

# BIOBENCH™

## Version 1.2

These release notes contain installation instructions, list system requirements, and provide updated information to help you begin using BioBench 1.2. This document also includes a list of the new features of this version and an overview of plug-in analysis.

## Minimum System Requirements

---

Your computer must have the following software and hardware to work with BioBench:

- Microsoft Windows 2000/NT/Me/9x.
- Personal computer using at least a Pentium class chip with floating-point capability. National Instruments (NI) recommends a 200 MHz Pentium II or higher microprocessor with floating-point capability.
- CD drive.
- VGA or higher resolution video adapter.
- Minimum of 64 MB of RAM.
- 25 MB of free hard disk space.
- Microsoft-compatible mouse.
- NI-DAQ 6.9.1 or later.
- LabVIEW RunTime Engine 6.0.
- LabVIEW 6.0 or later, for plug-in analysis routines.

# Using BioBench with National Instruments Data Acquisition Hardware

---

If you are using BioBench with NI data acquisition (DAQ) hardware, you must use NI-DAQ version 6.9.1 or later. NI-DAQ is the driver software that ships with your data acquisition hardware. Refer to the *NI-DAQ User Manual for PC Compatibles* for more information about NI-DAQ.

## Installing BioBench

---

### Installing from a CD



**Note** NI recommends that you do *not* run other applications while you install BioBench.

Complete the following steps to install BioBench.

1. Insert the BioBench CD into your CD drive.
  - a. Wait several seconds for the autorun utility on the CD to launch the BioBench installation program.
  - b. If the BioBench installation program does not start, select **Start»Run**. Type `x:\setup\setup` in the **Run** dialog box, where *x* is the letter of your CD drive. Click the **OK** button.
2. Follow the instructions that appear on the screen.
3. If the BioBench installer does not find Adobe Acrobat Reader installed on your computer, the installer prompts you to install it. You must have the Adobe Acrobat Reader to view the online version of the BioBench documentation.

If you choose not to install the Adobe Acrobat Reader during the BioBench installation, you can install it later by running `acrobat\rs405eng.exe`. The installer places the file in your BioBench folder during installation.

### Installing on a Network

Contact NI for licensing information if you plan to run BioBench on a network.

### What the Setup Program Installs

The setup program for BioBench installs the BioBench software and supporting data files.



**Note** Do *not* rename the subdirectories `data`, `examples`, `lib`, `plugin`, `notch`, `software`, or `stimulus`. BioBench must retain this folder hierarchy to work correctly.

## Printing in BioBench

---

BioBench front panels print as bitmap images by default. If your printer supports postscript or color printing, you can modify lines in your `biobench.ini` file to use postscript or color printing when printing the BioBench screens.

The `biobench.ini` file is located in your BioBench directory. Be sure to save a backup copy of your `biobench.ini` file before you try to modify it.

You can edit your `biobench.ini` file with any word processing or text editing software, such as Notepad. Restart BioBench for the new settings to take effect after you modify any information in your `biobench.ini` file.

### Enabling Postscript Printing

Modify the `postScriptPrinting` line in your `biobench.ini` file exactly as shown in the following line to enable postscript printing.

```
postScriptPrinting=True
```

If you enable postscript printing and your printer does not support it, your printout consists of a single garbled line of postscript text.

### Enabling Color Printing

Color and grayscale printing are disabled by default. Modify the `colorPrinting` line in your `biobench.ini` file exactly as shown in the following line to enable color and grayscale printing.

```
colorPrinting=True
```

You can conserve toner by leaving grayscale printing disabled.



**Caution** Do *not* change any other lines in the `biobench.ini` file. Changing the `biobench.ini` file might cause BioBench to operate incorrectly.

# What's New in BioBench 1.2

---

The following features are new in BioBench 1.2:

- Using plug-in custom analysis routines—BioBench 1.2 offers a data interface through which you can process data provided by BioBench in a variety of ways and observe the results in the BioBench Analysis view. You need LabVIEW 6.0 or later to build VIs for plug-in analysis routines.
- Support for more National Instruments DAQ devices—BioBench 1.2 uses NI Measurement & Automation Explorer (MAX) to get the hardware configuration information of National Instruments DAQ devices. If you have MAX installed on your computer, and MAX supports the NI DAQ devices installed on your computer, BioBench 1.2 can identify the DAQ devices and work with them. To view a list of the DAQ devices installed on your computer, select **Configure»Data Source** to access the **Select Data Source** dialog box. Click the **National Instruments Data Acquisition Hardware** radio button, and you can see the names of NI DAQ devices that are listed in the **NI Data Acquisition Hardware** selection box. If the DAQ hardware you expected to see is not listed, make sure that it is properly installed and configured using MAX.
- Standard LabVIEW menus—Standard LabVIEW menus are used instead of the menu rings used in BioBench 1.0.
- Exporting Analysis Data—The peak detection and histogram tables can be saved as a text file.
- Saving as a New File—Select **File»Save as New File** to save all the data displayed in the Analysis view, including the records of the analysis steps taken, to a new file.
- Event List—Click the **Event** button, located at the top of the Analysis view to see a list of recorded events. The **Pause** event is also recorded during data acquisition.
- Scrolling Event Marks—Click the arrow button on the right side of the Event Profile to move the cursor to the next event or the last event. Using event marks, you can click through all the important points in the data without scrolling through the whole file.
- Real-Time Analysis—BioBench 1.2 has added real-time analysis tools in the Data Acquisition view. The real-time analysis tools include digital filters and math functions to add, subtract, multiply, and divide channels or channels with constants. Select **Configuration»Settings»Signal Process** to select filters and transform methods. You also can click the **Filter** radio and **Transform** radio buttons above the graph you want to process to set the filters and transforms parameters.

- Setting Decimation Proportion—You can set the decimation proportion when data is saved to spreadsheet.
- Inserting hot PCMCIA hardware—A PCMCIA DAQ card that is inserted while the laptop is powered on can be recognized while Biobench is running. Refresh MAX first after the PCMCIA card is inserted hot. Select **Configuration»Select Data Source** to select the hardware you want to use. If MAX is not refreshed, Biobench cannot recognize your hardware, and an error occurs.
- Retaining settings after channel configuration—When the channels are configured and calibrated, some settings are retained, including the channel scan list, the channels displayed in the acquisition screen, the max and min settings for all the channels, the settings for the alarms, and the settings for the stimulus.
- Exporting data to Excel directly—Select **File»Export to Spreadsheet»Excel format** to export selected data to Excel from any Analysis view directly.
- Displaying both absolute time and relative time—In BioBench 1.2, the time axis at the bottom of the Analysis view and Acquisition view indicates both the relative time and absolute time from the beginning of data acquisition.

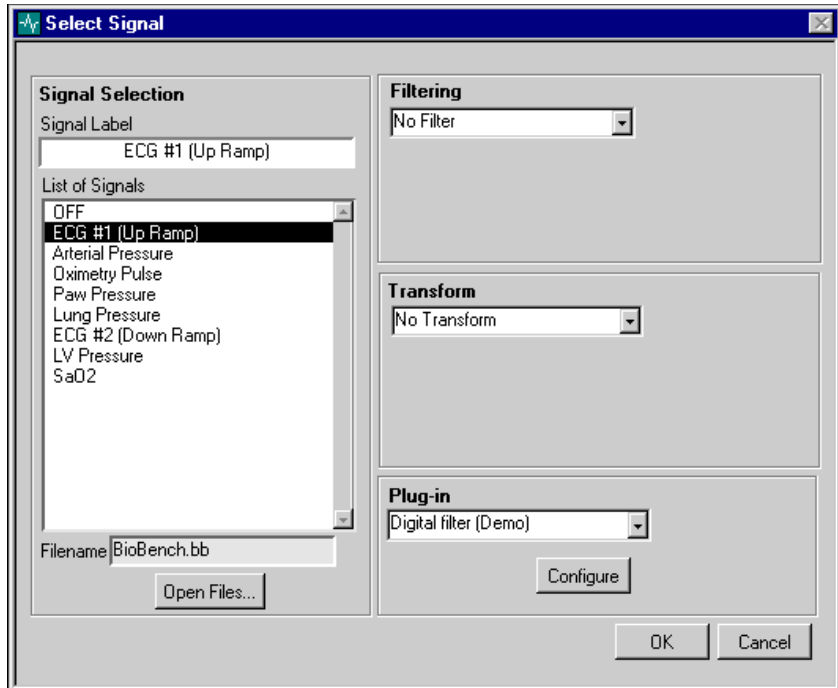
## Plug-In Analysis

---

To increase flexibility for processing data, BioBench 1.2 offers a plug-in architecture to allow implementation of custom LabVIEW-built analysis functions. Through this data interface, you can process data provided by BioBench and observe the results in the BioBench Analysis view. Based on the dynamically loading mechanism of LabVIEW, the plug-in architecture enables you to associate your own subVIs with BioBench. By calling those subVIs dynamically, BioBench can process data based on the requirements of the customer and avoid rewriting or recompiling its source code. Refer to the LabVIEW demo of the plug-in located in `\labview\Examples\Viserver\plugins.llb` for more information about plug-in analysis.

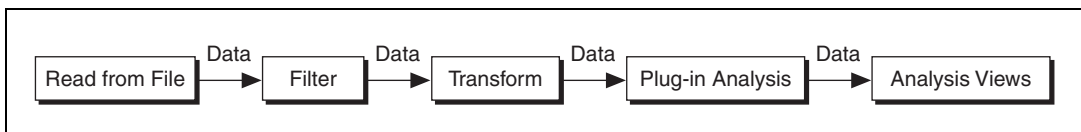
## Applying Plug-In Analysis

To choose a plug-in analysis method for a graph, click the corresponding label located in the upper left corner of each graph. The **Select Signal** dialog box appears as shown in the following illustration.



Select the type of plug-in option you want to use from the **Plug-in** list. If the selected plug-in VI is configurable, a **Configure** button appears. Click the **Configure** button to launch a configuration VI to configure the parameters of the plug-in VI. If you cannot configure the plug-in VI, the **Configure** button does not appear.

Data is processed according to the following flow chart.



**Note** You can perform one filter, one transform and/or plug-in analysis operation on any signal. BioBench processes all the data with filter and transform methods before carrying out plug-in analysis.

## Components of Plug-in Architecture

The plug-in architecture includes two subVIs, a plug-in VI, and a configuration VI. The plug-in VI is used for data analysis, while the configuration VI sets parameters for data analysis. The global variable `plugins_parameter.glo` in the `\BioBench\plugin\pluginutils.llb` library transfers data between two VIs. There are two controls, **Display Name** and **Config VI Name**, in the plug-in VI. **Display Name** denotes the display name of this plug-in VI, which is enumerated by BioBench in the plug-in pull-down list. **Config VI Name** denotes the name of the configuration VI. BioBench calls the configuration VI according to **Config VI Name**, which is an empty string if configuration is not needed. The plug-in VI and the configuration VI must be saved in the same VI library (`.llb`) in the `\BioBench\plugin` folder. BioBench identifies plug-in architecture automatically and calls a plug-in VI and/or a configuration VI according to the needs of data processing.

## Creating a Plug-In Custom Analysis VI

The following example shows you how to create a Plug-in Custom Analysis VI. First, the example generates a Plug-in VI and multiplies the inputs by 2. Next, the example generates a Configuration VI, which sets the multiplier of the Plug-in VI. Refer to `\BioBench\plugin\demo\Multiply by Const.llb` for an example of a plug-in custom analysis VI.

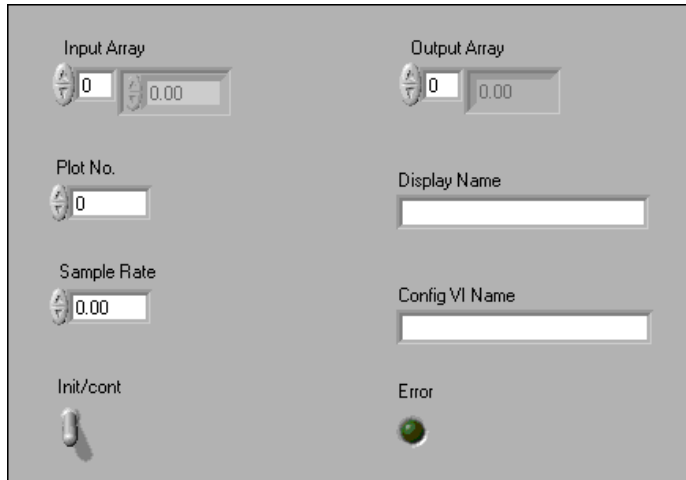


**Note** You need a general familiarity with the LabVIEW development environment to follow the next example.

Complete the following steps to create plug-in custom analysis routine.

## Building the Front Panel

1. Open the Plug-in Template VI from the \BioBench\plugin\plug-in template\ folder. Review the controls and indicators on the front panel shown in the following illustration.



2. All of the following controls and indicators are required:
  - **Display Name**—Display name of this plug-in VI. In plug-in analysis, BioBench enumerates the display name in plug-in selection pull-down list for choice. The label of the **Display Name** control must be `Display Name` so that BioBench can identify it correctly. However, the content of **Display Name** can be any desired string. For example, you can use the name of the plug-in VI.
  - **Config VI Name**—If you want the parameters of Plug-in VI to be changeable and configurable, you need a configuration VI. Enter the configuration VI you wish to load in the string indicator labeled `Config VI Name`. The label of the `Config VI Name` indicator must be `Config VI Name` so that BioBench can identify it correctly. The content of this string indicator must be the same as the name of configuration VI.
  - **Input Array**—Data to be processed and output from BioBench. Data type must be single precision (SGL).
  - **Plot No.**—Plot number selected when you click on the Signal label type in BioBench. Data type must be long (I32).

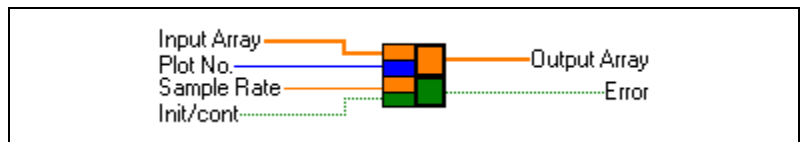


- **Sample Rate**—Sample rate received from information on data file. Data type must be single precision (SGL).
- **Init/cont**—Controls the initialization of the internal states. To process a large data sequence which has been split into smaller blocks, set this control to FALSE for the first block, and set this control to TRUE for continuous processing of all remaining blocks.
- **Output Array**—Data processed by plug-in custom analysis, output into BioBench. Data type must be single precision (SGL).
- **Error**—Indicates whether there are errors in the course of data processing. Output into BioBench.



**Note** Data type of all the controls and indicators must be consistent with what is specified. Otherwise, BioBench cannot recognize it and errors occur.

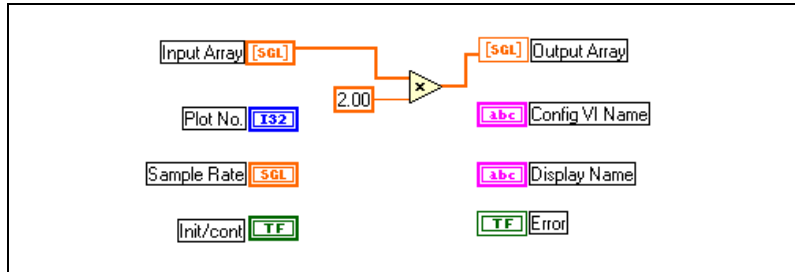
3. Enter *Multiply By Const (Configurable)* in the **Display Name** control, which represents the Plug-in Analysis name. Leave **Config VI Name** control blank, which means that configuring the parameters of the plug-in VI is not needed.
4. Right-click the **Display Name** control and select **Data Operations» Make Current Value Default** from the shortcut menu. Repeat this procedure with **Config VI Name** control.
5. Open the connector pane. Notice that the pattern has four inputs and two outputs.
6. The **Input Array**, **Plot No.**, **Sample Rate**, and **Init/cont** controls are connected to the four connectors on the left side of the icon. The **Output Array** and **Error** indicators are connected to the two output connectors on the right side of the icon, as shown in the following illustration.



## Building the Block Diagram

The following example multiplies the input array by 2. However, you can analyze data with any method you wish. Refer to `BioBench\plugin` for more plug-in examples.

7. Select **Window»Show Diagram** to view the block diagram. Build the block diagram according to the following illustration.



Multiply function (**Functions»Numeric** palette). This function calculates the product of two inputs.



Numeric constant (**Functions»Numeric** palette).

8. Save the VI as `Multiply by Const.vi` in the `\BioBench\plugin\Multiply by Const.llb`.

## Completing the Plug-in Analysis VI

Now you have created a simple plug-in analysis.

9. Launch BioBench 1.2 and set the Analysis view to the current view.
10. Select **File»Open** and select `BioBench.bb` from the **Open** dialog box.
11. Click the **ECG** signal label to open the **Select Signal** dialog box. Then click the **Plug-in** pull-down menu to see that the display name of the plug-in analysis `Multiply by Const (Configurable)` has been enumerated.
12. Select the plug-in analysis and click the **OK** button to return to the Analysis view. The value of ECG has been multiplied by 2.



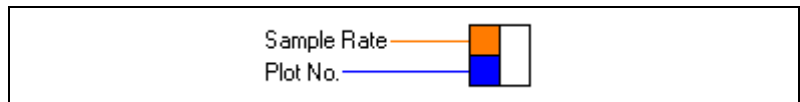
**Note** You must save the plug-in VI in a VI library in the `plugin` folder. BioBench looks for the plug-in VI in that `plugin` folder.

## Creating a Configuration VI

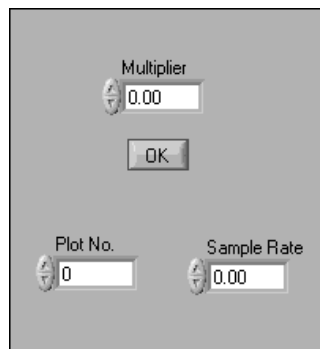
Complete the following steps to build a configuration VI that configures the parameters of plug-in VI. Refer to `\BioBench\plugin\demo\Multiply by Const (Configurable).llb` for an example of a configuration VI.

### Building the Front Panel

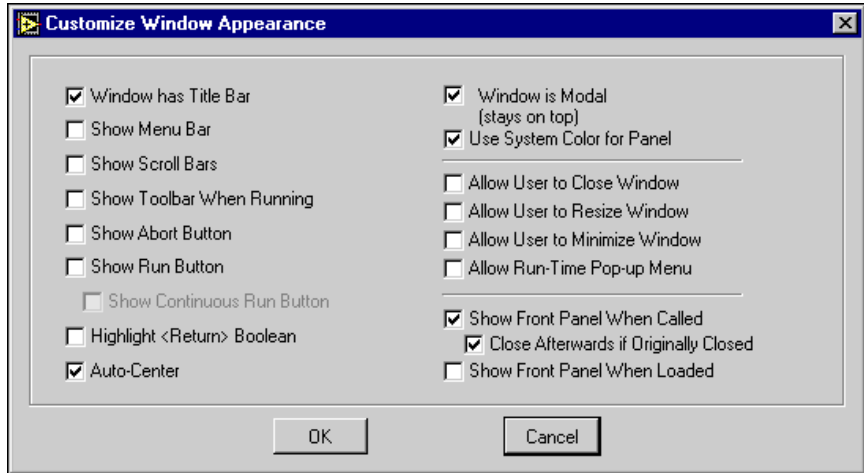
1. Create a new VI. On the front panel, add a numeric control called `Sample Rate` and a numeric control named `Plot No.`. The data types of the two controls are the same as those mentioned in the plug-in VI. These two controls are essential in the configuration VI.
2. Add a digital control and an **OK** button to the front panel. Label the digital control `Multiplier`, which are the parameters that should be configured in the configuration VI. Refer to [Creating a Plug-In Custom Analysis VI](#) section in this document for more information about the data types of these controls.
3. Build the connector pane. Choose the pattern with two inputs. Connect the controls as shown in the following illustration.



4. Your front panel should look similar to the following illustration.

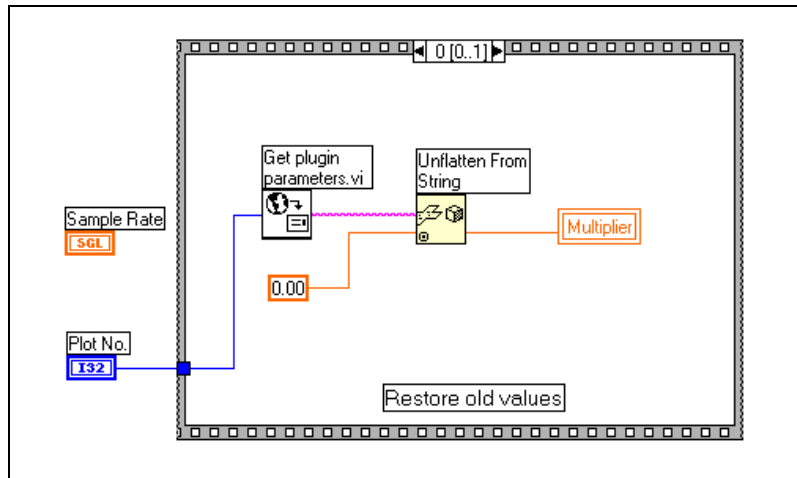


5. To customize the window appearance, select **File»VI Properties** to display the **VI Properties** dialog box. Select **Window Appearance** from the **Category** pull-down menu. Click the **Customize** button to view the **Customize Window Appearance** dialog box. Place a checkmark next to the options shown in the following illustration. Click the **OK** button to enable these customizations.



## Building the Block Diagram

6. Select **Window»Show Diagram** to view the block diagram. Build Frame 0 as shown in the following illustration.



Get Plugin Parameters VI, available in `Biobench\plugin\pluginutils.llb` (**Functions»Select a VI** palette). This VI gets the previous value of parameters from `plug-in information.glo` to initialize the multiplier control.



Unflatten From String (**Functions»Advanced»Data Manipulation** palette). This function converts anything to a string of binary values.

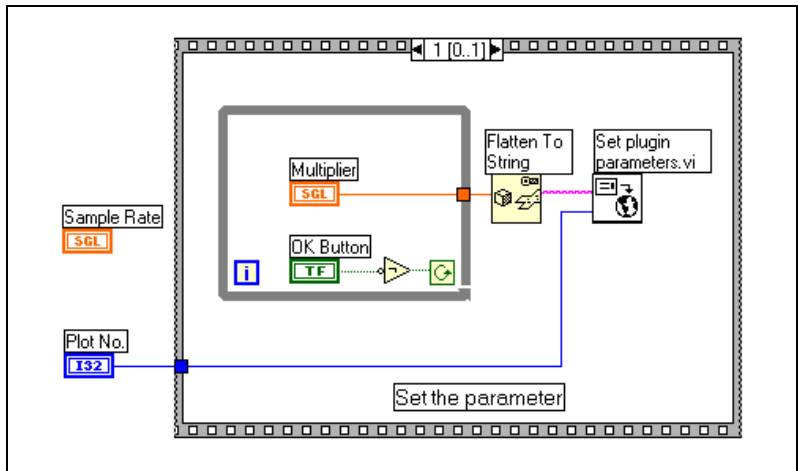
0.00

Numeric Constant (**Functions»Numeric** palette). To change the data type of the constant, right-click on the numeric constant and select **Representation»Single Precision (SGL)** from the shortcut menu. The data type of the number constant must be the same as the **Multiplier** control.

Multiplier

Local variable of **Multiplier** control. To create the local variable, right-click on the block diagram and select **Functions»Structures»Local Variable**. Place the local variable on the block diagram. Right-click the local variable and select **Select Item»Multiplier**.

7. Build Frame 1 as shown in the following illustration.



Set Plugin Parameters VI, available in `BioBench\plugin\pluginutils.llb` (**Functions»Select a VI** palette). This VI sets the value of the multiplier control into `plug-in information.glo` to configure plug-in parameters.



Flatten to String (**Functions»Advanced»Data Manipulation** palette). This function converts binary strings to any data type. It is the reverse operation of Unflatten from String.



Not Function (**Functions»Boolean** palette). This function inverts the value of the **OK** button.

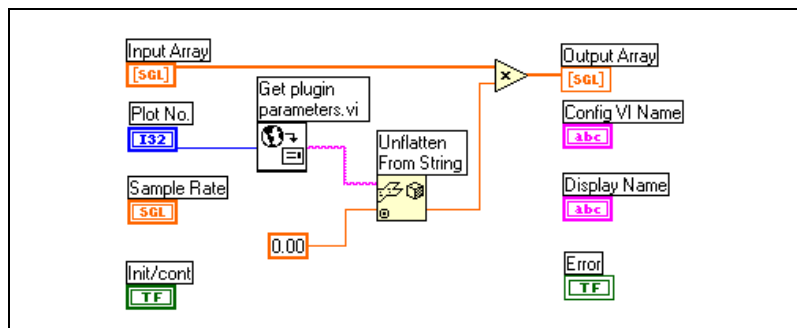
The global variable `plug-in information.glo` located in `BioBench\plugin\pluginutils.llb` is an array of binary strings that saves the configured parameters by **Plot No.**, aiding data transfer between plug-in VI and configuration VI.



**Note** NI recommends that you load the Set Plugin Parameters VI to set parameters of plug-in VI and load the Get Plugin Parameters VI to get parameters of plug-in VI. The two VIs are placed in the BioBench\plugin\pluginutils.llb library. Refer to BioBench\plugin for examples of these VIs.

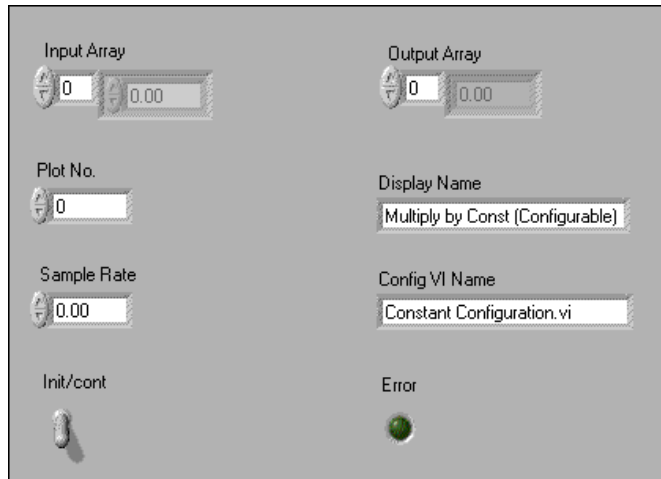
## Completing the Configuration VI

8. Resize the front panel area to hide the **Plot No.** and **Sample Rate** controls and only display the **Multiplier** control and the **OK** button to simulate a dialog box.
9. Save the VI as `Constant configuration.vi` in the `Biobench\plugin\Multiply by Const (Configurable).llb`. You must save the configuration VI in the same VI library in which the plug-in VI is saved. The name of the configuration VI must be consistent with the content of control labeled `Config VI Name` in the plug-in VI. Refer to the [Creating a Plug-In Custom Analysis VI](#) section of this document for more information about `Config VI Name`.
10. Open the plug-in VI you created earlier in LabVIEW and modify the block diagram as shown in the following illustration.



11. On the front panel, type `Constant Configuration.vi` in the **Config VI Name** control.

- Right-click the **Config VI Name** control and select **Data Operation» Make Current Value Default** from the shortcut menu. Your front panel should look similar to the following illustration.



- Now you can configure the parameters for the plug-in VI. The parameters can be any type, such as a string, number, array, cluster and so on. However, you must first convert the parameters to a string of binary values using the Flatten to String function and save the changes to `plug-in information.glo`. The plug-in VI also must unflatten the parameters from `plug-in information.glo` to read the effective values. Refer to `BioBench\plugin` for more examples of the configuration VI.
- Save the VI in `BioBench\plugin\*.11b`. You must save the configuration VI in the same VI library in which the plug-in VI is saved. The name of the configuration VI must be consistent with the name of the Config VI Name indicator in the plug-in VI. Refer to step 1 of the [Creating a Plug-In Custom Analysis VI](#) section of this document for more information about Config VI Name.
- Run BioBench 1.2 and enter the Analysis view. Select the **Multiply by Const (configurable)** plug-in VI in the **Select Signal** dialog box. The **Configure** button appears. Click the **Configure** button to load the configuration VI you built.
- Type `10.00` in the **Multiplier** control. Click the **OK** button to close the Configuration VI. Click the **OK** button in the **Select Signal** dialog box to return to the Analysis view. The value of ECG has been multiplied by 10.00.