LabVIEW™ Express VI Development Toolkit User Guide

Version 1.0

The LabVIEW Express VI Development Toolkit allows you to create and edit Express VIs, which you can distribute to users for building applications easily.

This user guide contains five parts, each of which explains different aspects of how Express VIs work. Each part includes exercises that help you develop Express VIs. Throughout the exercises, you will learn how to use the Express VI Development Toolkit to create, edit, and distribute Express VIs.

Contents

August 2003

323413A-01

ni.com

Part One: About Express VIs

Objective To learn about the structure and functionality of Express VIs, including how they differ from VIs.

You can complete the exercises in this section in approximately 15 minutes.

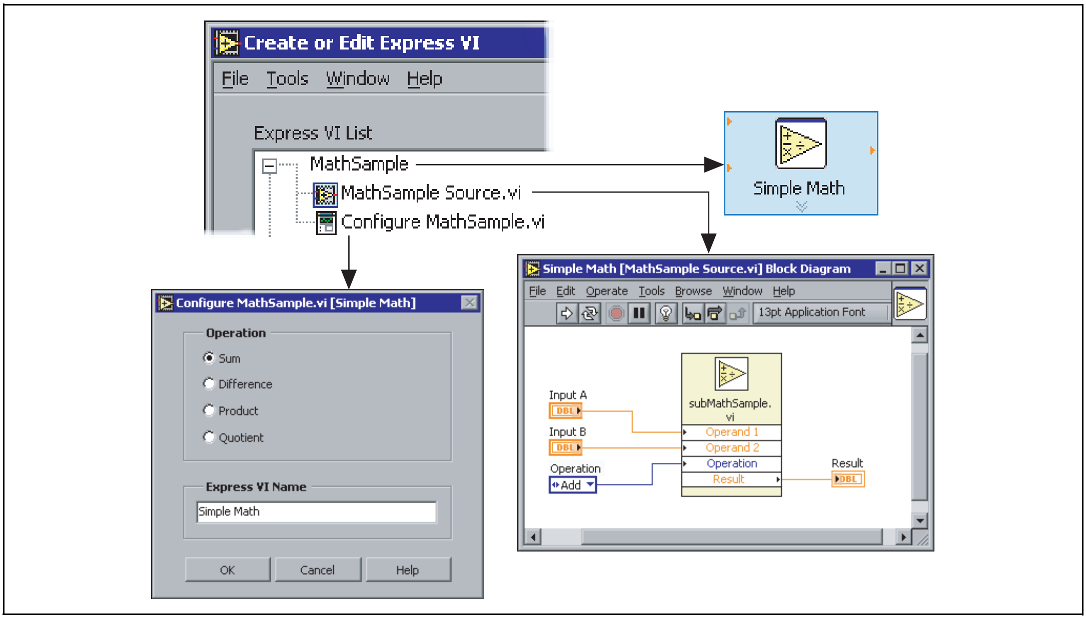
An Express VI is a VI that allows users to interactively configure its settings through a dialog box. An Express VI consists of the following components:

- Configuration dialog box VI—Allows users to configure settings for the run-time behavior of the Express VI.
- Source VI—Contains the code for the Express VI. The source VI also contains a link to the configuration dialog box VI.

The Express VI Development Toolkit saves both components in a library using the following naming conventions.

Component	Source VI Equivalent	Configuration Dialog Box VI Equivalent
File Library	*ExpressVIName*Source.llb	*ExpressVIName*Config.llb
VI Filename	*ExpressVIName*Source.vi	Configure *ExpressVIName*.vi
SubVI Filename	sub*ExpressVIName*.vi	genHelp*ExpressVIName*.vi

Figure 1 shows a graphical representation of an Express VI, using the MathSample Express VI as an example.



Operation

● Sum

○ Difference

○ Product

○ Quotient

Express VI Name: Simple Math

Operation: Add

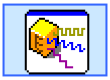
Figure 1. Express VI and Its Components

The Express VI Developmenbt Toolkit saves source VIs and configuration dialog box VIs in different LLBs that are stored together in the file system. If the source VI and configuration dialog box VI share subVIs, you can save the subVIs to a third LLB or place them in the most logical location.

The source VI contains a link to its associated configuration dialog box VI. If you move the configuration dialog box VI to a different location relative to `vi.lib` or `user.lib`, the link breaks, and the Express VI prompts users to find the configuration dialog box VI.

Examining an Express VI

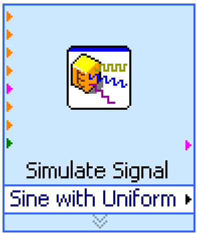
The Simulate Signal Express VI is included with LabVIEW to make generating signals easy. Complete the following steps to examine the Simulate Signal Express VI and learn how users interact with Express VIs.



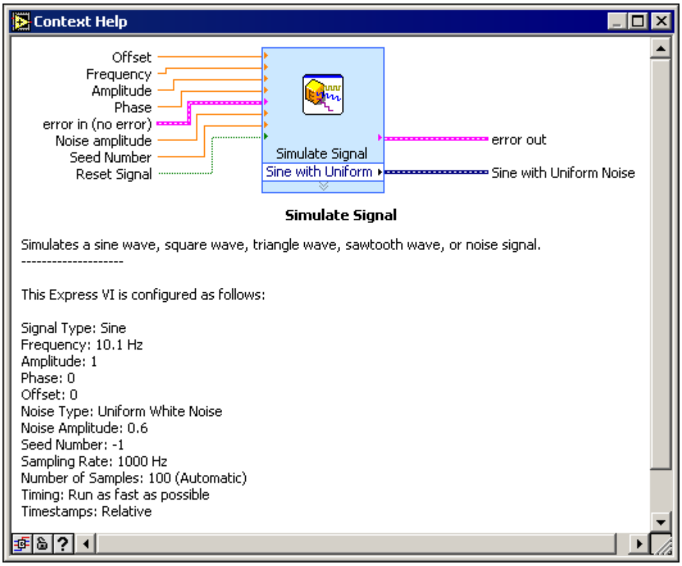
1. From the **Functions** palette, search for the Simulate Signal Express VI, shown at left. Notice that Express VIs on the palette have blue borders.
2. Place the Simulate Signal Express VI on an empty block diagram. The **Configure Simulate Signal** dialog box opens. The configuration dialog box allows users to interactively configure properties of the Express VI. Notice the **Result Preview** graph.

Note The configuration dialog box for an Express VI opens when you place an Express VI on the block diagram, when you double-click an Express VI, or when you right-click the Express VI and select **Properties** from the shortcut menu.

3. In the dialog box, place a checkmark in the **Add noise** checkbox. Notice the change in the **Result Preview** graph.
4. Click the **OK** button. The Express VI incorporates the dialog box settings you selected and uses those settings when the VI runs.



5. Notice that the Express VI on the block diagram expands to display the **Sine with Uniform** output, as shown at left. The Sine with Uniform output is an example of an expandable terminal, which is an Express VI terminal that you can configure from the block diagram.
6. Drag the arrow at the bottom of the Express VI to display the entire set of expandable terminals.
7. Press the <Ctrl-H> keys to display the **Context Help** window if it is not open already. The **Context Help** window shows the current configuration for the Express VI, as shown in Figure 2.



Simulate Signal

Simulates a sine wave, square wave, triangle wave, sawtooth wave, or noise signal.

This Express VI is configured as follows:

Signal Type: Sine
Frequency: 10.1 Hz
Amplitude: 1
Phase: 0
Offset: 0
Noise Type: Uniform White Noise
Noise Amplitude: 0.6
Seed Number: -1
Sampling Rate: 1000 Hz
Number of Samples: 100 (Automatic)
Timing: Run as fast as possible
Timestamps: Relative

Figure 2. Context Help Window for an Instance of the Simulate Signal Express VI

Some of the parameters listed in the **Context Help** window, such as **Timestamps**, do not appear as inputs or outputs on the Express VI.

8. Double-click the Express VI to display the configuration dialog box. Notice the **Time Stamps** radio buttons.

Terminals that you can configure from the dialog box are called configurable terminals. A terminal can be both configurable and expandable, which means you can configure it from both locations.

Note Values you wire directly to Express VI terminals take precedence over values you select in the configuration dialog box. For example, if you set **Amplitude** to `5.0` in the configuration dialog box of the Simulate Signal Express VI, but on the block diagram you wire a value of `10.0` to the **Amplitude** terminal, the output signal uses 10.0 for amplitude.

Examining Multiple Instances of an Express VI

When you place an Express VI on the block diagram, an instance of the Express VI is embedded in that particular block diagram. The settings that you select in the configuration dialog box affect only that instance of the Express VI. Changing the configuration dialog box settings for one instance of the Express VI does not affect other instances.

Complete the following steps to examine and compare different instances of the same Express VI.

1. Place a Simulate Signal Express VI on an empty block diagram.
2. From the **Configure Simulate Signal** dialog box, select settings to generate a sine waveform with uniform white noise.
3. Place another Simulate Signal Express VI on the block diagram.
4. From the **Configure Simulate Signal** dialog box, select settings to generate a square waveform with Gaussian white noise.
5. Notice the differences between the two instances of the Express VI. The first instance displays the **Simulate Signal** title and **Sine with Uniform** output. The second instance displays the **Simulate Signal2** title and **Square with Gaussian** output, as shown in Figure 3.

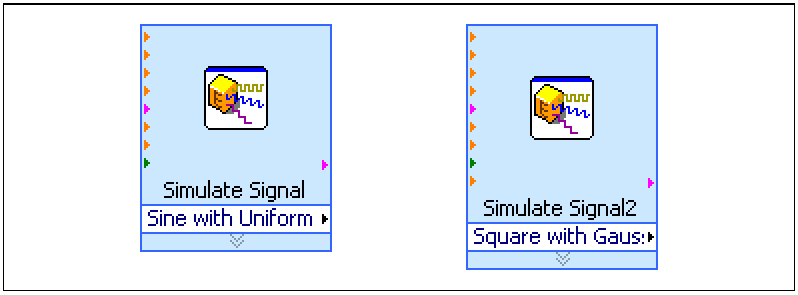


Figure 3. Separate Instances of an Express VI

The two instances of the Express VI are independent of each other. Each instance of an Express VI has a different name and can include different numbers and types of expandable terminals. The **Context Help** window updates with the different settings for each instance.

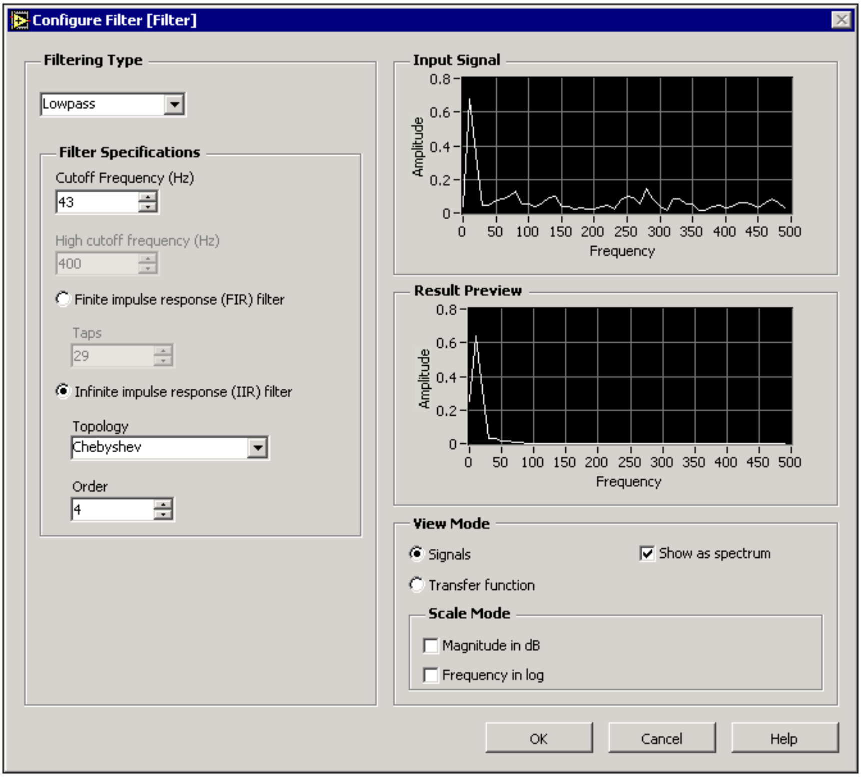
When you save a VI that contains an Express VI, the Express VI is also saved inside the VI. When you open the VI, the Express VIs within it load from the same file as the VI and not from the Express source VI file. Therefore, you must place all the functionality of the Express VI in a subVI that you can update as necessary.

You can convert an Express VI into a standard VI. Right-click the Express VI and select **Open Front Panel** from the shortcut menu. After you convert an Express VI into a standard VI, you cannot access the configuration dialog box or convert the VI back to the Express VI.

Advantages of Using Express VIs

The primary benefit of Express VIs is their interactive configurability. Express VIs are useful when you want to give users a VI or library of VIs for building their own applications easily with minimal programming expertise.

The configurability of Express VIs provides an interactive way to determine settings for operations that the user might not fully understand. For example, you might need a digital filter for a signal, but a library containing dozens of filter VIs does not help you choose the correct filter response or help you decide which parameters most directly affect the specific signal. However, the Filter Express VI allows you to interactively select the type of filter, manipulate filter parameters, and view the filter response in different ways. For example, you can use the **Result Preview** graph in the configuration dialog box to examine the results of different filters and filter parameters directly on the signal, as shown in Figure 4. After you run the Filter Express VI, the new signal displays as default data in the **Input Signal** and **Result Preview** graphs on the configuration dialog box.



Configure Filter [Filter]

Filtering Type

Lowpass

Filter Specifications

Cutoff Frequency (Hz) 43

High cutoff frequency (Hz) 400

○ Finite impulse response (FIR) filter

Taps 29

● Infinite impulse response (IIR) filter

Topology Chebyshev

Order 4

View Mode

● Signals

○ Transfer function

☑ Show as spectrum

Scale Mode

□ Magnitude in dB

□ Frequency in log

Figure 4. Using the Configuration Dialog Box to Examine a Noisy Sine Wave

Another advantage of Express VIs is that separate instances of an Express VI function independently. For example, if you place a VI in five different areas on the same block diagram, the result is five exact copies of the VI. The source code, default values, and front panel remain the same for all five copies. However, if you place an Express VI in five areas on a block diagram, the result is five separate Express VIs with different names, all independently configurable.

Caveats for Using Express VIs

Express VIs do not provide run-time interactive configuration for VIs. If you need run-time reconfiguration, build an application with a user interface that contains features similar to a configuration dialog box.

Express VIs are maximized for ease of use. If you need an application to run with memory restrictions or high execution speeds, use standard VIs.

Part Two: Learning the Basics of Express VI Creation

Objective To learn the fundamentals of creating an Express VI from an existing VI.

To create and distribute an Express VI, you must complete the following stages of development.

- Design the general framework for the Express VI. You can create an Express VI from an existing VI, an existing Express VI, or a blank VI.
- Design the configuration dialog box, including the user interface.
- Associate online help files with the Express VI.
- Place the Express VI in the LabVIEW directory structure.
- Add the Express VI to a new or existing palette.

You can complete the exercises in this section in approximately 30 minutes.

Refer to the `examples\ExpressVIDev` directory for the solution to these exercises.

Examining the Existing VI

In this exercise, you will create an Express VI that performs an arithmetic operation using an operator specified in the configuration dialog box. Use the Simple Calculator VI included with the Express VI Development Toolkit to create the Express VI.

Before you convert the Simple Calculator VI to an Express VI, review the VI to understand how it works, so you can examine how the functionality changes after you convert it to an Express VI.

Complete the following steps to examine the Simple Calculator VI.

1. Open the Simple Calculator VI located in the `examples\ExpressVIDev` directory. Notice the controls and indicators that appear on the front panel. You can enter two operands and the operator, and the VI performs the calculation.
2. Move the cursor over the VI icon. Notice the connector pane layout in the **Context Help** window, particularly the number and type of inputs and outputs.
3. Close the VI.

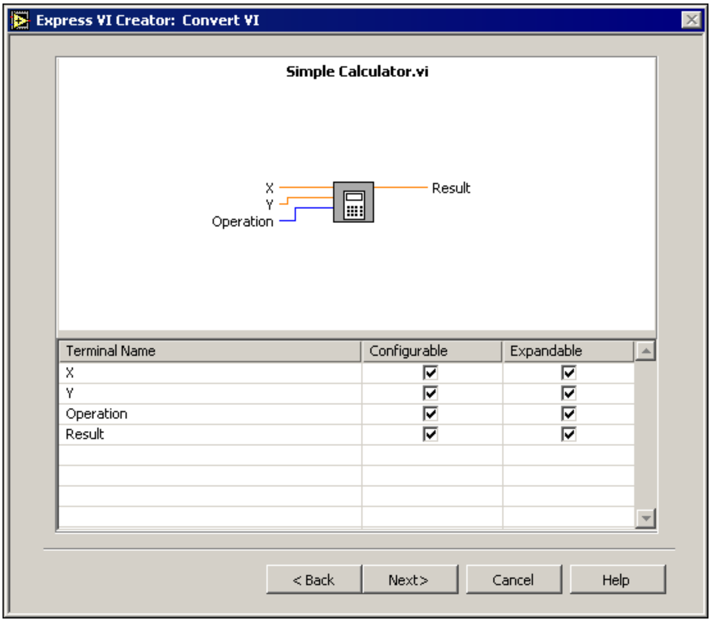
Converting a Standard VI to an Express VI

Complete the following steps to create an Express VI from the Simple Calculator VI.

1. Select **Tools»Create or Edit Express VI**.
2. From the **Create or Edit Express VI** dialog box, click the **New** button.
3. Select the **Create from a VI** option. Click the **Next** button.
4. Select the Simple Calculator VI in the `examples\ExpressVIDev` directory. Click the **OK** button.

   You can set Express VI terminals as configurable and/or expandable. The **Express VI Creator: Convert VI** dialog box maps the terminals to several elements on the block diagram of the configuration dialog box VI. The **Configurable** column in the dialog box corresponds to the **Control Data** array on the block diagram. The **Expandable** column in the dialog box corresponds to the **Terminals to Grow** array. You will learn more about these arrays in the *Completing the Configuration Dialog Box VI* section.

5. Keep the default checkmarks in the checkboxes in the **Configurable** and **Expandable** columns, as shown in Figure 5. This makes all the terminals configurable and expandable. Click the **Next** button.



Terminal Name	Configurable	Expandable
X	☑	☑
Y	☑	☑
Operation	☑	☑
Result	☑	☑

Figure 5. Mapping VI Terminals to Express VI Configurable or Expandable Terminals

6. Type `EasyMath` in the **Select a name for the Express VI** text box.
7. Save the Express VI in the `user.lib\_express` directory.

Tip Express VIs appear automatically on the **Express User Libraries** palette if you save them in the `user.lib\_express` directory.

8. Click the **Finish** button. The EasyMath Express VI appears in the **Express VI List** in the **Create or Edit Express VI** dialog box, as shown in Figure 6.

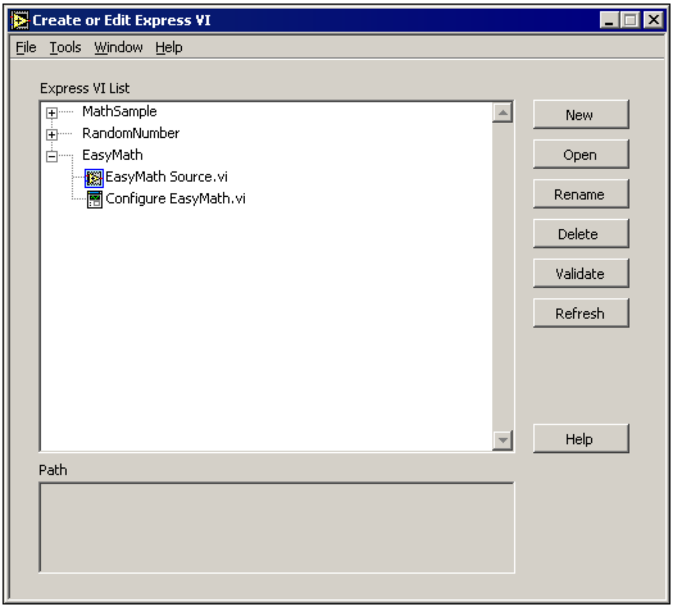


Figure 6. Selecting Express VI Components to Edit

9. Expand the EasyMath Express VI in the **Express VI List**. The source VI and the configuration dialog box VI are under the Express VI. Notice how the **Express VI List** displays the Express VI structure and names. The name of the the Express VI itself is on the top level.

Note Click the **Rename** button if you want to rename the Express VI and its components at the same time. Do not rename the source VI or the configuration dialog box VI except through the **Rename** button. Naming conventions provide a link between Express VI components, so improper renaming breaks the Express VI.

Examining the Source VI

To complete the EasyMath Express VI, you do not need to edit the source VI. However, you should examine the source VI structure since you might have to edit source VIs for other Express VIs you create.

Complete the following steps to examine the EasyMath Source VI.

1. Double-click the EasyMath source VI to open it.
2. Notice that the VI contains all the controls and indicators from the connector pane of the VI from which it was created, as well as the same connector pane layout.

Tip When you create an Express VI from an existing standard VI, the source VI automatically contains the controls and indicators you select as expandable, the connector pane, and the VI icon.

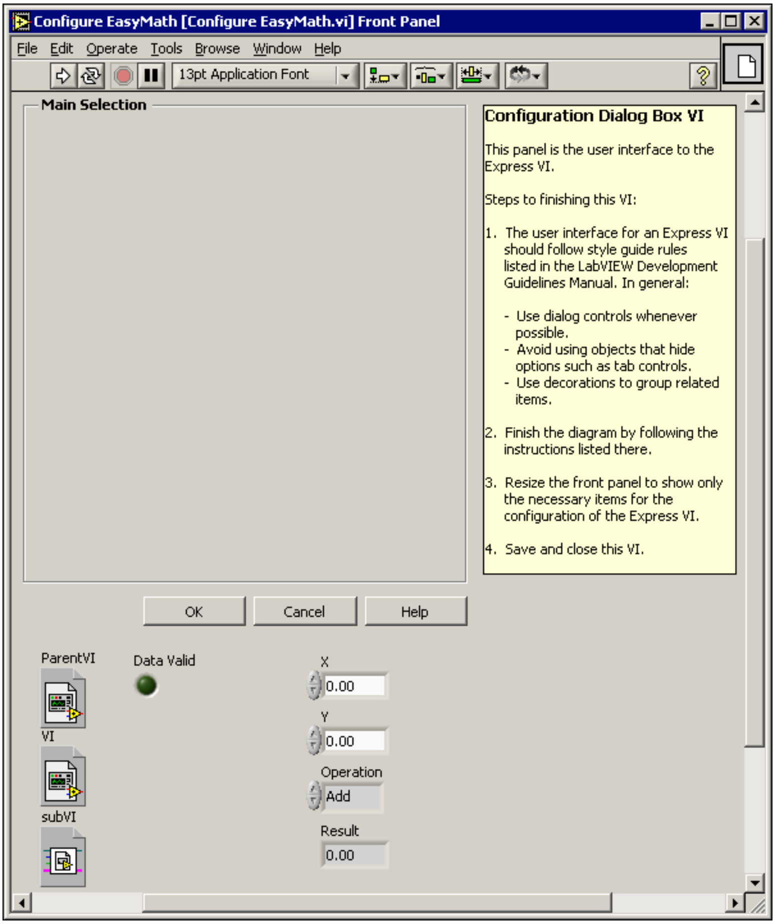
3. Notice the VI icon. The icon you use in the source VI becomes the Express VI icon.
4. Close the EasyMath Source VI.

Completing the Configuration Dialog Box VI

Edit the configuration dialog box VI for the EasyMath Express VI so users can select a mathematical operation from the configuration dialog box that specifies the type of calculation they want to perform.

Complete the following steps to examine and build the front panel of the EasyMath configuration dialog box VI.

1. From the **Create or Edit Express VI** dialog box, open the EasyMath configuration dialog box VI. Notice that the controls and indicators from the Simple Calculator VI that you selected as configurable appear automatically on the front panel, as shown in Figure 7.


Configure EasyMath [Configure EasyMath.vi] Front Panel
File Edit Operate Tools Browse Window Help
13pt Application Font
Main Selection
OK
Cancel
Help
ParentVI
VI
subVI
Data Valid

Configuration Dialog Box VI

This panel is the user interface to the Express VI.

Steps to finishing this VI:

1. The user interface for an Express VI should follow style guide rules listed in the LabVIEW Development Guidelines Manual. In general:
   - Use dialog controls whenever possible.
   - Avoid using objects that hide options such as tab controls.
   - Use decorations to group related items.
2. Finish the diagram by following the instructions listed there.
3. Resize the front panel to show only the necessary items for the configuration of the Express VI.
4. Save and close this VI.

X 0.00

Y 0.00

Operation Add

Result 0.00

Figure 7. Configure EasyMath VI Front Panel

2. On the front panel, move the **Operation** control into the **Main Selection** frame. The **Main Selection** frame contains the user interface objects for the configuration dialog box.
3. Resize the front panel to display only the **Main Selection** frame and the **OK**, **Cancel**, and **Help** buttons, as shown in Figure 8.

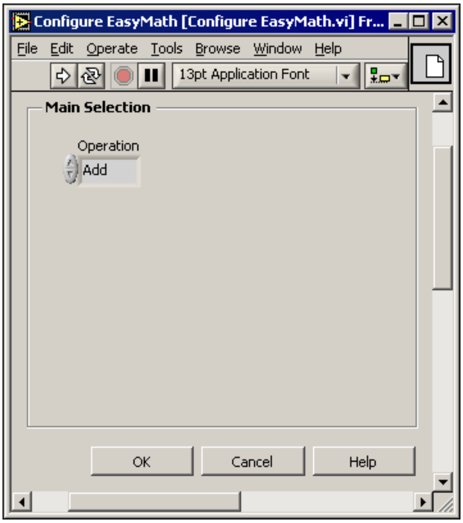

Configure EasyMath [Configure EasyMath.vi] Fr...
File Edit Operate Tools Browse Window Help
13pt Application Font
Main Selection
Operation
Add
OK
Cancel
Help

Figure 8. Setting the Configuration Dialog Box Size

4. Save the VI.

Complete the following steps to build the block diagram of the Configure EasyMath VI.

1. Open the block diagram.
2. Notice the three sections of the block diagram: Initialize, Main While Loop, and Reconfigure. Each section represents a different stage of the configuration dialog box process. You will learn more about these sections in the *Exploring the Configuration Dialog Box Block Diagram* section.
3. In the Initialize section, remove the the **X** and **Y** elements from the **Control Data** array. The **Control Data** array lists the configurable terminals for the Express VI, which appear in the configuration dialog box. The **X** and **Y** controls should not be configurable because you wire them to the Express VI.

Tip To delete an element in an array, right-click the element and select **Data Operations» Delete Element** from the shortcut menu.

The completed Initialize section should resemble Figure 9.

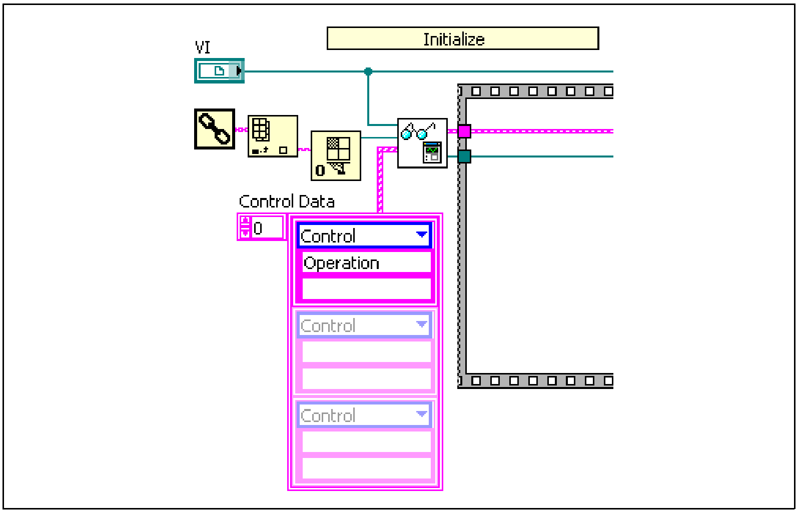


Figure 9. Removing Expandable Terminals from the **Control Data** Array

4. In the Reconfigure section, remove the **Operation** element from the **Terminals to Grow** array. The **Terminals to Grow** array lists the expandable terminals for the Express VI, which appear after you select settings in the configuration dialog box. The **Operation** terminal should not be expandable because you set the control from the configuration dialog box of the Express VI.

   The completed Reconfigure section should resemble Figure 10.

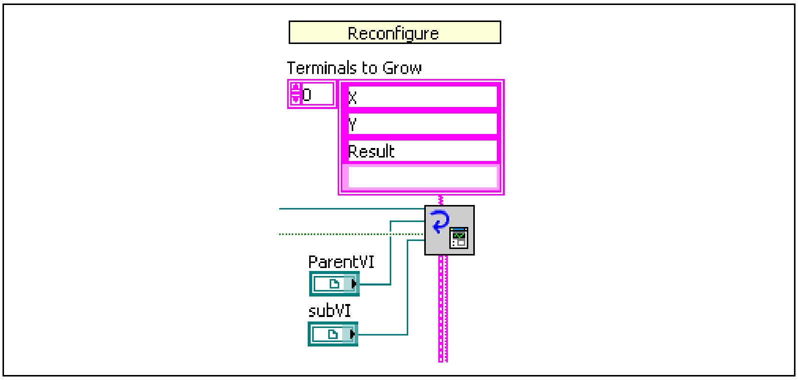
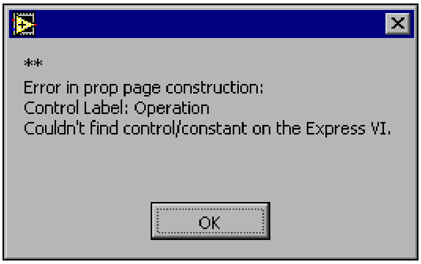


Figure 10. Removing Configurable Terminals from the **Terminals to Grow** Array

5. Save and close the VI.

Note If you run the configuration dialog box VI by itself instead of launching it from the Express VI, you receive an error message like the one shown in Figure 11. Disregard this error, which occurs because you are not running the VI as part of the Express VI.



```
**
Error in prop page construction:
Control Label: Operation
Couldn't find control/constant on the Express VI.
```

Figure 11. Configuration Dialog Box VI Error Message

Using the EasyMath Express VI

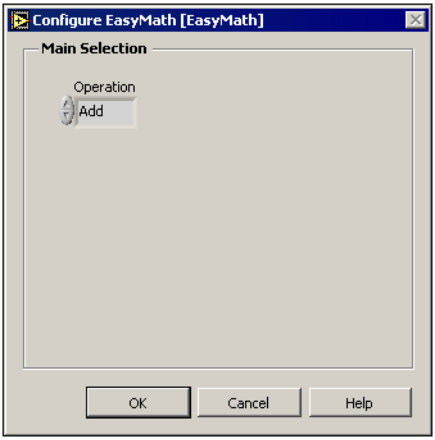
Complete the following steps to use the EasyMath Express VI for simple calculations.

1. Open a new VI.
2. Select the EasyMath Express VI from the **Express User Libraries** palette and place it on the block diagram.

Note When you add an Express VI to the `user.lib\_express` directory, it appears on the **Express User Libraries** palette. If the **paletteMenu** subpalette does not appear on the **Express User Libraries** palette, change the palette view to refresh the palettes. You might need to restart LabVIEW to view new Express VIs on a palette.

On the **Express User Libraries** palette, a **paletteMenu** subpalette appears for each LLB or directory in the `user.lib\_express` directory that does not start with an underscore. As a result, the palette for the EasyMath Express VI contains both the source VI and its subVI. You can edit the palette to remove the subVI. You will learn about editing palettes in the *Editing Express VI Palette Views* section.

The configuration dialog box opens automatically. Notice that the default operation is **Add** because it is the first operation in the enumerated control list. You will learn how to set defaults in the *Main While Loop Section* section. The window size is the one you resized, as shown in Figure 12.


Configure EasyMath [EasyMath]

Main Selection

Operation: Add

Figure 12. EasyMath Express VI Configuration Dialog Box

3. Set the **Operation** control to **Multiply**.
4. Click the **OK** button to close the configuration dialog box. Notice that the Express VI on the block diagram has grown to display the **X**, **Y**, and **Result** terminals. Because you included those terminals in the **Terminals to Grow** array, they are expanded.

Note When you close the configuration dialog box, an error message related to context help appears. Refer to the *Using the genHelp SubVI to Display Current Configuration Settings in the Context Help Window* section for more information about editing the genHelp VI.

5. Create controls and indicators from the expandable terminals on the Express VI and enter values in them.
6. Run the VI.
7. Double-click the Express VI to open the configuration dialog box. Change the **Operation** control setting to **Divide**. Click the **OK** button.
8. Run the VI again to observe the change in behavior.

Part Three: Developing a Full-Featured Express VI

Objective To learn how to develop an Express VI with configurable and expandable terminals already set, that automatically updates changes in the **Context Help** window, and that contains custom titles.

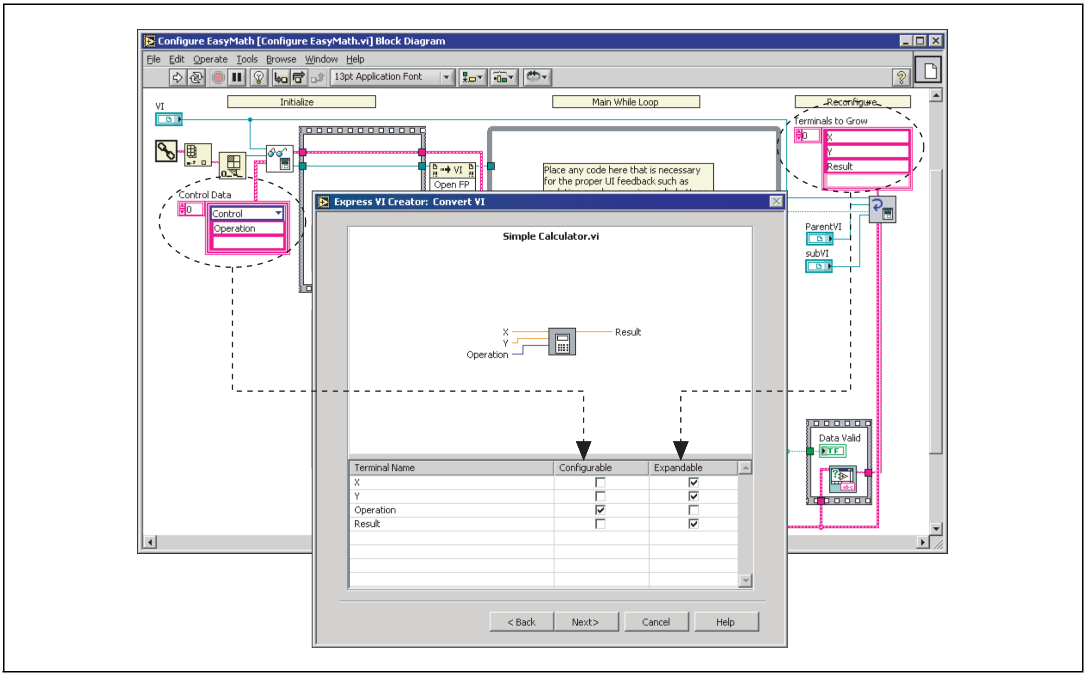
You can customize an Express VI to add features, to make it easier for users to configure, or to use advanced settings. For instance, you can edit the configuration dialog box of the EasyMath Express VI to use constants when calculating.

You can complete the exercises in this section in approximately 90 minutes.

Selecting Configurable or Expandable Terminal Options

You can set terminals as configurable and/or expandable when you use the **Express VI Creator** dialog boxes to convert a standard VI to an Express VI. This eliminates the need to delete elements in the **Control Data** and **Terminals to Grow** arrays, as you did in *Part Two: Learning the Basics of Express VI Creation*.

The **Configurable** column in the **Express VI Creator: Convert VI** dialog box corresponds to the **Control Data** array on the block diagram. The **Expandable** column in the dialog box corresponds to the **Terminals to Grow** array on the block diagram, as shown in Figure 13.



Terminal Name	Configurable	Expandable
X	☐	☑
Y	☐	☑
Operation	☑	☐
Result	☐	☑

Figure 13. Configurable and Expandable Terminal Options

Before you create an Express VI, use the following table to determine which options work best for each terminal.

Terminal Options Selected	Result	Effect on Users
Configurable and Expandable (default)	Adds an element to the **Terminals to Grow** array. If the terminal is a control, an element is added to the **Control Data** array and a control to the source VI.	Users need either to configure the control in the configuration dialog box or to wire a value to the Express VI.
Configurable	Adds an element to the **Control Data** array and a constant to the source VI.	Users need to configure the control in the configuration dialog box.
Expandable	Adds an element to the **Terminals to Grow** array and a control/indicator to the source VI.	Users need to wire a value to the Express VI.
None	Does not modify either Express VI component.	Users can keep the default value.

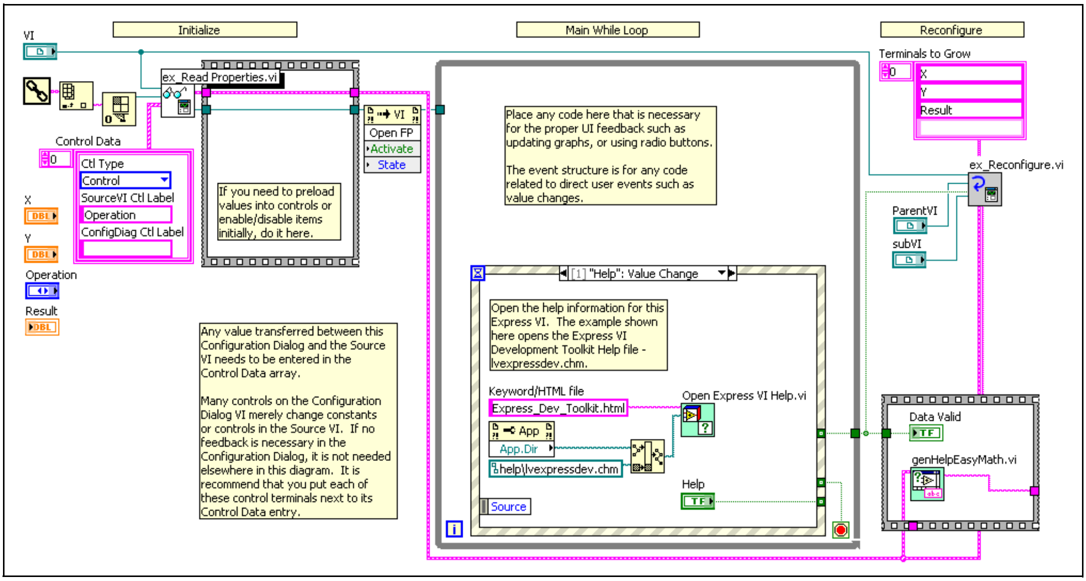
Complete the following steps to set configurable and expandable terminals for an Express VI from the **Create or Edit Express VI** dialog box.

1. Convert the Simple Calculator VI to an Express VI as described in the *Converting a Standard VI to an Express VI* section, but with the following changes:
   - From the **Express VI Creator: Convert VI** dialog box, remove the checkmarks from the checkboxes in the **Configurable** column for the **X**, **Y**, and **Result** terminals. Remove the checkmark from the checkbox in the **Expandable** column for the **Operation** control.
   - Save the Express VI as `EasyMath2`.
2. Open the EasyMath2 configuration dialog box VI.
3. Complete the front panel as described in the *Completing the Configuration Dialog Box VI* section.
4. On the block diagram, notice that the **Control Data** array and the **Terminals to Grow** array include the controls and indicators you selected. You do not need to delete or edit individual elements of the arrays.

Exploring the Configuration Dialog Box Block Diagram

Although you do not have to edit the **Control Data** and **Terminals to Grow** arrays for configurable and expandable terminals because you set those options in the **Express VI Creator: Convert VI** dialog box, you still can edit the block diagram if you want to add or change other aspects of the EasyMath2 Express VI.

The block diagram is divided into three sections: Initialize, Main While Loop, and Reconfigure, as shown in Figure 14.



Place any code here that is necessary for the proper UI feedback such as updating graphs, or using radio buttons.

The event structure is for any code related to direct user events such as value changes.

If you need to preload values into controls or enable/disable items initially, do it here.

Any value transferred between this Configuration Dialog and the Source VI needs to be entered in the Control Data array.

Many controls on the Configuration Dialog VI merely change constants or controls in the Source VI. If no feedback is necessary in the Configuration Dialog, it is not needed elsewhere in this diagram. It is recommend that you put each of these control terminals next to its Control Data entry.

Open the help information for this Express VI. The example shown here opens the Express VI Development Toolkit Help file - lvexpressdev.chm.

Figure 14. Initialize, Main While Loop, and Reconfigure Sections of the Block Diagram

Initialize Section

The Initialize section of the block diagram includes the **Control Data** array and an empty sequence structure. The **Control Data** array lists the configurable terminals for the source VI. The sequence structure allows you to set default values for configurable controls.

The control names in the array match the control names in the source VI. The array must include all controls that transfer values between the configuration dialog box and the Express VI . The Express VI reads the control values, generates context help information for them, and writes them back to the Express VI.

Each element in the **Control Data** array is a cluster that contains the following objects:

- **Ctl Type**—Specifies the type of control in the source VI, such as a front panel control, a constant, the Express VI title, or a dynamic data type control.
- **SourceVI Ctl Label**—Specifies the item label as it appears on the front panel of the source VI.
- **ConfigDiag Ctl Label**—Specifies the item label as it appears on the front panel of the configuration dialog box VI. An empty string indicates the label is the same as the source VI label.

Tip Right-click the **Control Data** array and select **Visible Items»Label** from the shortcut menu to view the labels for each array element.

Complete the following steps in the Initialize section of the block diagram to change the default operation of the EasyMath2 Express VI from **Add** to **Multiply**.

1. Create a Property Node from the **Operation** terminal and place it in the sequence structure. Refer to the *LabVIEW Help* for information about using Property Nodes.
2. Set the Property Node so that the default math operation for the VI is **Multiply**. The completed Initialize section should resemble Figure 15.

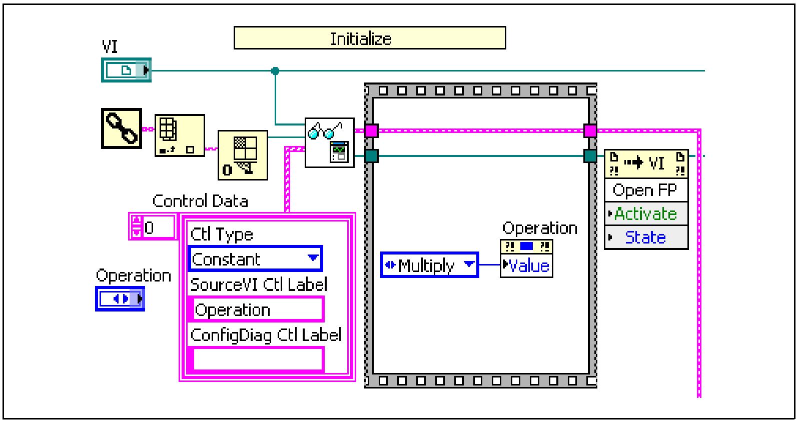


Figure 15. Initialize Section of the Block Diagram

Main While Loop Section

The Main While Loop section of the block diagram handles user interface events. This section corresponds to the area of the front panel that displays default values and values from the most recent iteration of the Express VI.

Because the configuration dialog box VI waits for user events, the main object in the While Loop is an Event structure. The Event structure waits for the user to click one of the three buttons that all dialog boxes include: **OK**, **Cancel**, and **Help**. Clicking either the **OK** or **Cancel** button ends the loop. Clicking the **OK** button causes the configuration dialog box VI to save the new configuration. Clicking the **Cancel** button closes the configuration dialog box without saving any changes. Clicking the **Help** button keeps the configuration dialog box open but also opens a referenced online help file.

The Main While Loop section of the configuration dialog box VI allows users to enter data that the VI writes in the Reconfigure section. When users select the settings they want from the Express VI configuration dialog box, the Main While Loop section processes the selections. The Main While Loop section disables controls if necessary, updates radio buttons, and updates feedback indicators, such as graphs of sample or preview data.

Complete the following steps to complete the Main While Loop section of the Configure EasyMath2 VI.

1. Add an event case to the existing Event structure to respond to the **Operation** control value changing. Refer to the *LabVIEW Help* for information about adding event cases.
2. Move the **Operation** control into the new event case. The Main While Loop section should resemble Figure 16.

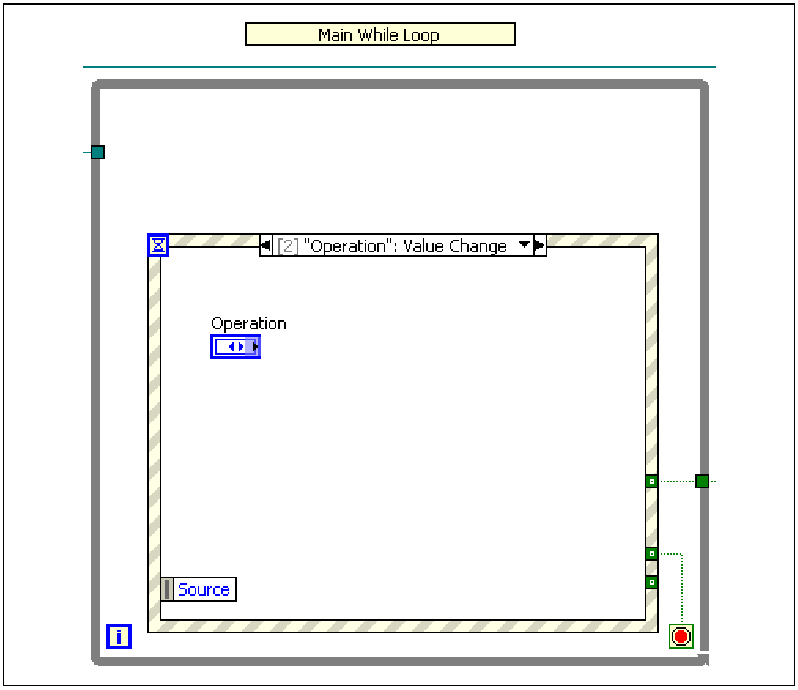


Figure 16. Main While Loop Section of the Block Diagram

3. Save the VI.

The event case you created is a placeholder that you can use with advanced features of a configuration dialog box, such as the **Preview Data** graph in the Simulate Signal Express VI.

Reconfigure Section

The Reconfigure section of the block diagram includes the **Terminals to Grow** array and the genHelp VI. The **Terminals to Grow** array lists the expandable terminals for the Express VI. The genHelp VI updates the **Context Help** window with the current configuration of the Express VI. You will edit the genHelp VI in the *Using the genHelp SubVI to Display Current Configuration Settings in the Context Help Window* section.

You also can use the Reconfigure section to rename inputs and outputs and to wire controls and indicators to the terminal pane.

Using the genHelp SubVI to Display Current Configuration Settings in the Context Help Window

The genHelp VI, shown at left, appears as a subVI in the configuration dialog box VI for all Express VIs you create using the Express VI Development Toolkit. As a subVI, it is named genHelp*ExpressVIName*. For example, the Configure EasyMath2 VI includes the genHelpEasyMath2 subVI. You can edit the genHelpVI so that the **Context Help** window displays the current configuration of the Express VI.

Complete the following steps to edit the genHelpEasyMath2 VI so the **Context Help** window updates automatically.

1. In the Reconfigure section of the Configure EasyMath2 block diagram, double-click the genHelpEasyMath2 subVI to open it.
2. On the block diagram of the genHelpEasyMath2 VI, enter the text `Operation Selected:` followed by a space in the **Context Help Name** constant.
3. Enter `Operation` in the **Control Name** constant. Delete all other text in the constant.

Note When entering text in the **Control Name** constant, the control name must exactly match the name as it appears in the **Control Data** array, including the correct case, or an error occurs when you run the Express VI.

The **Operation** control is an enumerated type control, so you must enter the enum strings that represent the four possible operation types into the genHelpEasyMath2 VI.

4. Right-click the **Enum & T/F Strings** input of the ex_PPGetProp subVI and select **Create»Constant** from the shortcut menu.
5. Expand the resulting array to show four elements.
6. Enter the enumerated strings into the array in the exact order they appear in the **Operation** control (`Add`, `Subtract`, `Multiply`, `Divide`). The genHelpEasyMath2 VI uses the enum values to retrieve the appropriate string, such as `Multiply`, and display it in the **Context Help** window.
7. Wire the **Enum Result** terminal to the **Help Description** indicator. The block diagram should resemble Figure 17.

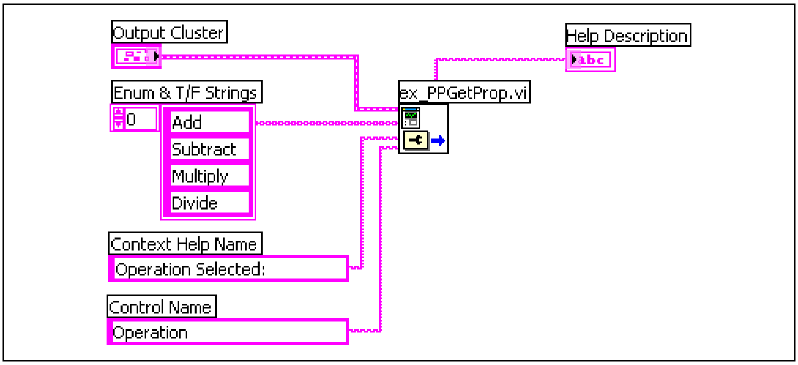


Figure 17. genHelpEasyMath2 Block Diagram Edited to Display Current Configuration Settings

8. Save and close the genHelpEasyMath2 VI.
9. Save and close the Configure EasyMath2 VI.

Updating the Context Help Window

Complete the following steps to update the **Context Help** window after you edit the genHelpEasyMath2 VI.

1. Place the EasyMath2 Express VI on a block diagram.
2. In the configuration dialog box, set the **Operation** control to **Add**.
3. Click the **OK** button to close the configuration dialog box. Notice the text and the configuration data in the **Context Help** window.
4. Double-click the Express VI to open the configuration dialog box.
5. Set the **Operation** control to **Subtract**.
6. Click the **OK** button to close the configuration dialog box. Notice the updated information in the **Context Help** window. The **Context Help** window updates each time you close the configuration dialog box.

Adding Custom Titles to Express VI Instances

You can create custom titles for instances of an Express VI by adding a string control to the Express VI.

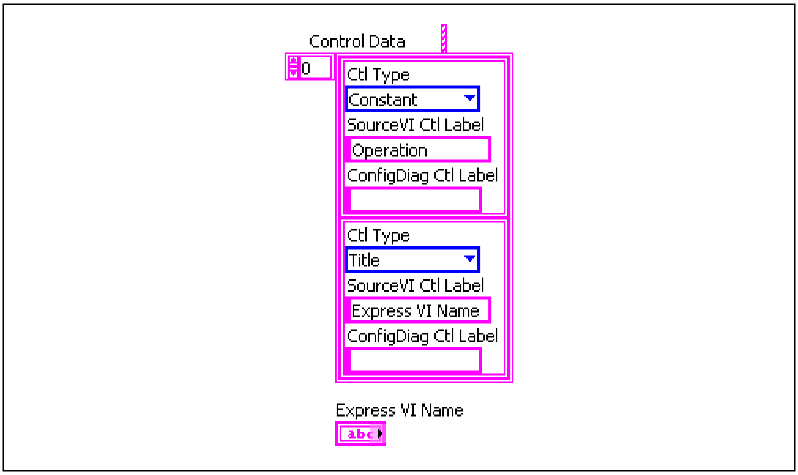
Complete the following steps to add the option for custom titles to the EasyMath2Express VI.

1. Open the Configure EasyMath2 VI.
2. Place a dialog string control located on the **Dialog Controls** palette on the front panel in the **Main Selection** frame. Label it `Express VI Name`.

Tip Dialog controls use platform-specific drawing functions, so they are useful in making dialog boxes look and behave like standard dialog boxes for each platform.

3. In the Initialize section of the block diagram, locate the first unused element in the **Control Data** array and select **Title** as the control type. Enter `Express VI Name` in **SourceVI Ctl Label**. You must enter the exact string, including the correct case, as it appears in the string control label on the front panel. To ensure accuracy, you can copy and paste from the front panel label. Adding this element adds a configurable terminal to the Express VI.

   The **Control Data** array should resemble Figure 18.



Control Data

0

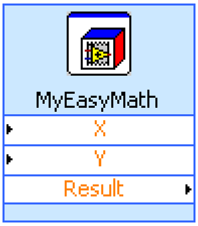
Ctl Type
Constant
SourceVI Ctl Label
Operation
ConfigDiag Ctl Label

Ctl Type
Title
SourceVI Ctl Label
Express VI Name
ConfigDiag Ctl Label

Express VI Name

Figure 18. Adding the Title Element in the **Control Data** Array

4. Save and close the Configure EasyMath2 VI.
5. Place the EasyMath2 Express VI on a block diagram.
6. In the configuration dialog box, enter `MyEasyMath` in the **Express VI Name** control.



7. Click the **OK** button to close the configuration dialog box. Notice that the new title appears on the Express VI, shown at left, and in the **Context Help** window.

Adding a Constant Option in the Configuration Dialog Box

Instead of inputting two numbers to perform a mathematical operation with the EasyMath2 Express VI, you can edit the Express VI to make one of the two numbers a constant.

Open the Configure EasyMath2 VI and complete the following steps to make the second operand for math operations a constant in the EasyMath2 Express VI.

Building the Front Panel

On the front panel, complete the following steps to add new controls to the configuration dialog box.

1. Place a dialog checkbox control located on the **Dialog Controls** palette in the **Main Selection** frame. Label the control `Use Constant`.
2. Right-click the control and select **Visible Items»Boolean Text** from the shortcut menu to hide the text next to the control.
3. Place a dialog numeric control in the **Main Selection** frame. Label the control `Constant`.

Building the Initialize Section

In the Initialize section of the block diagram, complete the following steps to map the existing **Y** control to the constant.

1. Add an element to the **Control Data** array. Select a control type of **Control**, enter `Y` as **SourceVI Ctl Label**, and enter `Constant` as **ConfigDiag Ctl Label**.
2. Create a Property Node from the **Constant** terminal and place it in the sequence structure. Expand the node to display two properties.
3. Set the first property of the Property Node as **Disabled** with a constant of `2`, and the second property as **Value** with a constant of `NaN`.

Building the Main While Loop Section

In the Main While Loop section, complete the following steps to build the **Use Constant** option.

1. Place the **Use Constant** terminal above the Event structure.
2. Place the Select function on the block diagram and wire it to the **Use Constant** terminal to disable the constant when **Use Constant** is FALSE.
3. Add an event to the Event structure for the **Use Constant: Value Change** event.

Building the Reconfigure Section

In the Reconfigure section, complete the following steps so the **Terminals to Grow** array displays only the **X** and **Result** terminals when you use a constant.

1. Enclose the **Terminals to Grow** array in a Case structure.
2. Duplicate the case.
3. Exchange the contents of the False case with the duplicate case.
4. Delete the empty duplicate case.
5. In the True case, remove the array element for the **Y** terminal.
6. Wire the **Use Constant** terminal in the Main While Loop to the Case structure.
7. Save the Configure EasyMath2 VI. The block diagram should resemble Figure 19.

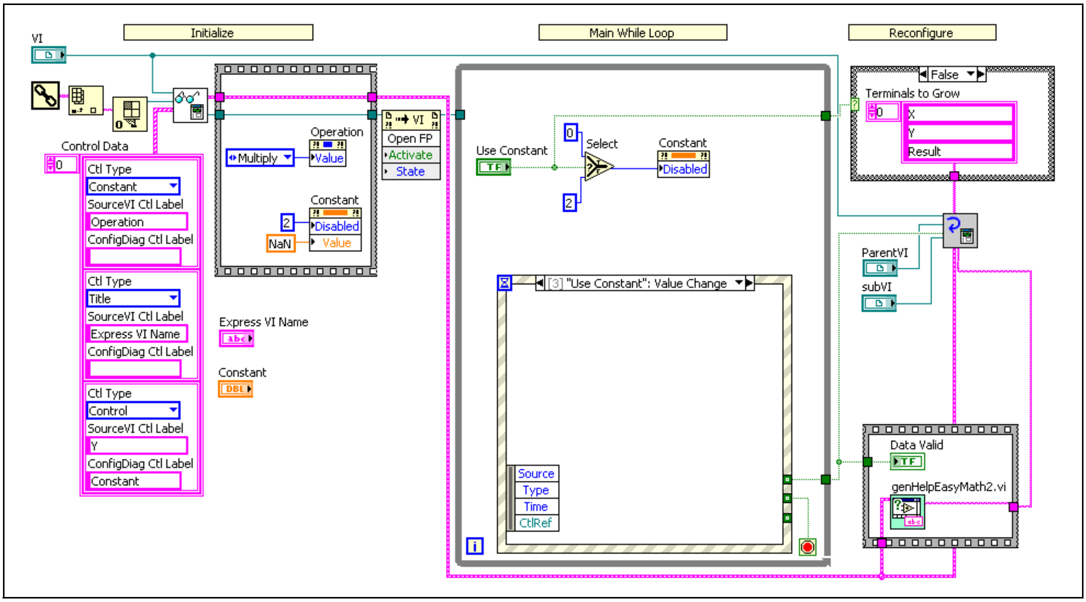


Figure 19. Configure EasyMath2 Block Diagram with Option for Using Constant

Updating the genHelp SubVI to Display the Constant

Complete the following steps to display the constant in the **Context Help** window.

1. Open the genHelpEasyMath2 VI.
2. On the block diagram, copy the ex_PPGetProp VI and place the copy to the right of the original subVI.

Note The ex_PPGetProp VI appends the control value to the help string. Use a separate copy of the ex_PPGetProp VI for each control that contains a value you want to display in the **Context Help** window. The ex_PPGetProp VI concatenates the help for each control, so you must wire the output from one copy of the VI to the input of the subsequent VI.

3. On the copy of the VI, create a string constant from the **Prefix** terminal. Enter `Constant Value:` with a space at the end in the string constant.
4. On the copy of the VI, create a string constant from the **Control:** terminal. Enter `Y` in the string constant.
5. Wire the **Enum Result** output from the first VI to the **Previous String** input of the copy of the VI.
6. On the copy of the VI, wire the **Output Cluster** input to the same terminal as the first VI.
7. Place the Search and Replace String function on the block diagram.
8. Wire the Search and Replace String function to search the **dbl String** output of the copy of the VI for the string `NaN` and replace it with the string `--Not Used--`.
9. Delete the wire from the first ex_PPGetProp VI to the **Help Description** indicator.
10. Wire the **result string** output of the Search and Replace function to the **Help Description** indicator.

    The completed block diagram should resemble Figure 20.

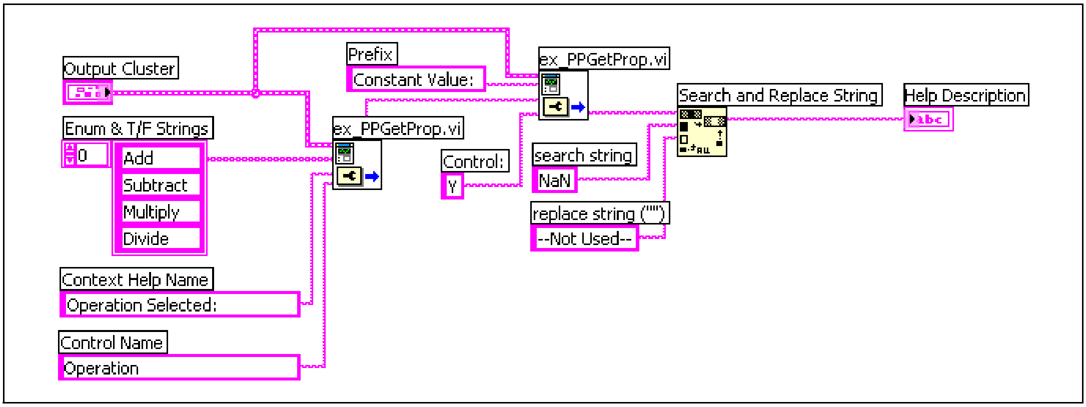


Figure 20. Updating the genHelpEasyMath2 VI to Display Multiple Configuration Values

11. Save and close all subVIs and VIs.

Testing the EasyMath2 Express VI

Complete the following steps to test the EasyMath2 Express VI to make sure the constant option works correctly and appears in the **Context Help** window.

1. Place the EasyMath2 Express VI on a block diagram.
2. In the configuration dialog box, set **Operation** to **Subtract** and enter a constant of `-12`. Click the **OK** button.
3. Notice that the Express VI on the block diagram displays only the **X** and **Result** terminals. The **Context Help** window shows the current configuration.
4. Create controls and indicators from the expandable terminals on the Express VI.
5. Run the VI.
6. Place another EasyMath2 Express VI on the block diagram.
7. In the configuration dialog box, set **Operation** to **Divide** and do not enter a constant. Click the **OK** button.
8. Notice that this copy of the Express VI displays the **Y** terminal. Also, notice that the configuration in the **Context Help** window is current.

Note The **Terminals to Grow** array determines how Express VI terminals expand. However, edits to the array cannot cause expanded terminals to disappear or change order. As a result, terminals that users expand and wire override values that users select in the configuration dialog box.

Editing Express VI Palette Views

When you create an Express VI, the source VI's subVI appears on the palette by default, although you are unlikely ever to use the subVI. Edit the palettes to display only the Express VIs and to remove additional layers of subpalettes so the Express VIs are easier to access. You can edit custom palette views with Express VIs in the same way you edit palettes with other objects.

Complete the following steps to create a subpalette that includes only the EasyMath2 Express VI.

1. Select **Tools»Advanced»Edit Palette Views** to display the **Edit Controls and Functions Palettes** dialog box. The **Controls** and **Functions** palettes appear so you can edit the palettes directly.
2. Navigate to the **Express User Libraries** palette. Right-click the **paletteMenu** icon for the palette that contains the EasyMath2 Express VI and select **Rename Submenu** for the shortcut menu. The **Edit Menu Item Name** dialog box appears.

3. Enter `My Express VIs` and click the **OK** button.
4. On the **My Express VIs** palette, right-click the subEasyMath2 VI icon, shown at left, and select **Delete Item** from the shortcut menu.
5. Right-click the space that the deleted VI occupied and select **Delete Empty Slot** from the shortcut menu. The EasyMath2 Express VI moves to the left of the palette.

Complete the following steps to move the EasyMath Express VI to the **My Express VIs** palette.

1. On the **My Express VIs** palette, right-click the space next to the EasyMath2 Express VI and select **Insert»VI(s)**.
2. Navigate to the `user.lib/_express` directory. Open the `EasyMathSource.llb` file.
3. From the file dialog box, double-click the EasyMath Source VI. The EasyMath Express VI appears on the palette next to the EasyMath2 Express VI.

Note When you place an Express VI on a palette, the blue box around the icon might appear to be missing. However, after you save changes to the palette, the Express VI looks correct.

4. From the **Edit Controls and Functions Palettes** dialog box, click the **Save Changes** button.

Refer to the *LabVIEW Help* for more information about creating and editing custom palette views.

Part Four: Considerations for Designing and Distributing Express VIs

Consider the following design guidelines and techniques when you create and edit Express VIs.

Express Source VI

When building Express VIs, use only controls and indicators that exist on the **Controls** palettes. If a user right-clicks an Express VI terminal and selects **Create»Control/Indicator** from the shortcut menu, LabVIEW creates a copy of the control/indicator on the front panel of the Express source VI. Use a 1D array of waveforms for multivalue inputs. For multivalue outputs, use a waveform graph. Otherwise, use a numeric control for scalar values, an LED control for Boolean values, and a string control for strings. Do not modify the controls and indicators.

If you are creating an Express VI that contains several memory-intensive subVIs that do not enable simultaneously, call them by reference. However, if you include the Express VI in an executable application created with LabVIEW Application Builder, you must manually add the subVIs called by reference to the build.

When creating Express VIs, you can include objects that are not part of the core LabVIEW development environment, such as LabVIEW module or toolkit VIs, drivers, or other software. However, Express VIs that include these objects can run only on computers on which those objects are present.

Configuration Dialog Box VI

The configuration dialog box VI contains the Express VI user interface. Design user interfaces that are consistent with standard dialog boxes but also efficient and easy for users to understand. The Express VI Development Toolkit defines only the basic **OK**, **Cancel**, and **Help** buttons and the template for the block diagram.

When designing a configuration dialog box, keep the following guidelines in mind:

- Use controls and indicators from the **Dialog Controls** palette as much as possible or copy and paste items from other configuration dialog boxes.
- Use system colors and fonts.
- Minimize the number of hidden selections, such as hidden controls, tab controls, or multiple pages. If you cannot create a configuration dialog box without hidden objects, use a second page or a tab control.

- Keep the user interface style consistent. For example, the **OK**, **Cancel**, and **Help** buttons should appear in the lower right corner of a configuration dialog box.

Refer to the *LabVIEW Development Guidelines* manual for more information about maintaining a consistent style when creating dialog boxes.

Using the Dynamic Data Type

Many of the Express VIs included with LabVIEW contain dynamic data type controls. The dynamic data type accepts data from and sends data to the following data types, where the scalar data type is a floating-point number or a Boolean value:

- 1D array of waveforms
- 1D array of scalars
- 1D array of scalars—most recent value
- 1D array of scalars—single channel
- 2D array of scalars—columns are channels
- 2D array of scalars—rows are channels
- Single scalar
- Single waveform

The dynamic data type control is located in the `vi.lib\Waveform\WDTCtls.llb` with the filename `dynamicData.ctl`. Refer to the National Instruments Web site at `ni.com` for more information about using dynamic data type controls in Express VIs.

Using Express VI Support VIs

Express VIs you build using the Express VI Development Toolkit automatically contain a number of subVIs that you can use to add functionality. Refer to the *LabVIEW Express VI Development Toolkit Help* for more information about specific support VIs.

Exploring Existing, Sample, and New Express VIs

Express VIs and their components are located in the following directories:

- `vi.lib\express` contains the Express VIs included with LabVIEW. The directory contains subdirectories of Express VIs that correspond to the subpalettes on the **Functions** palette.
- `vi.lib\express\express shared` contains the support VIs called by Express VIs and configuration dialog box VIs. Use the VIs in this directory to perform operations such as handling radio buttons and generating context help.
- `user.lib\_express\ExpressVIDevSamples` contains the example Express VIs belonging to the Express VI Development Toolkit. Use these examples as a starting place for the Express VIs you build.
- `user.lib\toolname\_express` contains Express VIs you create with the Express VI Development Toolkit and save in the default location. **(Mac OS and UNIX)** The default location for Express VIs you create is the LabVIEW default data directory because not all users can write to the `user.lib` directory.

Note The underscore before `express` prevents LabVIEW from automatically adding VIs to the **Express User Libraries** palette. If you want to add Express VIs to a palette, you must create a new palette or edit an existing one.

Deploying Express VIs to Other Computers

If you want to move Express VIs to other computers, move the two VI libraries for the Express VI and any related `.mnu` files. When you create Express VIs that you will deploy on other computers, you must use the same directory structure that the target computers use. The components of an Express VI link to each other through both relative and specific paths. If the directory structure is different on each computer to which you deploy, the links break, and the Express VI prompts users to find the configuration dialog box VI.

Part Five: Where to Go from Here

In addition to this user guide, the Express VI Development Toolkit provides sample Express VIs that you can explore to learn more about creating and editing Express VIs.

You can open components of the sample Express VIs from the **Create or Edit Express VI** dialog box. The Express VIs are located on the **Express VI Development Samples** subpalette of the **Express User Libraries** palette.

The *LabVIEW Express VI Development Toolkit Help* includes procedures for many aspects of Express VI development. Access the *LabVIEW Express VI Development Toolkit Help* by selecting **Help»VI, Function, and How-To Help** and clicking the **Toolsets** book in the table of contents.

Refer to the NI Developer Zone at `ni.com/zone` for more information about advanced techniques for Express VI development.

323413A-01 Aug03