

Bergmans Mechatronics LLC

LabSocket

User Guide

LabSocket-Basic System



June 2014

Copyright © 2014 Bergmans Mechatronics LLC

BML Document BML-2014-102.4

Table of Contents

1. Introduction	1
1.1 LabSocket System	1
1.2 About this Guide	1
1.3 Evaluation vs Commercial Packages.....	2
1.4 Standalone Executables	2
1.5 Contact	2
2. System Requirements	3
2.1 Browsers	3
2.2 LabVIEW Host Platform.....	3
2.3 LabSocket Server Virtual Machine Host.....	3
2.4 LabSocket Server Ports	3
3. System Overview	4
4. Setup Instructions	6
4.1 LabSocket Server Virtual Machine Installation	6
4.2 LabSocket-Basic Software Installation	8
4.3 Uninstalling LabSocket Software (Optional).....	9
5. LabSocket Application Development	10
5.1 Simple Demonstration Application	10
5.2 Working with Images.....	16
6. Example Files	18
7. LabSocket Start VI	20
8. User Administration	23

1. Introduction

1.1 LabSocket System

The LabSocket system enables remote access to LabVIEW applications from desktop or mobile web browsers without the need for browser plugins or a client-side run time engine. The system automatically creates a web page representation of a LabVIEW application ("Target VI") front panel and dynamically synchronizes the contents of the web page and front panel.

The system is available in three different versions, each of which offers distinct capabilities:

Version	Description
LabSocket-Basic	One or more browsers connect to a single Target VI
LabSocket-MultiClient	Each browser connects to a unique instance of a copy of a Target VI
LabSocket-RT	One or more browsers connect to a single LabVIEW Real-Time Target VI

A key technology used in the system is the HTML5 WebSocket interface that enables continuous, bi-directional communication with a web browser. The use of this interface contributes to the name of the system: "LabSocket = LabVIEW + WebSocket".

An important component of the system is the "LabSocket Server" software that acts as a bridge between LabVIEW and the browser. This software consists primarily of an HTTP server and message broker. The two main implementations, of this software are:

1. a customer-operated Virtual Machine (most popular for production); and,
2. an existing cloud server (for getting started quickly with the system).

A notable benefit of the Virtual Machine implementation is that it enables the entire LabSocket system to be easily set up within a customer LAN.

1.2 About this Guide

This User Guide describes the setup and operation of the LabSocket-Basic package. The guide assumes that the user is familiar with LabVIEW but has only a basic level of Linux and server virtualization experience.

Estimated system installation time is: 20 minutes if using the existing cloud server implementation of the LabSocket Server; or, 45 minutes for the LabSocket Server Virtual Machine implementation.

1.3 Evaluation Package

A fully functional evaluation LabSocket-Basic package is available for download via the LabVIEW VI Package Manager or the LabVIEW Tools Network (See Section 4.2 of this Guide). The package may be used in evaluation mode for up to 30 days.

1.4 Standalone Executables

Standalone executables may be created with VIs incorporating the LabSocket-Basic and LabSocket-MultiClient software.

1.5 Contact

For assistance with system setup or questions about the system, contact:

John Bergmans
Bergmans Mechatronics LLC
e-mail: jbergmans@bergmans.com

2. System Requirements

2.1 Browsers

Browsers that use the LabSocket system must be compatible with the HTML5 WebSocket interface standard. Fortunately, most modern mobile and desktop browsers are compatible with this standard. A complete list of WebSocket-compatible browsers is shown at: <http://caniuse.com/websockets>

For users who require support for older browsers such as Internet Explorer 9.0 and earlier, a commercial gateway product can be provided to emulate the behavior of the WebSocket interface.

2.2 LabVIEW Host Platform

LabVIEW	LabVIEW 2012 or later. Support for earlier versions of LabVIEW is available upon request.
Minimum Hardware	2.3 GHz CPU and 4 GB memory
Supported Operating Systems	<ul style="list-style-type: none">• Windows (XP or later)• Mac OS X (10.6.8 or later)

2.3 LabSocket Server Virtual Machine Host

The LabSocket Server Virtual Machine (VM) operates within the free Oracle VirtualBox virtualization application. Platform requirements for the LabSocket Server Virtual Machine (VM) host are

Minimum Hardware	2.3 GHz CPU and 4 GB memory
Supported Operating Systems	<ul style="list-style-type: none">• Windows, Mac OS X, Linux, Solaris• Details: https://www.virtualbox.org/manual/ch01.html-hostsupport

Note that it is feasible to install VirtualBox and VM on the LabVIEW Host Platform.

The LabSocket Server VM can be supplied as a vSphere VM upon request.

2.4 LabSocket Server Ports

LabSocket Server ports 80 and 61614 must be accessible to the browser and LabSocket Server port 61613 must be accessible to the LabVIEW Host Platform.

3. System Overview

To access a LabVIEW application, or "Target VI", over the Web, the LabSocket system operates "Support VIs" on the LabVIEW Host Platform in parallel to the Target VI, as shown in Figure 1.

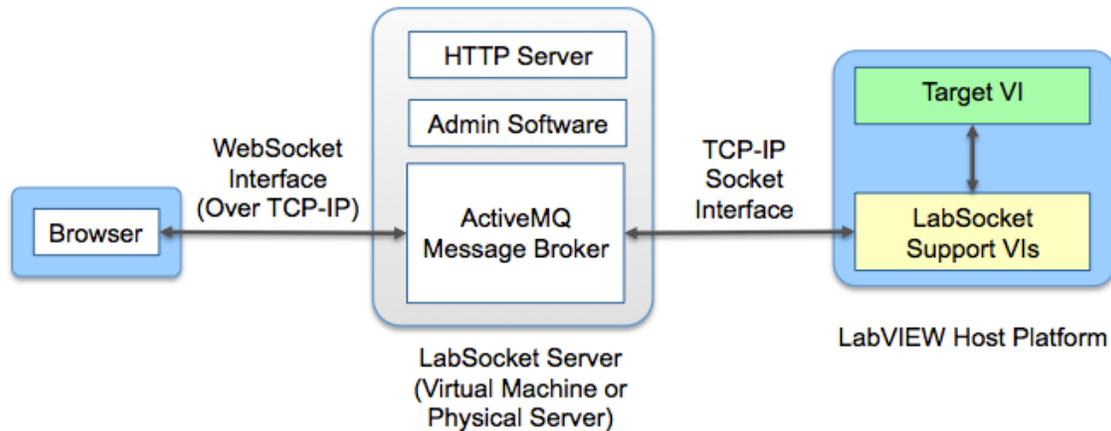


Figure 1. LabSocket System Overview

The LabSocket Support VIs operate unobtrusively and perform two key functions:

1. Upon starting the system, the Support VIs perform a screenscrape of the Target VI front panel. This function involves generating a web page using HTML and JavaScript code representing the Target VI front panel. This code is then posted to an HTTP server on the LabSocket Server platform.
2. After performing the screen scrape, the Support VIs enter Dynamic Synchronization Mode. In this mode, the properties of the elements on the front panel of the Target VI are continuously synchronized with those in the browser.

An ActiveMQ Message broker transfers data between the LabSocket Support VIs and the browser. A key feature of the broker is that it is able to establish a WebSocket interface to the browser. This interface enables continuous, bidirectional communications with the browser.

Administrative software on the LabSocket Server platform manages the browser code that is received from the LabSocket Support VIs.

The two main physical implementations of the LabSocket Server software are:

1. A customer-operated Virtual Machine (most popular for production). A pre-configured VirtualBox Virtual Machine (VM) is available for use with evaluation and commercial versions of the LabSocket software.

The use of a VM enables the entire LabSocket system to be easily set up within a customer LAN and assures customers complete control and ownership of the system and their data. The VM may be downloaded from the National Instruments LabVIEW Tools Network repository (See Section 4.1, below).

2. An existing Cloud Server (Quick start option) - This option is available for free for use with the LabSocket evaluation software and enables developers to get started quickly with the use of the system.

The use of both of these configurations of the system is described in this document.

Commercial users can also use one of the following two LabSocket Server implementations. Please contact BML to discuss and request these options.

1. Customer Cloud Server - A pre-configured cloud server instance that contains the LabSocket Server software can be leased for the exclusive use of a customer.
2. Native Operation - Free, detailed instructions are provided for setting up the individual LabSocket Server components directly on a customer's Windows, Mac or Linux platform.

4. Setup Instructions

This section provides step-by-step instructions on the installation of the LabSocket-Basic system. The instructions consist of two parts: i) LabSocket Server Virtual Machine installation; and, ii) LabSocket-Basic evaluation software installation.

Instructions to optionally uninstall the LabSocket-Basic software are also presented.

4.1 LabSocket Server Virtual Machine Installation

NOTE - If you plan to use the LabSocket Server software on BML's existing Cloud Server, proceed to Section 4.2 to begin installation of the LabSocket-Basic evaluation software.

1. Download and install the Oracle VirtualBox virtualization application (<http://virtualbox.org>) onto the physical machine that will act as the host platform for the Virtual Machine (VM). This host can use a Windows, Mac OSX or Linux operating system and may be either the LabVIEW Host PC or a separate physical machine.
2. Download the LabSocket Server VM Open Virtualization Format Archive (OVA) file named "LabSocket Server v2_5.ova" from the National Instruments LabVIEW Tools Network repository (http://www.ni.com/gate/gb/GB_EVALTLKTLABSOCKET/US) or from the Dropbox link provided by Bergmans Mechatronics.

(The LabSocket Server VM OVA file contains a Virtual Machine with Ubuntu 13.10 operating system, Apache HTTP server, ActiveMQ message broker and administrative software. All of this software is preconfigured and will automatically begin execution when the VM is started.)

3. Start VirtualBox
4. Select **File > Import Appliance...**
5. Navigate to the location of the OVA file and import it
6. From the Oracle VM Virtual Manager window, select the newly imported server and press the **Start** button. A new terminal window will open.

7. At the prompt, log in the VM using the username “labsocket” and password “labsocket” (no quotes for either value).
8. To enhance the security of the VM, change your password by entering the command

```
passwd
```

Enter and re-enter a new password of your choice when prompted.

9. Type the following command to obtain the IP address of the VM

```
ifconfig
```

Note the IP address for the "eth0" interface. This is the IP address that will be used later for communication with the LabSocket software on this VM.

10. The HTTP server, message broker and administrative software are all launched automatically when the VM is started. There is therefore no additional work required in the VM. End this terminal session using the command

```
logout
```

11. The LabSocket Server VM is now ready for use and will continue to operate in the background until it is stopped using the VirtualBox Manager window.

4.2 LabSocket-Basic Software Installation

1. Start LabVIEW on your LabVIEW Host Platform.
2. Install LabSocket-Basic VIPM package
 - a. Start VIPM by selecting **Tools > VI Package Manager...**
 - b. In the JKI Package Manager window select the VIPM file named **LabSocket-Basic**
 - c. In the LabSocket Package window, click **Install**
 - d. VIPM will display **LabSocket-Basic** as the only item in a list of Products to be installed. Click **Continue**
 - e. After installation, LabSocket will be displayed as the only item in a list of results of the last action. Click **Finish**
 - f. In the LabSocket Package window, select **File > Close**
 - g. The LabSocket package should now be visible in the JKI VI Package Manager window.
 - h. In the JKI VI Package Manager window, select **File > Exit**
 - i. Exit LabVIEW
3. Restart LabVIEW. Installation of the LabSocket-Basic package is now complete.

Alternate VIPM File Source

NOTE - The LabSocket VIPM file can optionally be downloaded from the LabVIEW Tools Network FTP directory:
http://download.ni.com/evaluation/labview/lvtn/vipm/packages/labsocket_basic_eval/. In this case, substitute Step 2b above with the following two steps:

- i. In the JKI Package Manager window select **File > Open Package File(s)**.
- ii. Select downloaded vip file downloaded. Click **Open**.

4.3 Uninstalling LabSocket Software (Optional)

1. Start LabVIEW
2. Remove LabSocket VIPM Package
 - a. Start VIPM by selecting **Tools > VI Package Manager...**
 - b. In the LabSocket Package Manager window, scroll through the list of packages to find the **LabSocket** package
 - c. Right-click on the **LabSocket** package
 - d. Select **Uninstall...**
 - e. VIPM will display LabSocket as the only item in a list of Products to be un-installed. Click **Continue**
 - f. After installation, LabSocket will be shown as having been un-installed. Click **Finish**
 - g. In the JKI VI Package Manager window, select **File > Exit**
 - h. Exit LabVIEW
3. Restart LabVIEW. Uninstallation of the LabSocket-Basic evaluation package is now complete.

5. LabSocket Application Development

5.1 Simple Demonstration Application

This section demonstrates how to use the LabSocket system to access a simple Target VI from a browser.

1. Create a new LabVIEW VI. From the main LabVIEW window, select **File > New VI**
2. A key component of the LabSocket system is a VI named **LabSocket Start.vi**. This VI calls the LabSocket Support VIs which in turn
 - i) perform the front panel screenscrape; and,
 - ii) synchronize the front panel and browser controls and indicators.**Right-click** in the block diagram of the new VI and select **Addons>LabSocket > LabSocket Start.vi** (Figure 2). Then, drag and drop this VI into the block diagram of the new VI.

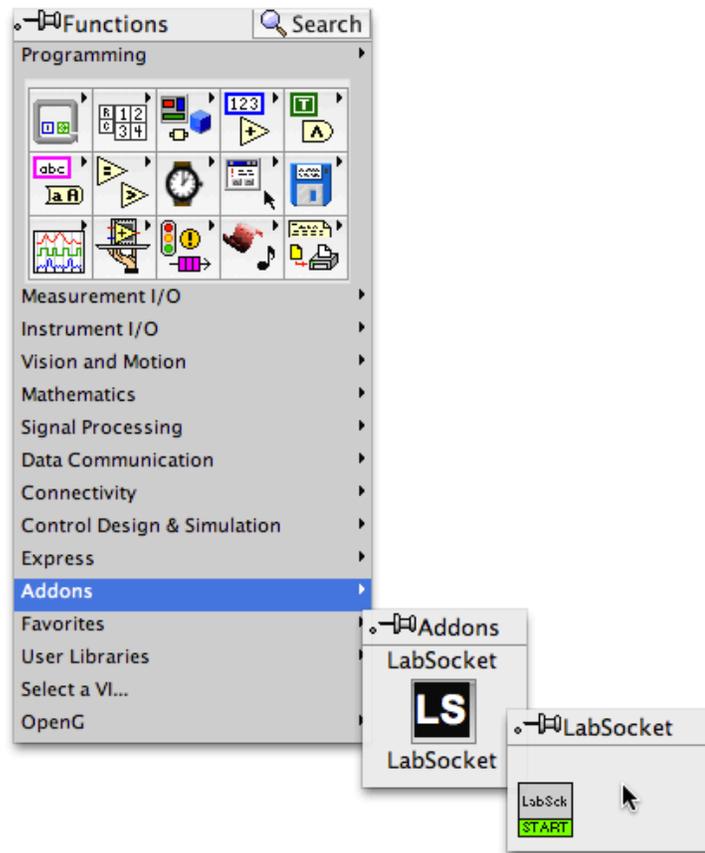
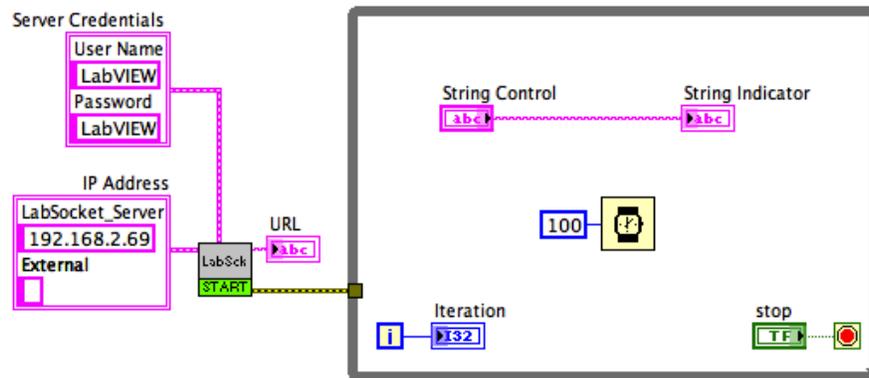


Figure 2. Selecting LabSocket Start.vi from Functions Palette

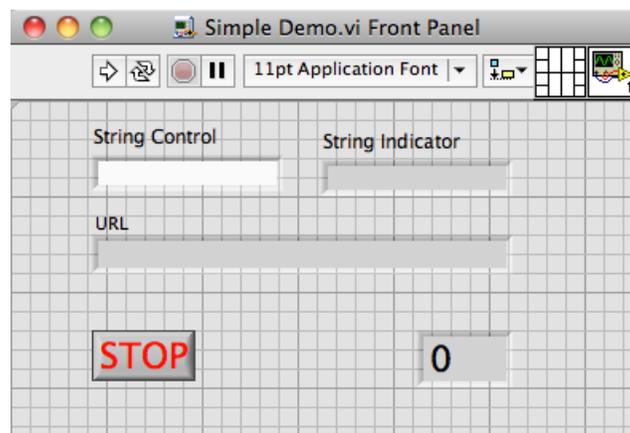
3. Wire the following three terminals of **LabSocket Start.vi** (Figure 3):
 - **IP Address** – a cluster containing two strings is wired to this input:
 - *LabSocket_Server* is the IP address of the LabSocket Server. Two options are available here:
 - i) Enter the IP Address of the LabSocket Server Virtual Machine relative to the LabVIEW Host platform. (Recall that this IP address value was obtained in Section 4.1, step 9).
 - OR
 - ii) Enter the string “labsocket.com” (without quotes) to use the LabSocket Server software on BML’s existing cloud server. This option is intended for users who want to quickly start using the system and not for production purposes.
 - *External* is the IP address of the LabSocket Server VM relative to the browser. Leave this value blank if this address is the same as the *LabSocket_Server* IP address or you’re using BML’s cloud server.
- **Server Credentials** – a cluster of two text strings each containing the string “LabVIEW” is wired to this input. This is the default value for this input and matches the default settings of the message broker. (See Section 8. User Administration for instructions on changing the message broker credentials)
- **URL** – wire a string indicator to this terminal. This indicator will display the URL at which the Target VI may be accessed with a browser. The URL is based on the filename of the VI. If “labsocket.com” is used as the LabSocket Server IP address, a random four-digit number is appended to the URL.

Details about the terminals of **LabSocket Start.vi** are available in Section 6

4. Add a while loop, a Wait VI, and simple controls and indicators as shown below (Figure 3). Wire the **error out** terminal of **LabSocket Start.vi** to the edge of the while loop. This wire forces **LabSocket Start.vi** to complete execution before the while loop starts.



a) Block Diagram



b) Front Panel

Figure 3. SimpleDemo.vi

5. Set the mechanical action of buttons to either "Switched when Pressed" or "Switched when Released". Save the file to disk. In this guide, the filename **Simple Demo.vi** is used, although any name is acceptable.
6. Start the VI by pressing the VI **Run** button.
7. If this is the first time "LabSocket Start.vi" has run on this platform, you will be prompted to enter a license string. Enter the license string provided to you by Bergmans Mechatronics..
8. The **LabSocket Status** window will appear. The URL at which the Target VI may be accessed will be displayed in the Log page of this window (Figure 4). This URL will match that output on the URL terminal of **LabSocket Start.vi**.

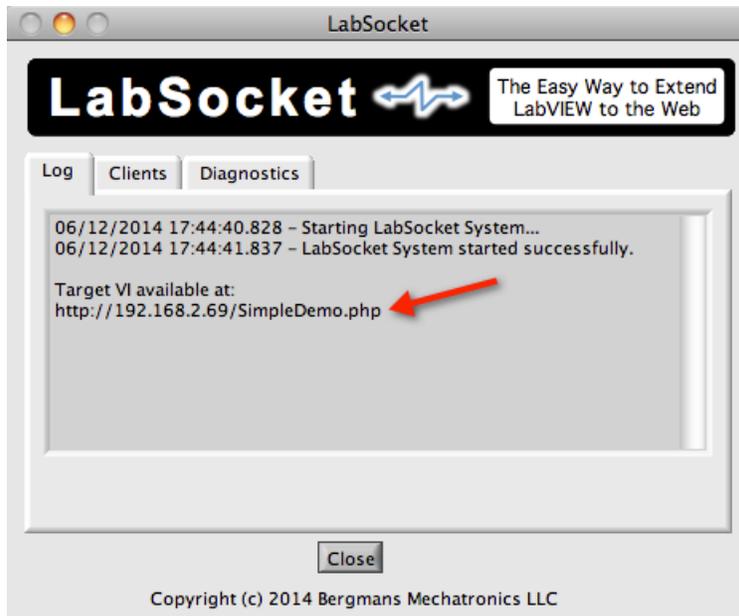


Figure 4. LabSocket Status Window (Log Page)

9. When a browser is pointed to the URL displayed in the **LabSocket Status** window, it will automatically display a representation of the Front Panel. All supported controls and indicators in the front panel will also become synchronized with those in the browser (Figures 5 and 6). As part of the process of connecting the browser to the VI, a pop-up window appears in the browser displaying the Application name and browser-specific “Client ID” number of the browser.

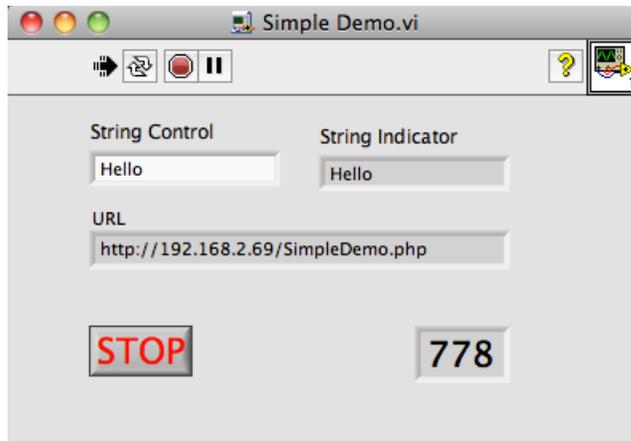


Figure 5. Front Panel of Target VI “SimpleDemo.vi”

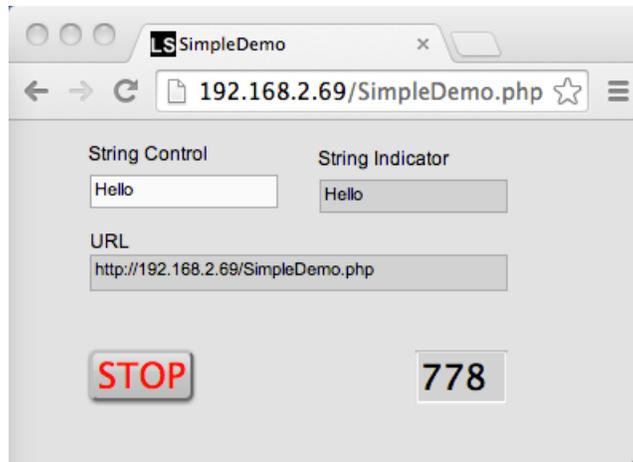


Figure 6. Browser Representation of Target VI “SimpleDemo.vi”

10. The **LabSocket Status** window mentioned earlier features two other pages: The Clients page (Figure 7) displays the Client ID of each connected browser in the “Browser ID” column. The “Last Heartbeat Echo” column displays the time of the last response from the browser to periodic test signals that are transmitted to each connected browser. (If an echo signal is not returned from a browser in greater than 10 seconds, the client is considered inactive and removed from the list of clients in the Status window.)

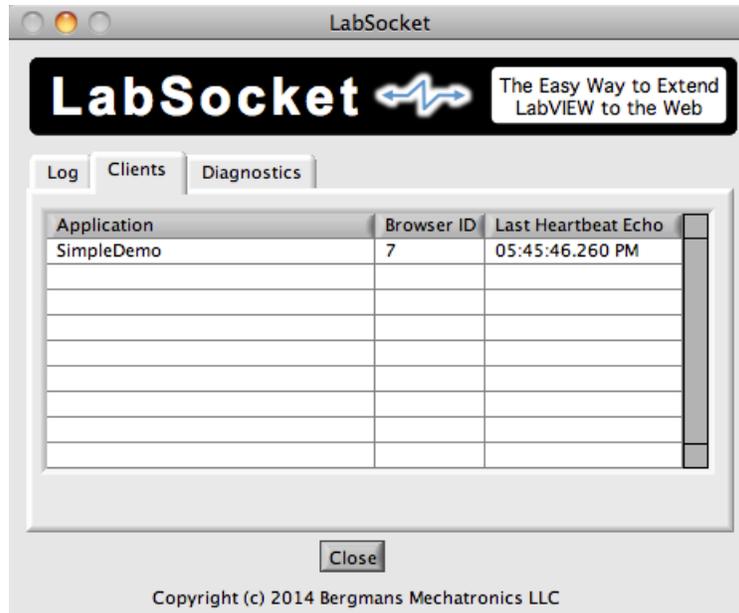


Figure 7. LabSocket Status Window – Clients Page

The Diagnostics page (Figure 8) of the **LabSocket Status** window displays a “Loop Period” value. This value is the period of time required to update the front panel of the Target VI with any changes sent from the browser plus the time required to detect changes in the Target VI front panel and to send these changes to each connected browser client.

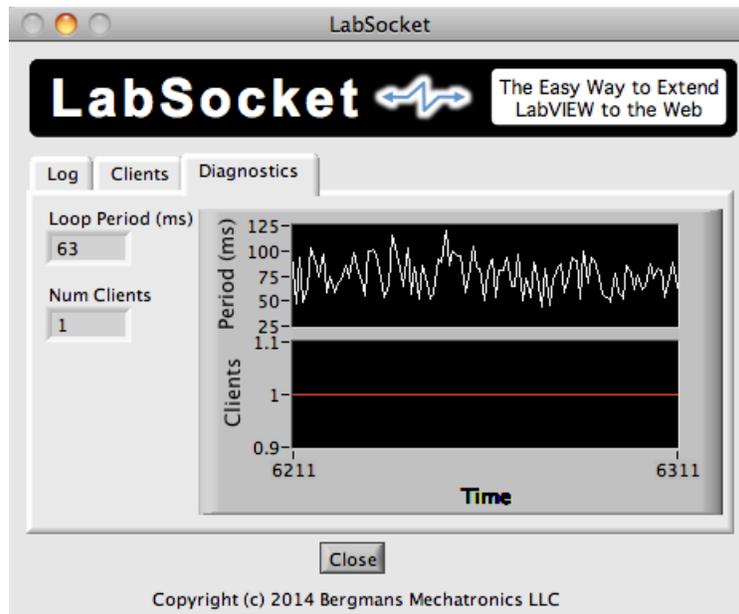


Figure 8. *LabSocket Status Window – Diagnostics Page*

The Diagnostics page also includes a display of the number of clients currently connected to the Target VI

11. Click **Close** to close the Status window and discontinue the synchronization of the Target VI and browser. Closing the Status window will not stop the Target VI and stopping the Target VI does not close the Status window.

5.2 Working with Images

A “PNG to Picture Converter” tool is provided with the LabSocket system to enable images contained in PNG files to be easily rendered in the browser. The process for working with images is as follows.

1. From the menu bar, select **Tools>PNG to Picture Converter...**
2. Select the PNG file of interest.

3. A new, unnamed VI will appear. Copy the picture control from this VI to the front panel of the Target VI.
4. Right-click on the picture. Select **Data Operations>Make Current Value Default**
5. Save the Target VI
6. Run the Target VI (assuming the **LabSocket Start.vi** and other components necessary for a practical application are already present on the VI block diagram).

A simple Target VI using a picture element and its representation in a browser is shown in Figure 9.

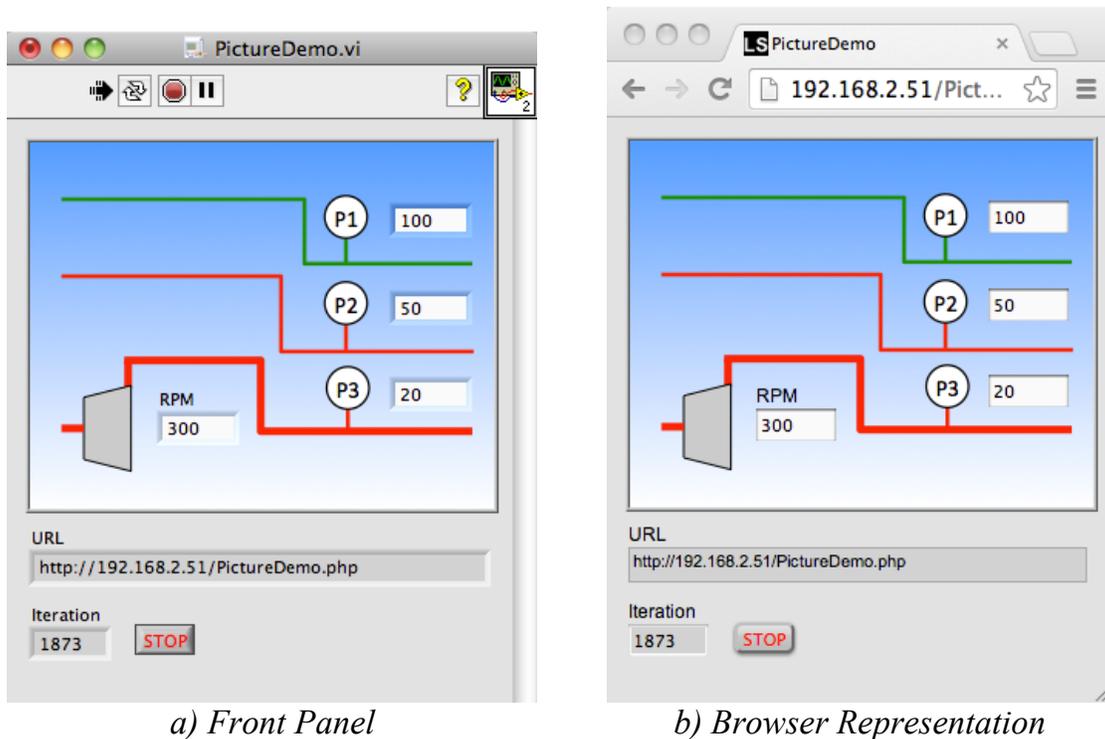


Figure 9. PictureDemo.vi Illustrating Rendering of Picture Elements in Browser

Controls and indicators may be superimposed in front of the picture element. Note that elements in the browser are displayed from back to front in order of increasing tabbing order value (ie tabbing order value 0 is at the back of the browser representation, with other elements displayed in front of it). Similarly, for elements in tab pages, elements are displayed from front to back in order of increasing tab “Control Order”.

6. Example Files

The LabSocket VIPM package installation process in Section 4.2 automatically installs a set of example Target VIs that illustrate some of the capabilities the system. These VIs may be accessed by selecting **Help > Find Examples...** from the main LabVIEW window. Select the **Directory Structure** radio button. In the directory structure, double-click on the **LabSocket** directory, to locate the examples (Figure 10). A description of each example VI is presented in Table 1.

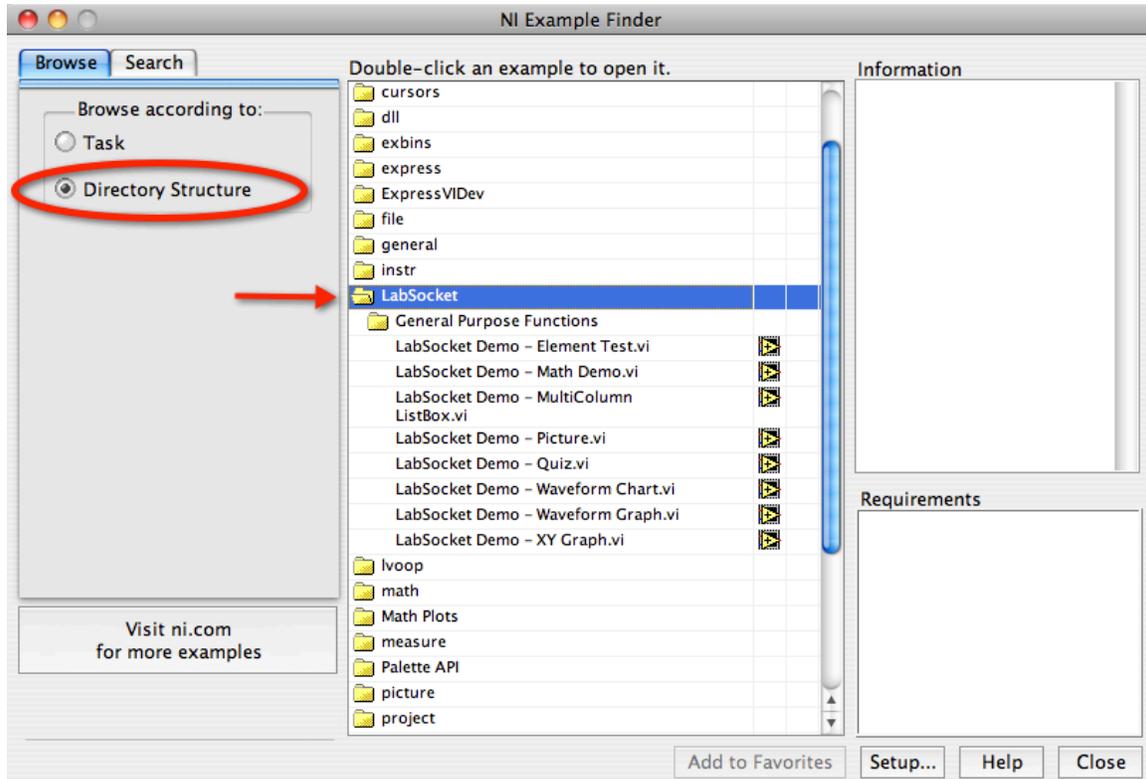


Figure 10. Example Files Supplied with LabSocket System

Table 1. Example VIs Supplied with LabSocket System

Filename	Description
LabSocket Demo - Element Test.vi	<p>Demonstrates synchronization of Front Panel elements values with elements in browser. Elements include:</p> <ul style="list-style-type: none"> • XY Graph • Picture • MultiColumn ListBox • Tabs <p>Also demonstrates support for dynamic replication in the browser of Front Panel element properties such as:</p> <ul style="list-style-type: none"> • button visibility, color, text, and disabled state • numeric control and indicator visibility and numeric control disabled state • string control and indicator visibility and background color
LabSocket Demo - Math Demo.vi	<p>Illustrates how to use the LabSocket system in a multiple-VI system.</p> <ul style="list-style-type: none"> • The front panel of the main Target VI is accessible over the web • Another VI displays the Target VI URL and while a third VI displays a Stop button for the application. Both of these VIs are only accessible on the LabVIEW Host Platform.
LabSocket Demo - MultiColumn ListBox.vi	<p>Demonstrates the eight combinations of MultiColumn ListBoxes that are possible based on i) row visibility; ii) column header visibility; and, iii) presence of vertical scrollbar.</p>
LabSocket Demo - Picture.vi	<p>Demonstrates Picture control support</p>
LabSocket Demo - Quiz.vi	<p>Illustrates how to set up a simple quiz application. Includes dynamic updates of button text and text indicator background color.</p>
LabSocket Demo - Waveform Chart.vi	<p>Demonstration of support for Waveform Charts</p>
LabSocket Demo - Waveform Graph.vi	<p>Demonstration of support for Waveform Graphs</p>
LabSocket Demo - XY Graph.vi	<p>Demonstration of support for XY Graphs</p>

7. LabSocket Start VI

The *LabSocket Start.vi* launches the routines that perform the Front Panel screenscrape and synchronize the front panel and browser. Terminals for this VI are shown in Figure 11. A description of each terminal is presented in Table 2. Connections to all terminals are recommended, but not required.

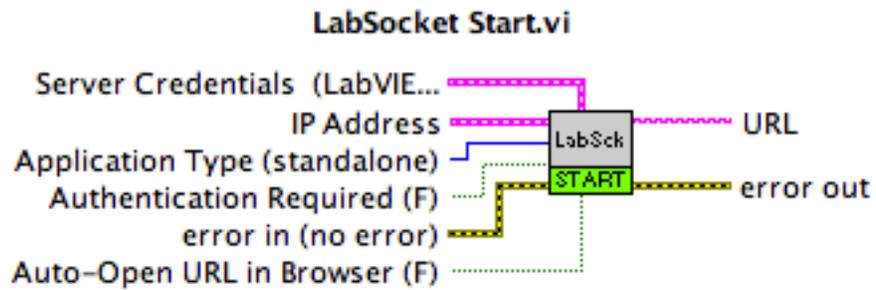


Figure 11. LabSocket Start VI

Table 2. LabSocket Start VI Terminals

Terminal	Description	Default Input Value
Server Credentials	Cluster containing username and password for access to LabSocket Server	LabVIEW / LabVIEW
LabSocket Server IP Address	<p>Cluster containing two strings:</p> <ul style="list-style-type: none"> • <i>LabSocket_Server</i> – IP Address of LabSocket Server relative to LabVIEW Host platform. Section 4.1, Step 9 describes how to determine this value. • <i>External</i> – IP Address of LabSocket Server relative to browser. Leave blank if same as <i>LabSocket_Server</i> IP address. <p>To use the LabSocket server software on the cloud server at labsocket.com during initial evaluation testing, set <i>LabSocket_Server</i> to “labsocket.com” (no quotes) and leave <i>External</i> blank</p>	XX.XX.XX.XX / blank
Application Type	<ul style="list-style-type: none"> • standalone – Target VI is rendered alone in a web page • moodle_integration – Target VI is rendered within an iFrame in a Moodle course web page. Contact Bergmans Mechatronics for more information on this option. 	standalone
Authentication Required	<p>User authentication requirement for browser access</p> <ul style="list-style-type: none"> • True – A pop-up window appears in browser when web page loads. User must enter credentials before web page becomes active (Figure 12) • False – User credentials are not requested. (In this case, the browser sends the credentials “client / client” to the message broker. Unless these credential settings have been changed, browser access will always be permitted.) <p>See Section 8 for additional details about user administration and authentication.</p>	False
error in	Errors that occur before <i>LabSocket Start.vi</i> is called	no error
Auto-Open URL in Browser	<ul style="list-style-type: none"> • True – Web page automatically opened in browser • False – User must manually point browser to URL of Target VI 	False
error out	Errors that occur prior to calling or within <i>LabSocket Start.vi</i>	NA
URL	URL for access to Target VI	NA

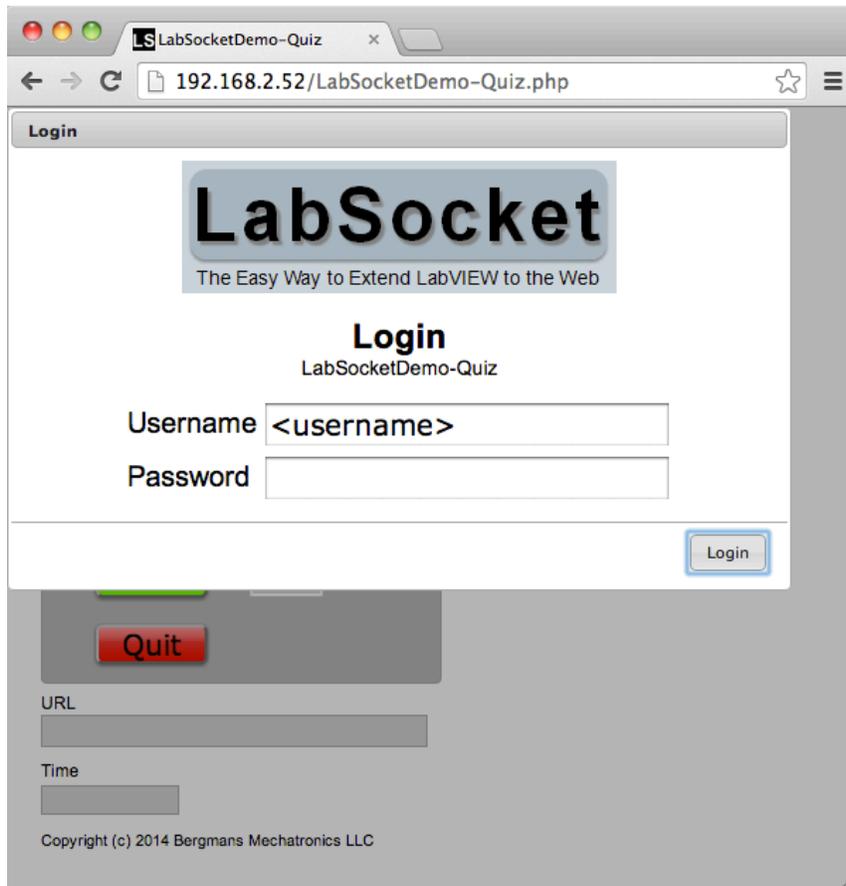


Figure 12. Login Window That Appears When Authentication Required=T

8. User Administration

All applications and users connecting to the ActiveMQ message broker require proper credentials before transmitting or receiving data via the broker. By default, the broker loads credentials from a configuration file on startup. This section describes how to modify the credentials in this configuration file.

LDAP User Authentication Option

NOTE - The message broker can optionally be configured to validate user credentials stored in a Lightweight Directory Access Protocol (LDAP) server. Benefits of the use of an LDAP server are:

- users can be authenticated using credentials in a customer's existing LDAP database
- administrators can manage users without restarting the broker
- a browser-based interface such as phpLDAPadmin can be used to manage user data
(http://phpldapadmin.sourceforge.net/wiki/index.php/Main_Page)

Contact Bergmans Mechatronics for details on implementing LDAP user management or to request a VM containing an OpenLDAP server.

In the instructions below, two locations within the VM directory structure are defined as follows:

```
$LABSOCKET_HOME = /home/labsocket  
$ACTIVEMQ_HOME = /home/labsocket/apache-activemq-5.9.0
```

To modify the credentials in the ActiveMQ configuration file:

1. Log in to LabSocket Server VM using an ssh client or the VirtualBox terminal window.
2. Navigate to `$ACTIVEMQ_HOME/conf`
3. Open **activemq-LabSocket.xml** in the vi editor
4. The `<simpleAuthenticationPlugin>` section beginning on line 47 defines the username and passwords for each user that is permitted to access the ActiveMQ message broker.

The default users are described in Table 3.

Table 3. Default User Definitions of ActiveMQ Message Broker in LabSocket Server VM

Username	Description
LabVIEW	name used by LabSocket Support VIs. This username and its associated password are wired to the Server Credentials input of LabSocket Start.vi
LabSocketPHP	name used by PHP-based administrative program. Credentials for this client are defined in the file <code>\$LABSOCKET_HOME/LabSocket_php/credentials.txt</code>
client	remote browser user
client2	remote browser user

- To add, delete or change user credentials, edit the settings in the `<simpleAuthenticationPlugIn>` section as required. If changing the credentials for the LabSocket Support VIs or PHP Administrative program, but sure to change the credential settings supplied by these clients (ie. Inputs to **LabSocket Start.vi** or contents of `$LABSOCKET_HOME/LabSocket_php/credentials.txt`.)
- Save and exit **activemq-LabSocket.xml**.

To activate the new user settings, restart the Virtual Machine using the command

```
sudo shutdown -r now
```

Enter the password for the user `labsocket` if prompted.