

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

Controlling Timing

You might want to control the speed at which a process executes, such as the speed at which data values are plotted to a chart. You can use a Wait function in the loop to wait an amount of time in milliseconds before the loop re-executes.

Auto-Indexing Loops

If you wire an array to a For Loop or While Loop input tunnel, you can read and process every element in that array by enabling auto-indexing.

Refer to the *Arrays* section of Chapter 9, *Grouping Data Using Strings, Arrays, and Clusters*, for more information about arrays.

When you wire an array to an input tunnel on the loop border and enable auto-indexing on the input tunnel, elements of that array enter the loop one at a time, starting with the first element. When auto-indexing is disabled, the entire array is passed into the loop. When you auto-index an array output tunnel, the output array receives a new element from every iteration of the loop. Therefore, auto-indexed output arrays are always equal in size to the number of iterations. For example, if the loop executes 10 times, the output array has 10 elements. If you disable auto-indexing on an output tunnel, only the element from the last iteration of the loop passes to the next node on the block diagram.

Right-click the tunnel at the loop border and select Enable Indexing or Disable Indexing from the shortcut menu to enable or disable auto-indexing. Auto-indexing for While Loops is disabled by default.

A bracketed glyph appears on the loop border to indicate that auto-indexing is enabled. The thickness of the wire between the output tunnel and the next node also indicates the loop is using auto-indexing. The wire is thicker when you use auto-indexing because the wire contains an array, instead of a scalar.

The loop indexes scalar elements from 1D arrays, 1D arrays from 2D arrays, and so on. The opposite occurs at output tunnels. Scalar elements accumulate sequentially into 1D arrays, 1D arrays accumulate into 2D arrays, and so on.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Auto-Indexing to Set the For Loop Count

If you enable auto-indexing on an array wired to a For Loop input terminal, LabVIEW sets the count terminal to the array size so you do not need to wire the count terminal. Because you can use For Loops to process arrays an element at a time, LabVIEW enables auto-indexing by default for every array you wire to a For Loop. Disable auto-indexing if you do not need to process arrays one element at a time.

If you enable auto-indexing for more than one tunnel or if you wire the count terminal, the count becomes the smaller of the choices. For example, if two auto-indexed arrays enter the loop, with 10 and 20 elements respectively, and you wire a value of 15 to the count terminal, the loop executes 10 times, and the loop indexes only the first 10 elements of the second array. As another example, if you plot data from two sources on one graph and you want to plot the first 100 elements, wire 100 to the count terminal. If one of the data sources includes only 50 elements, the loop executes 50 times and indexes only the first 50 elements. Use the Array Size function to determine the size of arrays.

Auto-Indexing with While Loops

If you enable auto-indexing for an array entering a While Loop, the While Loop indexes the array the same way a For Loop does. However, the number of iterations a While Loop executes is not limited by the size of the array because the While Loop iterates until a specific condition occurs. When a While Loop indexes past the end of the input array, the default value for the array element type passes into the loop. You can prevent the default value from passing into the While Loop by using the Array Size function. The Array Size function indicates how many elements are in the array. Set up the While Loop to stop executing when it has iterated the same number of times as the array size.



Caution Because you cannot determine the size of the output array in advance, enabling auto-indexing for the output of a For Loop is more efficient than with a While Loop. Iterating too many times can cause your system to run out of memory.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using Loops to Build Arrays

In addition to using loops to read and process elements in an array, you also can use the For Loop and the While Loop to build arrays. Wire the output of a VI or function in the loop to the loop border. If you use a While Loop, right-click the resulting tunnel and select Enable Indexing from the shortcut menu. On the For Loop, indexing is enabled by default. The output of the tunnel is an array of every value the VI or function returns after each loop iteration.

Refer to the *Arrays* section of Chapter 9, *Grouping Data Using Strings, Arrays, and Clusters*, for more information about arrays.

Refer to the `labview\examples\general\arrays llb` for examples of building arrays.

Shift Registers and the Feedback Node in Loops

Use shift registers or the Feedback Node with For Loops or While Loops to transfer values from one loop iteration to the next.

Shift Registers

Use shift registers when you want to pass values from previous iterations through the loop to the next iteration. A shift register appears as a pair of terminals, shown as follows, directly opposite each other on the vertical sides of the loop border.



The terminal on the right side of the loop contains an up arrow and stores data on the completion of an iteration. LabVIEW transfers the data connected to the right side of the register to the next iteration. After the loop executes, the terminal on the right side of the loop returns the last value stored in the shift register.

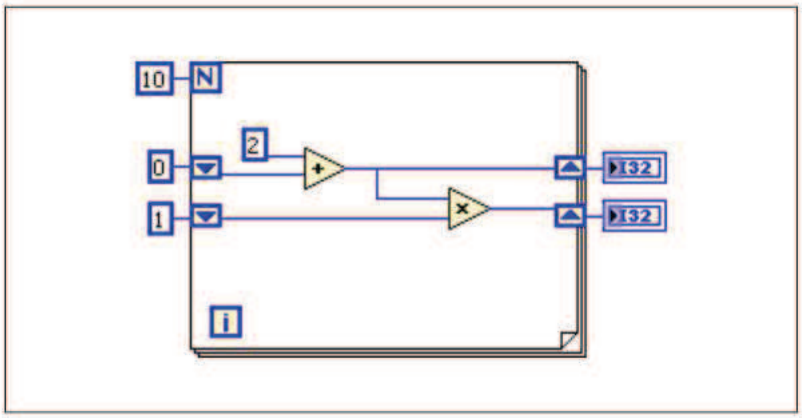
Create a shift register by right-clicking the left or right border of a loop and selecting Add Shift Register from the shortcut menu.

A shift register transfers any data type and automatically changes to the data type of the first object wired to the shift register. The data you wire to the terminals of each shift register must be the same type.

You can add more than one shift register to a loop. If you have multiple operations that use previous iteration values within your loop, use multiple

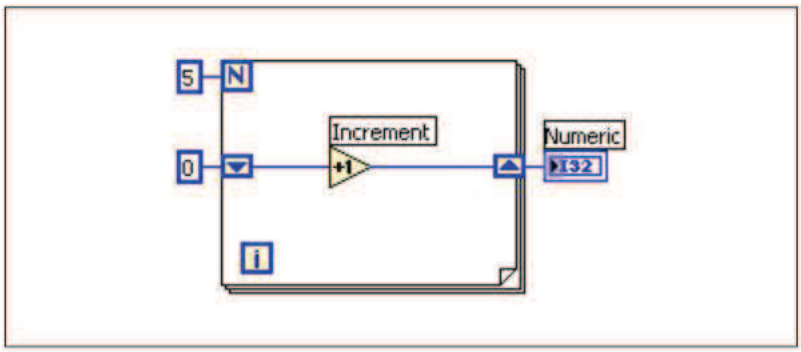
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

shift registers to store the data values from those different processes in the structure, as shown in the following figure.



Initializing Shift Registers

Initializing a shift register resets the value the shift register passes to the first iteration of the loop when the VI runs. Initialize a shift register by wiring a control or constant to the shift register terminal on the left side of the loop, as shown in the following figure.

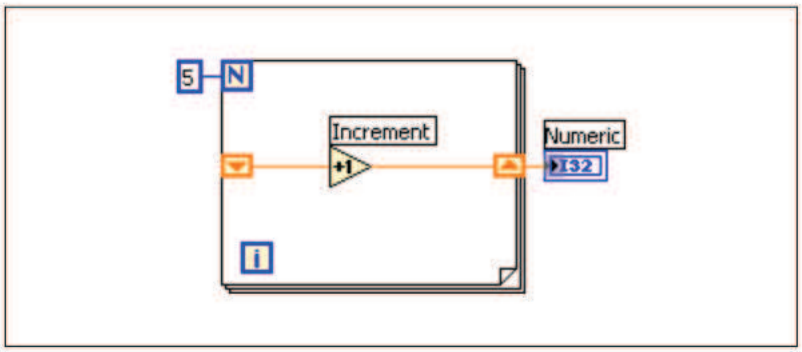


In the previous figure, the For Loop executes five times, incrementing the value the shift register carries by one each time. After five iterations of the For Loop, the shift register passes the final value, `5`, to the indicator and the VI quits. Each time you run the VI, the shift register begins with a value of `0`.

If you do not initialize the shift register, the loop uses the value written to the shift register when the loop last executed or the default value for the data type if the loop has never executed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Use an uninitialized shift register to preserve state information between subsequent executions of a VI. The following figure shows an uninitialized shift register.



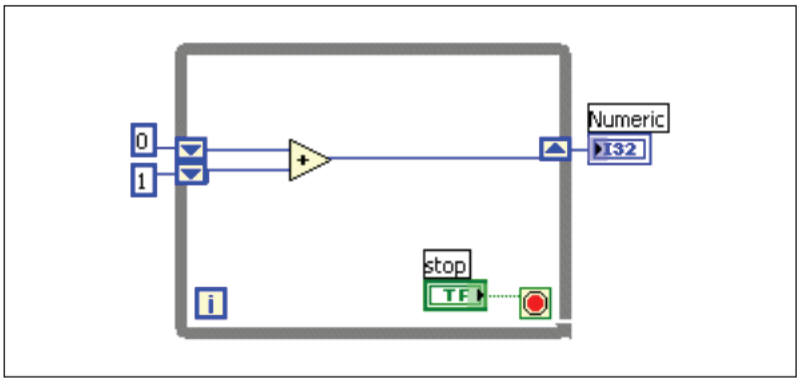
In the previous figure, the For Loop executes five times, incrementing the value the shift register carries by one each time. The first time you run the VI, the shift register begins with a value of 0, which is the default value for a 32-bit integer. After five iterations of the For Loop, the shift register passes the final value, `5`, to the indicator, and the VI quits. The next time you run the VI, the shift register begins with a value of `5`, which was the last value from the previous execution. After five iterations of the For Loop, the shift register passes the final value, `10`, to the indicator. If you run the VI again, the shift register begins with a value of `10`, and so on. Uninitialized shift registers retain the value of the previous iteration until you close the VI.

Stacked Shift Registers

Stacked shift registers let you access data from previous loop iterations. Stacked shift registers remember values from multiple previous iterations and carry those values to the next iterations. To create a stacked shift register, right-click the left terminal and select Add Element from the shortcut menu.

Stacked shift registers can occur only on the left side of the loop because the right terminal transfers the data generated only from the current iteration to the next iteration, as shown in the following figure.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



If you add another element to the left terminal in the previous figure, values from the last two iterations carry over to the next iteration, with the most recent iteration value stored in the top shift register. The bottom terminal stores the data passed to it from the previous iteration.

Feedback Node

The Feedback Node, shown as follows, appears automatically in a For Loop or While Loop when you wire the output of a node or group of nodes to the input of that node or group of nodes.



You also can select the Feedback Node on the Functions palette and place it inside a For Loop or While Loop. Use the Feedback Node to avoid long wires across loops.

Right-click the Feedback Node and select Initializer Terminal from the shortcut menu to add the initializer terminal to the loop border to initialize the loop. When you select the Feedback Node on the Functions palette or if you convert an initialized shift register to a Feedback Node, the loop appears with an initializer terminal. Initializing a Feedback Node resets the initial value the Feedback Node passes to the first iteration of the loop when the VI runs. If you do not initialize the Feedback Node, the Feedback Node passes the last value written to the node or the default value for the data type if the loop has never executed. If you do not wire the input of the initializer terminal, each time the VI runs, the initial input of the Feedback Node is the last value from the previous execution.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Replace a shift register with a Feedback Node by right-clicking the shift register and selecting Replace with Feedback Node from the shortcut menu. Replace a Feedback Node with shift registers by right-clicking the Feedback Node and selecting Replace with Shift Register from the shortcut menu.

Default Data in Loops

While Loops produce default data when the shift register is not initialized.

For Loops produce default data if you wire 0 to the count terminal of the For Loop or if you wire an empty array to the For Loop as an input with auto-indexing enabled. The loop does not execute, and any output tunnel with auto-indexing disabled contains the default value for the tunnel data type. Use shift registers to transfer values through a loop regardless of whether the loop executes.

Refer to the *LabVIEW Quick Reference Card* for more information about default values for data types.

Case, Sequence, and Event Structures

Case, Stacked Sequence, Flat Sequence, and Event structures contain multiple subdiagrams. A Case structure executes one subdiagram depending on the input value passed to the structure. A Stacked Sequence structure and a Flat Sequence structure execute all their subdiagrams in sequential order. An Event structure executes its subdiagrams depending on how the user interacts with the VI.

Case Structures

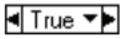
A Case structure, shown as follows, has two or more subdiagrams, or cases.



Only one subdiagram is visible at a time, and the structure executes only one case at a time. An input value determines which subdiagram executes. The Case structure is similar to switch statements or if...then...else statements in text-based programming languages.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The case selector label at the top of the Case structure, shown as follows, contains the name of the selector value that corresponds to the case in the center and decrement and increment arrows on each side.



Click the decrement and increment arrows to scroll through the available cases. You also can click the down arrow next to the case name and select a case from the pull-down menu.

Wire an input value, or selector, to the selector terminal, shown as follows, to determine which case executes.



You must wire an integer, Boolean value, string, or enumerated type value to the selector terminal. You can position the selector terminal anywhere on the left border of the Case structure. If the data type of the selector terminal is Boolean, the structure has a TRUE case and a FALSE case. If the selector terminal is an integer, string, or enumerated type value, the structure can have any number of cases.

Specify a default case for the Case structure to handle out-of-range values. Otherwise, you must explicitly list every possible input value. For example, if the selector is an integer and you specify cases for 1, 2, and 3, you must specify a default case to execute if the input value is 4 or any other unspecified integer value.

Case Selector Values and Data Types

You can enter a single value or lists and ranges of values in the case selector label. For lists, use commas to separate values. For numeric ranges, specify a range as `10 20`, meaning all numbers from 10 to 20 inclusively. You also can use open-ended ranges. For example, `100` represents all numbers less than or equal to 100, and `100` represents all numbers greater than or equal to 100. You also can combine lists and ranges, for example `5, 6, 7 10, 12, 13, 14`. When you enter values that contain overlapping ranges in the same case selector label, the Case structure redisplays the label in a more compact form. The previous example redisplays as `..10, 12..14`. For string ranges, a range of `a c` includes all of `a` and `b`, but not `c`. A range of `a c, c` includes the ending value of `c`.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If you enter a selector value that is not the same type as the object wired to the selector terminal, the value appears red to indicate that you must delete or edit the value before the structure can execute, and the VI will not run. Also, because of the possible round-off error inherent in floating-point arithmetic, you cannot use floating-point numbers as case selector values. If you wire a floating-point value to the case, LabVIEW rounds the value to the nearest even integer. If you type a floating-point value in the case selector label, the value appears red to indicate that you must delete or edit the value before the structure can execute.

Input and Output Tunnels

You can create multiple input and output tunnels for a Case structure. Inputs are available to all cases, but cases do not have to use each input. However, you must define each output tunnel for each case. When you create an output tunnel in one case, tunnels appear at the same position on the border in all the other cases. If even one output tunnel is not wired, all output tunnels on the structure appear as white squares. You can define a different data source for the same output tunnel in each case, but the data types must be compatible for each case. You also can right-click the output tunnel and select Use Default If Unwired from the shortcut menu to use the default value for the tunnel data type for all unwired tunnels.

Using Case Structures for Error Handling

When you wire an error cluster to the selector terminal of a Case structure, the case selector label displays two cases—`Error` and `No Error`—and the border of the Case structure changes color—red for `Error` and green for `No Error`. If an error occurs, the Case structure executes the `Error` subdiagram.

Refer to the *Handling Errors* section of Chapter 6, *Running and Debugging VIs*, for more information about handling errors.

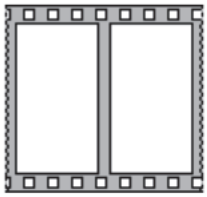
Sequence Structures

A sequence structure contains one or more subdiagrams, or frames, that execute in sequential order. Within each frame of a sequence structure, as in the rest of the block diagram, data dependency determines the execution order of nodes. Sequence structures are not used commonly in LabVIEW.

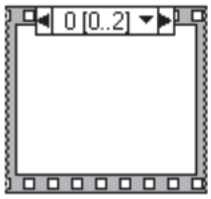
There are two types of sequence structures—the Flat Sequence structure and the Stacked Sequence structure.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The Flat Sequence structure, shown as follows, displays all the frames at once and executes the frames from left to right and when all data values wired to a frame are available, until the last frame executes. The data values leave each frame as the frame finishes executing.



The Stacked Sequence structure, shown as follows, stacks each frame so you see only one frame at a time and executes frame 0, then frame 1, and so on until the last frame executes.



To take advantage of the inherent parallelism in LabVIEW, avoid overusing sequence structures. Sequence structures guarantee the order of execution and prohibit parallel operations. For example, asynchronous tasks that use I/O devices, such as PXI, GPIB, serial ports, and DAQ devices, can run concurrently with other operations if sequence structures do not prevent them from doing so.

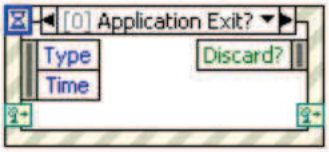
When you need to control the execution order, consider establishing data dependency between the nodes. For example, you can use flow-through parameters such as error I/O to control the execution order.

Refer to the *Handling Errors* section of Chapter 6, *Running and Debugging VIs*, for more information about error I/O. Refer to the *Flow-Through Parameters* section of Chapter 5, *Building the Block Diagram*, for more information about flow-through parameters.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Event Structures

An Event structure, shown as follows, has one or more subdiagrams, or event cases, exactly one of which executes when the structure executes.



The Event structure waits until an event happens, and then executes the appropriate case to handle that event. Events can originate from the user interface, external I/O, or other parts of the application. User interface events include mouse clicks, key presses, and so on. External I/O events include hardware timers or triggers that signal when data acquisition completes or when an error condition occurs. You can generate other types of events programmatically and use them to communicate with different parts of the application. LabVIEW supports user interface and programmatically generated events but does not support external I/O events.

Note The Event structure is available only in the LabVIEW Full and Professional Development Systems. You can run a VI built with event-driven programming features in the LabVIEW Base Package, but you cannot reconfigure the event-handling components.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9

Grouping Data Using Strings, Arrays, and Clusters

Use strings, arrays, and clusters to group data. Strings group sequences of ASCII characters. Arrays group data elements of the same type. Clusters group data elements of mixed types.

Grouping Data with Strings

A string is a sequence of displayable or non-displayable ASCII characters. Strings provide a platform-independent format for information and data. Some of the more common applications of strings include the following:

- Creating simple text messages.
- Passing numeric data as character strings to instruments and then converting the strings to numeric values.
- Storing numeric data to disk. To store numeric data in an ASCII file, you must first convert numeric data to strings before writing the data to a disk file.
- Instructing or prompting the user with dialog boxes.

On the front panel, strings appear as tables, text entry boxes, and labels. LabVIEW includes built-in VIs and functions you can use to manipulate strings, including formatting strings, parsing strings, and other editing.

Strings on the Front Panel

Use the string controls and indicators to simulate text entry boxes and labels.

Refer to the *String Controls and Indicators* section of Chapter 4, *Building the Front Panel*, for more information about string controls and indicators.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

String Display Types

Right-click a string control or indicator on the front panel to select from the display types shown in the following table. The table also shows an example message in each display type.

Display Type	Description	Message
Normal Display	Displays printable characters using the font of the control. Non-displayable characters generally appear as boxes.	`There are four display types.` `\ is a backslash.`
'\' Codes Display	Displays backslash codes for all non-displayable characters.	`There\sare\sfour\sdisplay\stypes.\n\\\sis\sa\sbackslash.`
Password Display	Displays an asterisk (*) for each character including spaces.	`**************************` `*****************`
Hex Display	Displays the ASCII value of each character in hex instead of the character itself.	`5468 6572 6520 6172 6520 666F 7572 2064 6973 706C 6179 2074` `7970 6573 2E0A 5C20 6973 2061 2062 6163 6B73 6C61 7368 2E`

Tables

Use the table control to create a table on the front panel. Each cell in a table is a string, and each cell resides in a column and a row. Therefore, a table is a display for a 2D array of strings.

Refer to the Arrays section of this chapter for more information about arrays.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Editing, Formatting, and Parsing Strings

Use the String functions to edit strings in ways similar to the following:

- Search for, retrieve, and replace characters or substrings within a string.
- Change all text in a string to upper case or lower case.
- Find and retrieve matching patterns within a string.
- Retrieve a line from a string.
- Rotate and reverse text within a string.
- Concatenate two or more strings.
- Delete characters from a string.

Refer to the *LabVIEW Style Checklist* in the *LabVIEW Help* for more information about minimizing memory usage when editing strings programmatically. Refer to the `labview\examples\general\strings.llb` for examples of using the String functions to edit strings.

Formatting and Parsing Strings

To use data in another VI, function, or application, you often must convert the data to a string and then format the string in a way that the VI, function, or application can read. For example, Microsoft Excel expects strings that include delimiters, such as tabs, commas, or blank spaces. Excel uses the delimiter to segregate numbers or words into cells.

For example, to write a 1D array of numeric values to a spreadsheet using the Write to Binary File function, you must format the array into a string and separate each numeric with a delimiter, such as a tab. To write an array of numeric values to a spreadsheet using the Write To Spreadsheet File VI, you must format the array with the Array To Spreadsheet String function and specify a format and a delimiter.

Use the String functions to perform tasks similar to the following:

- Extract a subset of strings from a string.
- Convert data into strings.
- Format a string for use in a word processing or spreadsheet application.

Use the File I/O VIs and functions to save strings to text and spreadsheet files.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Format Specifiers

In many cases, you must enter one or more format specifiers in the format string parameter of a String function to format a string. A format specifier is a code that indicates how to convert numeric data to or from a string. LabVIEW uses conversion codes to determine the textual format of the parameter. For example, a format specifier of `%x` converts a hex integer to or from a string.

Grouping Data with Arrays and Clusters

Use the array and cluster controls and functions to group data. Arrays group data elements of the same type. Clusters group data elements of mixed types.

Arrays

An array consists of elements and dimensions. Elements are the data that make up the array. A dimension is the length, height, or depth of an array. An array can have one or more dimensions and as many as $(2^{31}) - 1$ elements per dimension, memory permitting.

You can build arrays of numeric, Boolean, path, string, waveform, and cluster data types. Consider using arrays when you work with a collection of similar data and when you perform repetitive computations. Arrays are ideal for storing data you collect from waveforms or data generated in loops, where each iteration of a loop produces one element of the array.

Restrictions

You cannot create arrays of arrays. However, you can use a multidimensional array or create an array of clusters where each cluster contains one or more arrays. Also, you cannot create an array of subpanel controls, tab controls, .NET controls, ActiveX controls, charts, or multiplot XY graphs.

Refer to the *Clusters* section of this chapter for more information about clusters.

Indexes

To locate a particular element in an array requires one index per dimension. In LabVIEW, indexes let you navigate through an array and retrieve elements, rows, columns, and pages from an array on the block diagram.

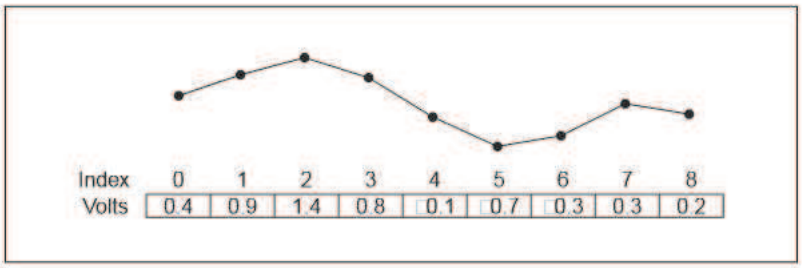
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Examples of Arrays

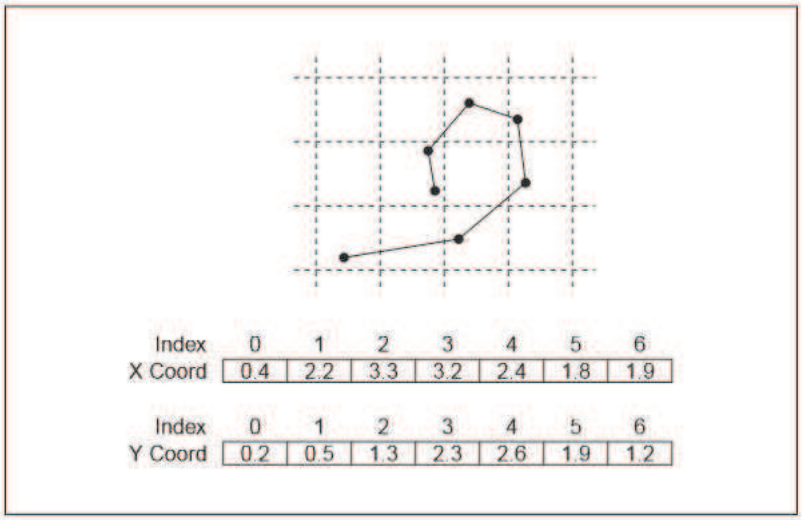
An example of a simple array is a text array that lists the nine planets of our solar system. LabVIEW represents this as a 1D array of strings with nine elements.

Array elements are ordered. An array uses an index so you can readily access any particular element. The index is zero-based, which means it is in the range 0 to $n - 1$, where n is the number of elements in the array. For example, $n = 9$ for the nine planets, so the index ranges from 0 to 8. Earth is the third planet, so it has an index of 2.

Another example of an array is a waveform represented as a numeric array in which each successive element is the voltage value at successive time intervals, as shown in the following figure.

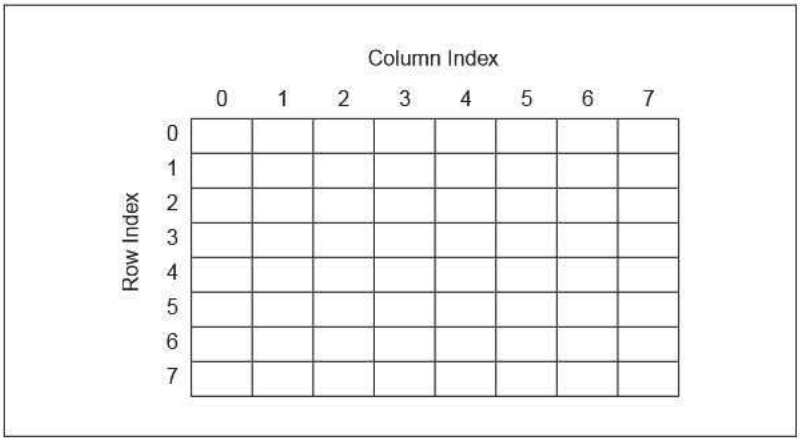


A more complex example of an array is a graph represented as an array of points where each point is a cluster containing a pair of numeric values that represent the X and Y coordinates, as shown in the following figure.



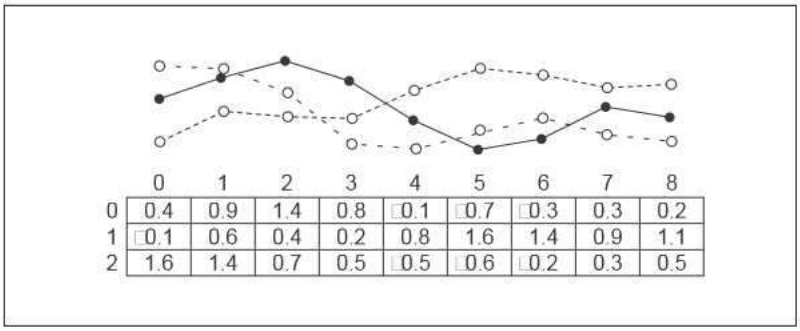
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The previous examples use 1D arrays. A 2D array stores elements in a grid. It requires a column index and a row index to locate an element, both of which are zero-based. The following figure shows an 8 column by 8 row 2D array, which contains 8 × 8 = 64 elements.



For example, a chessboard has eight columns and eight rows for a total of 64 positions. Each position can be empty or have one chess piece. You can represent a chessboard as a 2D array of strings. Each string is the name of the piece that occupies the corresponding location on the board or an empty string if the location is empty.

You can generalize the previous 1D array examples to two dimensions by adding a row to the array. The following figure shows a collection of waveforms represented as a 2D array of numeric values. The row index selects the waveform, and the column index selects the point on the waveform.

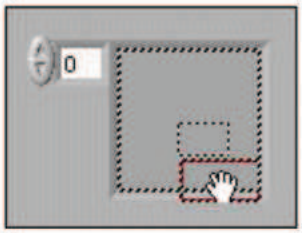


Refer to the `labview\examples\general\arrays llb` for examples of using arrays.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Creating Array Controls, Indicators, and Constants

Create an array control or indicator on the front panel by placing an array shell on the front panel, as shown in the following figure, and dragging a data object or element, which can be a numeric, Boolean, string, path, refnum, or cluster control or indicator, into the array shell.



The array shell automatically resizes to accommodate the new object.

To create an array constant on the block diagram, select an array constant on the Functions palette, place the array shell on the block diagram, and place a string constant, numeric constant, or cluster constant in the array shell. You can use an array constant to store constant data or as a basis for comparison with another array.

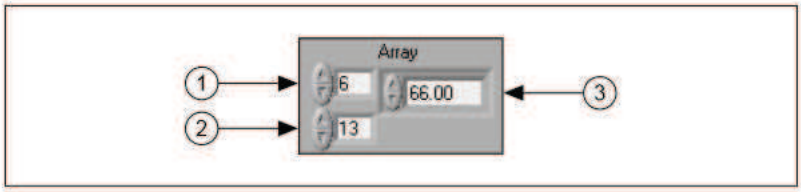
Creating Multidimensional Arrays

To create a multidimensional array on the front panel, right-click the index display and select Add Dimension from the shortcut menu. You also can resize the index display until you have as many dimensions as you want. To delete dimensions one at a time, right-click the index display and select Remove Dimension from the shortcut menu. You also can resize the index display to delete dimensions.

To display a particular element on the front panel, either type the index number in the index display or use the arrows on the index display to navigate to that number.

For example, a 2D array contains rows and columns. As shown in the following figure, the upper display of the two boxes on the left is the row index and the lower display is the column index. The combined display to the right of the row and column displays shows the value at the specified position. The following figure shows that the value at row 6, column 13, is 66.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



1 Row index	3 Value at row, column
2 Column index	

Rows and columns are zero-based, meaning the first column is column 0, the second column is column 1, and so on. Changing the index display for the following array to row 1, column 2 displays a value of 6.

0	1	2	3
4	5	6	7
8	9	10	11

If you try to display a column or row that is out of the range of the array dimensions, the array control appears dimmed to indicate that there is no value defined, and LabVIEW displays the default value of the data type. The default value of the data type depends on the data type of the array.

Use the Positioning tool to resize the array to show more than one row or column at a time.

Array Functions

Use the Array functions to create and manipulate arrays. For example, you can perform tasks similar to the following:

- Extract individual data elements from an array.
- Insert, delete, or replace data elements in an array.
- Split arrays.

Use the Build Array function to build an array programmatically. You also can use a loop to build an array.

Refer to the *Using Loops to Build Arrays* section of Chapter 8, *Loops and Structures*, for information about using loops to build arrays.

Refer to the *LabVIEW Style Checklist* in the *LabVIEW Help* for more information about minimizing memory usage when using Array functions in a loop.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

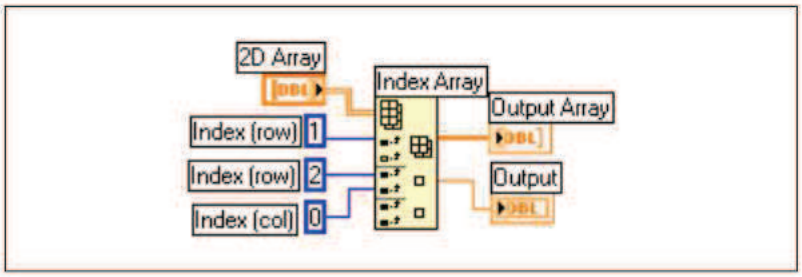
Automatically Resizing Array Functions

The Index Array, Replace Array Subset, Insert Into Array, Delete From Array, and Array Subset functions automatically resize to match the dimensions of the input array you wire. For example, if you wire a 1D array to one of these functions, the function shows a single index input. If you wire a 2D array to the same function, it shows two index inputs—one for the row and one for the column.

You can access more than one element, or subarray (row, column, or page), with these functions by using the Positioning tool to manually resize the function. When you expand one of these functions, the functions expand in increments determined by the dimensions of the array wired to the function. If you wire a 1D array to one of these functions, the function expands by a single index input. If you wire a 2D array to the same function, the function expands by two index inputs—one for the row and one for the column.

The index inputs you wire determine the shape of the subarray you want to access or modify. For example, if the input to an Index Array function is a 2D array and you wire only the row input, you extract a complete 1D row of the array. If you wire only the column input, you extract a complete 1D column of the array. If you wire the row input and the column input, you extract a single element of the array. Each input group is independent and can access any portion of any dimension of the array.

The block diagram shown in the following figure uses the Index Array function to retrieve a row and an element from a 2D array.



To access multiple consecutive values in an array, expand the Index Array function, but do not wire values to the index inputs in each increment. For example, to retrieve the first, second, and third rows from a 2D array, expand the Index Array function by three increments and wire 1D array indicators to each sub-array output.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Default Data in Arrays

Indexing beyond the bounds of an array produces the default value for the array element parameter. You can use the Array Size function to determine the size of the array.

You can index beyond the bounds of an array inadvertently by indexing an array past the last element using a While Loop, by supplying too large a value to the index input of an Index Array function, or by supplying an empty array to an Index Array function.

Refer to the *Auto-Indexing Loops* section of Chapter 8, *Loops and Structures*, for more information about indexing. Refer to the *LabVIEW Quick Reference Card* for more information about default values for data types.

Clusters

Clusters group data elements of mixed types. An example of a cluster is the LabVIEW error cluster, which combines a Boolean value, a numeric value, and a string. A cluster is similar to a record or a struct in text-based programming languages.

Refer to the *Error Clusters* section of Chapter 6, *Running and Debugging VIs*, for more information about using error clusters.

Bundling several data elements into clusters eliminates wire clutter on the block diagram and reduces the number of connector pane terminals that subVIs need. The connector pane has, at most, 28 terminals. If your front panel contains more than 28 controls and indicators that you want to pass to another VI, group some of them into a cluster and assign the cluster to a terminal on the connector pane.

Most clusters on the block diagram have a pink wire pattern and data type terminal. Clusters of numeric values, sometimes referred to as points, have a brown wire pattern and data type terminal. You can wire brown numeric clusters to Numeric functions, such as Add or Square Root, to perform the same operation simultaneously on all elements of the cluster.

Order of Cluster Elements

Although cluster and array elements are both ordered, you must unbundle all cluster elements at once or use the Unbundle By Name function to access specific cluster elements. Clusters also differ from arrays in that they are a fixed size. Like an array, a cluster is either a control or an indicator. A cluster cannot contain a mixture of controls and indicators.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Cluster elements have a logical order unrelated to their position in the shell. The first object you place in the cluster is element 0, the second is element 1, and so on. If you delete an element, the order adjusts automatically. The cluster order determines the order in which the elements appear as terminals on the Bundle and Unbundle functions on the block diagram. You can view and modify the cluster order by right-clicking the cluster border and selecting Reorder Controls In Cluster from the shortcut menu.

To wire clusters to each other, both clusters must have the same number of elements. Corresponding elements, determined by the cluster order, must have compatible data types. For example, if a double-precision floating-point numeric value in one cluster corresponds in cluster order to a string in the another cluster, the wire on the block diagram appears broken and the VI does not run. If the numeric values are different representations, LabVIEW coerces them to the same representation.

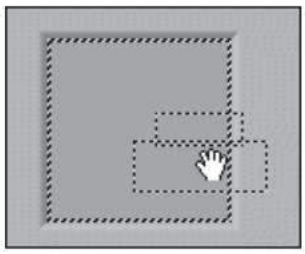
Cluster Functions

Use the Cluster functions to create and manipulate clusters. For example, you can perform tasks similar to the following:

- Extract individual data elements from a cluster.
- Add individual data elements to a cluster.
- Break a cluster out into its individual data elements.

Creating Cluster Controls, Indicators, and Constants

Create a cluster control or indicator on the front panel by placing a cluster shell on the front panel, as shown in the following figure, and dragging a data object or element, which can be a numeric, Boolean, string, path, refnum, array, or cluster control or indicator, into the cluster shell.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To create a cluster constant on the block diagram, select a cluster constant on the Functions palette, place the cluster shell on the block diagram, and place a string constant, numeric constant, or cluster constant in the cluster shell. You can use a cluster constant to store constant data or as a basis for comparison with another cluster.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

10

Graphs and Charts

After you acquire or generate data, use a graph or chart to display data in a graphical form.

Graphs and charts differ in the way they display and update data. VIs with a graph usually collect the data in an array and then plot the data to the graph. This process is similar to a spreadsheet that first stores the data then generates a plot of it. When the data is plotted, the graph discards the previous data and displays only the new data. You typically use a graph with fast processes that acquire data continuously.

In contrast, a chart appends new data points to those points already in the display to create a history. On a chart, you can see the current reading or measurement in context with data previously acquired. When more data points are added than can be displayed on the chart, the chart scrolls so that new points are added to the right side of the chart while old points disappear to the left. You typically use a chart with slow processes in which only a few data points per second are added to the plot.

Types of Graphs and Charts

LabVIEW includes the following types of graphs and charts:

- Waveform Graphs and Charts—Display data typically acquired at a constant rate.
- XY Graphs—Display data acquired at a non-constant rate and data for multivalued functions.
- Intensity Graphs and Charts—Display 3D data on a 2D plot by using color to display the values of the third dimension.
- Digital Waveform Graphs—Display data as pulses or groups of digital lines.
- **(Windows)** 3D Graphs—Display 3D data on a 3D plot in an ActiveX object on the front panel.

Refer to `labview\examples\general\graphs` for examples of graphs and charts.

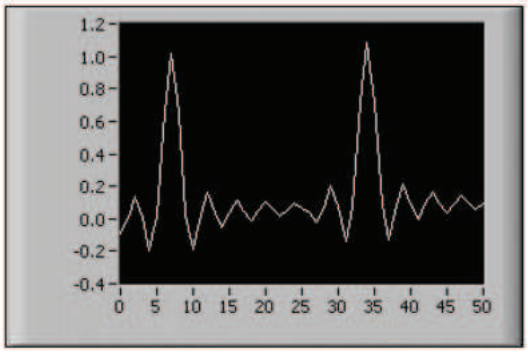
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Waveform Graphs and Charts

LabVIEW includes the waveform graph and chart to display data typically acquired at a constant rate.

Waveform Graphs

The waveform graph displays one or more plots of evenly sampled measurements. The waveform graph plots only single-valued functions, as in $y = f(x)$, with points evenly distributed along the x-axis, such as acquired time-varying waveforms. The following figure shows an example of a waveform graph.



The waveform graph can display plots containing any number of points. The graph also accepts several data types, which minimizes the extent to which you must manipulate data before you display it.

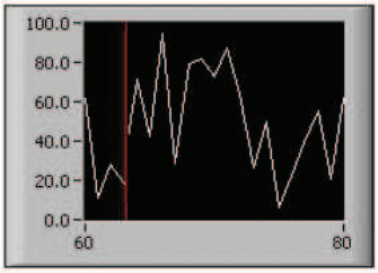
Note Use the digital waveform graph to display digital data. Refer to the *Digital Waveform Graphs* section of this chapter for more information about the digital waveform graph and the data types it accepts.

Refer to the Waveform Graph VI in the `labview\examples\general\graphs\gengraph.llb` for examples of the data types that a waveform graph accepts.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Waveform Charts

The waveform chart is a special type of numeric indicator that displays one or more plots of data typically acquired at a constant rate. The following figure shows an example of a waveform chart.



The waveform chart maintains a history of data, or buffer, from previous updates. Right-click the chart and select Chart History Length from the shortcut menu to configure the buffer. The default chart history length for a waveform chart is 1,024 data points. The frequency at which you send data to the chart determines how often the chart redraws.

Refer to the `labview\examples\general\graphs\charts llb` for examples of the waveform chart.

Waveform Data Type

The waveform data type carries the data, start time, and delta *t* of a waveform. You can create a waveform using the Build Waveform function. Many of the VIs and functions you use to acquire or analyze waveforms accept and return waveform data by default. When you wire waveform data to a waveform graph or chart, the graph or chart automatically plots a waveform based on the data, start time, and delta *x* of the waveform. When you wire an array of waveform data to a waveform graph or chart, the graph or chart automatically plots all waveforms.

Refer to the *Digital Waveform Data Type* section of this chapter for more information about the digital waveform data type.

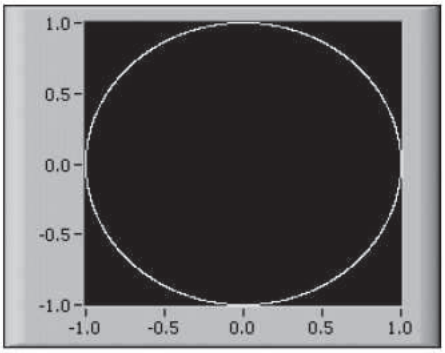
XY Graphs

The XY graph is a general-purpose, Cartesian graphing object that plots multivalued functions, such as circular shapes or waveforms with a varying time base. The XY graph displays any set of points, evenly sampled or not.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

You also can display Nyquist planes, Nichols planes, S planes, and Z planes on the XY graph. Lines and labels on these planes are the same color as the Cartesian lines, and you cannot modify the plane label font.

The following figure shows an example of an XY graph.



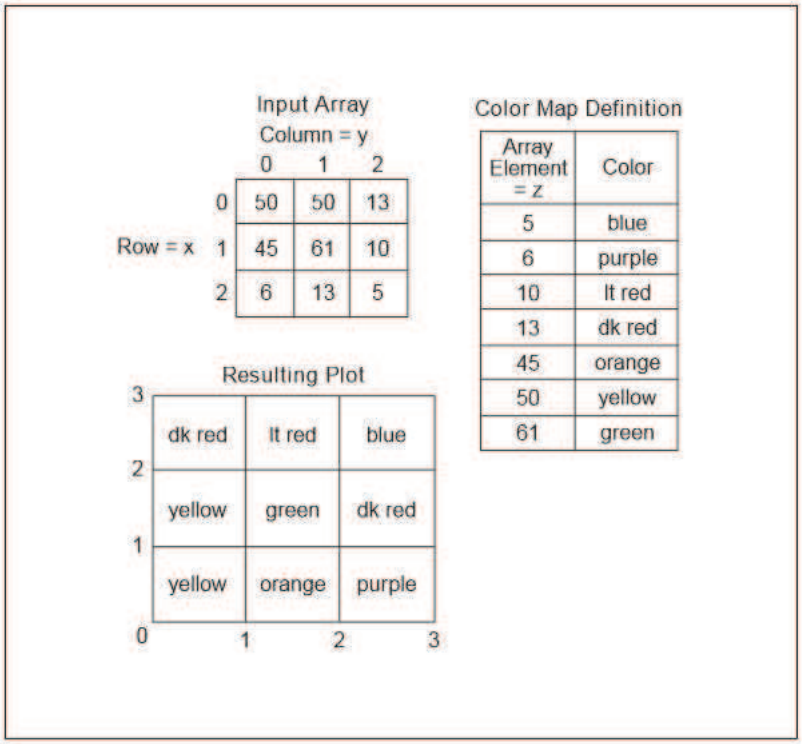
The XY graph can display plots containing any number of points. The XY graph also accepts several data types, which minimizes the extent to which you must manipulate data before you display it.

Refer to the XY Graph VI in the `labview\examples\general\graphs\gengraph.llb` for an example of an XY graph.

Intensity Graphs and Charts

Use the intensity graph and chart to display 3D data on a 2D plot by placing blocks of color on a Cartesian plane. For example, you can use an intensity graph or chart to display patterned data, such as temperature patterns and terrain, where the magnitude represents altitude. The intensity graph and chart accept a 3D array of numbers. Each number in the array represents a specific color. The indexes of the elements in the 2D array set the plot locations for the colors. The following figure shows the concept of the intensity chart operation.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



The rows of the data pass into the display as new columns on the graph or chart. If you want rows to appear as rows on the display, wire a 2D array data type to the graph or chart, right-click the graph or chart, and select `Transpose Array` from the shortcut menu.

The array indexes correspond to the lower left vertex of the block of color. The block of color has a unit area, which is the area between the two points, as defined by the array indexes. The intensity graph or chart can display up to 256 discrete colors.

Refer to the `labview\examples\general\graphs\intgraph llb` for examples of intensity graphs and charts.

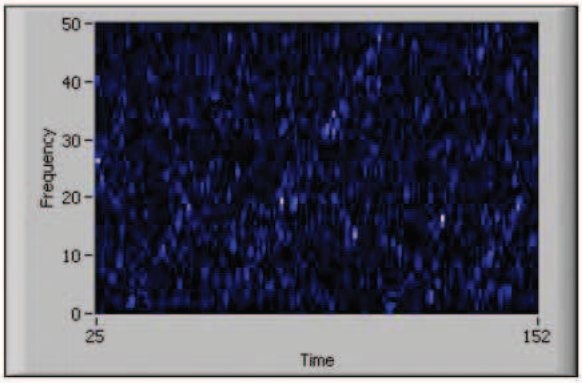
Intensity Charts

After you plot a block of data on an intensity chart, the origin of the Cartesian plane shifts to the right of the last data block. When the chart processes new data, the new data values appear to the right of the old data values. When a chart display is full, the oldest data values scroll off the left side of the chart. This behavior is similar to the behavior of a strip chart.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Refer to the *Configuring Chart Update Modes* section of this chapter for more information about the strip chart.

The following figure shows an example of an intensity chart.



The intensity chart shares many of the optional parts of the waveform chart, including the scale legend and graph palette, which you can show or hide by right-clicking the chart and selecting Visible Items from the shortcut menu. In addition, because the intensity chart includes color as a third dimension, a scale similar to a color ramp control defines the range and mappings of values to colors.

Refer to the *Using Color Mapping with Intensity Graphs and Charts* section of this chapter for information about color mapping.

Like the waveform chart, the intensity chart maintains a history of data, or buffer, from previous updates. Right-click the chart and select Chart History Length from the shortcut menu to configure the buffer. The default size for an intensity chart is 128 data points. The intensity chart display can be memory intensive.

Intensity Graphs

The intensity graph works the same as the intensity chart, except it does not retain previous data values and does not include update modes. Each time new data values pass to an intensity graph, the new data values replace old data values. Like other graphs, the intensity graph can have cursors. Each cursor displays the *x*, *y*, and *z* values for a specified point on the graph.

Refer to the *Using Graph Cursors* section of this chapter for information about cursors.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using Color Mapping with Intensity Graphs and Charts

An intensity graph or chart uses color to display 3D data on a 2D plot. When you set the color mapping for an intensity graph or chart, you configure the color scale of the graph or chart. The color scale consists of at least two arbitrary markers, each with a numeric value and a corresponding display color. The colors displayed on an intensity graph or chart correspond to the numeric values associated with the specified colors. Color mapping is useful for visually indicating data ranges, such as when plot data exceeds a threshold value.

You can set the color mapping interactively for the intensity graph and chart the same way you define the colors for a color ramp numeric control.



Note The colors you want the intensity graph or chart to display are limited to the exact colors and number of colors your video card can display. You also are limited by the number of colors allocated for your display.

Refer to the Create IntGraph Color Table VI in the `labview\examples\general\graphs\intgraph.llb` for an example of color mapping.

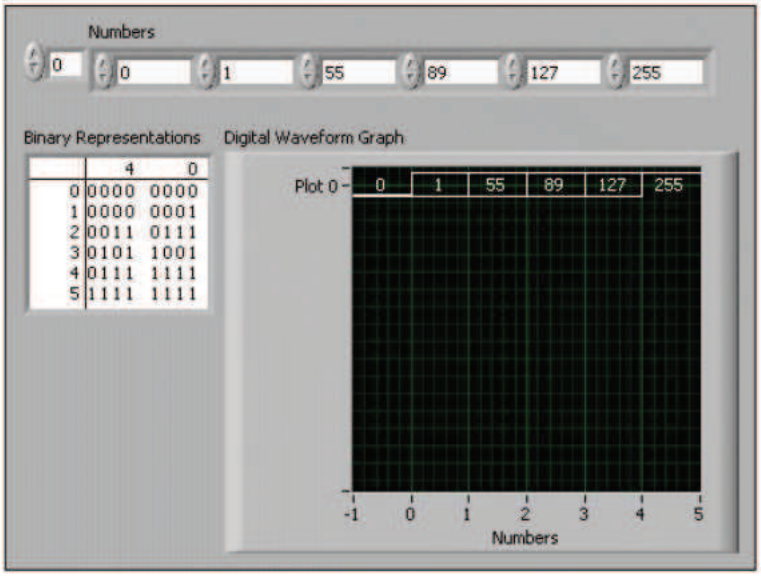
Digital Waveform Graphs

Use the digital waveform graph to display digital data, especially when you work with timing diagrams or logic analyzers.

The digital waveform graph accepts the digital waveform data type, the digital data type, and an array of those data types as an input. By default, the digital waveform graph collapses digital buses, so the graph plots digital data on a single plot. If you wire an array of digital data, the digital waveform graph plots each element of the array as a different plot in the order of the array.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

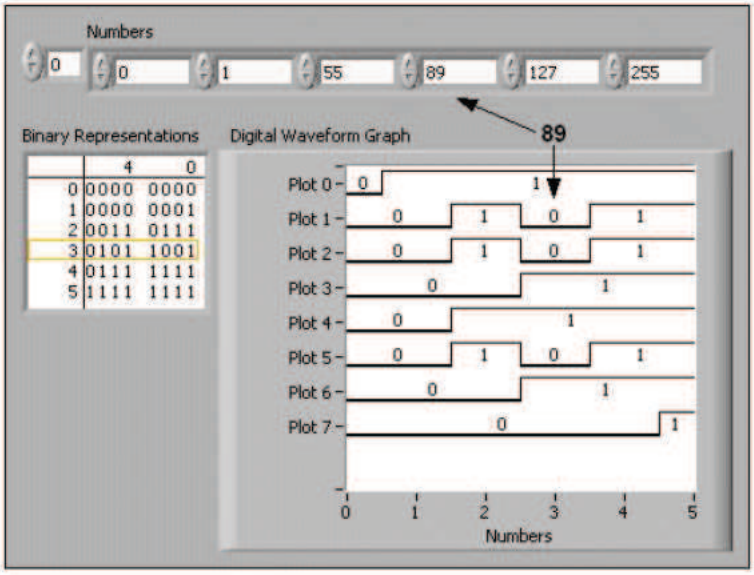
The digital waveform graph in the following front panel plots digital data on a single plot. The VI converts the numbers in the Numbers array to digital data and displays the binary representations of the numbers in the Binary Representations digital data indicator. In the digital graph, the number 0 appears without a top line to symbolize that all the bit values are zero. The number 255 appears without a bottom line to symbolize that all the bit values are 1.



Right-click the y-scale and select Expand Digital Buses from the shortcut menu to plot each sample of digital data. Each plot represents a different bit in the digital pattern.

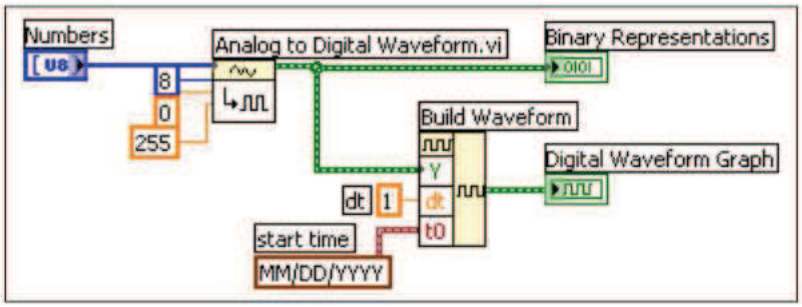
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The digital waveform graph in the following front panel displays the six numbers in the Numbers array.



The Binary Representations digital indicator displays the binary representations of the numbers. Each column in the table represents a bit. For example, the number 89 requires 7 bits of memory (the 0 in column 7 indicates an unused bit). Point 3 on the digital waveform graph plots the 7 bits necessary to represent the number 89 and a value of 0 to represent the unused eighth bit on plot 7.

The following VI converts an array of numbers to digital data and uses the Build Waveform function to assemble the start time, delta *t*, and the numbers entered in a digital data control and to display the digital data.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Refer to the *Digital Data Control* section of Chapter 4, *Building the Front Panel*, for more information about the digital data control.

Refer to the `labview\examples\general\graphs\DWDT Graphs.llb` for examples of the digital waveform graph.

Digital Waveform Data Type

The digital waveform data type carries start time, delta *x*, the data, and the attributes of a digital waveform. You can use the Build Waveform function to create a digital waveform. When you wire digital waveform data to the digital waveform graph, the graph automatically plots a waveform based on the timing information and data of the digital waveform. Wire digital waveform data to a digital data indicator to view the samples and signals of a digital waveform.

Refer to the *Waveform Data Type* section of this chapter for more information about the waveform data type.

3D Graphs

For many real-world data sets, such as the temperature distribution on a surface, joint time-frequency analysis, and the motion of an airplane, you need to visualize data in three dimensions. With the 3D graphs, you can visualize three-dimensional data and alter the way that data appears by modifying the 3D graph properties.

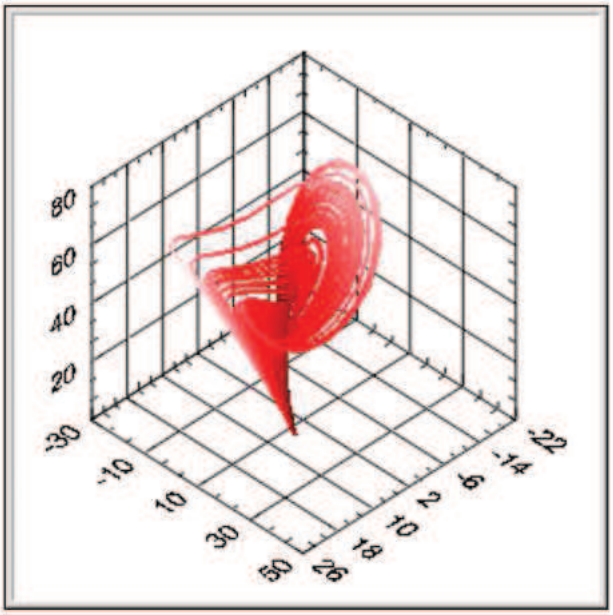
Note The 3D graph controls are available only on Windows in the LabVIEW Full and Professional Development Systems.

LabVIEW includes the following types of 3D graphs:

- 3D Surface Graph—Draws a surface in 3D space.
- 3D Parametric Surface Graph—Draws a parametric surface in 3D space.
- 3D Curve Graph—Draws a line in 3D space.

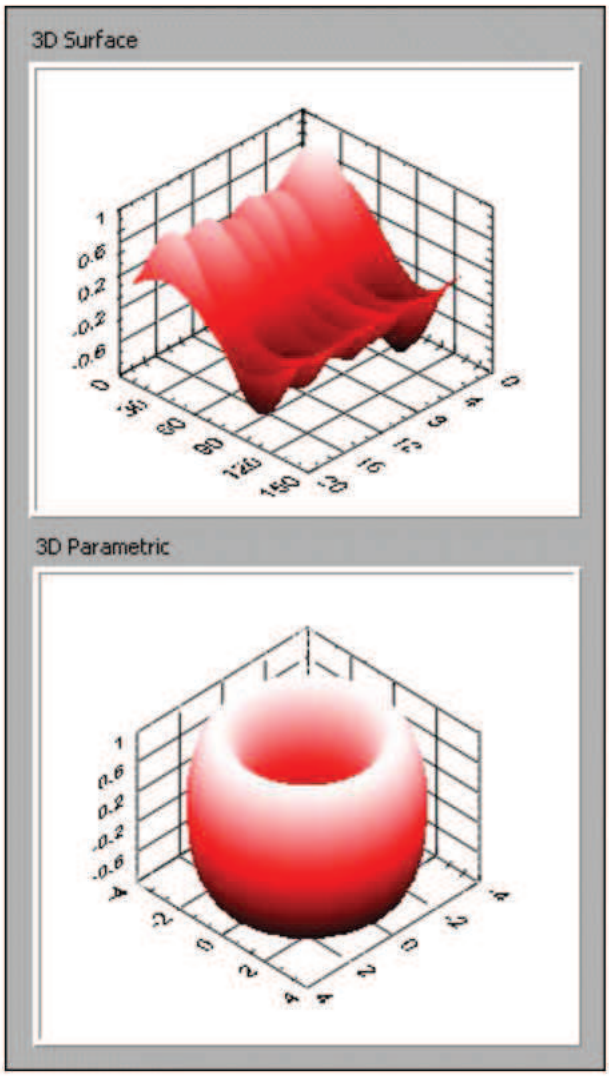
Use the 3D graphs in conjunction with the 3D Graph VIs to plot curves and surfaces. A curve contains individual points on the graph, each point having an *x*, *y*, and *z* coordinate. The VI then connects these points with a line. A curve is ideal for visualizing the path of a moving object, such as the flight path of an airplane. The following figure shows an example of a 3D curve graph.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



A surface plot uses *x*, *y*, and *z* data to plot points on the graph. The surface plot then connects these points, forming a three-dimensional surface view of the data. For example, you could use a surface plot for terrain mapping. The following figure shows examples of a 3D surface graph and a 3D parametric surface graph.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



The 3D graphs use ActiveX technology and VIs that handle 3D representation. When you select a 3D graph, LabVIEW places an ActiveX container on the front panel that contains a 3D graph control. LabVIEW also places a reference to the 3D graph control on the block diagram. LabVIEW wires this reference to one of the three 3D Graph VIs.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Customizing Graphs and Charts

Each graph and chart includes many options that you can use to customize appearance, convey more information, or highlight data. Although graphs and charts plot data differently, they have several common options that you access from the shortcut menu. However, some options are available only for a specific type of graph or chart.

Refer to the *Customizing Graphs* and *Customizing Charts* sections of this chapter for more information about the options that are available only on graphs or only on charts.

Using Multiple X- and Y-Scales

All graphs support multiple x- and y-scales, and all charts support multiple y-scales. Use multiple scales on a graph or chart to display multiple plots that do not share a common x- or y-scale. Right-click the scale of the graph or chart and select Duplicate Scale from the shortcut menu to add multiple scales to the graph or chart.

Autoscaling

All graphs and charts can automatically adjust their horizontal and vertical scales to fit the data you wire to them. This behavior is called autoscaling. Right-click the graph or chart and select X Scale»AutoScale X or Y Scale»AutoScale Y from the shortcut menu to turn autoscaling on or off. By default, autoscaling is enabled for the graph or chart. However, autoscaling can slow performance.

Use the Operating tool or the Labeling tool to change the horizontal or vertical scale directly.

Formatting X- and Y-Scales

Use the Format and Precision page of the Properties dialog box to specify how the scales of the x-axis and y-axis appear on the graph or chart.

By default, the x-scale is configured to use floating-point notation and have a label of `Time`, and the y-scale is configured to use automatic formatting and have a label of `Amplitude`. To configure the scales for the graph or chart, right-click the graph or chart and select Properties from the shortcut menu to display the Graph Properties dialog box or Chart Properties dialog box.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Use the Format and Precision page of the Properties dialog box to specify a numeric format for the scales of a graph or chart. Click the Scales tab to rename the scale and to format the appearance of the axis scale. By default, a graph or chart scale displays up to six digits before automatically switching to exponential notation.

On the Format and Precision page, select Advanced editing mode to display the text options that let you enter format strings directly. Enter format strings to customize the appearance and numeric precision of the scales.

Using the Graph Palette

Use the graph palette, shown as follows, to interact with a graph or chart while the VI is running.



With the graph palette, you can move cursors, zoom, and pan the display. Right-click the graph or chart and select Visible Items»Graph Palette from the shortcut menu to display the graph palette. The graph palette appears with the following buttons, in order from left to right:

- Cursor Movement Tool (graph only)—Moves the cursor on the display.
- Zoom—Zooms in and out of the display.
- Panning Tool—Picks up the plot and moves it around on the display.

Click a button in the graph palette to enable moving the cursor, zooming the display, or panning the display. Each button displays a green LED when it is enabled.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Customizing Graph and Chart Appearance

Customize the appearance of a graph or chart by showing or hiding options. Right-click the graph or chart and select Visible Items from the shortcut menu to display or hide the following options:

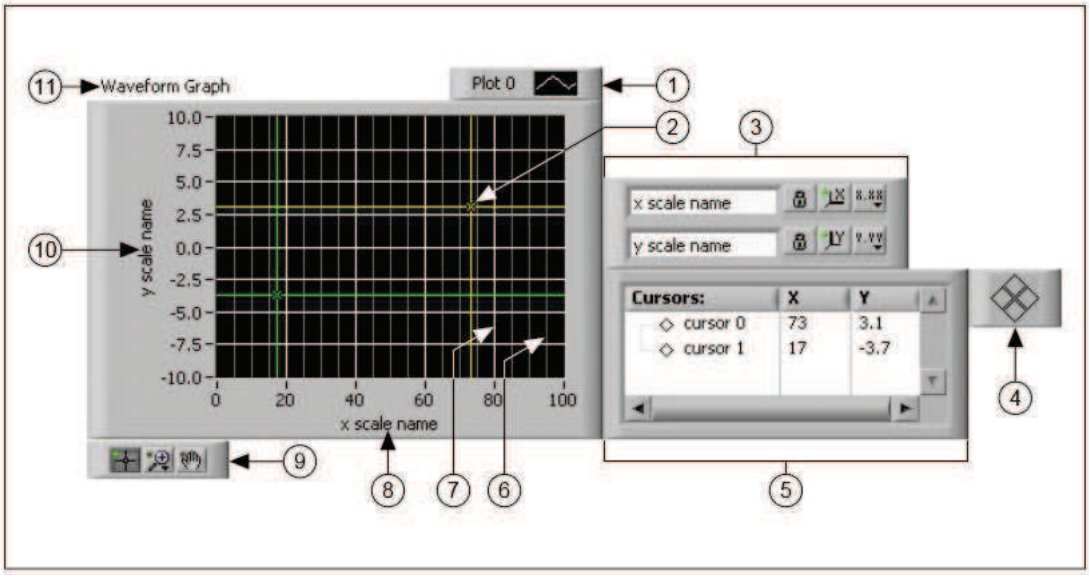
- Plot Legend—Defines the color and style of plots. Resize the legend to display multiple plots.
- Scale Legend—Defines labels for scales and configures scale properties.
- Graph Palette—Allows you to move the cursor and zoom and pan the graph or chart while a VI runs.
- X Scale and Y Scale—Formats the x- and y-scales.

  Refer to the *Formatting X- and Y-Scales* section of this chapter for more information about formatting scales.
- Cursor Legend (graph only)—Displays a marker at a defined point coordinate. You can display multiple cursors on a graph.
- X Scrollbar—Scrolls through the data in the graph or chart. Use the scroll bar to view data that the graph or chart does not currently display.
- Digital Display (waveform chart only)—Displays the numeric value of the chart.

Customizing Graphs

Each graph includes options that you can use to customize the graph to match your data display requirements. For example, you can modify the behavior and appearance of graph cursors or configure graph scales. The following figure shows the elements of a graph.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



1	Plot legend	4	Cursor mover	7	Grid mark	10	Y-scale
2	Cursor	5	Cursor legend	8	X-scale	11	Label
3	Scale legend	6	Minor-grid mark	9	Graph palette		

You add most of the items listed in the legend above by right-clicking the graph, selecting Visible Items from the shortcut menu, and selecting the appropriate element. Right-click the graph and select the option from the shortcut menu to set the graph option.

Using Graph Cursors

Use a cursor on a graph to read the exact value of a point on a plot or a point in the plot area. The cursor value displays in the cursor legend.

Right-click the graph and select Visible Items»Cursor Legend from the shortcut menu to view the cursor legend. Add a cursor to the graph by right-clicking anywhere in the cursor legend, selecting Create Cursor, and selecting a cursor mode from the shortcut menu.

The cursor position is defined by the cursor mode. The cursor includes the following modes:

- Free—Moves the cursor freely within the plot area, regardless of plot positions.
- Single-Plot—Positions the cursor only on the plot that is associated with the cursor. You can move the cursor along the associated plot.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Right-click the cursor legend row and select Snap To from the shortcut menu to associate one or all plots with the cursor.

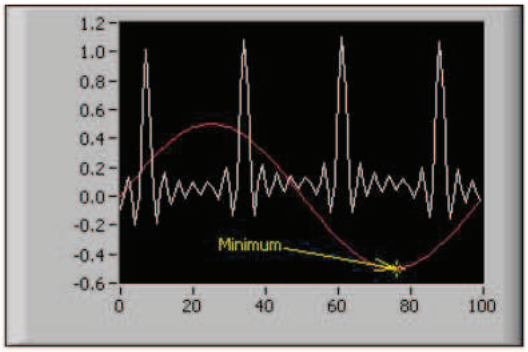
- Multi-Plot—Positions the cursor only on a specific data point in the plot area. The multi-plot cursor reports values at the specified x-value for all of the plots with which the cursor is associated. You can position the cursor on any plot in the plot area. Right-click the cursor legend row and select Snap To from the shortcut menu to associate one or all plots with the cursor. This mode is valid only for mixed signal graphs.

Note You cannot change the mode of a cursor after you create it. You must delete the cursor and create another cursor.

You can customize the appearance of the cursor in several ways. You can label the cursor on the plot, specify the color of the cursor, and specify line, point, and cursor style. Right-click the cursor legend row and select items from the shortcut menu to customize the cursor.

Using Graph Annotations

Use annotations on a graph to highlight data points in the plot area. The annotation includes a label and an arrow that identifies the annotation and data point. A graph can have any number of annotations. The following figure shows an example of a graph using annotations.



Right-click the graph and select Data Operations»Create Annotation from the shortcut menu to display the Create Annotation dialog box. Use the Create Annotation dialog box to specify the annotation name and how the annotation snaps to plots in the plot area.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Use the Lock Style pull-down menu in the Create Annotation dialog box to specify how the annotation snaps to plots in the plot area. The Lock Style component includes the following options:

- Free—Allows you to move the annotation anywhere in the plot area. LabVIEW does not snap the annotation to any plots in the plot area.
- Snap to All Plots—Allows you to move the annotation to the nearest data point along any plot in the plot area.
- Snap to One Plot—Allows you to move the annotation only along the specified plot.

You can customize the behavior and appearance of the annotation in several ways. You can hide or show the annotation name or arrow in the plot area, specify the color of the annotation, and specify line, point, and annotation style. Right-click the annotation and select options from the shortcut menu to customize the annotation.

To delete the annotation, right-click the annotation and select Delete Annotation from the shortcut menu. Right-click the graph and select Data Operations»Delete All Annotations from the shortcut menu to delete all annotations in the plot area.

Customizing 3D Graphs

The 3D graphs have many options that you can use to customize them, including 3D plot styles, scale formatting, grids, and plot projection. Because the 3D graphs use ActiveX technology and VIs that handle 3D representation, you set options for the 3D graphs differently than you set options for other graphs. While creating an application, use the ActiveX Property Browser to set properties for a 3D graph. Right-click the 3D graph and select Property Browser from the shortcut menu to display the ActiveX Property Browser.

If you want to allow users to change common properties at run time or you need to set a property programmatically, use the 3D Graph Properties VIs.

Customizing Charts

Unlike the graph, which displays new data that overwrites any stored data, the chart updates periodically and maintains a history of the data previously stored.

You can customize the chart to match your data display requirements. Options available for all charts include a scroll bar, the scale legend, the graph palette, a digital display, and representation of scales with respect to

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

time. You also can modify the behavior of chart history length, update modes, and plot displays.

Configuring Chart History Length

LabVIEW stores data points already added to the chart in a buffer, or the chart history. The default size for a chart history buffer is 1,024 data points. Right-click the chart and select Chart History Length from the shortcut menu to configure the history buffer. You can view previously collected data using the chart scroll bar. Right-click the chart and select Visible Items»X Scrollbar from the shortcut menu to display a scroll bar.

Note Large chart history values can be memory intensive.

Configuring Chart Update Modes

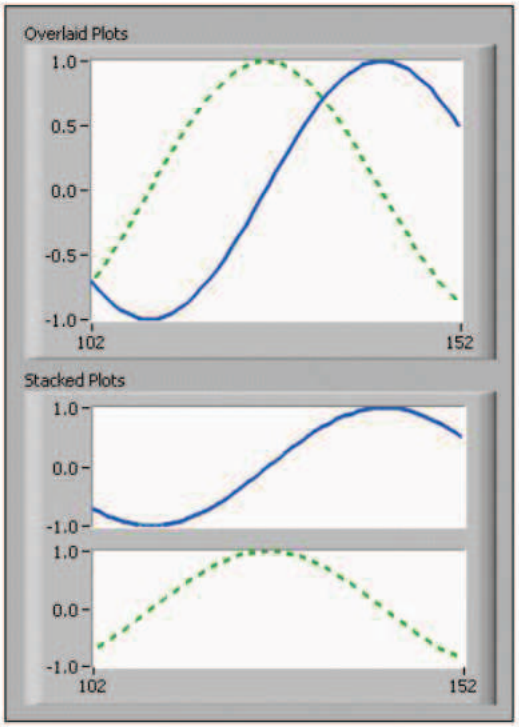
You can configure how the chart updates to display new data. Right-click the chart and select Advanced»Update Mode from the shortcut menu to set the chart update mode. The chart uses the following modes to display data:

- Strip Chart—Shows running data continuously scrolling from left to right across the chart with old data on the left and new data on the right. A strip chart is similar to a paper tape strip chart recorder. Strip Chart is the default update mode.
- Scope Chart—Shows one item of data, such as a pulse or wave, scrolling partway across the chart from left to right. For each new value, the chart plots the value to the right of the last value. When the plot reaches the right border of the plotting area, LabVIEW erases the plot and begins plotting again from the left border. The retracing display of a scope chart is similar to an oscilloscope.
- Sweep Chart—Works similarly to a scope chart except it shows the old data on the right and the new data on the left separated by a vertical line. LabVIEW does not erase the plot in a sweep chart when the plot reaches the right border of the plotting area. A sweep chart is similar to an EKG display.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using Overlaid and Stacked Plots

You can display multiple plots on a chart by using a single vertical scale, called overlaid plots, or by using multiple vertical scales, called stacked plots. The following figure shows examples of overlaid plots and stacked plots.



Right-click the chart and select Stack Plots from the shortcut menu to view the chart plots as multiple vertical scales. Right-click the chart and select Overlay Plots to view the chart plots as a single vertical scale.

Refer to the Charts VI in the `labview\examples\general\graphs\charts.llb` for examples of different kinds of charts and the data types they accept.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

11

File I/O

File I/O operations pass data to and from files. Use the File I/O VIs and functions on the File I/O palette to handle all aspects of file I/O, including the following:

- Opening and closing data files.
- Reading data from and writing data to files.
- Reading from and writing to spreadsheet-formatted files.
- Moving and renaming files and directories.
- Changing file characteristics.
- Creating, modifying, and reading a configuration file.

You can open, read or write, and close a file using a single VI or function. You also can use a function to control each step in the process. Use the Read From Measurement File Express VI and the Write To Measurement File Express VI to read data from and write data to `lvm` or `tdm` files.

Refer to the *Using Storage VIs* section of this chapter for more information about `tdm` files.

Basics of File I/O

A typical file I/O operation involves the following process.

1. Create or open a file. Indicate where an existing file resides or where you want to create a new file by specifying a path or responding to a dialog box to direct LabVIEW to the file location. After the file opens, a refnum represents the file.

   Refer to the *References to Objects or Applications* section of Chapter 4, *Building the Front Panel*, for more information about refnums.
2. Read from or write to the file.
3. Close the file.

File I/O VIs and some File I/O functions, such as the Read from Text File and Write to Text File functions, can perform all three steps for common

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

file I/O operations. The VIs and functions designed for multiple operations might not be as efficient as the functions configured or designed for individual operations.

Many File I/O VIs and functions contain flow-through parameters, typically a refnum or path, that return the same value as the corresponding input parameter.

Refer to the *Flow-Through Parameters* section of Chapter 5, *Building the Block Diagram*, for more information about flow-through parameters.

Choosing a File I/O Format

The VIs on the File I/O palette you use depend on the format of the files. You can read data from or write data to files in three formats—text, binary, and datalog. The format you use depends on the data you acquire or create and the applications that will access that data.

Use the following basic guidelines to determine which format to use:

- If you want to make your data available to other applications, such as Microsoft Excel, use text files because they are the most common and the most portable.
- If you need to perform random access file reads or writes or if speed and compact disk space are crucial, use binary files because they are more efficient than text files in disk space and in speed.
- If you want to manipulate complex records of data or different data types in LabVIEW, use datalog files because they are the best way to store data if you intend to access the data only from LabVIEW and you need to store complex data structures.

Text files typically take up more memory than binary and datalog files if the data is not originally in text form, such as graph or chart data, because the ASCII representation of data usually is larger than the data itself. For example, you can store the number –123.4567 in 4 bytes as a single-precision floating-point number. However, its ASCII representation takes 9 bytes, one for each character.

In addition, it is difficult to randomly access numeric data in text files. Although each character in a string takes up exactly 1 byte of space, the space required to express a number as text typically is not fixed. To find the ninth number in a text file, LabVIEW must first read and convert the preceding eight numbers.

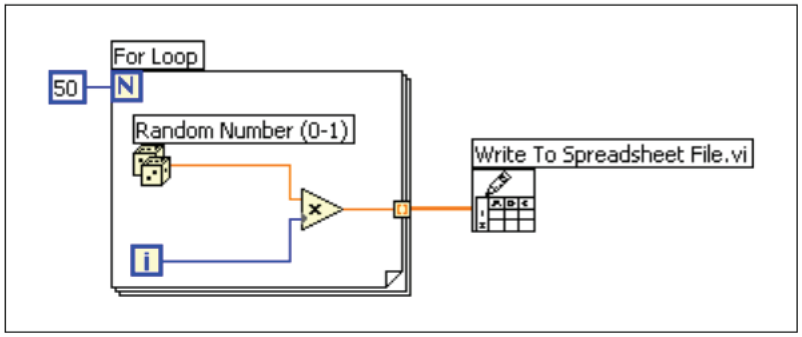
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using VIs and Functions for Common File I/O Operations

The File I/O palette includes VIs and functions designed for common file I/O operations, such as writing to or reading from the following types of data:

- Numeric values to or from spreadsheet text files
- Characters to or from text files
- Lines from text files
- Data to or from binary files

The following block diagram shows how to use the Write To Spreadsheet File VI to send numbers to a tab-delimited spreadsheet file. When you run this VI, LabVIEW prompts you to write the data to an existing file or to create a new file.



The open, read, and write functions expect a file path input. If you do not wire a file path, a dialog box appears for you to specify a file to read from or write to.

The File I/O palette includes functions to control each file I/O operation individually. Use these functions to create or open a file, read data from or write data to the file, and close the file. You also can use them to perform the following tasks:

- Create directories.
- Move, copy, or delete files.
- List directory contents.
- Change file characteristics.
- Manipulate paths.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

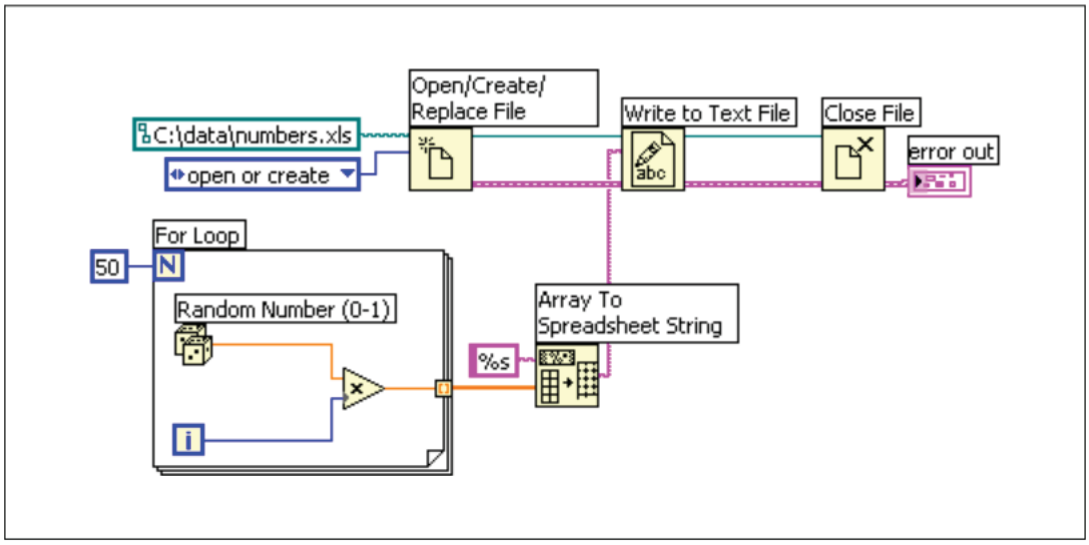
A path, shown as follows, is a LabVIEW data type that identifies the location of a file on disk.



The path describes the volume that contains the file, the directories between the top-level of the file system and the file, and the name of the file. Enter or display a path using the standard syntax for a given platform with the path control or indicator.

Refer to the *Path Controls and Indicators* section of Chapter 4, *Building the Front Panel*, for more information about path controls and indicators.

The following block diagram shows how to use File I/O functions to send numeric data to a tab-delimited spreadsheet file. When you run this VI, the Open/Create/Replace File function opens the `numbers xls` file. The Write to Text File function writes the string of numbers to the file. The Close File function closes the file. If you do not close the file, the file stays in memory and is not accessible from other applications or to other users.



Compare the previous block diagram to the Write to Spreadsheet VI, which completes the same task. The previous block diagram uses individual functions for each file operation, including using the Array To Spreadsheet String function to format the array of numbers as a string. The Write To Spreadsheet File VI completes multiple file operations, including opening the file, converting the array of numbers to a string, and closing the file.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Refer to the Write Datalog File Example VI in the `labview\examples\file\datalog llb` for an example of using File I/O VIs and functions for advanced operations.

You also can use File I/O functions for disk streaming operations, which save memory resources by reducing the number of times the function interacts with the operating system to open and close the file. Disk streaming is a technique for keeping files open while you perform multiple write operations, for example, within a loop. Wiring a path control or a constant to the Write to Text File function, the Write to Binary File function, or the Write to Spreadsheet File VI adds the overhead of opening and closing the file each time the function or VI executes. VIs can be more efficient if you avoid opening and closing the same files frequently.

To create a typical disk-streaming operation, place the Open/Create/Replace File function before a loop, the read or write function in the loop, and the Close File function after the loop so continuous writing to a file can occur within the loop without the overhead associated with opening and closing the file in each iteration.

Disk streaming is ideal in lengthy data acquisition operations where speed is critical. You can write data continuously to a file while acquisition is still in progress. For best results, avoid running other VIs and functions, such as Analysis VIs and functions, until you complete the acquisition.

Using Storage VIs

Use the Storage VIs on the Storage palette to read and write waveforms and waveform properties to binary measurement files (`tdm`). Use `tdm` files to exchange data between NI software, such as LabVIEW and DIAdem.

Note The Storage VIs are available only on Windows.

The Storage VIs combine waveforms and waveform properties to form channels. A channel group organizes a set of channels. A file includes a set of channel groups. If you store channels by name, you can quickly append data to or retrieve data from an existing channel. In addition to numeric values, the Storage VIs support arrays of strings and arrays of time stamps. A reference number represents files, channel groups, and channels on the block diagram.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

You also can use the Storage VIs to query files to obtain channel groups or channels that meet conditions you specify.

If the system requirements change during development and you need to add additional data to a file, you can change the format of the file without causing the file to become unusable.

Refer to the `labview\examples\file\storage llb` for examples of using the Storage VIs.

You also can use the Read From Measurement File Express VI and the Write To Measurement File Express VI to read data from and write data to `tdm` measurement files.

Creating Text and Spreadsheet Files

To write data to a text file, you must convert your data to a string. To write data to a spreadsheet file, you must format the string as a spreadsheet string, which is a string that includes delimiters, such as tabs.

Refer to the *Formatting and Parsing Strings* section of Chapter 9, *Grouping Data Using Strings, Arrays, and Clusters*, for more information about formatting strings.

Writing text to text files requires no formatting because most word processing applications that read text do not require formatted text. To write a text string to a text file, use the Write to Text File function, which automatically opens and closes the file.

Use the Write to Binary File function to create platform-independent text files. Use the Read from Binary File function to read platform-independent text files.

Refer to *Creating Binary Files* section for more information about binary files.

Use the Write To Spreadsheet File VI or the Array To Spreadsheet String function to convert a set of numbers from a graph, a chart, or an acquisition into a spreadsheet string.

Reading text from a word processing application might result in errors because word processing applications format text with different fonts, colors, styles, and sizes that the File I/O VIs cannot process.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If you want to write numbers and text to a spreadsheet or word processing application, use the String functions and the Array functions to format the data and to combine the strings. Then write the data to a file.

Refer to the *Editing, Formatting, and Parsing Strings* and *Array Functions* sections of Chapter 9, *Grouping Data Using Strings, Arrays, and Clusters*, for more information about using these functions to format and combine data.

Formatting and Writing Data to Files

Use the Format Into File function to format string, numeric, path, and Boolean data as text and to write the formatted text to a file. Often you can use this function instead of separately formatting the string with the Format Into String function and writing the resulting string with the Write to Text File function.

Refer to the *Formatting and Parsing Strings* section of Chapter 9, *Grouping Data Using Strings, Arrays, and Clusters*, for more information about formatting strings.

Scanning Data from Files

Use the Scan From File function to scan text in a file for string, numeric, path, and Boolean values and then convert the text into a data type. Often you can use this function instead of reading data from a file with the Read from Binary File function or Read from Text File function and scanning the resulting string with the Scan From String function.

Creating Binary Files

Use the Write to Binary File function to create a binary file. You can wire any data type to the data input of the Write to Binary File function. Use the Read from Binary File function to specify the data type of the data in the file you want to read by wiring a control or constant of that type to the data type input of the Read from Binary File function. You can use the Write to Binary File and Read from Binary File functions to read from and write to text files created on a different operating system.

Creating Datalog Files

You can create and read datalog files by using the Datalog functions on the Datalog palette to acquire data and write the data to a file.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

You do not have to format the data in a datalog file. However, when you write or read datalog files, you must specify the data type. For example, if you acquire a temperature reading with the time and date the temperature was recorded, you write the data to a datalog file and specify the data as a cluster of one number and two strings. Refer to the Simple Temp Datalogger VI in the `labview\examples\file\datalog llb` for an example of writing data to a datalog file.

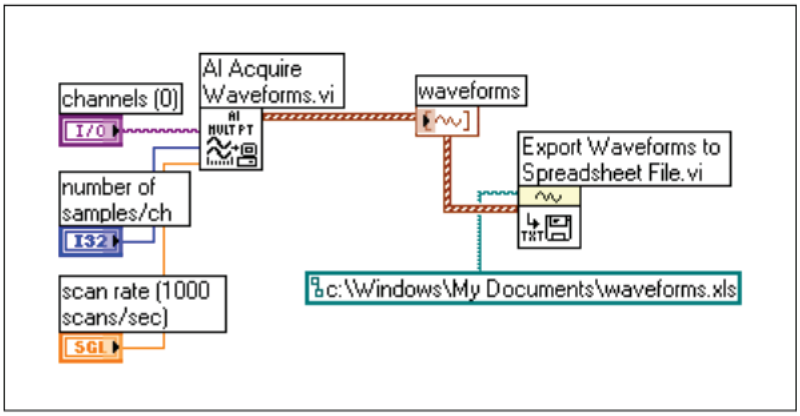
If you read a file that includes a temperature reading with the time and date the temperature was recorded, you specify that you want to read a cluster of one number and two strings. Refer to the Simple Temp Datalog Reader VI in the `labview\examples\file\datalog llb` for an example of reading a datalog file.

Writing Waveforms to Files

Use the Write Waveforms to File and Export Waveforms to Spreadsheet File VIs to send waveforms to files. You can write waveforms to spreadsheet, text, or datalog files.

If you expect to use the waveform you create only in a VI, save the waveform as a datalog file (`log`).

The following VI acquires multiple waveforms, displays them on a graph, and writes them to a spreadsheet file.



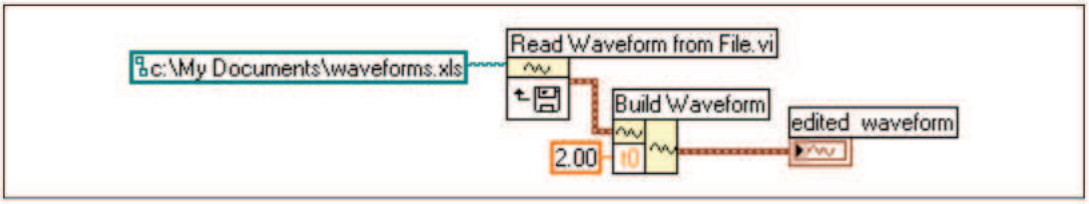
You also can use the Storage VIs on the Storage palette or the Write To Measurement File Express VI to write waveforms to files.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Reading Waveforms from Files

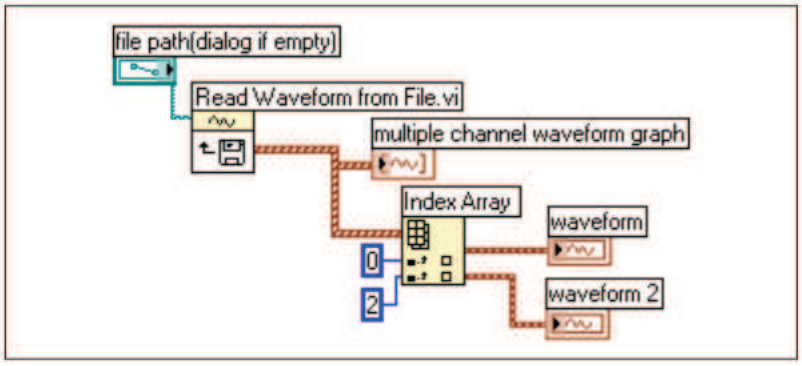
Use the Read Waveform from File VI to read waveforms from a file. After you read a single waveform, you can add or edit waveform data components with the Build Waveform function, or you can extract waveform attributes with the Get Waveform Attribute function.

The following VI reads a waveform from a file, edits the t0 component of the waveform, and plots the edited waveform to a graph.



The Read Waveform from File VI also reads multiple waveforms from a file. The VI returns an array of waveform data, which you can display in a multiplot graph. If you want to access a single waveform from a file, you must index the array of waveform data, as shown in the following block diagram. The VI accesses a file that includes multiple waveforms. The Index Array function reads the first and third waveforms in the file and plots them on two separate waveform graphs.

Refer to the *Arrays* section of Chapter 9, *Grouping Data Using Strings, Arrays, and Clusters*, for more information about indexing arrays. Refer to the *Waveform Graphs* section of Chapter 10, *Graphs and Charts*, for more information about waveform graphs.



You also can use the Storage VIs on the Storage palette or the Read From Measurement File Express VI to read waveforms from a file.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

12

Documenting and Printing VIs

You can use LabVIEW to document and print VIs.

Document a VI to record information about the block diagram and/or the front panel at any stage of development.

Some options for printing VIs are more appropriate for printing information about VIs, and others are more appropriate for reporting the data and results the VIs generate. Several factors affect which printing method you use, including if you want to print manually or programmatically, how many options you need for the report format, if you need the functionality in the stand-alone applications you build, and on which platforms you run the VIs.

Documenting VIs

You can use LabVIEW to document a finished VI and create instructions for users of VIs. You can view documentation within LabVIEW, print it, and save it to HTML, RTF, or text files.

To create effective documentation for VIs, create VI and object descriptions.

Create descriptions for VIs and their objects, such as controls and indicators, to describe the purpose of the VI or object and to give users instructions for using the VI or object. You can view the descriptions in LabVIEW, print them, or save them to HTML, RTF, or text files.

Create, edit, and view VI descriptions by selecting File»VI Properties and selecting Documentation from the Category pull-down menu. Create, edit, and view object descriptions by right-clicking the object and selecting Description and Tip from the shortcut menu. Tip strips are brief descriptions that appear when you move the cursor over an object while a VI runs. If you do not enter a tip in the Description and Tip dialog box, no tip strip appears. The VI or object description appears in the Context Help window when you move the cursor over the VI icon or object, respectively.

Note You cannot enter a description for a VI or function located on the Functions palette.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Printing VIs

You can use the following primary ways to print VIs:

- Select File»Print Window to print the contents of the active window.
- Select File»Print to print more comprehensive information about a VI, including information about the front panel, block diagram, subVIs, controls, VI history, and so on.
- Programmatically print a VI window or programmatically print or save a report that contains VI documentation or data the VI returns.

Select File»Print to print VI documentation or save it to HTML, RTF, or text files. You can select a built-in format or create a custom format for documentation. The documentation you create can include the following items:

- Icon and connector pane
- Front panel and block diagram
- Controls, indicators, and data type terminals
- Labels and captions for controls and indicators
- VI and object descriptions
- VI hierarchy
- List of subVIs
- Revision history

Note The documentation you create for certain types of VIs cannot include all the previous items. For example, a polymorphic VI does not have a front panel or a block diagram, so you cannot include those items in the documentation you create for a polymorphic VI.

You can use the HTML or RTF files LabVIEW generates to create your own compiled help files. **(Windows)** You can compile the individual HTML files LabVIEW generates into an HTML Help file. **(Mac OS)** You can use the individual HTML files LabVIEW generates in Apple Help.

You can compile the RTF files LabVIEW generates into a **(Windows)** WinHelp or **(Linux)** HyperHelp file.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A

Technical Support and Professional Services

Visit the following sections of the National Instruments Web site at ni com for technical support and professional services:

- Support—Online technical support resources at ni com/support include the following:
  - Self-Help Resources—For answers and solutions, visit the award-winning National Instruments Web site for software drivers and updates, a searchable KnowledgeBase, product manuals, step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on.
  - Free Technical Support—All registered users receive free Basic Service, which includes access to hundreds of Application Engineers worldwide in the NI Developer Exchange at ni com/exchange. National Instruments Application Engineers make sure every question receives an answer.

    For information about other technical support options in your area, visit ni com/services or contact your local office at ni com/contact.
- Training and Certification—Visit ni com/training for self-paced training, eLearning virtual classrooms, interactive CDs, and Certification program information. You also can register for instructor-led, hands-on courses at locations around the world.
- System Integration—If you have time constraints, limited in-house technical resources, or other project challenges, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit ni com/alliance.

If you searched ni com and could not find the answers you need, contact your local office or NI corporate headquarters. Phone numbers for our worldwide offices are listed at the front of this manual. You also can visit the Worldwide Offices section of 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.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Glossary

Symbol	Prefix	Value
y	yocto	10^{-24}
z	zepto	10^{-21}
a	atto	10^{-18}
f	femto	10^{-15}
p	pico	10^{-12}
n	nano	10^{-9}
μ	micro	10^{-6}
m	milli	10^{-3}
c	centi	10^{-2}
d	deci	10^{-1}
da	deka	10^{1}
h	hecto	10^{2}
k	kilo	10^{3}
M	mega	10^{6}
G	giga	10^{9}
T	tera	10^{12}
P	peta	10^{15}
E	exa	10^{18}
Z	zetta	10^{21}
Y	yotta	10^{24}

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Numbers/Symbols

□	Infinity.
□	Delta; difference. *x* denotes the value by which *x* changes from one index to the next.
□	Pi.
1D	One-dimensional.
2D	Two-dimensional.
3D	Three-dimensional.

A

A	Amperes.
active window	Window that is currently set to accept user input, usually the frontmost window. The title bar of an active window is highlighted. Make a window active by clicking it or by selecting it from the Windows menu.
application	Application created using the LabVIEW Development System and executed in the LabVIEW Run-Time System environment.
application instance	Instance of LabVIEW created for each target in a LabVIEW project. When you open a VI from the Project Explorer window, the VI opens in the application instance for the target. LabVIEW also creates a main application instance, which contains open VIs that are not part of a project and VIs that you did not open from a project. *See also* target.
array	Ordered, indexed list of data elements of the same type.
array shell	Front panel object that houses an array. An array shell consists of an index display, a data object window, and an optional label. It can accept various data types.
artificial data dependency	Condition in a dataflow programming language in which the arrival of data, rather than its value, triggers execution of a node.
ASCII	American Standard Code for Information Interchange.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

auto-indexing	Capability of loop structures to disassemble and assemble arrays at their borders. As an array enters a loop with auto-indexing enabled, the loop automatically disassembles it extracting scalars from 1D arrays, 1D arrays extracted from 2D arrays, and so on. Loops assemble data values into arrays as data values exit the loop in the reverse order.
autoscaling	Ability of scales to adjust to the range of plotted values. On graph scales, autoscaling determines maximum and minimum scale values.

B

block diagram	Pictorial description or representation of a program or algorithm. The block diagram consists of executable icons called nodes and wires that carry data between the nodes. The block diagram is the source code for the VI. The block diagram resides in the block diagram window of the VI.
Boolean controls and indicators	Front panel objects to manipulate and display Boolean (TRUE or FALSE) data.
breakpoint	Pause in execution used for debugging.
Breakpoint tool	Tool to set a breakpoint on a VI, node, or wire.
broken Run button	Button that replaces the Run button when a VI cannot run because of errors.
broken VI	VI that cannot run because of errors; signified by a broken arrow in the broken Run button.
buffer	Temporary storage for acquired or generated data.
Bundle function	Function that creates clusters from various types of elements.

C

case	One subdiagram of a Case structure.
Case structure	Conditional branching control structure that executes one of its subdiagrams based on the input to the Case structure. It is the combination of the IF, THEN, ELSE, and CASE statements in control flow languages.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

channel	1. Physical—a terminal or pin at which you can measure or generate an analog or digital signal. A single physical channel can include more than one terminal, as in the case of a differential analog input channel or a digital port of eight lines. A counter also can be a physical channel, although the counter name is not the name of the terminal where the counter measures or generates the digital signal. 2. Virtual—a collection of property settings that can include a name, a physical channel, input terminal connections, the type of measurement or generation, and scaling information. You can define NI-DAQmx virtual channels outside a task (global) or inside a task (local). Configuring virtual channels is optional in Traditional NI-DAQ (Legacy) and earlier versions, but is integral to every measurement you take in NI-DAQmx. In Traditional NI-DAQ (Legacy), you configure virtual channels in MAX. In NI-DAQmx, you can configure virtual channels either in MAX or in your program, and you can configure channels as part of a task or separately. 3. Switch—a switch channel represents any connection point on a switch. It can be made up of one or more signal wires (commonly one, two, or four), depending on the switch topology. A virtual channel cannot be created with a switch channel. Switch channels may be used only in the NI-DAQmx Switch functions and VIs.
chart	2D display of one or more plots in which the display retains a history of previous data, up to a maximum that you define. The chart receives the data and updates the display point by point or array by array, retaining a certain number of past points in a buffer for display purposes. *See also* scope chart, strip chart, and sweep chart.
checkbox	Small square box in a dialog box you can select or clear. Checkboxes generally are associated with multiple options that you can set. You can select more than one checkbox.
class	A category containing properties, methods, and events. Classes are arranged in a hierarchy with each class inheriting the properties and methods associated with the class in the preceding level.
cluster	A set of ordered, unindexed data elements of any data type, including numeric, Boolean, string, array, or cluster. The elements must be all controls or all indicators.
cluster shell	Front panel object that contains the elements of a cluster.
coercion	Automatic conversion LabVIEW performs to change the numeric representation of a data element.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

coercion dot	Appears on a block diagram node to alert you that you have wired data of two different numeric data types together. Also appears when you wire any data type to a variant data type.
Coloring tool	Tool to set foreground and background colors.
compile	Process that converts high-level code to machine-executable code. LabVIEW compiles VIs automatically before they run for the first time after you create or edit alteration.
conditional terminal	Terminal of a While Loop that contains a Boolean value that determines if the VI performs another iteration.
configuration utility	Refers to Measurement & Automation Explorer on Windows and configuration utilities for the instrument on Mac OS and Linux.
connector	Part of the VI or function node that contains input and output terminals. Data values pass to and from the node through a connector.
connector pane	Region in the upper right corner of a front panel or block diagram window that displays the VI terminal pattern. It defines the inputs and outputs you can wire to a VI.
constant	A terminal on the block diagram that supplies fixed data values to the block diagram. *See also* universal constant and user-defined constant.
Context Help window	Window that displays basic information about LabVIEW objects when you move the cursor over each object. Objects with context help information include VIs, functions, constants, structures, palettes, properties, methods, events, dialog box components, and items in the Project Explorer window.
control	Front panel object for entering data to a VI interactively or to a subVI programmatically, such as a knob, push button, or dial.
control flow	Programming system in which the sequential order of instructions determines execution order. Most text-based programming languages are control flow languages.
Controls palette	Palette that contains front panel controls, indicators, and decorative objects.
conversion	Changing the type of a data element.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

count terminal	Terminal of a For Loop whose value determines the number of times the For Loop executes its subdiagram.
current VI	VI whose front panel, block diagram, or Icon Editor is the active window.

D

DAQ	*See* data acquisition (DAQ) and NI-DAQ.
DAQ Assistant	A graphical interface for configuring measurement tasks, channels, and scales.
DAQ device	A device that acquires or generates data and can contain multiple channels and conversion devices. DAQ devices include plug-in drivers, PCMCIA cards, and DAQPad devices, which connect to a computer USB or 1394 (FireWire™) port. SCXI modules are considered DAQ devices.
data acquisition (DAQ)	1. Acquiring and measuring analog or digital electrical signals from sensors, acquisition transducers, and test probes or fixtures. 2. Generating analog or digital electrical signals.
data dependency	Condition in a dataflow programming language in which a node cannot execute until it receives data from another node. *See also* artificial data dependency.
data flow	Programming system that consists of executable nodes that execute only when they receive all required input data. The nodes produce output data automatically when they execute. LabVIEW is a dataflow system. The movement of data through the nodes determines the execution order of the VIs and functions on the block diagram.
data type	Format for information. In LabVIEW, acceptable data types for most VIs and functions are numeric, array, string, Boolean, path, refnum, enumerated type, waveform, and cluster.
datalog	To acquire data and simultaneously store it in a disk file. LabVIEW File I/O VIs and functions can log data.
datalog file	File that stores data as a sequence of records of a single, arbitrary data type that you specify when you create the file. Although all the records in a datalog file must be a single type, that type can be complex. For example, you can specify that each record is a cluster that contains a string, a number, and an array.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

default	Preset value. Many VI inputs use a default value if you do not specify a value.
device	An instrument or controller you can access as a single entity that controls or monitors real-world I/O points. A device often is connected to a host computer through some type of communication network. *See also* DAQ device and measurement device.
dialog box	Window that appears when an application needs further information to carry out a command.
dimension	Size and structure of an array.
directory	Structure for organizing files into convenient groups. A directory is like an address that shows the location of files. A directory can contain files or subdirectories of files.
discrete	Having discontinuous values of the independent variable, usually time.
drag	To use the cursor on the screen to select, move, copy, or delete objects.
drive	Letter in the range a-z followed by a colon ( ), to indicate a logical disk drive.
driver	Software that controls a specific hardware device, such as a DAQ device.

E

edit mode	When you can make changes to a VI.
empty array	Array that has zero elements but has a defined data type. For example, an array that has a numeric control in its data display window but has no defined values for any element is an empty numeric array.
error cluster	Consists of a Boolean status indicator, a numeric code indicator, and a string source indicator.
error in	Error cluster that enters a VI.
error message	Indication of a software or hardware malfunction or of an unacceptable data entry attempt.
error out	The error cluster that leaves a VI.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

event	Condition or state of an analog or digital signal.
execution highlighting	Debugging technique that animates VI execution to illustrate the data flow in the VI.
Express VI	A subVI designed to aid in common measurement tasks. You configure an Express VI using a configuration dialog box.

F

Flat Sequence structure	Program control structure that executes its subdiagrams in numeric order. Use this structure to force nodes that are not data dependent to execute in the order you want if flow-through parameters are not available. The Flat Sequence structure displays all the frames at once and executes the frames from left to right until the last frame executes.
For Loop	Iterative loop structure that executes its subdiagram a set number of times. Equivalent to text-based code: `For i = 0 to n – 1, do`
frame	Subdiagram of a Flat or Stacked Sequence structure.
free label	Label on the front panel or block diagram that does not belong to any other object.
frequency	f, the basic unit of rate, measured in events or oscillations per second using a frequency counter or spectrum analyzer. Frequency is the reciprocal of the period of a signal.
front panel	Interactive user interface of a VI. Front panel appearance imitates physical instruments, such as oscilloscopes and multimeters.
function	Built-in execution element, comparable to an operator, function, or statement in a text-based programming language.
Functions palette	Palette that contains VIs, functions, block diagram structures, and constants.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

G

G	Graphical programming language LabVIEW uses.
General Purpose Interface Bus	GPIB. Synonymous with HP-IB. The standard bus used for controlling electronic instruments with a computer. Also called IEEE 488 bus because it is defined by ANSI/IEEE Standards 488-1978, 488.1-1987, and 488.2-1992.
glyph	Small picture or icon.
GPIB	*See* General Purpose Interface Bus.
graph	2D display of one or more plots. A graph receives and plots data as a block.
graph control	Front panel object that displays data in a Cartesian plane.

H

handle	Pointer to a pointer to a block of memory that manages reference arrays and strings. An array of strings is a handle to a block of memory that contains handles to strings.
hex	Hexadecimal. Base-16 number system.
Hierarchy window	*See* VI Hierarchy window.

I

I/O	Input/Output. The transfer of data to or from a computer system involving communications channels, operator input devices, and/or data acquisition and control interfaces.
icon	Graphical representation of a node on a block diagram.
indicator	Front panel object that displays output, such as a graph or LED.
`1nf`	Digital display value for a floating-point representation of infinity.
instrument driver	A set of high-level functions that control and communicate with instrument hardware in a system.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

integer — Any of the natural numbers, their negatives, or zero.

intensity map/plot — Method of displaying three dimensions of data in a 2D plot with the use of color.

iteration terminal — Terminal of a For Loop or While Loop that contains the current number of completed iterations.

IVI — Interchangeable Virtual Instruments. A software standard for creating a common interface (API) to common test and measurement instruments.

L

label — Text object used to name or describe objects or regions on the front panel or block diagram.

Labeling tool — Tool to create labels and enter text into text windows.

LabVIEW — Laboratory Virtual Instrument Engineering Workbench. LabVIEW is a graphical programming language that uses icons instead of lines of text to create programs.

LED — Light-emitting diode.

legend — Object a graph or chart owns to display the names and plot styles of plots on that graph or chart.

library — *See* LLB or project library.

listbox — Box within a dialog box that lists all available choices for a command. For example, a list of filenames on a disk.

LLB — LabVIEW file that contains a collection of related VIs for a specific use.

M

marquee — Moving, dashed border that surrounds selected objects.

matrix — A rectangular array of numbers or mathematical elements that represent the coefficients in a system of linear equations.

MAX — *See* Measurement & Automation Explorer.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Measurement & Automation Explorer	The standard National Instruments hardware configuration and diagnostic environment for Windows.
measurement device	A DAQ device such as the E Series multifunction I/O (MIO) device, the SCXI signal conditioning module, and the switch module.
menu bar	Horizontal bar that lists the names of the main menus of an application. The menu bar appears below the title bar of a window. Each application has a menu bar that is distinct for that application, although some menus and commands are common to many applications.
method	A procedure that is executed when an object receives a message. A method is always associated with a class.

N

NaN	Digital display value for a floating-point representation of <Not A Number>. Typically the result of an undefined operation, such as log(–1).
NI-DAQ	Driver software included with all NI DAQ devices and signal conditioning components. NI-DAQ is an extensive library of VIs and functions you can call from an application development environment (ADE), such as LabVIEW, to program an NI measurement device, such as the M Series multifunction I/O (MIO) DAQ devices, signal conditioning modules, and switch modules.
NI-DAQmx	The latest NI-DAQ driver with new VIs, functions, and development tools for controlling measurement devices. The advantages of NI-DAQmx over earlier versions of NI-DAQ include the DAQ Assistant for configuring channels and measurement tasks for a device for use in LabVIEW, LabWindows™/CVI™, and Measurement Studio; increased performance such as faster single-point analog I/O; NI-DAQmx simulation for most supported devices for testing and modifying applications without plugging in hardware; and a simpler, more intuitive API for creating DAQ applications using fewer functions and VIs than earlier versions of NI-DAQ.
node	Program execution element. Nodes are analogous to statements, operators, functions, and subroutines in text-based programming languages. On a block diagram, nodes include functions, structures, and subVIs.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

non-displayable characters	ASCII characters that cannot be displayed, such as null, backspace, tab, and so on.
numeric controls and indicators	Front panel objects to manipulate and display numeric data.

O

object	Generic term for any item on the front panel or block diagram, including controls, indicators, nodes, wires, and imported pictures.
one-dimensional	Having one dimension, as in the case of an array that has only one row of elements.
Operating tool	Tool to enter data into controls or to operate them.
operator	Person who initiates and monitors the operation of a process.

P

palette	Displays objects or tools you can use to build the front panel or block diagram.
picture	Series of graphics instructions that a picture indicator uses to create a picture.
pixel	Smallest unit of a digitized picture.
plot	Graphical representation of an array of data shown either on a graph or a chart.
point	Cluster that contains two 16-bit integers that represent horizontal and vertical coordinates.
polymorphism	Ability of a node to automatically adjust to data of different representation, type, or structure.
Positioning tool	Tool to move and resize objects.
probe	Debugging feature for checking intermediate values in a VI.
Probe tool	Tool to create probes on wires.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

project	A collection of LabVIEW files and non-LabVIEW files that you can use to create build specifications and deploy or download files to targets.
Project Explorer window	Window in which you can create and edit LabVIEW projects.
project library	A collection of VIs, type definitions, shared variables, palette menu files, and other files, including other project libraries.
prototype	Simple, quick implementation of a particular task to demonstrate that the design has the potential to work. The prototype usually has missing features and might have design flaws. In general, prototypes should be thrown away, and the feature should be reimplemented for the final version.
pull-down menus	Menus accessed from a menu bar. Pull-down menu items are usually general in nature.
pulse	A signal whose amplitude deviates from zero for a short period of time.
PXI	PCI eXtensions for Instrumentation. A modular, computer-based instrumentation platform.

R

range	Region between the limits within which a quantity is measured, received, or transmitted. Expressed by stating the lower and upper range values.
rectangle	Cluster that contains four 16-bit integers. The first two values describe the vertical and horizontal coordinates of the top left corner. The last two values describe the vertical and horizontal coordinates of the bottom right corner.
refnum	Reference number. An identifier that LabVIEW uses as reference to an object such as a VI, application, or an ActiveX or .NET object. Use a refnum as an input parameter for a function or VI to perform an operation on the object.
representation	Subtype of the numeric data type, of which there are 8-, 16-, and 32-bit signed and unsigned integers, as well as single-, double-, and extended-precision, floating-point numbers.
resizing circles or handles	Circles or handles that appear on the borders of an object to indicate the points where you can resize the object.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ring control	Special numeric control that associates 32-bit integers, starting at 0 and increasing sequentially, with a series of text labels or graphics.
run mode	When a VI is running or reserved to run. A VI enters run mode when you click the Run or Run Continuously buttons on the front panel toolbar, the single-stepping buttons on the block diagram toolbar, or select Operate»Change to Run Mode. In run mode, all front panel objects have an abridged set of shortcut menu items. You cannot edit a VI while the VI runs.

S

sample	Single analog or digital input or output data point.
scalar	Number that a point on a scale can represent. A single value as opposed to an array. Scalar Boolean values and clusters are explicitly singular instances of their respective data types.
scale	Part of graph, chart, and some numeric controls and indicators that contains a series of marks or points at known intervals to denote units of measure.
scope chart	Numeric indicator modeled on the operation of an oscilloscope.
sequence structure	*See* Flat Sequence structure or Stacked Sequence structure.
shift register	Optional mechanism in loop structures to pass the value of a variable from one iteration of a loop to a subsequent iteration. Shift registers are similar to static variables in text-based programming languages.
shortcut menu	Menu accessed by right-clicking an object. Menu items pertain to that object specifically.
slider	Moveable part of slide controls and indicators.
source control	A solution to the problem of sharing VIs and controlling access to avoid accidental loss of data. You can use a source control provider to share files among multiple users, improve security and quality, and track changes to shared projects. Also called source code control.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Stacked Sequence structure	Program control structure that executes its subdiagrams in numeric order. Use this structure to force nodes that are not data dependent to execute in the order you want if flow-through parameters are not available. The Stacked Sequence structure displays each frame so you see only one frame at a time and executes the frames in order until the last frame executes.
string	Representation of a value as text.
string controls and indicators	Front panel objects to manipulate and display text.
strip chart	Numeric plotting indicator modeled after a paper strip chart recorder, which scrolls as it plots data.
structure	Program control element, such as a Flat Sequence structure, Stacked Sequence structure, Case structure, For Loop, or While Loop.
subdiagram	Block diagram within the border of a structure.
subVI	VI used on the block diagram of another VI. Comparable to a subroutine.
sweep chart	Numeric indicator modeled on the operation of an oscilloscope. It is similar to a scope chart, except that a line sweeps across the display to separate old data from new data.
syntax	Set of rules to which statements must conform in a particular programming language.

T

target	A device or machine on which a VI runs. You must use a LabVIEW project to work with an RT, FPGA, or PDA target.
terminal	Object or region on a node through which data values pass.
tip strip	Small yellow text banners that identify the terminal name and make it easier to identify terminals for wiring.
tool	Special cursor to perform specific operations.
toolbar	Bar that contains command buttons to run and debug VIs.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Tools palette	Palette that contains tools you can use to edit and debug front panel and block diagram objects.
top-level VI	VI at the top of the VI hierarchy. This term distinguishes the VI from its subVIs.
trigger	Any event that causes or starts some form of data capture.
tunnel	Data entry or exit terminal on a structure.
two-dimensional	Having two dimensions, as in the case of an array that has several rows and columns.

U

universal constant	Block diagram object you cannot edit that emits a particular ASCII character or standard numeric constant, for example, □.
user	*See* operator.
user-defined constant	Block diagram object that emits a value you set.

V

VI	*See* virtual instrument (VI).
VI Hierarchy window	Window that graphically displays the hierarchy of VIs and subVIs.
virtual instrument (VI)	Program in LabVIEW that models the appearance and function of a physical instrument.
Virtual Instrument Software Architecture	VISA. Single interface library for controlling GPIB, VXI, RS-232, and other types of instruments.
VISA	*See* Virtual Instrument Software Architecture.

W

waveform	Multiple voltage readings taken at a specific sampling rate.
waveform chart	Indicator that plots data points at a certain rate.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

While Loop	Loop structure that repeats a section of code until a condition occurs.
wire	Data path between nodes.
wire branch	Section of wire that contains all the wire segments from junction to junction, terminal to junction, or terminal to terminal if there are no junctions between.
wire segment	Single horizontal or vertical piece of wire.
wire stubs	Truncated wires that appear next to unwired terminals when you move the Wiring tool over a VI or function node.
Wiring tool	Tool to define data paths between terminals.
wizard	A dialog box with a sequence of pages through which you can move forward and backward as you fill in information.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Index

Numerics

3D graphs, 10-10

A

abridged menus, 3-4
adding
    space to front panel, 4-15
    terminals to functions, 5-5
additional documentation, 1-1
    See also related documentation
aligning objects, 4-14
annotations
    See also labeling
    using, 10-17
Application Builder
    readme, 1-3
application font, 4-16
arrays
    auto-indexing loops, 8-5
    building with loops, 8-7
    controls and indicators, 4-7
    creating constants, 9-7
    creating controls and indicators, 9-7
    default data, 9-10
    dimensions, 9-4
    examples of 1D arrays, 9-5
    examples of 2D arrays, 9-6
    indexes in multidimensional arrays, 9-4
    indexes on multidimensional arrays, 9-7
    restrictions, 9-4
    size of, 9-10
artificial data dependency, 5-11
auto-indexing
    default data, 8-11
    For Loops, 8-6
    While Loops, 8-6
automatic wiring, 5-8

B

binary
    creating files, 11-7
block diagram, 2-2
    adding terminals to functions, 5-5
    coercion dots, 5-9
    constants, 5-3
    data flow, 5-9
    data types, 5-2
    designing, 5-13
    displaying terminals, 5-1
    fonts, 4-16
    functions, 5-4
    labels, 4-16
    nodes, 5-3
    objects, 5-1
    options, 3-7
    removing terminals from functions, 5-5
    structures, 8-1
    terminals for controls and indicators, 5-1
    wiring automatically, 5-8
    wiring manually, 5-6
Boolean controls and indicators, 4-5
Breakpoint tool
    debugging VIs, 6-4
broken VIs
    common causes, 6-3
    correcting, 6-2
    displaying errors, 6-2
broken wires, 5-8
building
    block diagram, 5-1
    front panel, 4-1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

polymorphic VIs, 7-6
subVIs, 7-1
buttons
front panel, 4-5

C

Case structures
data types, 8-12
error handling, 6-7
executing, 8-11
selector terminals, 8-12
specifying a default case, 8-12
certification (NI resources), A-1
characters
formatting, 4-16
charts, 10-1
customizing appearance, 10-15
customizing behavior, 10-18
graph palette, 10-14
history length, 10-19
intensity, 10-4
multiple scales, 10-13
options, 10-13
overlaid plots, 10-20
scale formatting, 10-13
scrolling, 10-15
stacked plots, 10-20
types, 10-1
update mode, 10-19
waveform, 10-3
classic controls and indicators, 4-2
clusters
constants, 9-11
controls and indicators, 4-7
creating, 9-11
error, 6-6
order of elements, 9-11
wire patterns, 9-10
coercion dots, 5-9
color
high-color controls and indicators, 4-2
low-color controls and indicators, 4-2
mapping, 10-7
options, 3-7
coloring
front panel objects, 4-14
combo boxes, 4-6
communication
file I/O, 11-1
computer-based instruments
configuring, 1-4
conditional terminals, 8-3
configuring
front panel controls, 4-12
front panel indicators, 4-12
front panels, 4-13
VI appearance and behavior, 7-7
connecting terminals, 5-6
connector panes, 2-5
building, 7-3
printing, 12-2
constants, 5-3
arrays, 9-7
clusters, 9-11
containers, 4-9
subpanel controls, 4-9
tab controls, 4-9
Context Help window, 3-5
creating object descriptions, 12-1
creating VI descriptions, 12-1
continuously running VIs, 6-1
control flow programming model, 5-10
controls, 4-1
array, 4-7
Boolean, 4-5
changing to indicators, 4-13
classic, 4-2
cluster, 4-7
coloring, 4-14

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

data type terminals, 5-1
data types, 5-2
dialog, 4-2
displaying optional elements, 4-13
enumerated type, 4-8
grouping, 4-14
guidelines for using on front panel, 4-17
hiding, 4-13
high-color, 4-2
I/O name, 4-10
icons, 5-1
listbox, 4-7
locking, 4-14
low-color, 4-2
matrix, 4-7
modern, 4-2
navigating, 3-2
numeric, 4-3
on block diagram, 5-1
optional elements, 4-13
palette, 3-1
path, 4-7
printing, 12-2
refnum, 4-11
replacing, 4-13
resizing, 4-15
ring, 4-8
rotary, 4-4
scroll bar, 4-4
searching, 3-2
slide, 4-3
string, 4-6
string display types, 9-2
tab, 4-9
table, strings in, 9-2
terminals, 5-1
time stamp, 4-4
user interface design, 4-17
Controls palette, 3-1
  navigating, 3-2
  searching, 3-2
correcting
  broken VIs, 6-2
  broken wires, 5-8
  VIs with unexpected data, 6-3
count terminals, 8-2
  auto-indexing to set, 8-6
creating
  arrays, 9-7
  binary files, 11-7
  clusters, 9-11
  datalog files, 11-8
  icons, 7-2
  object descriptions, 12-1
  spreadsheet files, 11-6
  subVIs, 7-1
  subVIs from sections of a VI, 7-4
  text files, 11-6
  tip strips, 12-1
  user-defined constants, 5-3
  VI descriptions, 12-1
cursors
  graph, 10-16
customizing
  palettes, 3-7
  VI appearance and behavior, 7-7
  work environment, 3-7

D

DAQ
  passing channel names, 4-10
data dependency, 5-10
  artificial, 5-11
  missing, 5-11
data flow. *See* dataflow
data types, 5-2
  case selector values, 8-12
  control and indicator, 5-2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

default values, 5-2
printing, 12-2
waveform, 10-3
dataflow
observing, 6-3
dataflow programming model, 5-9
managing memory, 5-12
datalog files
creating, 11-8
reading from, 11-8
debugging
automatic error handling, 6-5
broken VIs, 6-2
error handling, 6-5
loops, 8-11
options, 3-7
single-stepping, 6-4
techniques, 6-3
undefined data, 5-3
using execution highlighting, 6-3
using the Breakpoint tool, 6-4
default cases, 8-12
default data
arrays, 9-10
loops, 8-11
default values
data types, 5-2
deleting
broken wires, 5-8
designing
block diagram, 5-13
dialog boxes, 4-17
front panel, 4-17
user interfaces, 4-17
diagnostic tools (NI resources), A-1
dialog boxes
controls, 4-2
designing, 4-17
font, 4-16
indicators, 4-2
labels, 4-2
ring controls, 4-8
dials
See also numeric
front panel, 4-4
digital data
digital waveform data type, 10-10
digital graphs, 10-7
digital waveform data type, 10-10
digital waveform graph
displaying digital data in, 10-7
dimensions
arrays, 9-4
disk space
options, 3-7
disk streaming, 11-5
displaying
errors, 6-2
optional elements in front panel objects, 4-13
terminals, 5-1
warnings, 6-2
distributing
objects on the front panel, 4-14
documentation, 1-1
See also related documentation
guide, 1-1
NI resources, A-1
using with other resources, 1-1
documenting VIs
creating object descriptions, 12-1
creating tip strips, 12-1
creating VI descriptions, 12-1
help files, 12-2
printing, 12-2
dots
coercion, 5-9
drivers (NI resources), A-1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

E

enumerated type controls, 4-8
errors
  automatically handling, 6-5
  broken VIs, 6-2
  checking for, 6-5
  clusters, 6-6
  codes, 6-6
  debugging techniques, 6-3
  displaying, 6-2
  finding, 6-2
  handling, 6-5
  handling automatically, 6-5
  handling using Case structures, 6-7
  handling using While Loops, 6-7
  I/O, 6-6
  list, 6-2
  methods to handle, 6-6
  window, 6-2
examples, 1-4
  NI resources, A-1
execution
  debugging VIs, 6-3
  flow, 5-9
  highlighting, 6-3
Express VIs and functions
  overview, 5-5

F

Feedback Node
  initializing, 8-10
  replacing with shift registers, 8-11
  selecting, 8-10
file I/O, 11-1
  advanced file functions, 11-3
  basic operation, 11-1
  creating binary files, 11-7
  creating datalog files, 11-8
  creating spreadsheet files, 11-6
  creating text files, 11-6
  disk streaming, 11-5
  formats, 11-2
  functions for common operations, 11-3
  paths, 11-4
  reading datalog files, 11-8
  reading waveforms, 11-9
  refnums, 11-1
  spreadsheet files, 11-6
  using Storage VIs, 11-5
  VIs for common operations, 11-3
  writing waveforms, 11-8
finding
  controls, VIs, and functions on the palettes, 3-2
  errors, 6-2
fixing
  VIs, 6-2
Flat Sequence structures
  executing, 8-14
floating-point numbers
  overflow and underflow, 5-3
flow of execution, 5-9
fonts
  application, 4-16
  dialog, 4-16
  options, 3-7
  settings, 4-16
  system, 4-16
For Loops
  auto-indexing, 8-6
  controlling timing, 8-5
  count terminals, 8-2
  default data, 8-11
  executing, 8-2
  iteration terminals, 8-2
  shift registers, 8-7
format string parameters, 9-4
formats for file I/O, 11-2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

formatting
specifiers in strings, 9-4
strings, 9-3
text on front panel, 4-16
free labels, 4-16
front panel, 2-2
adding space without resizing, 4-15
aligning objects, 4-14
changing controls to indicators, 4-13
changing indicators to controls, 4-13
coloring objects, 4-14
controls, 4-1
designing, 4-17
displaying optional object elements, 4-13
distributing objects, 4-14
fonts, 4-16
grouping objects, 4-14
hiding optional elements, 4-13
indicators, 4-1
labels, 4-16
loading in subpanel controls, 4-9
locking objects, 4-14
options, 3-7
overlapping objects, 4-9
replacing objects, 4-13
resizing objects, 4-15
spacing objects evenly, 4-14
terminals, 5-1
text characteristics, 4-16
full menus, 3-4
functions, 5-4
adding terminals, 5-5
navigating, 3-2
removing terminals, 5-5
searching, 3-2
Functions palette, 3-2
customizing, 3-7
navigating, 3-2
searching, 3-2

G

gauges
See also numeric
front panel, 4-4
getting started, 1-2
GPIB
configuring, 1-4
graph palette, 10-14
graphs, 10-1
3D, 10-10
annotating data points, 10-17
cursors, 10-16
customizing 3D, 10-18
customizing appearance, 10-15
customizing behavior, 10-15
intensity, 10-4
multiple scales, 10-13
options, 10-13
palette, 10-14
scale formatting, 10-13
scaling, 10-13
scrolling, 10-15
types, 10-1
waveform, 10-2
XY, 10-3
gray dots on block diagram, 5-9
grid, 4-14
options, 3-7
grouping
data in arrays, 9-4
data in clusters, 9-10
data in strings, 9-1
front panel objects, 4-14

H

hardware
configuring, 1-4
help
See also Context Help window

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

technical support, A-1
help files
creating, 12-2
HTML, 12-2
RTF, 12-2
help system
related documentation, 1-1
hiding
menu bars, 4-17
optional elements in front panel objects, 4-13
scroll bars, 4-17
hierarchy of VIs
printing, 12-2
viewing, 7-4
highlighting execution
debugging VIs, 6-3
history
charts, 10-19
options, 3-7
horizontal scroll bar, 4-4
HTML
help files, 12-2

I

I/O
See also file I/O
error, 6-6
name controls and indicators, 4-10
icons, 2-5
creating, 7-2
editing, 7-2
printing, 12-2
incrementally running VIs, 6-4
indexes
using on arrays, 9-4
indexing loops, 8-5
For Loops, 8-6
While Loops, 8-6
indicators, 4-1
array, 4-7
Boolean, 4-5
changing to controls, 4-13
classic, 4-2
cluster, 4-7
coloring, 4-14
data type terminals, 5-1
data types, 5-2
dialog, 4-2
displaying optional elements, 4-13
grouping, 4-14
guidelines for using on front panel, 4-17
hiding, 4-13
high-color, 4-2
I/O name, 4-10
icons, 5-1
locking, 4-14
low-color, 4-2
matrix, 4-7
modern, 4-2
numeric, 4-3
on block diagram, 5-1
optional elements, 4-13
path, 4-7
printing, 12-2
replacing, 4-13
resizing, 4-15
rotary, 4-4
scroll bar, 4-4
slide, 4-3
string, 4-6
string display types, 9-2
tab, 4-9
terminals, 5-1
time stamp, 4-4
user interface design, 4-17
infinite While Loops, 8-5
infinity floating-point value, 5-3

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

installing
LabVIEW, 1-2
instances of polymorphic VIs
See also polymorphic VIs
selecting manually, 7-5
instrument drivers (NI resources), A-1
instruments
configuring, 1-4
integers
overflow and underflow, 5-3
intensity charts, 10-4
color mapping, 10-7
options, 10-5
intensity graphs, 10-4
color mapping, 10-7
options, 10-6
introduction to LabVIEW, 1-1
iteration terminals
For Loops, 8-2
While Loops, 8-4
IVI
passing logical names, 4-10

K

knobs
See also numeric
front panel, 4-4
KnowledgeBase, A-1
known issues, 1-3

L

labeling
constants, 5-3
fonts, 4-16
labels
dialog box, 4-2
LabVIEW
customizing, 3-7
installing, 1-2
introduction, 1-1
options, 3-7
uninstalling, 1-2
launching
LabVIEW, 3-1
lights on front panel, 4-5
listboxes, 4-7
controls, 4-7
locking
front panel objects, 4-14
loops
auto-indexing, 8-5
building arrays, 8-7
controlling timing, 8-5
default data, 8-11
For (overview), 8-2
infinite, 8-5
shift registers, 8-7
While (overview), 8-3

M

mapping colors for intensity graphs and charts, 10-7
matrices
controls and indicators, 4-7
Measurement & Automation Explorer, 1-4
memory
coercion dots, 5-9
managing with dataflow programming model, 5-12
menu bars
hiding, 4-17
menus, 3-4
abridged, 3-4
combo boxes, 4-6
ring controls, 4-8
shortcut, 3-4
meters
See also numeric
front panel, 4-4

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

most recently used menu items, 3-4
MRU menu items, 3-4

N

naming
  VIs, 7-6
National Instruments support and services, A-1
navigating
  Controls and Functions palette, 3-2
Navigation window
  features, 3-6
needles
  accessing from the shortcut menu, 4-4
NI support and services, A-1
nodes, 2-4
  block diagram, 5-3
  execution flow, 5-10
not a number floating-point value, 5-3
numbers
  overflow and underflow, 5-3
numeric
  controls and indicators, 4-3
  formatting, 4-3
  symbolic values, 5-3

O

objects
  aligning, 4-14
  block diagram, 5-1
  changing controls to and from indicators, 4-13
  coloring on front panel, 4-14
  creating descriptions, 12-1
  creating tip strips, 12-1
  displaying optional elements, 4-13
  distributing, 4-14
  front panel and block diagram terminals, 5-1
  grouping on the front panel, 4-14
  hiding on front panel, 4-13
  labeling, 4-16
  locking on the front panel, 4-14
  optional elements, 4-13
  overlapping on front panel, 4-9
  printing descriptions, 12-2
  replacing on front panel, 4-13
  resizing on front panel, 4-15
  spacing evenly, 4-14
  wiring automatically on block diagram, 5-8
  wiring manually on block diagram, 5-6
options
  setting, 3-7
order of cluster elements, 9-11
order of execution, 5-9
overflow in numbers, 5-3
overlaid plots, 10-20
overlapping front panel objects, 4-9
owned labels, 4-16

P

palettes
  Controls, 3-1
  customizing, 3-7
  customizing Controls, 3-7
  customizing Functions, 3-7
  Functions, 3-2
  navigating, 3-2
  options, 3-7
  Tools, 3-3
parameter lists. *See* connector panes
parameters
  data types, 5-2
  flow-through, 5-12
paths
  controls and indicators, 4-7
  file I/O, 11-4
  options, 3-7

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

patterns
terminal, 7-3
performance
options, 3-7
pictures
ring controls, 4-8
plots
overlaid, 10-20
stacked, 10-20
polymorphic
building VIs, 7-6
VIs, 7-5
pop-up menus. *See* shortcut menus
preferences. *See* options
previous versions
saving VIs, 7-7
printing
documentation of VIs, 12-2
options, 3-7
programming examples, 1-4
NI resources, A-1
pull-down menus on front panel, 4-8

Q

quick reference card, 1-2

R

radio buttons controls, 4-5
reading
files, 11-1
refnums
controls, 4-11
file I/O, 11-1
related documentation, 1-1
See also documentation
release notes, 1-2
removing
broken wires, 5-8
terminals from functions, 5-5
repeating
blocks of code, 8-1
Repeat-Until Loops. *See* While Loops
replacing
front panel objects, 4-13
resizing
front panel objects, 4-15
revision history
printing, 12-2
ring controls, 4-8
rotary controls and indicators, 4-4
running VIs, 6-1
run-time
shortcut menus, 3-4

S

saving VIs
for previous versions, 7-7
scaling
graphs, 10-13
scope chart, 10-19
scroll bar controls, 4-4
scroll bars
hiding, 4-17
listboxes, 4-7
scrolling
charts, 10-15
graphs, 10-15
searching
for controls, VIs, and functions on the palettes, 3-2
selecting
wires, 5-8
selector terminal values, 8-12
sequence structures
comparing Flat to Stacked, 8-13
controlling execution order, 5-11
overusing, 8-14
setting
work environment options, 3-7

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

shift registers, 8-7
shortcut menus, 3-4
  in run mode, 3-4
shortened menus, 3-4
simple menus, 3-4
single-stepping
  debugging VIs, 6-4
sink terminals. *See* indicators
sizing. *See* resizing
slide controls and indicators, 4-3
  See also numeric
sliders
  adding, 4-3
snap-to grid, 4-14
software (NI resources), A-1
source code. *See* block diagram
source terminals. *See* controls
space
  adding to front panel or block diagram, 4-15
spacing objects evenly, 4-14
speed of execution
  controlling, 8-5
spreadsheet files
  creating, 11-6
stacked plots, 10-20
Stacked Sequence structures
  executing, 8-14
statements. *See* nodes
stepping through VIs
  debugging VIs, 6-4
strings, 9-1
  combo boxes, 4-6
  controls, 4-6
  display types, 9-2
  editing programmatically, 9-3
  formatting, 9-3
  formatting specifiers, 9-4
  indicators, 4-6
  tables, 9-2
strip chart, 10-19
structures, 8-1
  Case, 8-11
  Event, 8-15
  Flat Sequence, 8-14
  For Loops, 8-2
  on block diagram, 2-5
  Stacked Sequence, 8-14
  While Loops, 8-3
subpanel controls, 4-9
subroutines. *See* subVIs
subVIs, 7-1
  building, 7-1
  creating, 7-1
  creating from sections of a VI, 7-4
  hierarchy, 7-4
  polymorphic VIs, 7-5
support
  technical, A-1
sweep chart, 10-19
switches
  front panel, 4-5
symbolic numeric values, 5-3
system
  controls and indicators, 4-2
system font, 4-16

T

tab controls, 4-9
tables, 4-8
tanks
  See also numeric
  slide controls and indicators, 4-3
technical support, A-1
templates
  VIs, 7-1
terminals, 2-3
  adding to functions, 5-5
  auto-indexing to set count, 8-6
  block diagram, 5-1
  coercion dots, 5-9

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

conditional, 8-3
constants, 5-3
control and indicator, 5-1
count, 8-2
displaying, 5-1
iteration on For Loops, 8-2
iteration on While Loops, 8-4
patterns, 7-3
printing, 12-2
removing from functions, 5-5
selector, 8-12
wiring, 5-6
text
entry boxes, 4-6
formatting, 4-16
ring controls, 4-8
text files
binary format, 11-7
creating, 11-6
creating for multiple platforms, 11-7
thermometers
See also numeric
slide controls and indicators, 4-3
time stamp
See also numeric
controls and indicators, 4-4
timing
controlling, 8-5
tip strips
creating, 12-1
toolbars, 3-5
project, 3-5
tools
getting started, 1-4
palette, 3-3
training (NI resources), A-1
tree controls, 4-7
troubleshooting
See also debugging
NI resources, A-1
tunnels, 8-1
input and output, 8-13
type controls
enumerated, 4-8

U

undefined data, 5-3
arrays, 9-10
infinity, 5-3
not a number, 5-3
underflow in numbers, 5-3
unexpected data, 5-3
ungrouping
front panel objects, 4-14
uninstalling LabVIEW, 1-2
unlocking
front panel objects, 4-14
upgrade notes, 1-2
upgrading VIs, 7-7
user interface. *See* front panel
user manual, 1-2

V

versions
saving VIs for previous, 7-7
vertical scroll bar, 4-4
VI Hierarchy window
displaying, 7-4
printing, 12-2
VIs, 2-1
broken, 6-2
configuring appearance and behavior, 7-7
correcting, 6-2
creating descriptions, 12-1
debugging techniques, 6-3
documenting, 12-1
error handling, 6-5
examples, 1-4
hierarchy, 7-4

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

naming, 7-6
polymorphic, 7-5
printing, 12-2
running, 6-1
templates, 7-1
upgrading, 7-7
VISA
passing resource names, 4-10

W

warnings
displaying, 6-2
waveform
charts, 10-3
data type, 10-3
graphs, 10-2
reading from files, 11-9
writing to files, 11-8
Web resources, A-1
While Loops
auto-indexing, 8-6
conditional terminals, 8-3
controlling timing, 8-5
default data, 8-11
error handling, 6-7
executing, 8-3
infinite, 8-5
iteration terminals, 8-4
shift registers, 8-7
wires, 2-4
broken, 5-8
selecting, 5-8
wiring
automatically, 5-8
manually, 5-7
objects, 5-7
wizards, 1-4
work environment options
setting, 3-7
writing
files, 11-1

X

x-scales
multiple, 10-13
XY graphs, 10-3

Y

y-scales
multiple, 10-13

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP