.1, the X Scale Format button is in the Scale Legend. The Scale Legend can be made visible by selecting Visible Items>>Scale Legend in the XY Graph's pop-up menu.
Chapter 4

p. 110  Precision of Numeric Constant
        The default Digits of Precision for a Numeric Constant is 2. Thus when entering a number such as 0.001 into a Numeric Constant you may have to pop up on the icon and change Digits of Precision to 3 in the Format & Precision… selection in order to see the entire numerical value within the icon.

p. 120  Resizing Tool for Build Array
        To add inputs to Build Array, place Positioning Tool over icon until resizing circles appear at center of icon. Place cursor over one of these circles, then drag vertically to create as many inputs as needed.
Chapter 5

p. 138 Renaming Program within VI Library
   In v6.1, VIs within a VI Library can be easily renamed by selecting VI Library Manager… in the Tools pull-down menu.

p. 140 Assigning Terminals within Connector
   If you created Data Simulator by cloning Sine Wave, the terminals in the icon connector may retain their previous assignments. Before reassigning them, select Disconnect All Terminals in the icon pane's pop-up menu.

p. 141 Formula Node Exponentiation
   In v5, the ^ operator represented exponentiation within a Formula Node, so $X^4$ was written as $X^4$. In v6.1, the ^ operator represents the bitwise exclusive or (XOR) operation and the new operator for exponentiation is **, so $X^4$ is written as $X^{**}4$ as shown in the diagram below.
p. 162 Unwired Conditional Terminal in While Loop

In v5, if a While Loop's conditional terminal were left unwired, it would assume its default value of FALSE. In v6.1, the conditional terminal must always be explicitly wired to a Boolean icon. The conditional terminal in the block diagram of *Global Variable* then must be wired to a FALSE *Boolean Constant* as shown below.
Chapter 6

p. 172-173  Tunnel Color

In v6.1, the color of a tunnel indicates the data-type being passed into or out of a programming structure. On the block diagram for Odd?, the input tunnel on the Case Structure providing data from the Number of Points terminal is blue (indicating integer data-type) because the representation of Number of Points is I32. The output tunnel on this Case Structure is orange (indicating floating point data-type) because it is carrying the result of the Divide icon.

p. 174  Boolean Button

In v6.1, use one of the available switches such as the Push Button (rather than the no-longer available Labelled Round Button of v5) on the front panel of Even Ends.

p. 181  Formula Node Exponentiation

In v5, the ^ operator represented exponentiation within a Formula Node, so $X^2$ was written as $X^\wedge 2$. In v6.1, the ^ operator represents the bitwise exclusive or (XOR) operation and the new operator for exponentiation is **, so $X^2$ is written as $X**2$ as shown in the diagram below.

p. 183  Dot Product.vi

In v6.1, Dot Product.vi is found in Functions>>Mathematics>>Linear Algebra.
Y-Axis Scaling

In v5, the XY Graph's Palette provided control of axes scaling. In v6.1, the Scale Legend contains those controls.

Numeric Integration.vi

In v6.1, Numeric Integration.vi is found in Functions >> Mathematics >> Calculus.
In v6.1, you need to explicitly wire a False **Boolean Constant** to the conditional terminal in each While Loop on the block diagram of *Loop Timer-Data Dependency* as shown below.
You can also use single frame Sequence Structures rather than While Loops as shown below.
Chapter 8

p. 219-221  Slicing Array Using Index Array

In v6.1, **Index Array** is much smarter than in v5. Create the block diagram for *R-T Plot-Spreadsheet* as follows. First, put two **Index Array** icons on the block diagram as shown below.

Then wire Read From Spreadsheet File.vi's **all rows** output to the **n-dimensional array** input of each **Index Array** icon. The icons will then automatically resize themselves with a row (top) and column (bottom) index input as shown next.
Now you instruct each **Index Array** to slice off a desired column by leaving its row index unwired and wiring a **Numeric Constant** with the desired column number to the column index as show below. There is no need to pop up and **Disable Indexing** on the row index input when programming in v6.1.

With the addition of a **Bundle** icon, the diagram is then complete.

p. 222  General LS Linear Fit.vi

In v6.1, **General LS Linear Fit.vi** is found in **Functions>>Mathematics>> Curve Fitting.**
p. 229, 235  
Formula Node Exponentiation

In v5, the ^ operator represented exponentiation within a Formula Node, so \(\ln(R)^3\) was written as \((\ln(R)^3)\)^3. In v6.1, the ^ operator represents the bitwise exclusive or (XOR) operation and the new operator for exponentiation is **, so \(\ln(R)^3\) is written as \((\ln(R)^3)**3\).

On p. 229, the block diagram for Build H then is as shown in the diagram below.

On p. 235, the diagram is as below.
In v6.1, **General Polynomial Fit.vi** is found in **Functions>>Mathematics>>Curve Fitting**. Also, the **Enumerated Type** control of v5 is called the **Enum** control in v6.1.

**Nonlinear Lev-Mar Fit.vi** is found in **Functions>>Mathematics>>Curve Fitting**.

In v6.1, the **Unopened SubVIs** palette is found in the **Browse** pull-down menu.

**Target Fnc & Deriv NonLin.vi** is as below.

When wiring **Nonlinear Lev-Mar fit.vi**, you may wish to increase its **max iteration** input from the default of 200 to something bigger like 2000.
Chapter 9

p. 248  Real FFT.vi
In v6.1, Real FFT.vi is found in Functions>>Analyze>>Signal Processing>>Frequency Domain.

p. 250  Unopened SubVIs
In v6.1, the Unopened SubVIs palette is found in the Browse pull-down menu.

p. 258  Magnifying Glass
In v6.1, the control for the XY Graph's Magnifying Glass is in the Graph Palette. To make the Graph Palette visible, pop up on the XY Graph and select Visible Items>>Graph Palette.

p. 259  X-Axis Autoscaling
In v6.1, the control for the XY Graph's X-Axis Autoscaling is in Scale Legend. To make the Scale Legend visible, pop up on the XY Graph and select Visible Items>>Scale Legend.

p. 269  Scaled Time Domain Window.vi
In v6.1, Scaled Time Domain Window.vi is found in Functions>>Analyze>>Signal Processing>>Windows.

p. 272-273  Enum Control
The Enumerated Type control of v5 is called the Enum control in v6.1. This control is found in Controls>>Ring & Enum. Associated integers are made visible by popping up on the Enum control and selecting Visible Items>>Digital Display.
p. 277       Formula Node Exponentiation

In v5, the ^ operator represented exponentiation within a Formula Node, so \( A^2 \) was written as \( A^{\text{^2}} \). In v6.1, the ^ operator represents the bitwise exclusive or (XOR) operation and the new operator for exponentiation is **, so \( A^2 \) is written as \( A^{\text{**2}} \) as in the block diagram for "Estimated Frequency and Amplitude" as shown below.
Chapter 10

p. 283  DAQ Board Configuration Utility
For Windows machines, the configuration utility is called *Measurement and Automation Explorer (NI-MAX)*. For Macs, it's called *NI-DAQ Configuration Utility*.

p. 287  Device Number for DAQ Board
In v6.1, find the DAQ board's *device number* by selecting Tools>>Data Acquisition>>DAQ Channel Viewer... When the DAQ Channel Viewer appears, click on the Devices tab.

p. 288  Advanced Analog Input Subpalette
In v6.1, the *Advanced Analog Input VIs* are found in Functions>>Data Acquisition>>Analog Input>>Advanced Analog Input.

p. 289-290  Channel Scan List and Waveform Data-Type
In v5, the *channel scan list* input for AI Group Config.vi was an array of *String Constants*. In v6.1, *channel scan list* is an array of *DAQ Channel Name Constants*, where *DAQ Channel Name Constant* (found in Functions>>Data Acquisition) is a new data-type introduced in LabVIEW 6. When building the block diagram for *Simple Read-Advanced VIs*, the easiest way to get the correct data-type icon, of course, is to pop up on the *channel scan list* input and select Create>>Constant. If you want to construct this object yourself, first get an *Array Constant* from Functions>>Array, then stuff it with a *DAQ Channel Name Constant* from Functions>>Data Acquisition. Once constructed, pop up on the *DAQ Channel Name Constant* and select Allow Undefined Names. The v6.1 block diagram for *Simple Read-Advanced VIs* appears as follows.
In v6.1, when you first wire the **waveform data** output of **AI Single Scan.vi** to the **n-dimensional array** input of **Index Array**, the wire will default to the **waveform** data-type. **Waveform** is a new data-type introduced in LabVIEW 6, which bundles data and timing information together.

Since we simply want to pass the voltage data to **Index Array** (without any timing information such as the date upon which the data was taken), pop up on the **waveform data** output of **AI Single Scan.vi** and select **Select Type>>Scaled Array** as shown in the next picture.
The wire will then become thick and orange denoting it now is an array of floating-point numbers, i.e., the array of acquired voltage values.

p. 291 DAQ Channel Name Constant and Waveform Data-Type

In v6.1, the channel input for **AI Sample Channel.vi** should be a **DAQ Channel Name Constant** found in **Functions>>Data Acquisition**. Also the default data-type for the **sample** output is the **waveform** data-type, so you'll need to pop up of this output and select **Select Type>>Scaled Value**. The block diagram for **Simple Read-Easy I/O** appears as below.
Converting 2D Array into 1D Array

In v6.1, there is no need to pop up and select Disable Indexing on the row index input of Index Array when using this icon to slices a 1D column off of a 2D array.

Enum Control

The Enumerated Type control of v5 is called the Enum control in v6.1. This control is found in Controls>>Ring & Enum. Associated integers are made visible by popping up on the Enum control and selecting Visible Items>>Digital Display.

Triggered Read

Below is the completed v6.1 block diagram for Triggered Read. The differences from the completed v5 block diagram on p. 302 are (1) the channel scan list is an array of DAQ Channel Name Constants, (2) the waveform data output of AI Buffer Read.vi is initially in the default waveform data-type, so you must pop up on this output and select Select Type>>Scaled Array, and (3) you do not need to pop up and select Disable Indexing on the row index input of Index Array.

Case Structure

You need to pop up on the Case Structure and select Add Case After twice to make all four of the cases accessible on the block diagram of Number of Samples.
In v5, the channel input for AO Update Channel.vi was a String Constant. In v6.1, channel is an array of DAQ Channel Name Constant, where DAQ Channel Name Constant (found in Functions>>Data Acquisition) is a new data-type introduced in LabVIEW 6. When building the block diagram for Simple Write-Easy I/O, the easiest way to get the correct data-type icon, of course, is to pop up on the channel input and select Create>>Constant. If you want to wire the channel input manually, DAQ Channel Name Constant is found in Functions>>Data Acquisition. Once wired to the input, pop up on the DAQ Channel Name Constant and select Allow Undefined Names before entering the channel number (0 in the diagram below). The v6.1 block diagram for Simple Write-Easy I/O appears as follows.
Chapter 12

Determining Instrument's GPIB Address using Software

In the LabVIEW v6.1 Windows environment, if an instrument is connected to the GPIB, its GPIB address can be found by selecting **Tools>>Measurement & Automation Explorer…** (MAX). After MAX launches, select **Devices and Interfaces**, then click on your GPIB board's identifier. Finally, click on **Scan for Instruments** and the address will be displayed after the software finds the instrument.

GPIB 448.2 Icons

In v6.1, the **GPIB 488.2** icons are found in **Functions>>Instrument I/O>>GPIB>>GPIB 488.2**.

Exponential String to Number Conversion

In v5, a string representation of a number in exponential notation was converted to a floating-point number using **From Exponential/Fract/Eng**, which was found in **Functions>>String>>Additional String To Number Functions**. In v6.1, this icon is renamed **Fract/Exp String To Number** and is found in **Functions>>String>>String/Number Conversion**.

GPIB 448.2 Icons

In v6.1, the **GPIB 488.2** icons are found in **Functions>>Instrument I/O>>GPIB>>GPIB 488.2**.

Enum Control

The **Enumerated Type** control of v5 is called **Enum** in v6.1 and is found in **Controls>>Ring & Enum**. Also the **Labelled Square Button** of v5 is no longer available. In v6.1, use, for example, **Push Button** in **Controls>>Boolean**.

Pick Line & Append Changed to Pick Line

In v5, **Pick Line & Append** was found in **Functions>>String**. In v6.1, this icon is renamed **Pick Line** and is found in **Functions>>String>>Additional String Functions**.

Format & Append Changed to Format Value

In v5, **Format & Append** was found in **Functions>>String>>Additional String to Number Functions**. In v6.1, this icon is renamed **Format Value** and is found in **Functions>>String>>String/Number Conversion**.

Select & Append Changed to Append True/False String

In v5, **Select & Append** was found in **Functions>>String**. In v6.1, this icon is renamed **Append True/False String** and is found in **Functions>>String>>Additional String Functions**.
p. 371  Block Diagram for *Measurement Config*
In v6.1 the block diagram for *Measurement Config* appears as follows.

p. 377  Making Scrollbar Visible
In v6.1, pop up on *String Indicator* called *Spreadsheet* and select *Visible Items>>Scrollbar*. 