BridgeVIEW™

BridgeVIEW 2.1 Addendum

NATIONAL INSTRUMENTS™

March 1999 Edition
Part Number 321983B-01

Worldwide Technical Support and Product Information

`http://www.natinst.com`

National Instruments Corporate Headquarters

11500 North Mopac Expressway   Austin, Texas 78759-3504   USA   Tel: 512 794 0100

Worldwide Offices

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 011 284 5011,
Canada (Ontario) 905 694 0085, Canada (Québec) 514 694 8521, Denmark 45 76 26 00, Finland 09 725 725 11,
France 0 1 48 14 24 24, Germany 089 741 31 30, Hong Kong 2645 3186, India 91805275406,
Israel 03 6120092, Italy 02 413091, Japan 03 5472 2970, Korea 02 596 7456, Mexico (D.F.) 5 280 7625,
Mexico (Monterrey) 8 357 7695, Netherlands 0348 433466, Norway 32 84 84 00, Singapore 2265886,
Spain (Madrid) 91 640 0085, Spain (Barcelona) 93 582 0251, Sweden 08 587 895 00,
Switzerland 056 200 51 51, Taiwan 02 2377 1200, United Kingdom 01635 523545

For further support information, see the *Technical Support Resources* appendix of this manual.

Important Information

Warranty

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this manual is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

Except as specified herein, National Instruments makes no warranties, express or implied, and specifically disclaims any warranty of merchantability or fitness for a particular purpose. Customer's right to recover damages caused by fault or negligence on the part of National Instruments shall be limited to the amount theretofore paid by the customer. National Instruments will not be liable for damages resulting from loss of data, profits, use of products, or incidental or consequential damages, even if advised of the possibility thereof. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Trademarks

BridgeVIEW™, FieldPoint™, HiQ™, LabVIEW™, Lookout™, natinst.com™, NI-488.2™, NI-DAQ™, and SCXI™ are trademarks of National Instruments Corporation.

Product and company names mentioned herein are trademarks or trade names of their respective companies.

WARNING REGARDING MEDICAL AND CLINICAL USE OF NATIONAL INSTRUMENTS PRODUCTS

National Instruments products are not designed with components and testing for a level of reliability suitable for use in or in connection with surgical implants or as critical components in any life support systems whose failure to perform can reasonably be expected to cause significant injury to a human. Applications of National Instruments products involving medical or clinical treatment can create a potential for death or bodily injury caused by product failure, or by errors on the part of the user or application designer. Because each end-user system is customized and differs from National Instruments testing platforms and because a user or application designer may use National Instruments products in combination with other products in a manner not evaluated or contemplated by National Instruments, the user or application designer is ultimately responsible for verifying and validating the suitability of National Instruments products whenever National Instruments products are incorporated in a system or application, including, without limitation, the appropriate design, process and safety level of such system or application.

Contents

About This Addendum

Chapter 1
System Requirements, Installation, and Upgrade Information

Chapter 2
New Features in BridgeVIEW 2.1

Appendix A
Manual Clarifications and Additions

Appendix B
Technical Support Resources

Figures

Tables

About This Addendum

This addendum describes all of the new BridgeVIEW 2.1 features, compatibility issues, and configuration, installation, and upgrade information. Upgrade issues pertain to BridgeVIEW 2.1 only.

Note ***BridgeVIEW 2.1 ships with the same manual set that shipped with BridgeVIEW 2.0. This addendum updates that documentation. However, enhancements to BridgeVIEW 2.1 have rendered some information in the manuals incorrect. Please refer to Appendix A,*** *Manual Clarifications and Additions*, ***for corrections and important information.***

Regardless of whether you are a new or upgrade user, begin by reading Chapter 1, *System Requirements, Installation, and Upgrade Information*, before continuing with this installation.

If you are upgrading from a previous version of BridgeVIEW, carefully read the *Upgrading to BridgeVIEW 2.1* section in Chapter 1, *System Requirements, Installation, and Upgrade Information*.

Note ***BridgeVIEW is Year-2000 compliant. Because BridgeVIEW has never stored two-digit years, the change to 2000 does not affect any internal storage of dates.***

Conventions

This manual uses the following conventions:

<> Angle brackets enclose the name of a key on the keyboard—for example, <shift>. Angle brackets containing numbers separated by an ellipsis represent a range of values associated with a bit or signal name—for example, DBIO<3..0>.

[] Square brackets enclose optional items—for example, [`response`].

- A hyphen between two or more key names enclosed in angle brackets denotes that you should simultaneously press the named keys—for example, <Ctrl-Alt-Delete>.

» The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options»Substitute Fonts** directs you to pull down the **File** menu, select the **Page Setup** item, select **Options**, and finally select the **Substitute Fonts** options from the last dialog box.

✎ This icon to the left of bold italicized text denotes a note, which alerts you to important information.

💡 This icon to the left of bold italicized text denotes a tip, which alerts you to advisory information.

bold Bold text denotes the names of menus, menu items, parameters, dialog boxes, dialog box buttons or options, icons, windows, Windows 95 tabs, or LEDs.

italic Italic text denotes variables, emphasis, a cross reference, or an introduction to a key concept. This font also denotes text from which you supply the appropriate word or value, as in Windows 3.*x*.

`monospace` Text in this font denotes text or characters that you should literally enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames and extensions, and for statements and comments taken from programs.

`monospace italic` Italic text in this font denotes that you must enter the appropriate words or values in the place of these items.

paths Paths in this manual are denoted using backslashes (\) to separate drive names, directories, folders, and files.

1

System Requirements, Installation, and Upgrade Information

This chapter describes the system requirements for the BridgeVIEW software and contains installation instructions and updated documentation information.

Read the *System Requirements* section, then follow the instructions in the *Installing BridgeVIEW* section of this chapter.

System Requirements

BridgeVIEW is distributed in three different CD-ROM packages—the BridgeVIEW Base Package, the BridgeVIEW Full Development System, and the BridgeVIEW Professional Development System. All of the BridgeVIEW packages contain the BridgeVIEW Engine, configuration tools, and editing environment that you need to build measurement and automation systems.

The BridgeVIEW Full Development System (FDS) also includes the Industrial Automation Servers and the Advanced Analysis libraries.

The BridgeVIEW Professional Development System (PDS) includes the FDS, in addition to source code control utilities, Application Builder, metric utilities, and graphical differencing. If you use the Application Builder to create stand-alone executables that require the BridgeVIEW Engine, you must install the BridgeVIEW Run-Time System to run the executables.

The following table lists the minimum system requirements needed to run BridgeVIEW 2.1.

Table 1-1. System Requirements

Platform	Media and System Requirements	Important Notes
All versions	Distributed on CD. A separate CD contains the complete instrument driver library. You need a minimum of 180 MB of disk storage space for the FDS. You need a minimum of 24 MB of RAM.	National Instruments recommends 32 MB of RAM and a Pentium processor for this version to run effectively.
Windows 98	BridgeVIEW runs on any system that supports Windows 98.	
Windows 95	BridgeVIEW runs on any system that supports Windows 95.	National Instruments recommends at least 30 MB of swap space available on your system for this version to run effectively.
Windows NT	BridgeVIEW runs on Windows NT 4.0 Service Pack 3 or later. DEC Alpha, MIPS, and PowerPC 80*x*86 emulators must emulate 80386 instructions to run BridgeVIEW.	National Instruments recommends at least 30 MB of swap space available on your system for this version to run effectively.

Note ***The standard BridgeVIEW FDS installation requires approximately 180 MB of disk space. If you plan to install the NI-DAQ OPC Server, you need an additional 20 MB of disk space. The Industrial Automation Servers require approximately 10 MB.***

Upgrading to BridgeVIEW 2.1

If you are not upgrading from an earlier version of BridgeVIEW, skip to the *Installing BridgeVIEW* section later in this chapter. If you are upgrading from a version prior to 2.0, you can find upgrade information on your BridgeVIEW CD and also on our web site. The *BridgeVIEW 2.0 Upgrade Notes* are available as a Portable Document Format (PDF) file called

`Upgrade.pdf` in the `BridgeVIEW\manuals` directory.
You can also find this file on the BridgeVIEW support page at `http://www.natinst.com/support/`, in the Product Manuals Library. To view these files, you must have Adobe Acrobat Reader 3.0 or later installed. If you do not have Adobe Acrobat Reader 3.0 or later installed, you can download it from the Adobe Systems Incorporated web site at `http://www.adobe.com/`.

For more information about BridgeVIEW features, refer to the *BridgeVIEW User Manual* and *G Programming Reference Manual*. To access the extensive BridgeVIEW online documentation, select **Help» Online Reference**.

Converting VIs

Upgrading BridgeVIEW VIs is an automated process. When you open a VI created in a previous version, BridgeVIEW automatically converts and compiles the VI.

Conversion is a memory-intensive operation. When BridgeVIEW opens a VI saved in an earlier version, it loads all components of the converted VI (front panel, block diagram, and data) into memory, then compiles the VI in memory. BridgeVIEW also loads into memory the components of all subVIs that need to be converted.

To estimate the amount of memory required to convert VIs, total the amount of disk space that your VIs and all of their subVIs occupy. If these VIs are in VI libraries, add approximately 30 percent of the VI library size to the total, because the VIs are compressed within the library. The conversion process might require at least that total and an additional 3 MB of memory.

If your computer does not have enough memory to convert your VIs all at once, convert the VIs in stages, by components. Examine your hierarchy of VIs and begin by opening and saving subVIs in the lower levels of the hierarchy. Continue to the higher levels of the hierarchy. To convert a directory of VIs, select **File»Mass Compile**. However, this option converts the VIs in a directory or VI library in alphabetical order. If a high-level VI is encountered first, **Mass Compile** requires approximately the same amount of memory as described in the previous paragraph.

To monitor your memory usage, select **Help»About BridgeVIEW**. This option summarizes the amount of memory that you are using.

Upgrading Application Libraries and Toolkits

Most existing toolkits function with BridgeVIEW 2.1 without problems. However, you need to move the VIs so that they appear in the BridgeVIEW menus. BridgeVIEW 2.1 is compatible with toolkits designed for 2.0, with the following exceptions.

You must upgrade the following add-ons for compatibility with BridgeVIEW 2.1:

- **LabVIEW Application Builder**—You must upgrade to LabVIEW Application Builder 5.1. This upgrade is free to existing users of the LabVIEW Application Builder. If you have the Professional Development System, the new version of the Application Builder libraries is included in the installation.
- **Professional G Developers Toolkit**—If you have the Professional G Developers Toolkit, you must upgrade to version 5.1. This upgrade is free to existing users of the Professional G Developers Toolkit. If you have the Professional Development System, the new version of the Professional G Developers Toolkit is included in the installation.
- **LabVIEW Test Executive—**If you use LabVIEW Test Executive 5.0 or earlier, you must upgrade to LabVIEW Test Executive 5.1. This upgrade is free to existing users of LabVIEW Test Executive 5.0.

With minor exceptions, you can use the previous version of the following toolkit with BridgeVIEW 2.1:

- **Internet Developers Toolkit for G**—You can use the Internet Developers Toolkit 4.1 with BridgeVIEW 2.1, but you must delete `printvi.llb`, located in the `user.lib\internet\image` directory. If you upgrade to version 5.0 of this toolkit, this fix is made automatically. This upgrade is free to existing users of the Internet Developers Toolkit 4.1.

The following toolkits do not install VIs in a location that causes them to appear in the palettes. BridgeVIEW 2.1 updates these toolkits to version 2.0. To use the existing toolkits, move the VIs to `vi.lib\addons` or `user.lib` directories. Alternatively, select **Edit»Edit Control and Function Palettes** and add them to the palette of your choice.

- **Statistical Process Control Toolkit 1.0**
- **Proportional-Integral-Derivative (PID) Control Toolkit 1.0**—National Instruments recommends that you upgrade to version 2.0 of this toolkit.

If you are using the BridgeVIEW Full Development System or Professional Development System, you have two other toolkits—G Math and Picture Control. If you are using the BridgeVIEW Base Package, you can get these toolkits by upgrading to either the BridgeVIEW FDS or PDS.

Installing BridgeVIEW

To install BridgeVIEW, complete the following steps:

1. Insert the CD in your CD-ROM drive.
2. Run the BridgeVIEW installer, as follows:
    a. If your system has the AutoPlay feature enabled, the **Welcome to BridgeVIEW** screen appears a short time after you insert the CD.
    b. If your system does not use AutoPlay, run the following program:

    ```
    x:\bvsetup.exe
    ```

    where *x* is the letter of your CD-ROM drive.
3. Select an installation type. The installer offers several installation types: **Standard**, **Full**, **Minimum**, and **Custom**. National Instruments recommends the **Standard** installation.

Note ***If you plan to use National Instruments Data Acquisition (DAQ) devices, VISA, or GPIB instrumentation, you can select either the*** **Full** ***installation, which installs all necessary drivers and example programs, or the*** **Custom** ***installation, in which you select the specific items to install. If the installer detects a more recent version of a driver already installed on your system, use that driver.***

4. After you select an installation type, follow the instructions that appear on your screen to complete the installation.
5. The BridgeVIEW Help files contain 256-color graphics. Select **Control Panels»Display** to configure your video driver for at least 256 colors with 800 × 600 pixel resolution.

After you install BridgeVIEW, it is ready to run. If you selected the **Full** installation, you might need to restart your system so that it properly loads the DAQ, VISA, or GPIB drivers.

If you have one of the BridgeVIEW add-on toolsets, you can install those files at this time.

For More Information about BridgeVIEW

The *BridgeVIEW User Manual* and the *BridgeVIEW Online Reference*, which you can access by selecting **Help»Online Reference**, provide activities that illustrate basic BridgeVIEW concepts. If you want to complete these activities, copy the `BridgeVIEW\Activity` directory from the BridgeVIEW CD to your hard drive.

The BridgeVIEW documentation set, including the *LabVIEW Code Interface Reference Manual* and the *LabVIEW VXI VI Reference Manual*, is available in Portable Document Format (PDF) on the BridgeVIEW CD in the `manuals` directory. You can copy this directory or selected PDF files to the `BridgeVIEW\manuals` directory on your hard drive. To view these files, you must have Adobe Acrobat Reader 3.0 or later installed. If you do not have Adobe Acrobat Reader 3.0 or later installed, you can download it from the Adobe Systems Incorporated web site at `http://www.adobe.com/`.

Data Acquisition, VXI, and GPIB Installation Notes

All National Instruments GPIB interfaces and DAQ devices come with the drivers and other software that you need to use them. BridgeVIEW also comes with the drivers and other software that you need to use National Instruments hardware. While the drivers included with BridgeVIEW are the same NI-488.2 and NI-DAQ drivers that National Instruments includes with its GPIB and DAQ hardware, the version numbers might differ. The installer will warn you before you replace a later version of the drivers.

When you install BridgeVIEW, the installer places the application and most of the related files in the directory that you specify. The default name of this directory is `BridgeVIEW`. If you install DAQ or GPIB VIs, the installer places additional files in this directory, as described in the following sections.

To configure your hardware, use the National Instruments Measurement & Automation Explorer, which runs with BridgeVIEW. To start the Measurement & Automation Explorer, double-click on the **Measurement & Automation Explorer** icon on your desktop. For instructions on how to configure your particular DAQ device, refer to the Measurement & Automation Explorer Help.

For more information about the NI-DAQ driver, select **Start»Programs» BridgeVIEW»NI-DAQ Read Me File**.

Note ***National Instruments periodically updates drivers for GPIB and VXI. If you install new GPIB or VXI hardware for use with BridgeVIEW, the included drivers might overwrite those sent with BridgeVIEW. Compare the version numbers and use the driver with the higher number.***

Installing and Using BridgeVIEW 2.1 Servers

All BridgeVIEW development systems come with the following OPC servers for National Instruments hardware:

- NI-DAQ OPC Server (NI-DAQ 6.5 is included with BridgeVIEW 2.1)
- FieldPoint OPC Server
- FOUNDATION Fieldbus OPC Server—FOUNDATION Fieldbus driver software is not included, but is shipped with National Instruments Fieldbus hardware

The BridgeVIEW 2.1 FDS and PDS include a full set of servers for a large set of industrial automation devices. These servers are packed in two sets—Industrial Automation Servers and Lookout Protocol Drivers.

Industrial Automation Servers

The BridgeVIEW 2.1 FDS and PDS include the Industrial Automation Servers 2.1.

To install the servers shown in the following table, select the **Install Industrial Automation Servers** option during the BridgeVIEW installation, and select the specific servers that you want to install. The full set of Industrial Automation Servers requires 10 MB of disk space.

Table 1-2. Industrial Automation Servers

Server Name	Supported Protocols	Requirements
Allen-Bradley (requires RSLinx from Rockwell Software)	DH-485	Data Highway interface
	DH+	Data Highway+ interface
	Ethernet	Any Ethernet
	DF1	Any RS-232 or RS-485 interface

Table 1-2. Industrial Automation Servers (Continued)

Server Name	Supported Protocols	Requirements
Applicom (requires Applicom card and software)	Profibus FMS	Applicom PC1500PFB
	Profibus FDL	Applicom PC1500PFB
	Profibus DP	Applicom PC1500PFB
	Siemens MPI	Applicom PC1500PFB
	Siemens PPI	Applicom PC1500PFB
	Siemens 3964 and 3964R	Applicom PC1000, PC2000, PC2000ETH
	Siemens SINEC H1 Ethernet	Applicom PC2000ETH
	Siemens AS511	Applicom PC1000, PC2000, PC2000ETH
	Telemecanique Ethway	Applicom PC2000ETH
	Telemecanique Fipway	Applicom PC2000FIP
	Telemecanique Unitelway	Applicom PC1000, PC2000, PC2000ETH
	Allen-Bradley DF1	Applicom PC1000, PC2000, PC2000ETH
	Jbus	Applicom PC1000, PC2000, PC2000ETH
	Modbus	Applicom PC1000, PC2000, PC2000ETH
	SAIA Bus	Applicom PC1000, PC2000, PC2000ETH
	GE SNP-X	Applicom PC1000, PC2000, PC2000ETH
	SUCOMA	Applicom PC1000, PC2000, PC2000ETH
	Omron SYSMAC Way	Applicom PC1000, PC2000, PC2000ETH
	Siemens TI Direct	Applicom PC1000, PC2000, PC2000ETH
FieldPoint	FieldPoint	Any RS-232 or RS-485 interface
GE Fanuc	Ethernet	Any Ethernet
	CCM	Any RS-232 or RS-485 interface
Koyo (PLC Direct)	DirectNET/CCM/Hostlink	Any RS-232 or RS-485 interface
	Modbus RTU	Any RS-232 or RS-485 interface
Mitsubishi	Mitsubishi Dedicated Protocol 1 with checksum	Any RS-232 or RS-485 interface
Modicon Modbus	RTU	Any RS-232 or RS-485 interface
	Plus	Modicon SA85 interface
Omron	SYSMAC Hostlink	Any RS-232 or RS-485 interface

Table 1-2. Industrial Automation Servers (Continued)

Server Name	Supported Protocols	Requirements
Optomux	Optomux	Any RS-232 or RS-485 interface
Phillips	PPCCOM	Any RS-232 or RS-485 interface
PLC Direct (Koyo)	DirectNET/CCM/Hostlink	Any RS-232 or RS-485 interface
	Modbus RTU	Any RS-232 or RS-485 interface
PROFIBUS		Applicom card and software
Siemens	3964 with RK 512	Any RS-232 or RS-485 interface
	3964R with RK 512	Any RS-232 or RS-485 interface
	AS511	Any RS-232 or RS-485 interface
Telemecanique		Applicom card and software
TI	CCM/Hostlink	Any RS-232 or RS-485 interface

Lookout Protocol Drivers

The set of protocol drivers used with Lookout have been repackaged for BridgeVIEW as a single OPC server that supports a large number of industrial automation devices. The BridgeVIEW 2.1 FDS and PDS include this OPC server. For a list of the protocol drivers supported by the Lookout Protocol Drivers OPC server, refer to Table 1-3.

To install this server, select the **Install Industrial Automation Servers** option during the BridgeVIEW installation, and select the **Lookout Protocol Drivers** option. This set of drivers requires 10 MB of disk space.

To use this set of drivers in BridgeVIEW, select the OPC server named `National Instruments.OPCLookoutDrivers` from the set of servers. Then, run the OPC server executable and create a process file that includes objects for each driver you want to use, as described in the following section, *Configure the OPC Server*.

Only an OPC client or a user can run this OPC server. If a client runs it, it will always run in OPC client mode. If a user runs it, it can run in two modes—edit mode and OPC client mode. If the user starts the server and it is not already running via an OPC client, the server starts up in edit mode and the user can create or edit a process. If the server is already running via an OPC client and the user opens the server, it will be in OPC client mode and the user can only view and acknowledge alarms.

Configure the OPC Server

To configure the Lookout protocol drivers for this OPC server, complete the following steps:

1. Run the `lkopcsrv.exe` application.
2. Select **File»New** to create a new process.
3. Create a new object, for example a Modbus object. Configure the object and give it an object name. You can create multiple objects, and you need at least one object for each protocol driver you want to use from BridgeVIEW.
4. Select **File»Save** to save the process.
5. Select **Options»System»Startup process file**. Type the **fully-qualified** path name to the process file that you saved.
6. Close the server.

When BridgeVIEW launches the server, the server automatically uses that process file. The protocol driver items appear in BridgeVIEW with the naming convention *objectname.item name*. For more information, refer to the `lkopcsrv.hlp` help file.

The following table lists the protocol drivers supported by the Lookout Protocol Drivers OPC server.

Table 1-3. Lookout Protocol Drivers

Protocol Driver Name	Important Notes
National Instruments NI-DAQ	5.0 and later
National Instruments FieldPoint Serial	
National Instruments FOUNDATION Fieldbus	
National Instruments Profibus	Works with PN 777654-01
National Instruments Remote SCXI	
National Instruments SCXI	
Advantech PCL	
Allen-Bradley 1784-KT DH+	502, 505, SLC 500
Allen-Bradley 1784-KTX DH+	502, 505, SLC 500
Allen-Bradley Ethernet	502, 505, SLC 500
Allen-Bradley KE/KF Serial/DF1	502, 505, SLC 500
Allen-Bradley S-S Technologies 5136-D DH+	502, 505, SLC 500

Table 1-3. Lookout Protocol Drivers (Continued)

Protocol Driver Name	Important Notes
April 1000 PLCs Ethway	Applicom PC2000ETH
April 1000 PLCs FipWay	Applicom PC2000ETH
April 1000 PLCs Jbus	Applicom PC1000
Aquatrol W1300	Modbus
Aquatrol W1500	
ASCII general-purpose serial	32-bit only
Cutler-Hammer D50, D300 Serial	
CTI TCP/IP	For Siemens TI Series 500 PLCs
Delta-Tau	Motion control
Fisher ROC	
FOUNDATION Fieldbus	32-bit only
GE Fanuc CCM2	
GE Fanuc CCM3	
GE Fanuc Ethernet	32-bit only
GE Fanuc Modbus	
GE Fanuc SNPX	Series 90-30 and 90-70
Hitachi CPU22-02HB and CPU22-02HC Serial	
Klöckner-Moeller SUCOMA	Applicom PC1000
Koyo PLC Direct DirectNet	205, 405
Koyo PLC Direct Modbus	305, 405
Koyo PLC Direct Optomux	305, 405
Merlin Gerin BatiBUS	Applicom PC1000
Mitsubishi A-Series	
Mitsubishi FX	FX-24MR
Modicon Modbus	Daniel Floating Point
Modicon Modbus ASCII	
Modicon Modbus ASCII Slave	
Modicon Modbus Plus	SA85 card

Table 1-3. Lookout Protocol Drivers (Continued)

Protocol Driver Name	Important Notes
Modicon Modbus RTU	
Modicon Modbus RTU Slave	
Motorola Modbus	For Moscad
Omron C200H	
Omron CPM1	
Omron CQM	
OPC client	
Opto 22 Optumux	
Otic Fisher and Porter Data Link	Applicom PC1000
Philips PPC COMM	P8 Series
Profibus DP	SS-Tech 5136-PFB
	Works with PN 777654-01
Profibus FMS	Works with PN 777654-01
Profibus SINEC L2	
Profibus SINEC L2	S-S Technologies 5136-PFB
Reliance AutoMate	16-bit only
Reliance PC-Link	16-bit only
RKC Serial for REX-F400, -F700, and -F900 PLCs	
SAIA Bus	Applicom PC1000
Siemens 3964 and 3964R Native Serial	
Siemens AS511	Applicom PC1000
Siemens AS511 Native Serial	32-bit only
Siemens MPI	Works with PN 777654-01
Siemens PPI	Works with Applicom PC1000
	Works with PN 777654-01
Siemens PROFIBUS FDL	Works with PN 777654-01
Siemens PROFIBUS L2	Works with PN 777654-01
Siemens SINEC H1Ethernet	Applicom PC2000ETH

Table 1-3. Lookout Protocol Drivers (Continued)

Protocol Driver Name	Important Notes
Siemens TI Direct	Series 500 PLCs
Siemens TI Dynamic	8635 and 8640 RTUs
Siemens TI H1 for Series 500 PLCs	16-bit only
Siemens TI HostLink	205, 305, 405
Siemens TI Modbus	305, 405
Siemens TI TIWAY	Host Adapter PC Board
	Unilink Host Adapter
Siemens TI TIWAY 405	
Siemens TI TIWAY Autoset 1500	
Siemens TI TIWAY Facts 405	
Sixnet	
Spectrum	
Square D Serial	SCP1xx, SCP3xx-SCP7xx
Square D PowerLogic SY/LINK	SCP1xx, SCP3xx-SCP7xx
S-S Technologies 5136-SD	A-B DH or DH+
TCP/IP	Any compliant device (32-bit only)
Telemecanique Ethway	Applicom PC2000ETH
Telemecanique Unitelway	Applicom PC1000
Tesco Liquitronic 4	
Tesco Liquitronic 5	
Toshiba EX100	
Toshiba M Series	
Toshiba T Series	
UDP	Any compliant device (32-bit only)
Wizdom co-processor board	Motion control

Installing HiQ for Windows

The BridgeVIEW CD includes HiQ, a National Instruments mathematics application. If you install HiQ from the BridgeVIEW CD, a registration dialog box prompts you to enter a registration number. Type your BridgeVIEW registration number in the HiQ registration dialog box. BridgeVIEW 2.1 includes functionality that supports HiQ. If you want to use this functionality, but do not yet have HiQ, install HiQ after you install BridgeVIEW. For more information about this functionality, refer to the *HiQ and MATLAB Functionality* section in Chapter 2, *New Features in BridgeVIEW 2.1*.

Examples and Solutions for Your BridgeVIEW Programs

If you are using data acquisition (DAQ) or instrument I/O and want to find examples or generate solutions for your BridgeVIEW programs, launch the **DAQ Wizard** or the **Instrument Wizard** from the BridgeVIEW **Project** menu. For more information about these wizards, refer to Chapter 3, *Data Acquisition*, and Chapter 4, *Instrumentation*, of the *LabVIEW QuickStart Guide*.

To find any other type of example, use the Search Examples Help by selecting **Help»Search Examples**.

The `BridgeVIEW\examples` directory contains a VI named `readme.vi`. With this VI, you can find all the available examples. When you select a VI or a VI library, the documentation for that VI displays in the **Selection Description** text box. If the VI is open, you can also select **Window»Show VI Info** to view the documentation. Within `readme.vi`, you can open a selected VI by double-clicking on its name. You can also select **File»Open** and navigate to your VI.

Note ***Because the examples were reorganized with this release of BridgeVIEW, many paths to examples are not listed or are listed incorrectly in the BridgeVIEW documentation set. Some examples were moved from the*** `examples\G examples` ***directory to the*** `examples` ***directory. For more information about the updated activities and examples, refer to Chapter 2,*** *New Features in BridgeVIEW 2.1*.

Low-Level Register I/O for Windows 98/95

BridgeVIEW has two VIs—In Port and Out Port—that you can use to read or write hardware registers. These VIs work under Windows 98/95. Windows NT applications, however, cannot manipulate hardware directly. If you need to communicate with a hardware device in Windows NT, you must write a Windows NT driver.

Compatibility Issues between Versions 1.1 and 2.*x*

This section describes the compatibility issues between different BridgeVIEW versions.

Compatibility VIs for New Server Functionality

BridgeVIEW 2.*x* can act as a server, so you have expanded control over VIs. You can control VIs across a TCP/IP network and the ActiveX interface. BridgeVIEW includes Compatibility VIs for the VI Control VIs from previous versions. For information about using functionality from the VI Control VIs using the new server functions, open each VI Control VI and analyze the diagram use of the VI Server feature. You can copy this code to your BridgeVIEW applications.

Compatibility VIs for ActiveX Functions

The ActiveX functionality has expanded. The functions are more generic because BridgeVIEW 2.*x* can act as an ActiveX server, as well as a client. Compatibility VIs are provided for the ActiveX functions from previous versions. For more information about the new ActiveX functions, refer to the *Enhancements to ActiveX Automation* section in Chapter 2, *New Features in BridgeVIEW 2.1*.

Additional Resources

The BridgeVIEW documentation set, including the *LabVIEW Code Interface Reference Manual* and the *LabVIEW VXI VI Reference Manual*, is available in Portable Document Format (PDF) on the BridgeVIEW CD in the `manuals` directory. You can copy this directory or selected PDF files to the `BridgeVIEW\manuals` directory on your hard drive. To view these files, you must have Adobe Acrobat Reader 3.0 or later installed.

If you need to perform data acquisition, refer to the *LabVIEW Data Acquisition Basics Manual*. This manual contains important information about the DAQ VIs and examples you can use in BridgeVIEW. For information about a specific DAQ VI, refer to the *LabVIEW Function and VI Reference Manual* and the *BridgeVIEW Online Reference*, which you can access by selecting **Help»Online Reference**. For more information about new features and VIs, refer to Chapter 2, *New Features in BridgeVIEW 2.1*, in this addendum.

The DAQ examples folder contains `RUN_ME.LLB`, a VI library that includes an example Getting Started VI for analog input, analog output, digital I/O, and counters. You can use the `RUN_ME.LLB` examples as a starting place for using data acquisition with BridgeVIEW.

2

New Features in BridgeVIEW 2.1

This chapter describes the features added and the significant changes made between versions 2.0 and 2.1. To learn more about BridgeVIEW, refer to the extensive online documentation, which you can access by selecting **Help»Online Reference**.

Default Controls and Functions Palettes Updates

The BridgeVIEW 2.1 default **Controls** and **Functions** palettes have been reorganized to match the LabVIEW 5.1 **Controls** and **Functions** palettes more closely. The BridgeVIEW palettes contain four columns—the three columns on the left contain the functions shared between LabVIEW and BridgeVIEW, and the column on the right contains the industrial automation controls and VI functions that are specific to BridgeVIEW, such as the Tag VIs. Figure 2-1 shows the default **Controls** palette and Figure 2-2 shows the default **Functions** palette.

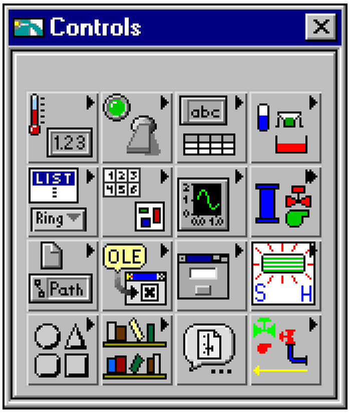


Figure 2-1. BridgeVIEW 2.1 Default Controls Palette

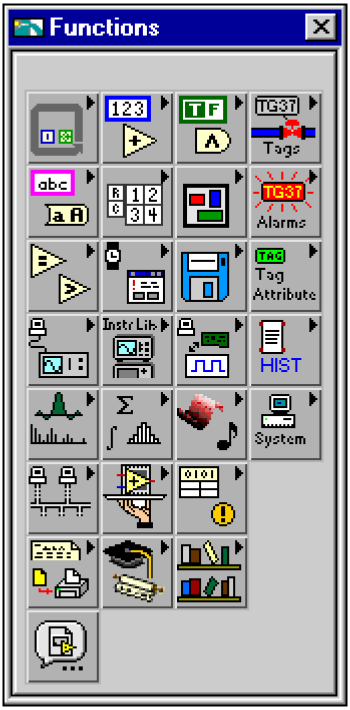


Figure 2-2. BridgeVIEW 2.1 Default Functions Palette

Automatic Tag Generation Enhancements

The BridgeVIEW Configuration Wizard, which automatically creates tags, now allows you to select specific items or sets of items from one or more servers. This enhancement gives you more control over the specific tags that are generated for each server.

To create tags using the Configuration Wizard, complete the following steps:

1. Click on the **Configuration Wizard** button in the **Tag Configuration Editor** dialog box. The **Select Items for Automatic Tag Generation** dialog box appears, as shown in Figure 2-3. The **Servers** tree view lists all servers in the system.

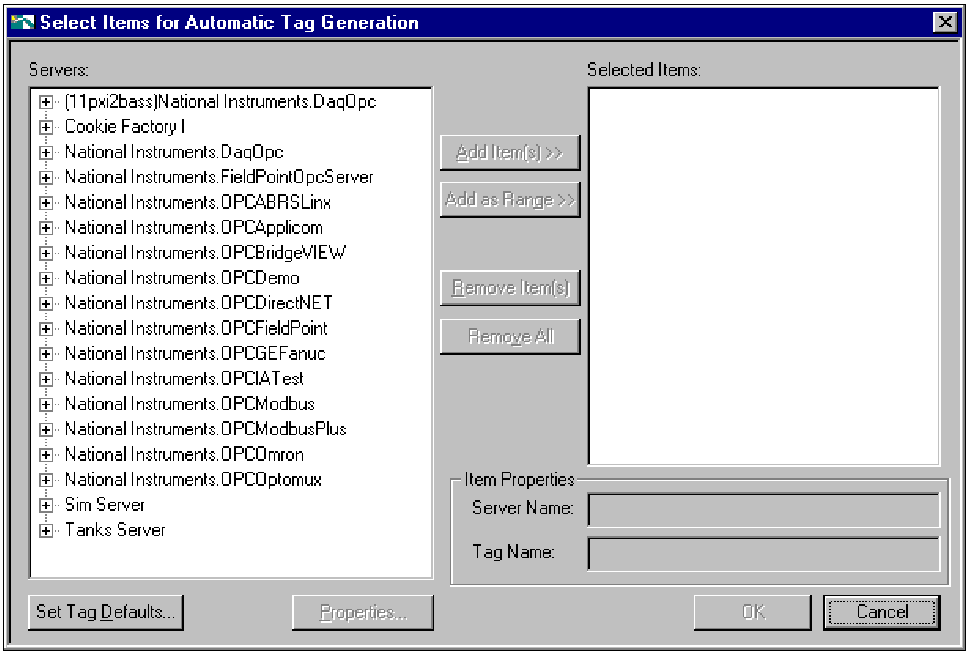

Select Items for Automatic Tag Generation
Servers:
(11pxi2bass)National Instruments.DaqOpc
Cookie Factory I
National Instruments.DaqOpc
National Instruments.FieldPointOpcServer
National Instruments.OPCABRSLinx
National Instruments.OPCApplicom
National Instruments.OPCBridgeVIEW
National Instruments.OPCDemo
National Instruments.OPCDirectNET
National Instruments.OPCFieldPoint
National Instruments.OPCGEFanuc
National Instruments.OPCIATest
National Instruments.OPCModbus
National Instruments.OPCModbusPlus
National Instruments.OPCOmron
National Instruments.OPCOptomux
Sim Server
Tanks Server
Selected Items:
Add Item(s) >>
Add as Range >>
Remove Item(s)
Remove All
Item Properties
Server Name:
Tag Name:
Set Tag Defaults...
Properties...
OK
Cancel

Figure 2-3. Select Items for Automatic Tag Generation Dialog Box

2. To show the devices and items for one of more servers, expand each server branch in the **Servers** tree view. If the **Servers** tree view lists item ranges instead of individual item names, skip to step 4.

3. Select the items that you want to create tags for and click on the **Add Item(s)** button. The Configuration Wizard moves the selected items to the **Selected Items** list box.

   If you select a branch in a hierarchical tree of items in the **Servers** tree view, all the items contained in that branch are moved to the **Selected Items** list box. If you select a server branch, all items for the server are moved to the **Selected Items** list box.

4. (Optional) Some OPC servers do not list individual item names in their hierarchical tree, but instead provide ranges for item names. This is common when the server contains a large set of items. These item ranges help you create specific item names. The format for the item ranges depends on the OPC server. If your server uses item ranges, you can generate tags from an item range, as follows:

   a. Select an item range and click on the **Add as Range** button.

   b. In the **Add Items in a Range** dialog box that appears, type the starting item name and set the number of items that you want to create.

The Configuration Wizard creates the item names, incrementing the trailing numbers in the starting item name. If you did not add a trailing number to the starting item name, the Configuration Wizard appends a zero to the first name and then increments trailing numbers in each subsequent name.

5. To remove individual items from the **Selected Items** list box, select the items and click on the **Remove Items(s)** button. To remove all of the items, click on the **Remove All** button.
6. (Optional) The Configuration Wizard uses the Tag Configuration defaults to set most of the tag parameter values. To change these defaults, click on the **Set Defaults** button.

Tip ***If you want to change the Tag Configuration defaults from the main* Tag Configuration Editor *dialog box before you run the Configuration Wizard, select* Configure»Default Parameters*.***

7. (Optional) The Configuration Wizard automatically creates I/O groups for each server and uses the Tag Configuration defaults to set the I/O group rate and deadband settings. The Configuration Wizard also sets the I/O group name to the server name. You can change the I/O group settings for each server, as follows:
   a. Select a server and click on the **Properties** button. The **Properties of Tags Generated for Device/Server** dialog box appears.
   b. Click on the **I/O Group** tab and type the I/O group settings that you want.
8. (Optional) The Configuration Wizard sets the tag name to the item name for each tag created. For non-OPC servers that have devices, the tag name contains both the device and item name if the server has more than one device. You can change the tag name format for each server, as follows:
   a. Select a server and click on the **Properties** button. The **Properties of Tags Generated for Device/Server** dialog box appears.
   b. Click on the **Tag Names** tab and set the tag name format.
9. When all the items that you want to create tags for are listed in the **Selected Items** list box, click on the **OK** button.

   The Tag Configuration Editor creates tags for each item and appends the tags to the current Tag Configuration (`.scf`) file.

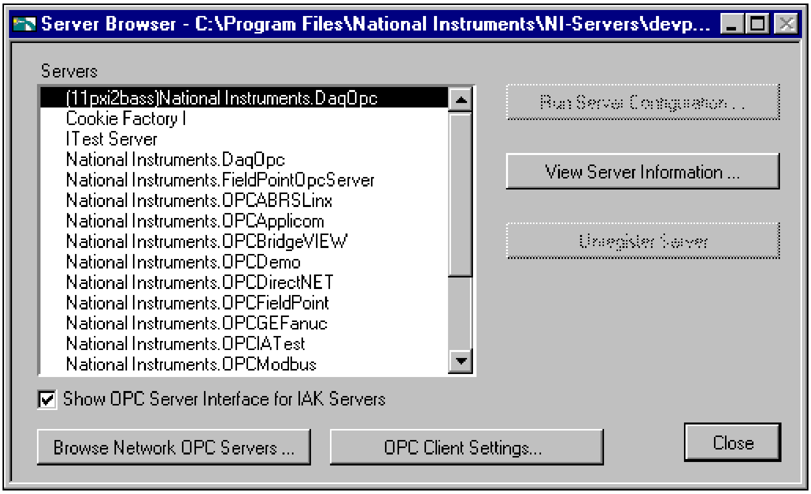
BridgeVIEW OPC Server Capability

BridgeVIEW 2.1 has OPC server capability. To install this capability, select the **OPC Server** option when you install BridgeVIEW. With this capability, any other OPC client application can read and write BridgeVIEW tags via its OPC server interface. BridgeVIEW appears in the system as an OPC server under the name `National Instruments.OPCBridgeVIEW`. When an OPC client application launches BridgeVIEW, the default Tag Configuration (`.scf`) file loads. If BridgeVIEW is already running when an OPC client attaches, the tags appear in the `.scf` file that is currently running.

The tag names in the `.scf` file appear as the OPC item names. To change the default `.scf` file within BridgeVIEW, select **Project»Tag Browser**, click on the **Browse** button, and select a `.scf` file. When you close the Tag Browser, BridgeVIEW prompts you whether to make this `.scf` file the default `.scf` file. If you want BridgeVIEW to use this `.scf` file when an OPC client launches BridgeVIEW, click on the **Yes** button.

BridgeVIEW OPC Client Changes

To configure OPC client features with BridgeVIEW, select **Projects» Server Browser**. The **Server Browser** dialog box appears, as shown in Figure 2-4.



Server Browser - C:\Program Files\National Instruments\NI-Servers\devp...

Servers

(11pxi2bass)National Instruments.DaqOpc
Cookie Factory I
ITest Server
National Instruments.DaqOpc
National Instruments.FieldPointOpcServer
National Instruments.OPCABRSLinx
National Instruments.OPCApplicom
National Instruments.OPCBridgeVIEW
National Instruments.OPCDemo
National Instruments.OPCDirectNET
National Instruments.OPCFieldPoint
National Instruments.OPCGEFanuc
National Instruments.OPCIATest
National Instruments.OPCModbus

☑ Show OPC Server Interface for IAK Servers

Run Server Configuration ...
View Server Information ...
Unregister Server
Browse Network OPC Servers ...
OPC Client Settings...
Close

Figure 2-4. Server Browser Dialog Box

Because the Industrial Automation (IAK) Servers have two interfaces that you can use in BridgeVIEW, only the **IAK Server** interface is shown in the **Servers** list in the Tag Configuration Editor. Also, the OPC names for the IAK Servers do not appear in the **Servers** list. However, you can check the **Show IAK Servers as OPC Servers** option in the **Server Browser** dialog box to show the OPC names for the IAK servers in the **Servers** list. Selecting this option allows you to use the OPC interface from BridgeVIEW. The item names available from the OPC server interface are not the same as the item names from the IAK interface.

To customize advanced options for the BridgeVIEW OPC client interface, click on the **OPC Client Settings** button. You can select any servers that require the synchronous OPC interface, change the maximum number of OPC items that display during an OPC browse, and change the length of the BridgeVIEW OPC write queue.

Dialog Box, Menu, and Window Enhancements

The following sections describe changes to BridgeVIEW dialog boxes, menu items, and the behavior or appearance of front panel objects and windows.

Scaling Front Panel Objects

With BridgeVIEW 2.1, you can set one particular front panel object or all objects on an entire front panel to scale automatically when the front panel window resizes. If you set a front panel object to scale with the window, the object resizes itself automatically in proportion to any changes in the front panel window size. Also, the other objects reposition themselves to remain consistent with their previous position on the front panel.

When you scale objects on the front panel, consider the following important information:

- BridgeVIEW scales objects automatically in the same way you resize the object manually. For example, because you can only resize numeric boxes horizontally, they can only scale horizontally.
- When BridgeVIEW scales a front panel object, the font size does not change.
- You cannot set more than one particular object on a front panel to resize automatically. You can set either *one* object or *all* objects on the front panel to resize automatically.

- After an object scales automatically, it might not scale back to its exact original size when you change the window back to its original size. However, you can select **Edit»Undo** to restore the window to its original size after BridgeVIEW scales the object automatically.
- To scale arrays, you can set scaling either on the array, or on the objects within the array, as follows:
  - When you set scaling on the array, BridgeVIEW adjusts the number of rows and columns that you can see within the array.
  - When you set scaling on the objects within the array, you always see the same number of rows and columns—though different sizes—within the array.

After you set an object to scale automatically, several regions on the front panel appear outlined by dotted lines. When you resize a window, the object selected to scale automatically repositions itself to remain consistent with its previous position within these regions. To see these regions outlined by dotted lines, do the following:

- Set one object on the front panel to scale automatically.
- Operate in edit mode.

Setting an Object to Scale

To set any front panel object to scale automatically when the front panel window resizes, complete the following steps:

1. Select the front panel object that you want to scale automatically.
2. Select **Edit»Scale Object With Window**. A checkmark appears beside this option when you select it.

Setting All Objects on a Front Panel to Scale

To set all objects in a front panel to scale automatically when the front panel window resizes, complete the following steps:

1. (Optional) If you want to set automatic scaling for all objects on the front panel, but you already set a single object on the front panel to scale, deselect automatic scaling for the particular object. To do so, select the front panel object and select **Edit»Scale Object With Window**. The checkmark beside this option disappears.
2. From the front panel of the VI, select **VI Setup**.
3. Under **Window Options**, select **Scale All Objects on Panel**.

Note *After you select* **Scale All Objects on Panel***, you cannot set or unset automatic scaling on an individual object on the front panel. If you have an individual object set to scale automatically, a dialog box appears that prompts you to choose between setting automatic scaling for either one particular object or all objects on the front panel.*

Defining a Minimum Window Size

To define a minimum window size for front panels, complete the following steps:

1. From the front panel of the VI, select **VI Setup**.
2. Under **Window Options**, locate the **Minimum Window Size** options.
3. Type the minimum **Width** and **Height**, in pixels, that you want to define for the window.

Maintaining Window Proportions with Monitor Resolution

BridgeVIEW can maintain its front panel window proportions relative to the resolution of your monitor. The percentage of the screen used by the VI front panel window stays the same, no matter what your screen resolution is. To do so, complete the following steps:

1. From the front panel of the VI, select **VI Setup**.
2. Under **Window Options**, select **Maintain Window Proportions with Monitor Resolution**.

Selecting this option maintains only window proportions. To maintain proportions of the objects, select **Scale All Objects on Panel**. You can also set a single object to scale when the window resizes. For instructions on how to do so, refer to the *Setting an Object to Scale* section earlier in this chapter.

Saving for a Previous Version

With BridgeVIEW 2.1, you can save your VIs for BridgeVIEW 2.0. Thus, you can upgrade to new versions and always have the capability to go back to your previous version, should you ever need to.

When you save a VI for the previous version, BridgeVIEW converts not just that VI, but all the VIs in its hierarchy, excluding `vi.lib` files. To save a hierarchy of VIs for a previous version of BridgeVIEW, complete the following steps:

1. From the VI at the top of your hierarchy, select **File»Save with Options**. The **Save with Options** dialog box appears.
2. Select the **Save for BridgeVIEW 2.0.x** option to save your VIs for the previous version.
3. Click on the **Save** button. The **Choose a Directory** dialog box appears.
4. Select the directory where you want to save the VIs.
5. Click on the **Save** button.

A VI occasionally uses functionality that is not available in the previous version. In such cases, BridgeVIEW converts the other VIs and reports what could not be converted. After the conversion, the report appears in the **Save for BridgeVIEW 2.0.x Warnings** dialog box. To acknowledge these warnings and close the dialog box, click on the **OK** button. To save the warnings to a text file, click on the **Save** button.

Accessing Recently Opened Files

BridgeVIEW 2.1 gives you easy access to recently opened files, instead of searching through various directories. To find a file that you opened recently, select **File»Recently Opened Files** and select a file to open it. The **Recently Opened Files** list includes the following file types:

- VI (`*.vi`)
- Control (`*.ctl`)
- VI template (`*.vit`)
- Control template (`*.ctt`)

The files are listed in chronological order, with the most recently opened file listed first. If the list is empty, the **Recently Opened Files** option appears disabled. The **Recently Opened Files** list displays up to 10 file names.

Note ***When two or more files have the same file name but reside in different directories, the* Recently Opened Files *list displays the full path of the file.***

Searching in BridgeVIEW

BridgeVIEW 2.1 includes new options designed to help you find VIs, subVIs, and text references more easily.

For more information about the **Find** dialog box, refer to the *Find Dialog Box* topic in the *BridgeVIEW Online Reference,* or refer to the *Finding VIs, Objects, and Text* section in Chapter 3, *Using SubVIs*, of the *G Programming Reference Manual*.

Find Dialog Box

To display the **Find** dialog box, select **Project»Find** or press <Ctrl-f>.

The **Find** dialog box now includes the **Include SubVIs** option in the **Search Scope** section. Select the **Include SubVIs** option to search for an object or text within the subVIs of the VI you currently have open. By default, this option is disabled.

Also, the **Search in Hierarchy Window** and **Search VIs in vi.lib** options are now named **Include Hierarchy Window** and **Include VIs in vi.lib**, respectively. The functionality of these two options has not changed.

Find All Instances and Search Results Dialog Box

To make it easier to search for all instances of a VI, BridgeVIEW now includes the **Find All Instances** option. You can search for all instances of a subVI or type definition (constant, control, or indicator). Also, you can search for all instances of a connector pane or hierarchy window object.

To find all instances of an object, right-click the object and do the following:

- If you are searching for a reference to a type definition, select **Find»All Instances**. For more information about type definitions, refer to the *Type Definitions* section in Chapter 24, *Custom Controls and Type Definitions*, of the *G Programming Reference Manual*.
- If you are searching for a reference to a subVI, right-click the VI icon in the block diagram, hierarchy window, or connector pane, and select **Find All Instances** from the list that appears.

If BridgeVIEW finds one or more references to the object, the **Search Results** dialog box appears. Otherwise, the object remains highlighted.

New VIs, Functions, and Controls

The following sections describe the new functionality in BridgeVIEW 2.1 for advanced mathematics, picture control, report generation, and image management.

Changes to Controls and Indicators

BridgeVIEW 2.1 improves the appearance of controls and indicators. Some controls in BridgeVIEW 2.1 look slightly different than in previous versions when you drop them on the front panel. However, controls on existing VIs that you created in previous versions are not updated. You will notice improvements such as the addition of color to many controls, but each of these controls works the same way. The following sections, *Labels* and *Dialog Controls*, describe the changes to control and indicator labels and the new dialog controls.

Labels

By default, when you create a new control or indicator, its name label includes the name of the type of control or indicator (such as *Slide* or *String*) and, if necessary, a number distinguishing it from other controls and/or indicators of the same kind. For example, if you place a slide control on the front panel, it is labeled *Slide*. If you place another slide control on the front panel, it is labeled *Slide 2*.

When you replace a control or indicator that has a default label with an object of a different type, the control or indicator is renamed as well. For example, if you replace a slide control labeled *Slide 2* with a knob control, the knob is labeled *Knob*. If a slide or control already exists on the front panel with the label *Knob*, the new knob is labeled *Knob 2*.

When you drop an object on the front panel, its name label is highlighted to allow you to type a different name. With BridgeVIEW 2.1, if you want to hide a name label right after you place it on the front panel, right-click the control or indicator and select **Show»Label**.

By default, name labels appear transparent. To display a name label in a raised box, select **Edit»Preferences**, select **Front Panel** from the drop-down menu, and deselect the **Use transparent name labels** option.

Dialog Controls

With BridgeVIEW 2.1, all the dialog controls are grouped together in a new **Dialog** palette, as shown in Figure 2-5. To view this palette, select **Controls»Dialog**.

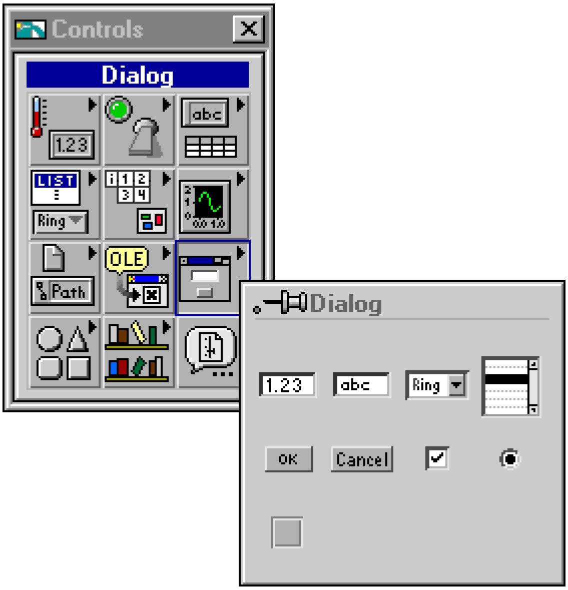


Figure 2-5. Dialog Palette

This palette includes the following dialog controls. These controls still exist in the palettes from previous versions of BridgeVIEW.

- Dialog Numeric Control
- Dialog String Control
- Dialog Ring
- Dialog Button
- Cancel Button
- Dialog Checkbox
- Dialog Radio Button

Also, this palette includes two new controls: the *Dialog Listbox* control and the *Dialog Recessed Frame* control. The Dialog Listbox control behaves like other list boxes, but functions and appears like the tabbed highlighting and bordering of the native system controls on Windows. The Dialog

Recessed Frame control behaves like other decoration objects in the **Decorations** palette. You can use it as a grouping border around other dialog controls.

3D Graph Controls

BridgeVIEW 2.1 adds a new way to represent data on your front panel—the 3D graph. The 3D graph uses ActiveX technology and new VIs that handle three-dimensional representation. You can set parameters for the 3D Graphs VIs to change behavior at run time, including setting basic, axes, grid, and projection properties.

For more information about the 3D Graphs VIs, refer to the *3D Graphs VIs* topic in the *BridgeVIEW Online Reference*.

Examples of the 3D Graphs VIs are in the `Examples\General\graphs\3dgraph.llb` directory.

Note ***The 3D Graphs VIs are available in the BridgeVIEW Full Development System and Professional Development System.***

Enhancements to Property and Invoke Nodes

With BridgeVIEW 2.1, VI Server and Application Class property and invoke nodes do not always need a wired refnum input or output. This makes it easier to program many common VI Server functions because you can drop fewer diagram objects.

If you do not wire inputs or outputs, BridgeVIEW uses default values. The default input value for the Application Class is your local BridgeVIEW application. The default value for the Virtual Instrument Class is your current VI—that is, the VI in which you have placed the property or invoke node.

Enhancements to VI Server Properties in Reserved VIs and Run-time Systems

In BridgeVIEW 2.1, the VI Server was enhanced so that you can set many more properties in run-time versions or while a VI is running. Also, the BridgeVIEW Help Window now describes each Virtual Instrument Class property and its limitations, if any. To open the Help Window, select **Help»Show Help**. Move your cursor over the property portion of any Virtual Instrument Class property node, and the description of that property appears in the Help Window.

You can set any of the following properties on running VIs and in run-time versions or while a VI is running:

- **AutoLogging»Log File Path**
- **AutoLogging»Log at Finish**
- **AutoLogging»Print at Finish**
- **Edit Mode on Open**
- **Execution»Close after Call**
- **Execution»Show Front Panel on Call**
- **Front Panel Window»Allow Runtime Popup**
- **Front Panel Window»AutoCenter**
- **Front Panel Window»Closeable**
- **Front Panel Window»Highlight Return Button**
- **Front Panel Window»Is Dialog**
- **Front Panel Window»Resizeable**
- **Front Panel Window»Show Menu Bar**
- **Front Panel Window»Show Scroll Bars**
- **Front Panel Window»Size to Screen**
- **Front Panel Window»Title Bar Visible**
- **Help»Document Path**
- **Help»Document Tag**
- **Tool Bar»Show Abort Button**
- **Tool Bar»Show Free Run Button**
- **Tool Bar»Show Run Button**
- **Tool Bar»Visible**
- **VI Description**

You also can set any of the following properties while a VI is running, but not in run-time versions:

- **Execution»Show Front Panel on Load**
- **Execution»Run When Opened**
- **Execution»Suspend on Call**
- **History»Always Add Comments at Save**
- **History»Prompt for Comments at Close**
- **History»Prompt for Comments at Save**

- **History»Record Application Comments**
- **History»Use Defaults**

Ring Enhancements

All ring and enum constants include a down arrow, which distinguishes them from numeric constants. Also, rings and enums on panels or block diagrams have scroll bars when the menus include many items. BridgeVIEW 2.1 menus, like list boxes, now support type completion. That is, when you type the first few characters, BridgeVIEW finds a matching item in the list. Use the <Tab> and <Shift-Tab> keys to move to the next and previous matching item, respectively.

IVI Instrument Drivers and ActiveX

Some parameters for ActiveX and IVI take a discrete list of valid values. Previously, when building ActiveX and IVI applications, you used numeric values to set parameters for properties and methods that control devices or programs. You had to type a particular numeric value in a numeric control or constant, which required you to know which numeric values corresponded to specific settings.

With BridgeVIEW 2.1, you do not need to know these values. You can select from a descriptive name in a ring to set parameter values. The selections available in the ring depend on the refnum passed to the node. Figures 2-6 and 2-7 show examples of these numeric and ring constants.

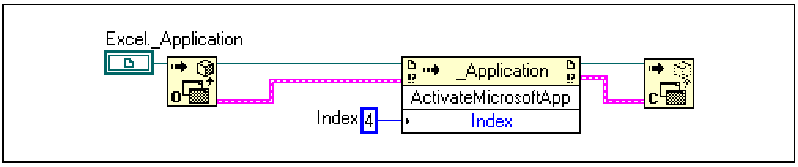


Figure 2-6. Setting a Data Value with a Numeric Constant

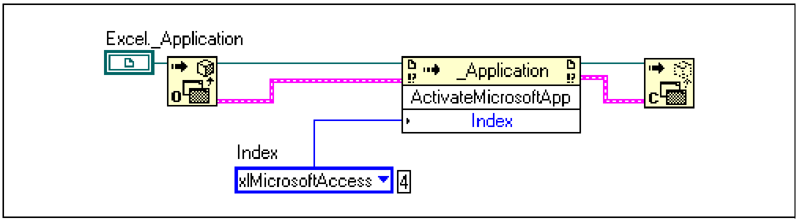


Figure 2-7. Setting a Data Value with a Ring Constant

In both examples, BridgeVIEW accesses the Microsoft Excel application and invokes a method. The **Index** parameter has several options: *MicrosoftAccess*, *MicrosoftFoxPro*, *MicrosoftMail*, *MicrosoftPowerPoint*, *MicrosoftProject*, *MicrosoftSchedulePlus*, and *MicrosoftWord*.

Previously, you had to know that the **Index** parameter corresponding to the *MicrosoftAccess* option has a data value of 4. With BridgeVIEW 2.1, you can select the *MicrosoftAccess* option from the drop-down menu in the ring constant.

To access the ring constant when you build an ActiveX or IVI application, right-click the parameter that accepts data values and select the **Create Constant** option. To see the corresponding numeric data value, right-click the ring constant and select **Show»Digital Display**.

ActiveX Enhancements

The following sections describe the ActiveX enhancements in BridgeVIEW 2.1.

Ring Enhancements

With BridgeVIEW 2.1, you do not have to use numeric values when setting parameters for properties and methods that control devices or programs. You can use a ring constant as well. For more information, refer to the previous section, *IVI Instrument Drivers and ActiveX*.

Support for ActiveX Events

With BridgeVIEW 2.1, you can access events associated with ActiveX objects.

In addition to accessing the properties and methods associated with an ActiveX object when you build an ActiveX server VI, you can also access the events. You can embed a control on a VI front panel, then execute code based on an event that occurs. For example, you can place a tree view control that lists the contents of a directory. Using events, you can specify that when the user double-clicks on a particular file, the contents of that file open.

To build VIs that accept events, use the ActiveX Event functions in the **Functions»Communications»ActiveX»ActiveX Events** palette.

Working with ActiveX Events

Use this basic procedure to create a VI designed to create and wait on an ActiveX event queue, then destroy the event queue. An *event queue* is a tag that corresponds to an internal list of events a control receives.

Note ***If you generated an Automation refnum using an Automation Client function, skip step 1.***

1. On the front panel, open the **ActiveX** palette and select the **Container** control.
2. Use the Create ActiveX Event Queue VI to create an ActiveX event queue, as follows:
   a. Wire the Automation refnum from the container terminal or an Automation function to the Create ActiveX Event Queue VI.
   b. Place the Wait On ActiveX Event VI on the block diagram.
   c. Wire the event queue to the Wait On ActiveX Event VI.
3. To destroy the event queue, use the Destroy ActiveX Event Queue VI.

ActiveX Event Functions

You can use the other ActiveX Event functions to pass an event queue from one task to another separate task. To access the ActiveX Event palette shown in Figure 2-8, select **Functions»Communications»ActiveX» ActiveX Events**.

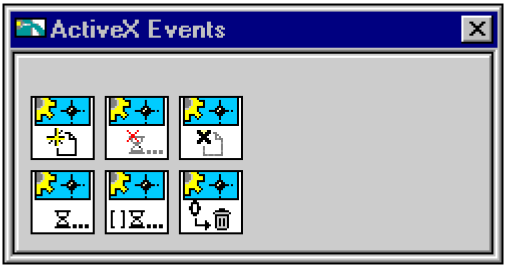


Figure 2-8. ActiveX Events Palette

For more information about the ActiveX Event functions, refer to the *ActiveX Event Functions* topic in the *BridgeVIEW Online Reference*.

Enhancements to ActiveX Automation

The Automation Open function now includes the optional **machine name** string input. Use **machine name** to specify the computer that you want to use to open an Automation Server object. If you do not specify a **machine name**, BridgeVIEW creates the object on the local machine. For more information, refer to the *Open Automation Refnum* description in Chapter 51, *ActiveX Automation Functions*, of the *LabVIEW Function and VI Reference Manual*.

BridgeVIEW 2.1 also supports enumerations in the ActiveX Invoke and Property Nodes. When you right-click an Invoke Node or Property Node terminal that is part of an enumeration or is a constant, you create a ring with the available enumeration values. For more information, refer to the *Ring Enhancements* section earlier in this chapter.

HiQ and MATLAB Functionality

The HiQ and MATLAB software packages help you organize and visualize real-world math, science, and engineering problems. You can use HiQ and MATLAB to express numeric formulas professionally. You can use BridgeVIEW to load and edit HiQ and MATLAB scripts into your block diagram code.

Note ***To use this new feature, you must install HiQ 4.1 or MATLAB 5.0 or later. You can install HiQ 4.1 from the BridgeVIEW CD. HiQ and MATLAB functionality is available in the BridgeVIEW Full Development System and Professional Development System.***

The new HiQ and MATLAB script nodes that work similarly to the formula node are in the **Functions»Mathematics»Formula** palette. You can place a script node on your block diagram and enter a script according to the syntax of HiQ or MATLAB. BridgeVIEW then communicates with that script server engine. When you create inputs and outputs on the script node, those values are passed between HiQ or MATLAB and BridgeVIEW. If you already have a script written, you can import it from HiQ or MATLAB. For more information, refer to the *Importing or Exporting a Script* section later in this chapter.

Although the basic functionality of the script node is similar to that of the formula node, the script node can handle more data types than the formula node. This script node supports the same HiQ or MATLAB data types that ActiveX supports.

Note ***Because of the nature of the HiQ and MATLAB script languages, you must select the BridgeVIEW data type for each terminal. For more information, refer to the*** *Configuring the Data Type of a Terminal* ***section later in this chapter.***

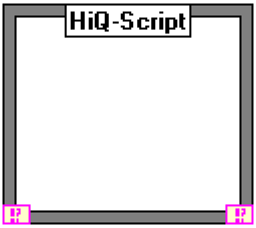
The two following sections, *Creating a HiQ Script* and *Creating a MATLAB Script*, contain instructions on how to create a script that does the following:

1. Generates a matrix of random values
2. Plots that information to a graph
3. Displays the graph in either product
4. Presents the generated random values on your front panel in BridgeVIEW

Creating a HiQ Script

To create and run a VI that uses a HiQ script, complete the following steps:

1. From the block diagram, select **Functions»Mathematics»Formula» HiQ-Script**. Place the node on the block diagram, and size it according to the amount of script that you want to include in the window. The HiQ script node appears, as follows.



2. With the operating tool, enter the script in the HiQ script node. For example, the following example HiQ script creates a matrix of random values, plots that information to a graph, and displays the matrix in HiQ:

```
a = random({50, 50});

g = createGraph(a);

createView(g, true);
```

You can use a HiQ script window to edit, compile, and run your script directly from HiQ to ensure that your script behaves properly, as follows. To access HiQ context-sensitive online help, place the cursor inside any function and press <F1>. You can use this information to help you build your script.

3. Right-click the HiQ script node and select **Edit In Server**. HiQ launches and a script window displays your HiQ script.
4. (Optional) Within HiQ, right-click and select **compile** from the pop-up menu. A message window displays any compile-time errors. However, you do not need to compile the script explicitly; HiQ compiles the script automatically when you run it.
5. Right-click and select **Run** from the pop-up menu. A message window displays any run-time errors.
6. Close the HiQ window to update and return to the HiQ script node on the BridgeVIEW block diagram.
7. To add inputs and outputs for variables, right-click the right side of the node frame and select **Add Input** or **Add Output**. Type a to add an output for the **a** variable that is included in your HiQ script. By default, your node already includes one input and one output terminal for the **error in** and **error out** parameters.
8. Verify the data type of the inputs and/or outputs. The error-checking input and output are already the correct data type. In HiQ, the default data type for any new input or output is **Real**. Right-click the **a** output and select **Choose Type**. From the submenu that appears, select **Real Matrix** from the available data types—**Integer**, **Real**, **Complex**, **Text**, **Integer Vector**, **Real Vector**, **Complex Vector**, **Integer Matrix**, **Real Matrix**, and **Complex Matrix**.
9. To create controls and indicators for each input and output, right-click the **a** output terminal and select **Create Indicator**. Right-click the **error out** output terminal and select **Create Indicator**.

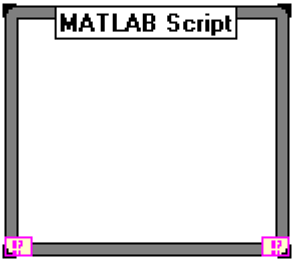
   Indicators for **a** and **error out** appear on the front panel, and terminals appear wired to these outputs on the block diagram.
10. Go to the front panel. Resize your **a** indicator so that you can see the generated numbers when you run the VI.
11. Run the VI. BridgeVIEW launches HiQ and a new HiQ window, labeled *G in Notebook1*, displays the matrix. The values that make up this matrix are displayed in the **a** indicator of your front panel in BridgeVIEW.

    The second parameter in the CreateView function specifies whether to pause execution of the HiQ script while the view is visible. The HiQ script does not run until you click on the **Continue** button in the HiQ window to dismiss the view.
12. Change the **true** parameter to **false** and re-run the VI to see the difference.

Creating a MATLAB Script

To create and run a VI that uses a MATLAB script, complete the following steps:

1. From the block diagram, select **Functions»Mathematics»Formula» MATLAB Script**. Place the node on the block diagram, and size it according to the amount of script that you want to include in the window. The MATLAB script node appears, as follows.

2. With the operating tool, enter the script in the MATLAB script node. For example, the following example MATLAB script creates a matrix of random values, plots that information to a graph, and displays the matrix in MATLAB:

   ```
   a=rand(50)

   surf(a)
   ```

3. To add inputs and outputs for variables, right-click the right side of the node frame and select **Add Input** or **Add Output**. Type `a` to add an output for the **a** variable that is included in your MATLAB script. By default, your node already includes one input and one output terminal for the **error in** and **error out** parameters.
4. Verify the data type of the inputs and/or outputs. The error-checking input and output are already the correct data type. In MATLAB, the default data type for any new input or output is **Real**. Right-click the **a** output and select **Choose Type**. From the submenu that appears, select **Real Matrix** from the available data types—**Real**, **Complex**, **Real Vector**, **Complex Vector**, **Real Matrix**, or **Complex Matrix**.
5. To create controls and indicators for each input and output, right-click the **a** output terminal and select **Create Indicator**. Right-click the **error out** output terminal and select **Create Indicator**.

   Indicators for **a** and **error out** appear on the front panel and terminals appear wired to these outputs on the block diagram.

6. Go to the front panel. Resize your **a** indicator so that you can see the generated numbers when you run the VI.
7. Run the VI. BridgeVIEW launches MATLAB and a new MATLAB window, labeled *Figure No. 1*, displays the matrix. The values that make up this matrix are displayed in the **a** indicator of your front panel in BridgeVIEW.

Importing or Exporting a Script

To import a script into a script node in BridgeVIEW, complete the following steps:

1. Right-click the HiQ or MATLAB script node.
2. Select the **Import** option.
3. In the **File** dialog box that appears, select the file that you want to import and click on the **Open** button. The script text appears in your node.

To export a script to BridgeVIEW from MATLAB or HiQ, complete the following steps:

1. Right-click the HiQ or MATLAB script node.
2. Select the **Export** option. The **Save Script as** dialog box appears.
3. Type the name that you want to save your file as or select the file that you want to overwrite.
4. Click on the **Save** button.

Note ***HiQ and MATLAB script files are text files. Text files usually have the*** `.txt` ***extension. However, HiQ files have a*** `.hqs` ***extension and MATLAB files have a*** `.m` ***extension.***

Selecting a Script Server

To change the server behind your script node, complete the following steps:

1. Right-click the HiQ or MATLAB script node.
2. Select **Choose Script Server»HiQ-Script** or **Choose Script Server» MATLAB Script**, depending on whether you want to use HiQ or MATLAB.

Some data types in HiQ are not recognized by MATLAB, and vice versa. If you change the script server from one of these products to the other after you assigned data types, these data types do not change automatically; the VI will be broken. When you click on the broken arrow in the toolbar of your VI, the **Error List** window reports these incorrect data types.

Scrolling through a HiQ or MATLAB Script

Like with a formula node, you can display a scroll bar within your script node by right-clicking the node and selecting **Show»Scrollbar**. A scroll bar appears on the right side of your script node.

Configuring the Data Type of a Terminal

Both HiQ and MATLAB are loosely typed script languages and do not determine the data type of a variable until after the script executes. Therefore, BridgeVIEW cannot determine a variable type in **Edit** mode. However, BridgeVIEW does query the script server to find out possible data types, and allows you to select the BridgeVIEW data type for each terminal.

Note ***If you do not correctly configure a variable data type, BridgeVIEW produces either an error or incorrect information at run time.***

To change the data type of an input or output terminal on a script node, complete the following steps:

1. Right-click the terminal of the input or output. A pop-up menu appears.
2. Select **Choose Type**. A list of the data types that are available for your script appears, depending on whether you use HiQ or MATLAB.
3. Select a data type.

Note ***BridgeVIEW recognizes all the data types that HiQ and MATLAB can use, although they might be named differently.***

The following table shows the BridgeVIEW data types and their corresponding data types in HiQ and MATLAB.

Table 2-1. HiQ and MATLAB Data Types in BridgeVIEW

BridgeVIEW Data Type	HiQ Data Type	MATLAB Data Type
I32	Integer	N/A
DBL	Real	Real
abc	Text	N/A
[I32]	Integer Vector	N/A
[DBL]	Real Vector	Real Vector
[I32]	Integer Matrix	N/A
[DBL]	Real Matrix	Real Matrix
CDB	Complex	Complex
[CDB]	Complex Vector	Complex Vector
[CDB]	Complex Matrix	Complex Matrix

Debugging a HiQ or MATLAB Script

To debug your script, use the following programming techniques:

- Before you import your script into BridgeVIEW, write and run it within the native engine (HiQ or MATLAB) to test and debug it. In HiQ, right-click the script node and select **Edit In Server** to debug in a native HiQ script window.
- Verify your data types. When you create a new input or output, make sure that the data type of the terminal is correct. Also, create controls and indicators for your inputs and outputs so that you can monitor what values are being passed between BridgeVIEW and the native engine. For more information, refer to the *Configuring the Data Type of a Terminal* section earlier in this chapter.
- Use the error-checking parameters to debug. Before you run any VI, create an indicator for the **error out** terminal on a script node, so that you can view the generated error information at run time.

Error Codes

The following table describes error messages that you might receive at run time when you use a script node.

Table 2-2. Script Node Run Time Error Messages

Error Code	Error Code Message	Description
1046	`edScriptCantInitServer`	BridgeVIEW did not initiate a new session with the server.
1047	`edScriptCantSetValue`	BridgeVIEW did not set the value of a variable to the server.
1048	`edScriptCantGetValue`	BridgeVIEW did not retrieve the value of a variable from the server.
1049	`edScriptCantSetScript`	BridgeVIEW did not set a script to the server.
1050	`edScriptExecError`	BridgeVIEW encountered a problem while executing the script. The server returns a string to report the problem.

Integration of Mathematics and Signal Processing VIs

BridgeVIEW 2.1 includes more VIs designed to help you solve advanced mathematics problems. You can use these VIs to manipulate mathematical formulas on the BridgeVIEW front panel, and then graphically assemble your problem-solving program on the BridgeVIEW block diagram.

Previously, most of this functionality was available only with the G Math Toolkit. This toolkit was integrated into the BridgeVIEW Full Development System.

Integrating these VIs restructures the BridgeVIEW **Functions** palette. The new **Mathematics** palettes includes the new Mathematics VIs, as well as some of the VIs and functions formerly found in the **Analysis** and **Structures** palettes.

The **Analysis** palette from previous versions of BridgeVIEW has been replaced with the **Signal Processing** palette. For information about the Signal Processing VIs, refer to the *Signal Processing Palette* section later in this chapter.

Mathematics Palette

For function and reference information about the Mathematics VIs and functions, refer to the *Mathematics VIs* topic in the *BridgeVIEW Online Reference*.

The following table lists the subpalettes from the **Functions»Mathematics** palette, and lists the location of the VIs or functions in previous versions of BridgeVIEW.

Table 2-3. Current and Previous Locations of Mathematics VIs

Mathematics Subpalettes	Previous Location
Formula	G Math Toolkit and **Structures** palette
1D and 2D Evaluation	G Math Toolkit
Calculus **Differential Equations** subpalette	G Math Toolkit
Probability and Statistics **Probability** subpalette **Analysis of Variance** subpalette	**Analysis** palette
Array Operations	**Analysis** palette
Curve Fitting	**Analysis** palette
Linear Algebra **Advanced Linear Algebra** subpalette **Complex Linear Algebra** subpalette	**Analysis** palette
Zeros	G Math Toolkit and **Analysis** palette
Optimization	G Math Toolkit
Numeric Functions	G Math Toolkit

Examples of the Mathematics VIs are in the `Examples\Math\math.llb` directory.

Signal Processing Palette

For function and reference information about the Signal Processing VIs, refer to the *Signal Processing VIs* topic in the *BridgeVIEW Online Reference*.

The following table lists the subpalettes from the **Functions»Signal Processing** palette, and lists the location of the VIs or functions in previous versions of BridgeVIEW.

Table 2-4. Current and Previous Locations of Signal Processing VIs

Signal Processing Subpalettes	Previous Location
Signal Generation	**Analysis** palette
Time Domain	**Analysis** palette, **Signal Processing** subpalette
Frequency Domain	G Math Toolkit and **Analysis** palette—combines the FFT and Power Spectrum VIs from the **Analysis** palette **Signal Processing** subpalette and the VIs from the G Math Toolkit **Transforms** palette
Measurement	**Analysis** palette
Filters	**Analysis** palette
Windows	**Analysis** palette

Examples of the Signal Processing VIs are in the `Examples\Math\sig_proc.llb` directory.

Integration of the Picture Control VIs

With BridgeVIEW 2.1, you can display complex images and graphs using functionality that was previously included only in the Picture Control Toolkit. Because the Picture Control Toolkit was integrated into the BridgeVIEW Full Development System, you can use any of the VIs in the Picture Control VI Library.

The Picture Control VIs include drawing operations that you can use to create diagrams and dynamically build images. You can create new front panel displays such as specialized bar graphs, pie charts, or Smith charts. You can also display and animate arbitrary objects such as robot arms, test equipment, or a two-dimensional display of a real-world process. For more information about these VIs, refer to the *Graphics and Sound VIs* topic in the *BridgeVIEW Online Reference*.

Examples of the Picture Control VIs are in the `Examples\Picture` directory.

Sound VIs

You can use the Sound VIs to integrate sound into your VIs. For example, you can create a beep to alert the user of an error, play a `.wav` file, and read and write sound data. To access the Sound VIs, select the **Functions»Graphics & Sounds»Sound** palette from the block diagram.

For more information, refer to the *Sound VIs* topic in the *BridgeVIEW Online Reference*. Examples of the Sound VIs are in the `Examples\Sound\sndExample.llb` directory.

Generating Reports in BridgeVIEW

You can create reports of any text-based information that a VI generates or that the user enters into a string parameter. You can also create a report from an array of two-dimensional numbers. For example, if you develop a BridgeVIEW program that tests functions, you can create a text-based report of which functions passed and which failed.

To create reports using the Report Generation VIs, select the **Functions»Report Generation** palette.

You can use the Report Generation VIs to do the following:

- Set up the report headers and footers, which can include date and time stamps
- Set character font, size, style, and color
- Set the report margins and tabs
- Define the information that appears on a particular line or page of the report
- Set the page orientation of the report—lengthwise or widthwise
- Include text from other files in the report
- Clear information from an existing report, to re-use the report's formatting in a new report
- Automatically print the report
- Delete the report after it is printed, which saves memory

Suggestions for Generating Reports in BridgeVIEW

Make sure that any information you want to print is formatted into a string, unless you want to print a 2D array of numbers as a table. If you have a lot of information and you want to write it to a report, you will likely use one or more of the String functions. For information about the String functions and how they format data, refer to Chapter 6, *String Functions,* of the *LabVIEW Function and VI Reference Manual*.

Easy Text Report VI Overview

The Easy Text Report VI creates a basic report from a block of text. Use this VI if you do not need to have control over all aspects of the report, and if you do not have a lot of varying data or a numeric table in the report.

You can use the Easy Text Report VI to specify the text font, set up the headers and footers, set margins, specify a printer, and set the page orientation. However, you cannot control where information is placed, append information from another file, or clear the report of font styles, headers and footers, or text. The Easy Text Report VI deletes the report after it is printed, which saves memory.

For more information, refer to the Easy Text Report VI description in the *Report Generation VI Descriptions* section later in this chapter.

Tip ***For an example of how you can use the other VIs in the* Report Generation *palette to create a report, refer to the block diagram of the Easy Text Report VI.***

Report In/Report Out Parameters

The **Report in** and **Report out** parameters link the VIs used to create a report. All Report Generation VIs have both these parameters, with the following exceptions:

- Easy Text Report VI has neither of these parameters.
- Dispose Report VI has only the **Report in** parameter.
- New Report VI has only the **Report out** parameter.

Use these parameters to link a report to the VIs that control the report appearance, data, and printing. You can use these parameters to add and control several different features on one report.

Tokens

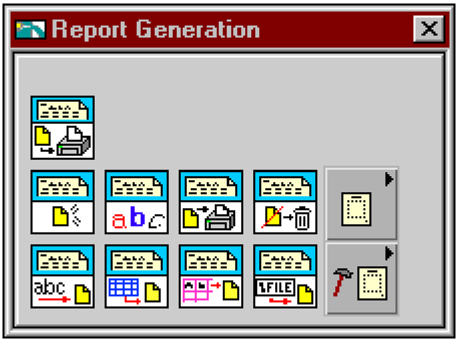
Tokens are strings you can enter that generate information automatically. You can use tokens to number pages and place a timestamp on a report, especially in the headers and footers. In any parameter that takes a string, you can enter several tokens. For example, if you are using the Easy Text Report VI, entering the token `<page>` in one of the footer strings prints a page number in the footer of every page of the report. The following table lists some useful tokens.

Table 2-5. Token Descriptions

Token	Description
`<page>`	Current page number
`<pages>`	Total number of pages
`<pagenofm>`	Current page number along with the total number of pages in the report. Example: 7 of 30
`<shortdate>`	Current date in the form *xx*/*xx*/*xx*. The month, date, and year order defaults to the date settings of your operating system. Example: 10/5/98
`<longdate>`	Current date in the form *Day*, *Month and Date*, *Year.* The month, date, and year order defaults to the date settings of your operating system. Example: Monday, October 05, 1998
`<time>`	Current time in the form *Hour:Minute:Second AM/PM*. This token defaults to the clock settings on your system, whether it uses a 12-hour or 24-hour clock. Example: 1:58:22 PM.

Report Generation VI Descriptions

This section describes the Report Generation VIs, including each parameter and its data type. To access these VIs from the block diagram, select **Functions»Report Generation**.



The **Report Generation** palette has two subpalettes—**Report Layout** and **Advanced Reports**.

Examples of the Report Generation VIs are in the `Examples\Reports\TextReportExample.llb` directory.

Report Generation Parameter Descriptions

This section describes the parameters used by the Report Generation VIs.

TF **append on new line? (F)**, if the `True` value is selected, appends the information onto a new line in the report. The default value is `False`.

abc **center footer text** is the information you want to appear in the center portion of the footer.

abc **center header text** is the information you want to appear in the center portion of the header.

[abc] **Column Headers** determines how each column is labeled in the table.

DBL **Column width** defines the width of each column in the report's table. The value you enter is in inches or centimeters, depending on the settings you enter in **measurement system** or **measurement system for columns**. The default value is 1.

error in

error out

file path is the path of the text file from which you want to pull information into your report. You must include the file's path in this parameter.

Font Settings allows you to set the font of your report.

Charset sets the character set used in the report (such as the set for a specific language).

Weight sets the characters' weight.

Name specifies the font used in the report. You can type in the name of any font available on your system.

Note *The name you enter must match the font name exactly.*

Size specifies the size of the font in number of points.

Font Settings indicates what font settings currently are used in a report.

Charset indicates the character set used in the report (such as the set for a specific language).

Weight indicates how dark bold characters appear.

Name indicates the font used in the report.

Size indicates the size of the font in number of points.

Footers allows you to set up what information appears in the footer of each page of the report. You can type in text, leave the parameters blank, or type in a token (see *Tokens*, earlier in this chapter, for information on the tokens you can enter).

left is the information you want to appear in the left side of the footer.

center is the information you want to appear in the center portion of the footer.

right is the information you want to appear in the right side of the footer.

format string determines how the numbers appear in each cell of the table, such as `01`, `1`, `1.0`, or `1.00`.

It is as defined by Format & Append:

```
[Str]%[-][0][Width][.Precision]Conversion[Str]
```

where `-` causes left justification and `0` pads with zeros.

Conversions: d (decimal), x (hexadecimal), o (octal), f (fractional), e (scientific), or g (scientific).

For example, if you wanted the numbers to appear in each as 1.00, you would use the default value: %.2f.

Refer to the *Format & Append and Format Strings Overview* topic in the *BridgeVIEW Online Reference* for more information on values you can enter in **format string**.

Headers allows you to set up what information appears in the header of each page of the report. You can type in text, leave the parameters blank, or type in a token (see *Tokens*, earlier in this chapter, for information on the tokens you can enter).

left is the information you want to appear in the left side of the header.

center is the information you want to appear in the center portion of the header.

right is the information you want to appear in the right side of the header.

left footer text is the information you want to appear in the left side of the footer.

left header text is the information you want to appear in the left side of the header.

Margins set where the information is placed on each page of the report. You set the margins with the following parameters:

left or **LeftMargin** sets the report's left margin in inches or centimeters.

DBL **right** or **RightMargin** sets the report's right margin in inches or centimeters.

DBL **top** or **TopMargin** sets the report's top margin in inches or centimeters.

DBL **bottom** or **BottomMargin** sets the report's bottom margin in inches or centimeters.

U16 **measurement system** has three options: **US**, which sets the margins in inches; **Metric**, which sets the margins in centimeters; and **Default**, which sets the margins in the measurement system set up on your computer.

U16 **measurement system for columns** determines whether the value you enter in **column width** is in inches or centimeters. There are three options: **US**, which sets the columns in inches; **Metric**, which sets the columns in centimeters; and **Default**, which sets the columns in the measurement system set up on your computer.

I32 **Number of Copies (1)** specifies how many copies of the report you want to print. If you do not specify a number, one copy is printed.

[DBL] **Numerical Data** is the numerical information included in the table. The information must be a 2D array.

U16 **orientation** specifies how the report appears when it is printed. You can choose from **portrait**, which prints the report widthwise, or **landscape**, which prints the report lengthwise.

abc **Printer Name** specifies the name of the printer that you want to print this report. If you do not enter a printer name, then this VI uses the default printer set up on your computer. If you do specify a printer name, you must have that printer configured for use with your computer. If you do not specify a printer, you must have a default printer specified on your system.

Report in links a report to the VIs used to control a report's appearance, data, and printing. This holds the report before the VI generates new data for the report.

Report out links a report to the VIs used to control a report's appearance, data, and printing. This holds the report after the VI generates new data for the report.

abc **right footer text** is the information you want to appear in the right side of the footer.

abc **right header text** is the information you want to appear in the right side of the header.

Row Headers determines how each row is labeled in the table.

Rows/Page determines how many rows appear on each printed page of the report. If set to `0`, as many rows as can fit on the page are printed and column headers are not repeated on each page.

Separate Page? Places the table on a new page of the report.

Text is the information you want to include in the report. Any information you want to include must be in a string.

Text Color (unchanged) lets you select the color of the text in the report. You can use the color box constant with this parameter (which you access from the **Functions»Numeric»Additional Numeric Constants** palette).

Text Color indicates the color currently used for the text in the report.

Text Data is the information you want to be printed in tabular form. If you are using a table control, pass the table's value to this parameter.

Text Options specifies how the text appears in the report.

- **Italic** determines whether subsequent text appears in italics in the report.
- **Strike Through** determines whether subsequent text appears with strikethroughs in the report.
- **Underline** determines whether subsequent text appears underlined in the report.
- **Bold** determines whether subsequent text appears bold in the report.

Text Options indicates what text options currently are set in a report.

- **Italic** indicates whether Italics currently are used in the report.
- **Strike Through** indicates whether strikethroughs currently are used in the report.
- **Underline** indicates whether underlines currently are used in the report.

TF **Bold** indicates whether bold currently is used in the report.

abc **Text to be Printed** is the information you want to include in the report. Any information you want to include must be in a string.

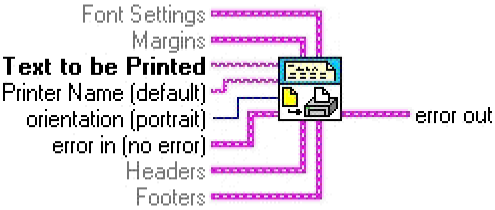
DBL **width** sets the report's tabs width in inches or centimeters. If you do not enter a tab length, the default value .25 is used.

Easy Text Report

Allows you to pass in a block of text (such as a log) along with optional formatting information, then print the report to a designated printer. (If you do not specify a printer, the default printer is used.)

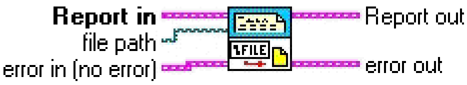
Note ***You cannot use this VI with any other NI Report VIs. Also notice that this VI does not give you fine-grain control over formatting your report. For more complicated reports (such as those with lots of different kinds of information), use the other VIs found in the* Report Generation *palette.***

You might find using tokens with this VI helpful, particularly in the **Headers** and **Footers** parameters. For example you might want to give the report a time stamp. See the *Tokens* topic, earlier in this chapter, for more information on some common tokens for this VI.



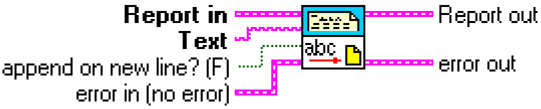
Append File to Report

Appends the text from a text file (`.txt`) into the current report. You must wire the **file path** (including its path) to the VI. The text is appended within the report.



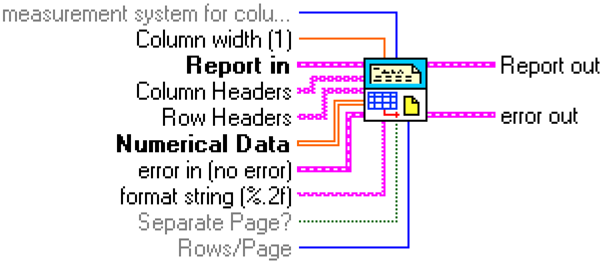
Append Report Text

Appends text to the selected report. The input into **Text** must be a string. The selected report is the one passed into **Report In**. You can append the text to the current position of the cursor in the report or on a new line.



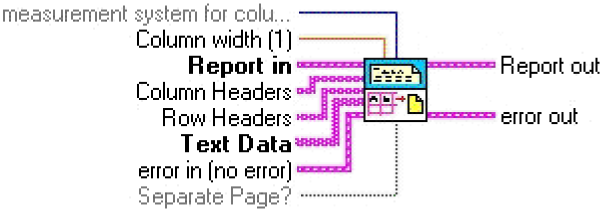
Append Numeric Table to Report

Takes a 2D array of numbers and appends it to a report as a table with the given column widths.



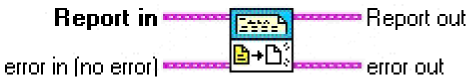
Append Text Table to Report

Takes a 2D array of strings and appends it to a report as a table with the given column widths.



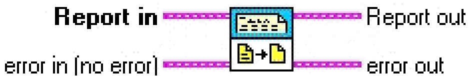
Clear Report

Clears the report of all text, headers, footers, and formatting information.



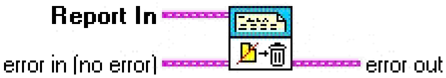
Clear Report Text

Clears the text and related formatting information from the report.



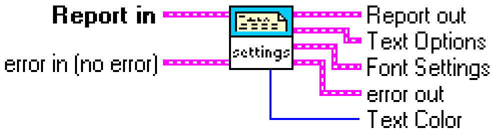
Dispose Report

Closes the report and releases its interface, which saves memory. No further operations are permitted on the report. However, you can create a new report. You can use this VI only as the last VI in the report function you are creating.



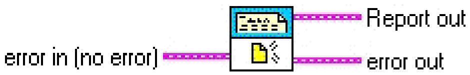
Get Report Settings

Retrieves information about the current font and text settings of a given report. The information is displayed on the front panel.



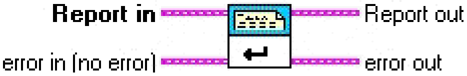
New Report

Creates a new report. You must use this VI to create a report if you do not use the Easy NI Report VI.



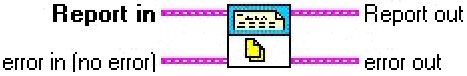
New Report Line

Starts a new line in the report to which you can append text or append another file.



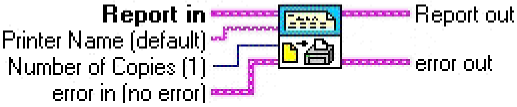
New Report Page

Adds a new page in the report to which you can append text or append another file.



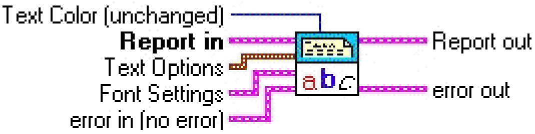
Print Report

Prints the report to a designated printer or to the default printer set up on your computer.



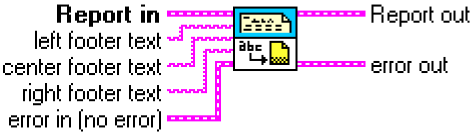
Set Report Font

Sets the font properties for the report, including those in the headers and footers. The available options include italic, bold, strikethrough, underline, color, font name, font size, character set and weight.



Set Report Footer Text

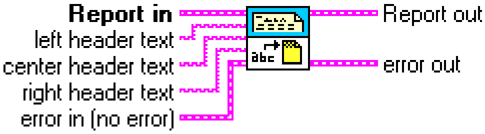
Sets the text for the report footer.



You might find that using tokens with this VI is helpful, particularly in the **Headers** and **Footers** parameters. For example, you might want to put a time stamp in the footer. See the *Tokens* topic, earlier in this chapter, for more information on some common tokens for this VI.

Set Report Header Text

Sets the text for the VI header.

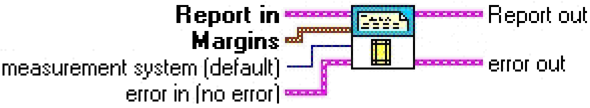


You might find that using tokens with this VI is helpful, particularly in the **Headers** and **Footers** parameters. For example, you might want to give the report a time stamp in the header. See the *Tokens* topic, earlier in this chapter, for more information on some common tokens for this VI.

Set Report Margins

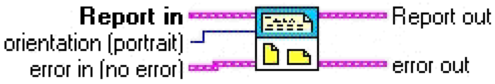
Sets the margins of the specified report.

Margins are defined by a cluster of four values: *left*, *right*, *top*, and *bottom* margins. Also, you must specify the measurement system (US, Metric, or Default) that determines your units of measurement. Selecting **US** gives you inches, selecting **Metric** gives you centimeters, and selecting **Default** gives you the units of the current measurement system configured on your computer.



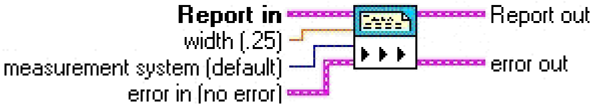
Set Report Orientation

Determines whether the report is printed in landscape or portrait orientation.



Set Report Tab Width

Sets the tab width in the report. To place a tab in your text, insert the token `<tab>` in the text string. Alternatively, you can use the Concatenate String function and insert a tab constant to build the text string.



Enhancements to DAQ

This section describes the data acquisition (DAQ) enhancements in BridgeVIEW 2.1.

DAQ Solution Wizard

You can now use the DAQ Solution Wizard even if you have no DAQ devices configured on your computer. However, without DAQ devices installed on your computer you cannot specify particular board options, and any opened solutions cannot be run until you install the appropriate DAQ device.

Support for NI-DAQ

BridgeVIEW 2.1 ships with NI-DAQ 6.5. The DAQ Channel Wizard has been upgraded in NI-DAQ 6.5 and now is integrated in with the Measurement & Automation Explorer, which allows you to configure and test your National Instruments products from one common application.

New Syntax Element for Nonsequentially Scanned SCXI Module Channels

If you operate a module in parallel mode, you can specify an SCXI channel either by specifying the corresponding onboard channels or by using the *SCXI channel syntax*. This syntax is described in the *SCXI Channel Addressing* section of Chapter 20, *Special Programming Considerations for SCXI*, in the *LabVIEW Data Acquisition Basics Manual*.

If you are scanning modules, you can scan an arbitrary number of channels for each module using channel list elements that are components of the SCXI channel syntax. Previously, however, the channels of each module needed to be scanned in consecutive, ascending order.

Now, with BridgeVIEW 2.1, you can scan modules randomly, provided that the module is capable of random scanning. The new syntax follows.

Channel List Element	Channel Specified
`OBz!SCx!MDy!(a,…,n)`	Channel `a` through `n` in the module in slot `y` of the chassis with ID `x` are multiplexed randomly onto onboard channel `z`.

For example, if you wanted to sequentially scan channels 2, 3, 4, and 5, you could use the following channel list element:

```
ob0!sc1!md2!(2,3,4,5)
```

You could also use the channel list element `ob0!sc1!md2!2:5`.

However, to scan the module’s channels randomly, you could use the following channel list element:

```
ob0!sc1!md2!(5,1,3,5)
```

You could also use a colon (:) in the list to scan a series of channels sequentially, as in the following channel list element:

```
ob0!sc1!md2!(2,1,4,7:11,13,15)
```

In addition, you can use semicolons instead of commas within the parentheses and achieve the same scan:

```
ob0!sc1!md2!(2;1;4;7:11;13;15)
```

Enhancements to VISA

BridgeVIEW 2.1 has two new VISA functions as well as a new VISA palette that contains the interface-specific functions. You can reach the new palette by selecting **Functions»Instrument I/O»VISA»Interface Specific**.

This palette replaces the **Functions»Instrument I/O»VISA»VISA Serial** palette. In versions of BridgeVIEW previous to BridgeVIEW 2.1, the following three VIs could be found on the **VISA Serial** palette but are now on the **Interface Specific** palette:

- VISA Set Serial Buffer Size
- VISA Flush Serial Buffer
- VISA Serial Break

In addition to the above existing functions, you can find the following two new functions on the **Interface Specific** palette:

- VISA GPIB Control REN
- VISA VXI Cmd or Query

The VISA GPIB Control REN function asserts or deasserts the GPIB Remote Enable interface line according to a specified mode. The VISA VXI Cmd or Query function sends a command or query, or receives a response to a query previously sent to the device.

You can find examples of the VISA VIs in the `Examples\Instr\visa.llb` directory.

Enhancements for Building and Distributing Applications

This section describes the new features for building and distributing applications now available in BridgeVIEW.

Note ***This functionality is available only in the BridgeVIEW Professional Development System or if you purchase the Application Builder libraries.***

Building Executable Programs

In BridgeVIEW 2.1, the process for building an application has been streamlined. Previously, to do so you had to save your VIs to a library, then build an application using the **Build Application** dialog box. Further, to build an installer you had to use the **Create Distribution Kit** dialog box.

Note ***If your application requires the BridgeVIEW Engine (that is, if it uses tags), your executable requires the BridgeVIEW Run-Time System.***

Now, in BridgeVIEW 2.1, you can use the **Build Application** dialog box to do all of these operations. You can configure the application to various settings within the tabs on the **Build Application** dialog box. After you define these settings, you can save them in a script so that you can easily rebuild the application.

Building an Application

To build an application in BridgeVIEW, complete the following steps:

1. Select **Project»Build Application**. The **Build Application** dialog box appears. The **Build Application** dialog box contains the following tabs: **Target**, **Source Files**, **VI Settings**, **App Settings**, and **Installer**. You can create a new build or load a build file that you created previously.
   - If you want to create a new application, proceed to step 2.
   - If you already have a build file, click **Load** and choose the `.bld` file to load. Then proceed to step 6.
2. From the **Target** tab, specify the following information:
   - **Application name**—The name of the application you are creating. This file should have a `.exe` extension.
   - **Destination directory**—The path and name of the directory in which to create and save your new application.
   - **Support file directory**—The path and name of the directory in which to save any support files.
3. Under the **Build** section of the **Target** tab, choose one of the following two options:
   - **Single application containing all VIs**—Check this option to create a single application containing all of your VIs.
   - **Small application with external file for subVIs**—Check this option if you want to keep the main application small.

4. From the **Source Files** tab, you can configure the VIs that make up your application. You can specify top-level VIs, dynamically loaded VIs, and additional non-VI files (such as readme files). You can update the file list automatically as VIs are added or removed from your hierarchies. Depending on what kind of source files you want to add, complete the following steps:

    a. If you want to add top-level VIs, click **Add Top Level VI**. The **Open** dialog box appears where you can enter a file name, or browse to find the VIs you want to add. When you select a top-level VI, BridgeVIEW automatically includes all its subVIs and related files (such as menu files or DLLs).

    b. If you want to add dynamic VIs, click **Add Dynamic VI**. If your VI dynamically calls any subVIs using the VI Server, BridgeVIEW cannot detect them automatically, so you must add them by using this option.

    c. If you want to add support files, click **Add Support File**. When you use this option, data files copy over to your application directory automatically. In addition to VI files (VIs, controls, menus, external subroutines, and so on), you can determine the set of DLLs referenced by your application. Because there are DLLs you might not want to redistribute, BridgeVIEW includes only those DLLs that are within the source hierarchy directories and the BridgeVIEW directory. If you want to include DLLs that are in the System directory, for example, you can include them as additional non-VI files manually.

    d. If you want to remove a file from the list, click the file to highlight it and click **Remove File**.

5. Click **Save** to save the information you have entered. The **Save As** dialog box appears. Enter a file name with a `.bld` extension to save the information you have entered into this dialog box.

6. Click **Build**. The **Build Status** dialog box appears.

7. After the build operation finishes, click **Done** to close the **Build Application** dialog box.

Customizing Application Features

1. If you want to customize some destination settings, select **Custom Destinations** from the **Source Files** tab. The **Destination Settings** dialog appears, in which you can configure the following settings:
   - You can modify your destination directory.
   - If you want to add a program item to your **Start** menu as part of an installer, select the **Create program item** checkbox and enter the name.
   - If you are creating an installer, you can specify how you want to **Replace existing files**. Select **Never**, **Ask**, **If Newer**, or **Always**, depending on how or if you want to be prompted.
2. From the **App Settings** tab, you can customize the features in your application. You can choose to customize icons and ActiveX server features.
   a. If you want to specify your own icon, click the **Custom icon** checkbox and designate the path to the icon.

   b. If you want to enable the ActiveX server, click the **Enable ActiveX server** checkbox. Your application can then respond to requests from ActiveX clients. The functionality of the ActiveX server in your application is a subset of the BridgeVIEW ActiveX server. When you build an application `myapp.exe`, an ActiveX type library `myapp.tlb` is also created along with the executable. The type library defines a createable class, *Application*, and a dispatch class, *Virtual Instrument*, and exports the properties and methods for these classes. You can find the Help for these properties and methods in `lvcomm.hlp` in the BridgeVIEW `Help` directory. When you distribute the application make sure the type library and the help file are located with the executable.

   When you assign the name of the application to the server name, your application is uniquely identified in the system registry. Once you build the application, you should run it at least once to enable registry with the system. After the application is registered, ActiveX clients access the server objects using server names. For example, if you specify the server name as `myapp`, clients instantiate an application object using the `myapp.application`.

Modifying VIs as Part of the Build

Use the **VI Settings** tab to specify the modifications to your VIs that you want to make part of the application build. You can choose to enable or disable various window option and execution option VI Setup settings. These settings apply to the build process only and do not affect your original source VIs.

BridgeVIEW removes debugging code, block diagrams, and unnecessary front panels, making your application as small as it can be. The removal of front panels is a new feature with BridgeVIEW 2.1. BridgeVIEW can detect which panels are necessary in almost all cases. However, if you open a front panel dynamically using the VI Server, you must specify that the panel is needed using the **VI Settings** tab.

You can edit only a single row at a time. By default, all unnecessary panels are removed. You can override the default and include the panel by setting the **Remove Panel** option to **No**.

To change settings, select a VI so that it is highlighted in the list. Click **Edit Build Settings**. The **Edit Build Settings** dialog box appears. For each setting you can choose **yes**, **no**, or **no change**. When you have made all the settings, click **Change**. Verify that all the settings are the way you want them for each VI in the build.

Creating an Installer

1. From the **Installer** tab, click the **Create installer** checkbox. Verify the following sections of this tabbed page. If you create an installer, the installer is written to the directory that contains your application. The disk images are created in a *disks* subdirectory of the destination directory that you specified on the **Target** tab. This directory will contain a setup program as well as files named `data.001`, `data.002`, and so on. If you plan to put the disk images on floppy, it is best to copy the `setup` and `data.001` files to the first floppy, copy the `data.002` file to the second floppy, and so on.
    - Installation name
    - Start menu program group
    - Default installation directory
    - Installation language
    - Media size
    - Extra space on first disk (KB)

The **Media size** item lets you specify how the file is to be segmented—for 720 KB, 1.2 MB, or 1.4 MB floppies. Even if you plan to distribute the files by CD, it is necessary to segment them. However, if you want to run the installer from a CD or from your drive, you can place all of the files in the same directory and run the setup program from that directory.

The **Extra space on first disk (KB)** item lets you reserve space on the first disk. You might reserve space on the first disk if you want to put a readme file on the first floppy.

2. Click the **Advanced** button. The **Advanced Installer Settings** dialog appears.
   a. If you would like to create an uninstaller, click the **Create uninstaller** checkbox.
   b. If you would like to run a program after the installation, click the **Run executable after installation** checkbox and enter the executable and command line argument information.
3. Select the **Run executable after installation** item if you want to run a program after the installation completes. Additionally, you can use this item to run a program that finishes the installation. For example, you might write a DOS batch program or a C program that modifies a `.ini` file or a `registry` file. Install the file as part of your installation and then run it afterwards to make the necessary modifications. The file that you run must be one of the files that you install.
4. If you choose to run an executable after the installation completes, you can use the **Command line arguments** to specify arguments passed to the program. In addition to specifying standard arguments, you can embed any of the following items in the command line argument string:

`%dest`	The application installation directory chosen by the user
`%src`	The directory that contains `setup.exe`
`%group`	The installation program group name
`%name`	The installation name

If any of these options are present at installation time, they are replaced with the proper values before the arguments are passed to the executable.

Run-Time Engine for the Application Builder

When you develop an executable program with BridgeVIEW and ship it to another computer, you must also install the BridgeVIEW Run-Time Engine. The computer on which the program runs must install this component using the BridgeVIEW Run-Time Engine Installer before the program executes.

If you distribute a program using Build Application, the Run-Time Engine is installed automatically.

This enhancement greatly reduces the size of the executable program.

Note ***After the Run-Time Engine is properly installed on a machine, it can run any executable program developed in BridgeVIEW.***

Enhancements to Networking

This section describes the networking enhancements in BridgeVIEW 2.1.

DataSocket VIs

DataSocket technology facilitates the exchange of data and information between an application and a number of different data sources and targets. These sources and targets include files and HTTP/FTP servers.

In BridgeVIEW 2.1, new VIs provide a simple yet intuitive way to access and use DataSocket technology within BridgeVIEW. You can create applications that share data among many different sources using a single Application Programming Interface (API).

For more information about the new DataSocket VIs, refer to the *DataSocket VIs* topic in the *BridgeVIEW Online Reference*. You can find examples of the DataSocket VIs in the `Examples\Comm\datasktx.llb` directory.

Internet/HTTP Services

Internet/HTTP services are available on all platforms and versions for BridgeVIEW 2.1, and give you capability to do the following:

- Publish HTML documents
- Generate HTML-readable images of a VI’s front panel across the Web

- Generate animated versions of a VI's front panel on the Web
- Use basic access control to limit which VIs may be viewed and by whom

Note ***If you want to control VIs from a browser, or you need to dynamically create HTML documents from your BridgeVIEW program, or you need more sophisticated access control, consider the Internet Toolkit, also available from National Instruments.***

The **Edit»Preferences** dialog box now includes three new list box options that you use to set up the built-in Web Server:

- **Web Server: Configuration**
- **Web Server: Browser Access**
- **Web Server: Visible VIs**

The Web Server

With the Web Server in BridgeVIEW, you can publish your VI front panels on the World Wide Web along with HTML documents.

The built-in Web Server is intended for users that need to make the panels of their running VIs visible on the Internet, but do not require sophisticated security features nor want to control VIs through the Internet.

The Web Server Configuration Dialog Box

You use the **Web Server Configuration** dialog box to set up the Web Server.

Access the **Web Server Configuration** dialog box by selecting **Edit»Preferences** and then **Web Server: Configuration** from the drop-down menu. The following table describes the available options.

Table 2-6. Web Server Configuration Dialog Box Options

Option	Description
Web Server Enabled	Enables the Web Server to publish front panel images and HTML documents. The default status is `Off`.
Root Directory	The directory that contains HTML documents published by the BridgeVIEW Web Server. The default path is `<BridgeVIEW>\www`.

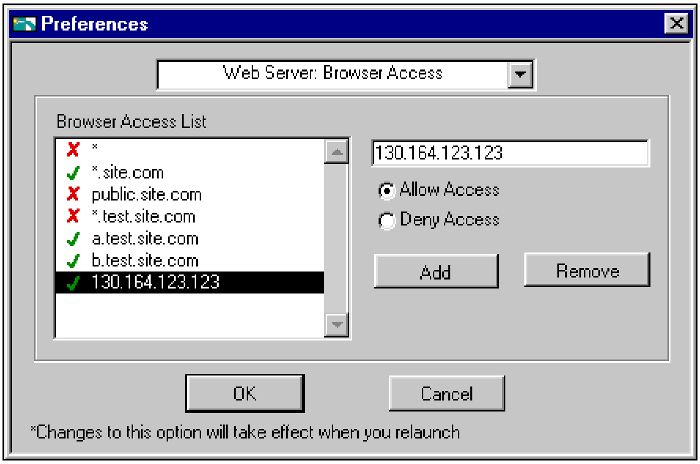
Table 2-6. Web Server Configuration Dialog Box Options (Continued)

Option	Description
HTTP Port	The TCP/IP port used to access the Web Server. The default port is `80`.
Timeout	How long (in seconds) the Web Server waits while reading a request before it times out. The default value is `60`.
Log File	The path to the data log file in which time-stamped information about connections is saved. The default path is `<BridgeVIEW>\www.log`.

The Web Server Browser Access Dialog Box

The **Web Server Browser Access** dialog box lists the remote computers on the Internet that are allowed to access the Web Server.

Access the **Web Server Browser Access** dialog box by selecting **Edit»Preferences** and selecting **Web Server: Browser Access** in the drop-down menu.



Browser Access List:

- ✗ *
- ✓ *.site.com
- ✗ public.site.com
- ✗ *.test.site.com
- ✓ a.test.site.com
- ✓ b.test.site.com
- ✓ 130.164.123.123

130.164.123.123

● Allow Access

○ Deny Access

*Changes to this option will take effect when you relaunch

Figure 2-9. Web Server Browser Access Dialog Box

The following table describes the available options.

Table 2-7. Web Server Browser Access Dialog Box Options

Option	Description
Browser Access List	Lists computers or domains that have access to the Web Server. Click and drag an entry in this list to move it in the access list. If an entry permits access to the Web Server from an address, a check mark appears next to its name. If an entry denies access to the Web Server, an **X** appears next to its name. If no symbol appears next to the entry, the syntax for the entry is incorrect.
Text box	Changes or adds an entry to the list. To add an entry, click the **Add** button and type in this box. To change an existing entry, select it from the **Access List** and edit it in this box.
Allow Access and **Deny Access**	Determines whether the current entry has access to the Web Server. Click the **Allow Access** radio button to grant a computer or a domain access to the Web Server. Click the **Deny Access** radio button to deny a computer or domain access to the Web Server.
Add	Adds a new entry to the **Access List** following the item currently highlighted in the **Access List**.
Remove	Removes the highlighted entry from the **Access List**.

When a browser attempts to connect to the Web Server, the server examines the entries in the **Browser Access List** to determine whether the computer is permitted access. If an entry in the list matches the computer's address, the server either permits or denies access, based on how you set up the entry. If a subsequent entry also matches the computer's address, that permission is used in place of the previous permission.

For example, if you give `a.test.site.com` and `b.test.site.com` access, but do not extend access to *all* addresses ending in `*.test.site.com` (where the `*` wildcard indicates all), the two computers still have access. If no entry matches the client address, access is denied. (See Table 2-8 for more information on the `*` wildcard and permitting matching access entries.)

To specify an Internet host address, enter its domain name or IP address. You can use the `*` wildcard when specifying Internet host addresses. For example, you can specify all hosts within the domain `domain.com` with the entry `*.domain.com`. You can specify all hosts in the subnet whose first two octets are `130.164` with the entry `130.164.*`. The entry `*` matches all addresses. The following table shows examples of TCP/IP Access List entries.

Table 2-8. Examples of Access List Entries

Access String	Matches
`*`	All hosts
`test.site.com`	The host whose domain name is `test.site.com`
`*.site.com`	All hosts whose domain names end with `.site.com`
`130.164.123.123`	The host with the IP address 130.164.123.123
`130.164.123.*`	All hosts whose IP addresses start with 130.164.123

In the *Web Server Browser Access Dialog Box* shown earlier in Figure 2-9, all hosts in the `site.com` domain have access to the server, with the exception of all hosts in the `test.site.com` domain. Additionally, the hosts `a.test.site.com`, `b.test.site.com`, and `130.164.123.123` also have access to the server. The host `public.site.com` does not have access, even though it is in the `site.com` domain.

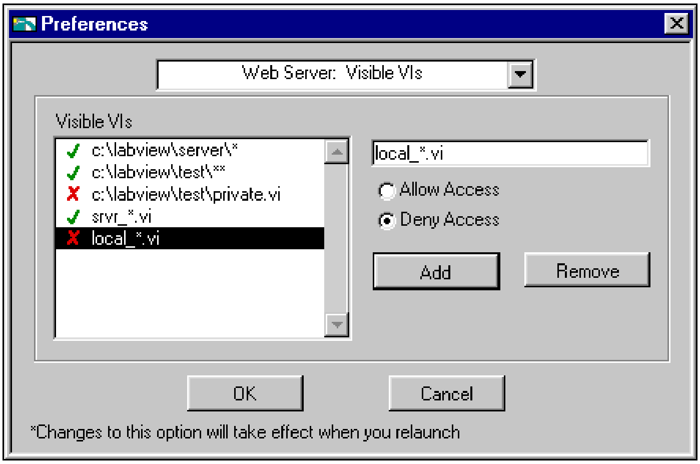
By default, all hosts have access to the Web Server.

Note ***If the Web Server runs on a system that does not have access to a DNS server, do not use domain name entries in the Access List. Requests to resolve the domain name or an IP address fail, slowing down the system. For performance reasons, place frequently matched entries toward the bottom of the Access List.***

The Web Server Visible VIs Dialog Box

Because the Web Server publishes front panels to the Web, it is best to specify which VI front panels you allow to be published. Through the **Web Server: Visible VIs** dialog box, you specify which front panels can be published to the Web.

You access the **Web Server Visible VIs** dialog box by selecting **Edit»Preferences** and selecting **Web Server: Visible VIs** in the drop-down menu.



Web Server: Visible VIs

Visible VIs

- ✓ c:\labview\server\*
- ✓ c:\labview\test\**
- ✗ c:\labview\test\private.vi
- ✓ srvr_*.vi
- ✗ local_*.vi

local_*.vi

○ Allow Access
● Deny Access

*Changes to this option will take effect when you relaunch

Figure 2-10. Web Server Visible VIs Dialog Box

The following table describes the available options.

Table 2-9. Web Server Visible VIs Dialog Box Options

Option	Description
Visible VIs	Specifies the VIs or groups of VIs whose front panels may be published by the Web Server. Click and drag an entry in this list to move it in the **Visible VIs** list. If an entry permits a VI or group of VIs to be seen, a check mark appears next to its name. If an entry denies such access, an **X** appears next to its name. If no symbol appears next to the entry, the entry's syntax is incorrect.
Text box	Changes or adds an entry to the list. To add an entry, click the **Add** button and type in this box. To change an existing entry, select it from the **Visible VIs** list and edit it in this box.
Allow Access and **Deny Access**	Determines whether the front panel of the current entry can be published by the Web Server. Click the **Allow Access** radio button if you want to allow the front panel of the VI or group of VIs to be published by the Web Server. Click the **Deny Access** radio button if you do not want to allow the front panel of the VI or group of VIs to be published by the Web Server.
Add	Adds a new entry to the **Visible VIs** list after the current item highlighted.
Remove	Removes the highlighted entry from the **Visible VIs** list.

Each entry in the **Visible VIs** list describes a VI name or a VI path and might contain wildcard characters. Entries that contain path separators are compared against VI paths, while entries that do not contain path separators are compared against VI names only.

When a web browser attempts to obtain a VI front panel image, the server examines the **Visible VIs** list to determine if it should grant access to the requested VI's image. If an entry in the list matches the requested VI, the Web Server either permits or denies access to that VI's image, based on how that entry is set up. If a subsequent entry also matches the VI, its access permission is used in place of the previous permission. If there is no VI in the list that matches the requested VI, access to the VI's image is denied.

You can use wildcard characters in the **Visible VIs** list so an entry in the list matches more than one VI. Use the wildcard characters shown in the following table.

Table 2-10. Wildcard Characters in Visible VIs List

Wildcard	Action
'?'	Matches exactly one arbitrary character, except for the path separator
'*'	Matches zero or more arbitrary characters, except for the path separator
'**'	Matches zero or more arbitrary characters, including the path separator

If you want to match a VI with a name that contains a wildcard character, you must escape that character using '`'. The following table shows examples of Visible VI list entries.

Table 2-11. Examples of Visible VI List Entries

VI Access String	Matches
*	All VIs
c:\usr\bridgeview\*	All VIs in the directory /usr/bridgeview/
c:\usr\bridgeview\**	All VIs in the directory /usr/bridgeview/ and any of its sub-directories
Test.vi	Any VI named Test.vi
OK\?	Any VI with the name OK?

In the *Web Server Visible VIs Dialog Box* shown earlier in Figure 2-10, all VIs in the c:\bridgeview\server directory have front panels that you can see on the web. All VIs in the c:\bridgeview\test directory and all its sub-directories are exported as well, with the exception of the VI c:\bridgeview\test\private.vi. Additionally, any VI that begins with the string srvr_ and ends with the string .vi is exported. No VI that begins with the string local_ and ends with the string .vi is exported, even if it is located within the c:\bridgeview\server directory.

By default, the front panel image of all VIs are visible.

Configuring the Web Server

To configure the Web Server, complete the following steps:

1. Select **Edit»Preferences** and choose **Web Server: Configuration** in the drop-down menu.

   See *The Web Server Configuration Dialog Box* section earlier in this chapter for more information on this dialog box.

2. Enter a **Root Directory**, which is the directory where the Web Server's HTML files are located. The default path is `<BridgeVIEW>\www`.
3. In **HTTP Port**, specify the TCP/IP port the server uses.

   The default port for HTTP is 80. You might specify a different port if another HTTP Server already uses port 80 on your machine or if you are on a system where you do not have permission to use reserved ports.

   If you use a non-default port, such as 8000, you must specify it on URLs that refer to your server, as shown in this example: `http://hostname:8000/index.htm`.

4. In **Timeout** specify the number of seconds the Web Server waits while reading a request before the server times out. The default value is `60`.
5. Specify a **Log File**, which is the path of the file where information about web connections is saved. The default path is `<BridgeVIEW>\www.log`.

Running the Web Server

You must run the Web Server in order to publish VI front panel images on the World Wide Web.

To run the Web Server, check **Web Server Enabled** in the **Preferences» Web Server: Configuration** dialog box.

Publishing Front Panel Images on the Web

To publish a VI front panel image on the World Wide Web, complete the following steps:

1. Run the Web Server by checking **Enable Web Server** in the **Preferences»Web Server: Configuration** dialog box.
2. Launch the VI(s) you want to publish to the web so they are in your system's memory.

Retrieving a Static Image of a VI Front Panel

Note ***To retrieve an HTML-Readable Image of a VI front panel, the VI must be in memory.***

To retrieve an HTML-readable image of a front panel, complete the following steps:

1. Open a Web browser.
2. Type in a URL that includes the address of the BridgeVIEW Web Server, the command for a static image (`.snap`), a question mark (`?`) to separate the URL from the parameters, and the name of the VI. The URL takes the following form:

   ```
   http://web.server.addr/.snap?VI_Name
   ```

You must encode the VI Name according to URL naming rules. Replace special characters with their hexadecimal value preceded by a percent (`%`) sign and replace spaces with a plus (`+`) sign.

For example, you can write the URL for the static image of the `Test Example.vi` as follows:

```
http://web.server.addr/.snap?Test+Example.vi
```

Following the VI name, you can add parameters that specify attributes for the image. See *Static Front Panel Image (.snap URL)* in the *What URLs Can I Use with My Front Panel Images?* section later in this chapter for more information on the characters you can enter in URLs to retrieve images.

Viewing an Animated Version of a Front Panel

To view an animated version of a front panel, complete the following steps:

1. Open a Web browser.
2. Type in a URL that includes the address of the BridgeVIEW Web Server, the command for an animated image (`.monitor`), a question mark (`?`) to separate the URL from the parameters, and the name of the VI. The URL takes the following form:

   ```
   http://web.server.addr/.monitor?VI_Name
   ```

You must encode the VI Name according to URL naming rules. Replace special characters with their hexadecimal value preceded by a percent (`%`) sign and replace spaces with a plus (`+`) sign.

With Netscape Navigator browsers, the Web Server uses the server-push method to implement animations of front panel images. During a server push, the server maintains an open connection and sends a new image after a predefined period of time. With other browsers, the Web Server uses the client-pull method to implement animation. During a client-pull animation, the browser intermittently sends a request for each image.

Following the VI name, you can add parameters that specify attributes for the image animation. See *Animated Front Panel Image (.monitor URL)* in the *What URLs Can I Use with My Front Panel Images?* section later in this chapter for more information on the characters you can enter in URLs to retrieve images.

Determining Which Front Panels are Visible

To make a VI's front panel visible across the Web, complete the following steps:

1. Open the **Web Server Visible VIs** dialog box by selecting **Edit»Preferences** and selecting **Web Server: Visible VIs** in the drop-down menu.
2. Choose **Add**.
3. To deny a specific VI's front panel from being viewed, type its path in the **Text box** and choose **Deny Access**. To allow a specific VI's front panel to be viewed, types its path in the **Text box** and choose **Allow access**.

Note ***You also can allow or disallow the viewing of whole groups of VI front panels by using wildcards. For more information about wildcards, see the*** *The Web Server Visible VIs Dialog Box* ***section earlier in this chapter.***

What URLs Can I Use with My Front Panel Images?

With the Web Server, you can publish images of your VI front panels on the World Wide Web. You do not need to modify the VIs to display their front panels.

Front Panel Image Formats

The Web Server can generate images of VI front panels in the Joint Photographic Experts Group (JPEG) and Portable Network Graphics (PNG) image formats.

The JPEG image format is a public domain image format that all current browsers support. It has been developed for the distribution of real-life images and photographs and uses a lossy compression algorithm to reduce the memory size of an image. When you use JPEG on images that contain lines and text, such as front panels, the resulting image often displays artifacts of the compression, such as fuzzy text or stray color pixels.

The PNG format is a recent public domain image format. The compression algorithm in this format is lossless, which produces PNG images exactly like the original images. PNG is designed to be the successor of the Graphics Interchange Format (GIF), which also uses lossless compression. PNG is an open standard that you also can use on true-color images. Internet Explorer 4.0.1 and Netscape Navigator 4.0.4 support the PNG format. Older browsers require a plug-in or an external application to view PNG images.

Static Front Panel Image (.snap URL)

The .snap URL signals the server to return a static image of the front panel of a VI currently in memory. The query parameters in the URL specify the VI name and the attributes of the image.

You must open the front panel of the VI to take snapshots for static images because closed front panels do not update the images of controls when the value changes.

The syntax to use in the URL for static front panel images is as follows (parameters in brackets [ ] are optional):

.snap?VI_Name

[**&type=type**]

[**&depth=depth**]

[**&quality=quality**]

[**&compression=compression**]

[**&refresh=refresh**]

[**&full=full**]

VI_Name is the name of the returned VI front panel. You must encode the VI name according to HTTP conventions. Replace special characters with `%`*`xx`*, where *`xx`* is the hexadecimal value of the character.

Type is the returned image type, either JPEG or PNG. If no **type** is specified, the default type is used.

Depth is the depth of the returned image. **Depth** can be 1, 4, 8, or 24 bits. If no **depth** is specified, the default depth is used.

Quality is the image quality and memory size of the JPEG front panel image. **Quality** can be between 0 and 100. If no **quality** is specified, the default quality is used.

Compression is the compression level used for compressing PNG images. **Compression** can be between 0 and 7. If no **compression** is specified, the default PNG compression is used.

Refresh is the maximum age of a cached image. If a cached image is older than refresh seconds, a new image is generated.

Full specifies whether to return the image of all controls or just the part visible in the window. Set **full** to **on** to indicate all controls and **off** to indicate the window content. If no **full** is specified, the image of the visible front panel in the window is returned.

The following are some examples on how you would use this syntax:

- To return the front panel image of the VI `My VI.vi` from the computer `foo` using the default image **type**, **depth**, and **quality**, use the following code:

  ```
  http://foo/.snap?My%20VI.vi
  ```

- To return the front panel image of the VI `Test 1.vi` from the computer `foo` using image **depth**=24 and image **type**=PNG, use the following code:

  ```
  http://foo/.snap?Test%201.vi&depth=24&type=png
  ```

- To embed the image of the VI `Example.vi`, in an HTML document on the same system, use the following code:

  ```
  <IMG SRC="/.snap?Example.vi">
  ```

- To embed the image of the VI `Example.vi` running on the computer `foo` in any HTML document, use the following code:

  ```
  <IMG SRC="http://foo/.snap?Example.vi">
  ```

Animated Front Panel Image (.monitor URL)

The .monitor URL signals the server to return an animated image of the front panel of a VI currently in memory. The query parameters in the URL specify the VI name, attributes of the animation, and attributes of the image.

For example, you can write the URL for the animated image of the VI `Test Example.vi`, which updates once every two seconds for three minutes, as follows:

```
http://web.server.addr/.monitor?Test+Example.vi&refresh=2&lifespan=180
```

For Netscape Navigator browsers, the server uses the server-push method of animation. The server accomplishes this animation by taking subsequent snapshots of the front panel image and sending them to the client. With other browsers, the server uses the client-pull method of animation.

You must open the front panel of the VI to take snapshots for animated images because closed front panels do not update the images of controls when the value changes.

The syntax to use in the URL for static front panel images is as follows (parameters in brackets [ ] are optional):

.monitor?VI_Name

[**&type=type**]

[**&depth=depth**]

[**&quality=quality**]

[**&compression=compression**]

[**&refresh=refresh**]

[**&full=full**]

[**&lifespan=lifespan**]

VI_Name is the name of the returned VI front panel. You must encode the VI name according to HTTP conventions. Replace special characters with %*xx*, where *xx* is the hexadecimal value of the character.

Type is the returned image type, either JPEG or PNG. If no **type** is specified, the default type is used.

Depth is the depth of the returned image. **Depth** can be 1, 4, 8, or 24 bits. If no **depth** is specified, the default depth is used.

Quality is the image quality and memory size of the JPEG front panel image. **Quality** can be between 0 and 100. If no **quality** is specified, the default quality is used.

Compression is the compression level used for compressing PNG images. **Compression** can be between 0 and 7. If no **compression** is specified, the default PNG compression is used.

Refresh is number of seconds between each succeeding image.

Full specifies whether to return the image of all controls or just the part visible in the window. Set **full** to **on** to indicate all controls and **off** to indicate the window content. If no **full** is specified, the image of the visible front panel in the window is returned.

Lifespan is the number of seconds the front panel animation lasts. Setting `Lifespan=0` implies that the animation continues until the browser cancels it. If no **lifespan** is specified, the default lifespan is used.

Note ***When using client-pull animations, the lifespan is ignored.***

The following are some examples on how you would use this syntax:

- To generate an animated front panel image of the VI `My VI.vi` from the computer `foo` using the default image **type**, **depth**, and **quality**, use the following code:

  ```
  http://foo/.monitor?My%20VI.vi
  ```

- To generate a 60-second animation of the front panel image of the VI `Test 1.vi` from the computer `foo` using the default image **type** and **quality** but using `refresh=5`, use the following code:

  ```
  http://foo/.monitor?Test%201.vi&refresh=5&lifespan=60
  ```

- To embed the image of the VI `Example.vi` in an HTML document on the same system, use the following code:

  ```
  <IMG SRC="/.monitor?Example.vi">
  ```

- To embed the image of the VI `Example.vi` running on the computer `foo` in any HTML document, use the following code:

  ```
  <IMG SRC="http://foo/.monitor?Example.vi">
  ```

Enhancements to Examples and Activities

You can run example VIs to help you get started with BridgeVIEW. To generate or find examples similar to your application, refer to the Solution Wizards or to the Search Examples online help file, which you can access from the BridgeVIEW startup dialog box.

In addition to the example VIs, a good way to get started with BridgeVIEW is to complete the activities available to you in the *BridgeVIEW User Manual* and in the *BridgeVIEW Online Reference*.

Note ***The*** *BridgeVIEW Online Reference* ***has the most up-to-date information regarding paths to VIs, functions, and controls.***

A

Manual Clarifications and Additions

This appendix clarifies and corrects information in the BridgeVIEW manual set. Because these manuals were not revised for BridgeVIEW 2.1, this appendix contains information relevant only to the previous release.

Enhancements to BridgeVIEW 2.1 have rendered some information in these manuals incorrect, particularly illustrations of palettes and navigation paths to functions and controls, including those in activities. See Chapter 2, *New Features in BridgeVIEW 2.1*, for updated information and descriptions of new features.

Using Remote OPC Servers

With BridgeVIEW 2.0, you could configure clients to use remote OPC servers on networked Windows NT/98/95 machines, without using the Windows `dcomcnfg.exe` utility and without registering the server on the client machine. With BridgeVIEW 2.1, to access a remote OPC server, complete the following steps:

1. In the Tag Configuration Editor, select **Servers»Browse Network OPC Servers**. The **Browse OPC Servers on Network** dialog box appears. You can also open this dialog box through the Server Browser utility.
2. (Optional) On the machine that contains the remote OPC server, use the `dcomcnfg.exe` utility to configure security options so that other machines can access it across the network.
3. Browse the network for OPC servers and add them to your local **Servers** list.

For more information about accessing remote OPC servers, refer to the *Using Remote OPC Servers* section in Chapter 8, *Servers*, of the *BridgeVIEW User Manual*.

Multithreading

Color of Code Interface and Call Library Function Nodes—The color of a code interface node (CIN) or Call Library Function node on a block diagram changes depending on whether BridgeVIEW considers it reentrant. If BridgeVIEW considers a CIN or Call Library Function node reentrant, BridgeVIEW assigns it the current primitive color (the default is pale yellow). If a CIN or Call Library Function node is not considered reentrant, its color is orange. This color designation exists on all platforms, even if the platform itself is not threaded.

ActiveX

ole_lv5container.dll—The ActiveX Container uses a DLL named `ole_lv5container.dll`, which is located in the `resource` directory. If you build an application that includes ActiveX controls and move it to another machine, you must install this file in the same directory as the built application or in the `System` directory. In the BridgeVIEW documentation, references to `ole_container.dll` should be `ole_lv5container.dll`.

Data Format—The compatibility VIs for the BridgeVIEW 1.*x* Automation functions require that you pass flattened data in the BridgeVIEW 1.*x* format. BridgeVIEW 2.*x* loads your BridgeVIEW 1.*x* VIs and automatically selects the **Convert 4.x Data** option for the Flatten To String and Unflatten From String functions, which are found in the **Advanced»Data Manipulations** palette.

Instrumentation

Signal Generator by Duration VI—The Signal Generator by Duration VI has been added to the **Signal Processing»Signal Generation** palette. This VI generates a signal with a shape given by the waveform type: sine, cosine, triangle, square, sawtooth, increasing ramp, or decreasing ramp.

CVI Function Panel Converter Changes—The improved CVI Function Panel Converter creates hierarchical text menus so you can find functions quickly. Two new options have been added to the CVI Function Panel Converter. These options are ON by default.

- **Map ViSession type to VISA Session RefNum**—This option specifies that instrument session numbers of type ViSession in the CVI Function Panel are converted to BridgeVIEW VISA RefNums in the resulting VI. Functions that contain the string `_init` in their name automatically register with the VISA refnum; functions that contain `_close` in their name automatically close the VISA refnum.

- **Create instr.lib menu mirroring CVI Class Hierarchy**—This option specifies that when converting a Function Panel file, a palette menu for the instrument is created in the **Instrument Drivers** menu. This menu is organized hierarchically according to the Function Panel Tree in the `.fp` file.

General Interface Features

Icon and Text Palettes

You can display palettes in three modes: Standard, All Icons, or All Text. Choose the palette display mode in the **Edit»Select Palette Set» Display Style** submenu.

In All Text mode, you right-click to access the **Controls** or **Functions** palette. These text palettes contain the names of options. Items in text palettes are organized in the same order as in the icon palettes when you read the icon palette row by row, left to right. Empty spaces in the icon palettes are skipped. Unlike icon palettes, you cannot tack down text palettes or subpalettes. In All Text mode, BridgeVIEW displays text palettes in the **Project** menu and the Find dialog box. In Standard or All Icons mode, BridgeVIEW displays icon palettes.

Standard is the default mode. In Standard mode, all palettes default to icon palettes, but you can edit individual palettes to display them as text palettes.

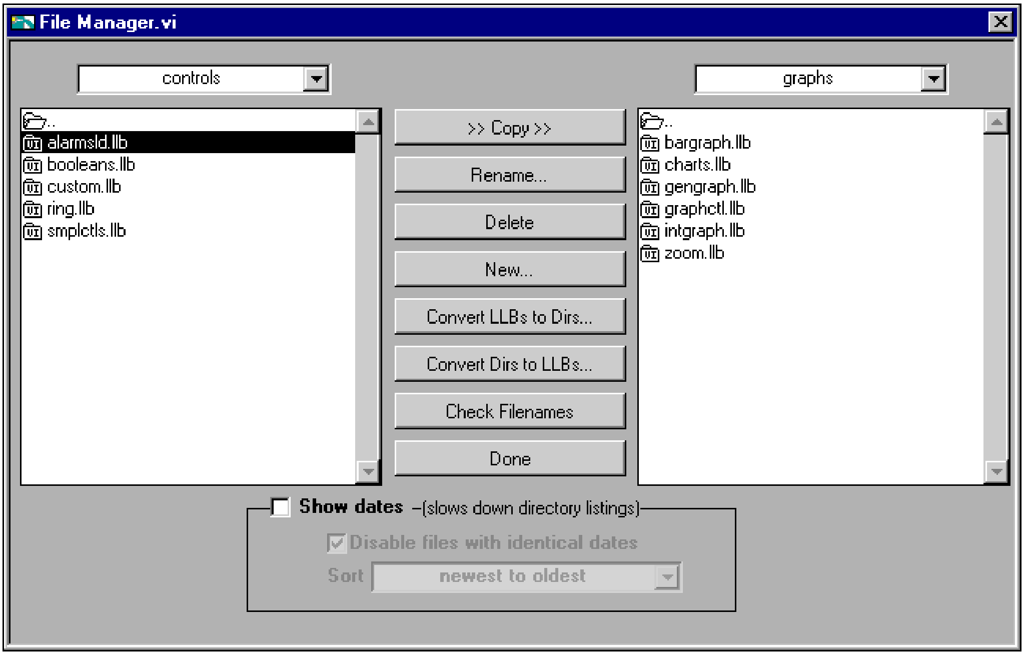
When you edit a palette by selecting **Edit**»**Edit Control & Function Palettes**, BridgeVIEW displays the palettes in All Icons mode. You cannot edit palettes in the other modes because they do not contain as much information (icon palettes have both icons and two-dimensional layout, while text palettes do not). To specify the mode for the Functions or Controls palette, right-click on the palette—but not on a subpalette icon—and select either **Icons** or **Text** from the **Standard Menu View** submenu. The mode you select affects only the menu you are editing.

File Manager Tool

You can use the File Manager tool to simplify copying, renaming, and deleting files within VI libraries (LLBs). You also can use this tool to create new LLBs and directories and convert LLBs to and from directories. To access the File Manager tool, select **Project»File Manager**.

To avoid performing a file operation on a VI already in memory, close all VIs that might be affected before using this tool.

In the File Manager dialog box, shown in the following figure, you can view two locations (directory or LLB) simultaneously. When you select a file, you can copy, rename, or delete it using the corresponding buttons between the two lists. Click on the **New** button to create a new directory or LLB.



File Manager.vi

controls

..
alarmsld.llb
booleans.llb
custom.llb
ring.llb
smplctls.llb

graphs

..
bargraph.llb
charts.llb
gengraph.llb
graphctl.llb
intgraph.llb
zoom.llb

□ Show dates –(slows down directory listings)

☑ Disable files with identical dates

Sort newest to oldest

Figure A-1. File Manager Tool Dialog Box

If you select an LLB, you can click **Convert LLBs to Dirs** to convert it to a directory. If you select a directory and click this button, the tool scans for all LLBs within that directory and gives you the option to convert them to directories. The new directory is created in the same location as the original LLB.

If you assign the new directory a name that differs from that of the original LLB, BridgeVIEW searches for the files that were within the LLB when calling a VI (even when the name is the same minus the `.llb` extension). When you convert an LLB to a directory, you have the option to back up the LLB (the `.llb` extension changes to `.llx`).

To convert a directory to an LLB, select a directory and click **Convert Dirs to LLBs**.

Click **Check Filenames** to scan a directory or VI library for platform-dependent filenames. The tool scans all filenames for invalid characters (:, \, /, ?, *, <, >, or |) and verifies filenames to be 31 characters or less (a limitation on the Macintosh). The **Check Filenames** option also scans files within LLBs. These files are portable, even if their names contain characters that are invalid on some platforms. By scanning within LLBs, this tool helps you detect potential problems if you move your files out of VI libraries.

Use the **Show dates** option at the bottom of the dialog box to display file modification dates next to each file. You can choose to sort the files alphabetically or by date and disable files with the same name and date in both directory listings. Use this technique when comparing two directories to determine whether any files have changed.

Other General Interface Features

Dragging and Dropping VI Icons—BridgeVIEW 2.0 simplified the creation of VI icons. By dragging an image file and dropping it onto the VI icon in the upper-right corner of a front panel, a 32-by-32 version of the image replaces the existing icon.

You can drag a VI icon from the icon pane in the upper-right corner to a block diagram to instantly create a subVI call. By pressing <Shift> while dragging the VI icon, you automatically wire the non-default values of the controls as constants for the subVI.

If the subVI already appears in a block diagram, pressing <Shift> while dragging onto the existing call updates the attached constants. A control at its default value discards the constant attached to the subVI, and an input wired to anything other than a constant is unaffected.

When you press <Shift> while double-clicking a subVI icon to open the subVI front panel, BridgeVIEW loads the values of the constants wired to the subVI into the front panel controls. All unwired controls retain the default values.

You also can use the drag-and-drop technique for global variables and custom controls. Additionally, you can drag a VI icon into a VI refnum on a front panel control to load VIs into memory dynamically, which is part of the VI Server functionality.

Print to RTF/HTML Feature—The Print to RTF/HTML feature can export graphics in uncompressed graphics interchange format (GIF). To use this feature, select **File»Print Documentation**, and select **RTF File** or **HTML File** from the **Destination** pull-down menu.

Configuration File VIs—The Configuration File VIs, which you can access from the **Functions»File I/O»Configuration File VIs** palette, provide tools for reading from and writing to a platform-independent configuration file similar in format to an initialization (`.ini`) file.

New Preferences Options—BridgeVIEW 2.*x* adds the following two options in the **Miscellaneous** view of the **Edit»Preferences** dialog box:

- **Automatically close VISA sessions**—Use this option to specify that VISA sessions, like file refnums, close automatically when the top-level VI goes idle. The default is ON, which closes VISA sessions automatically.
- **Treat read-only VI as locked**—Using this option, you can choose whether to treat read-only VIs as locked. You cannot edit locked VIs, but you can re-compile and execute them. By default the option is not selected so that read-only VIs appear normally. However, you cannot save the VI to the same location (the read-only file) unless you change the file permissions outside BridgeVIEW. This behavior is consistent with the behavior in previous versions of BridgeVIEW. When using the VI Server, the read-only status of files is ignored except when saving. This option is designed primarily to support the source-code control of the Professional G Developers Toolkit.

Execution System Selection—The default preferred execution system for a VI is **same as caller**. This setting allows the VI to run in the same execution system in which caller is running when the subVI call to the VI is made. The **same as caller** setting has the lowest run-time overhead. When you set a VI to **same as caller** and you run it at the top level, it runs in the **standard** execution system at its selected priority.

Icon Editor—The **Undo** button has been removed from the Icon Editor, but you can undo an action by selecting **Edit»Undo** or `<Ctrl-Z>`.

Offscreen Updates Default Value—The default value for offscreen updates is now ON instead of OFF.

Support for Template VIs and Controls

You can save commonly used VIs and controls as templates. To create a template VI, save a VI with a `.vit` extension (or `.ctt` extension for typedefs). When you open a template VI or control, the new file you create is named automatically using your template name and a number corresponding to the number of times it has been opened. When you finish editing the VI and save it, BridgeVIEW prompts you to enter a new name for the file.

To modify a template, open it, make your changes, then save over the `.vit` (or `.ctt`) file that you originally created.

Adding VIs to the Project and Help Menus

You can add VIs to the **Project** and **Help** menus by placing them inside the `Project` or `Help` directories in the `BridgeVIEW` directory. You can use this technique to provide quick access to VIs that act as tools in your system. National Instruments uses this feature to make the Tech Support VIs accessible from the **Help** menu. Also, if you have the Application Builder libraries installed, you can see a **Build Application** option in the **Project** menu.

Any VI placed at the top level of the `Project` or `Help` directory is appended directly to the corresponding menu. If you create a subdirectory, a submenu is appended.

Clarifications to the BridgeVIEW User Manual

Because the examples were reorganized with this release of BridgeVIEW, many paths to examples are not listed or are listed incorrectly in the BridgeVIEW documentation set. Some examples were moved from the `examples\G examples` directory to the `examples` directory. For more information about the updated activities and examples, refer to Chapter 2, *New Features in BridgeVIEW 2.1*.

In Chapter 9, *Creating VIs*, of the *BridgeVIEW User Manual*, the text and an illustration in Activity 9-3, *Create an Icon and Connector,* refer to an **Undo** button in the Icon Editor. The **Undo** button has been removed, but you can undo an action by selecting **Edit»Undo** or <Ctrl-Z>.

VISA Error Codes

The following table lists numeric VISA error codes that were not included in the printed documentation.

Table A-1. VISA Error Codes

Error Code	Error Name	Description
1073676443	VI_SUCCESS_SYNC	Operation completed successfully, but the operation was actually synchronous rather than asynchronous.
1073676442	VI_SUCCESS_NESTED_EXCLUSIVE	Operation completed successfully, and this session has nested exclusive locks.
1073676441	VI_SUCCESS_NESTED_SHARED	Operation completed successfully, and this session has nested shared locks.
1073676440	VI_SUCCESS_NCHAIN	Event handled successfully. Do not invoke any other handlers on this session for this event.
1073676424	VI_WARN_NSUP_BUF	The specified I/O buffer is not supported.
1073676421	VI_WARN_UNKNOWN_STATUS	The status code passed to the operation could not be interpreted.
1073676420	VI_WARN_NSUP_ATTR_STATE	Although the specified state of the attribute is valid, it is not supported by this resource implementation.
1073676418	VI_WARN_NULL_OBJECT	The specified object reference is uninitialized.
1073676416	VI_SUCCESS_QUEUE_NEMPTY	Wait terminated successfully on receipt of an event notification. There is at least one more event occurrence of the type specified by inEventType available for this session.
1073676413	VI_SUCCESS_DEV_NPRESENT	Session opened successfully, but the device at the specified address is not responding.
1073676407	VI_WARN_CONFIG_NLOADED	The specified configuration either does not exist or could not be loaded. VISA-specified defaults will be used.
1073676294	VI_SUCCESS_MAX_CNT	The number of bytes transferred is equal to the input count.
1073676293	VI_SUCCESS_TERM_CHAR	The specified termination character was read.
1073676292	VI_SUCCESS_QUEUE_EMPTY	Operation completed successfully, but queue was already empty.
1073676291	VI_SUCCESS_EVENT_DIS	Specified event is already disabled for at least one of the specified mechanisms.
1073676290	VI_SUCCESS_EVENT_EN	Specified event is already enabled for at least one of the specified mechanisms.

Table A-1. VISA Error Codes (Continued)

Error Code	Error Name	Description
-1073807202	VI_ERROR_LIBRARY_NFOUND	A code library required by VISA could not be located or loaded.
-1073807204	VI_ERROR_SESN_NLOCKED	The current session did not have a lock on the resource.
-1073807215	VI_ERROR_INV_MODE	Invalid mode specified.
-1073807229	VI_ERROR_INV_LENGTH	Invalid length specified.
-1073807240	VI_ERROR_INV_PARAMETER	The value of some parameter (which parameter is not known) is invalid.
-1073807246	VI_ERROR_RSRC_BUSY	The resource is valid, but VISA cannot currently access it.
-1073807247	VI_ERROR_USER_BUF	A specified user buffer is not valid or cannot be accessed for the required size.
-1073807248	VI_ERROR_NSUP_ALIGN_OFFSET	The specified offset is not properly aligned for the access width of the operation.
-1073807252	VI_ERROR_ASRL_OVERRUN	An overrun error occurred during transfer. A character was not read from the hardware before the next character arrived.
-1073807253	VI_ERROR_ASRL_FRAMING	A framing error occurred during transfer.
-1073807254	VI_ERROR_ASRL_PARITY	A parity error occurred during transfer.
-1073807263	VI_ERROR_NSYS_CNTLR	The interface associated with this session is not the system controller.
-1073807271	VI_ERROR_RESP_PENDING	A previous response is still pending, causing a multiple query error.
-1073807275	VI_ERROR_NSUP_VAR_WIDTH	Cannot support source and destination widths that are different.
-1073807278	VI_ERROR_INV_WIDTH	Invalid access width specified.
-1073807301	VI_ERROR_QUEUE_ERROR	Unable to queue the asynchronous operation.
-1073807303	VI_ERROR_IN_PROGRESS	Unable to queue the asynchronous operation because there is already an operation in progress.
-1073807312	VI_ERROR_ABORT	User abort occurred during transfer.
-1073807313	VI_ERROR_NENABLED	You must be enabled for events of the specified type in order to receive them.
-1073807315	VI_ERROR_QUEUE_OVERFLOW	The event queue for the specified type has overflowed (usually due to previous events not having been closed).
-1073807327	VI_ERROR_INV_ACCESS_KEY	The access key to the specified resource is invalid.

Table A-1. VISA Error Codes (Continued)

Error Code	Error Name	Description
-1073807328	VI_ERROR_INV_LOCK_TYPE	The specified type of lock is not supported by this resource.
-1073807333	VI_ERROR_INV_DEGREE	Specified degree is invalid.

B

Technical Support Resources

National Instruments offers technical support through electronic, fax, and telephone systems. The electronic services include our Web site, an FTP site, and a fax-on-demand system. If you have a hardware or software problem, please first try the electronic support systems. If the information available on these systems does not answer your questions, contact one of our technical support centers, which are staffed by applications engineers, for support by telephone and fax. To comment on the documentation supplied with our products, send e-mail to `techpubs@natinst.com`.

Web Site

The InstrumentationWeb address is `http://www.natinst.com`.

From this Web site you can connect to our Web sites around the world (`http://www.natinst.com/niglobal/`) and access technical support (`http://www.natinst.com/support/`).

FTP Site

To access our FTP site, log in to our Internet host, `ftp.natinst.com`, as `anonymous` and use your e-mail address, such as `yourname@anywhere.com`, as your password. The support files and documents are located in the `\support` directories.

Fax-on-Demand Support

Fax-on-Demand is a 24-hour information retrieval system containing a library of documents in English on a wide range of technical information. You can access Fax-on-Demand from a touch-tone telephone at 512 418 1111.

E-Mail Support

You can submit technical support questions to the applications engineering team through e-mail at `support@natinst.com`. Remember to include your name, address, and phone number so we can contact you with solutions and suggestions.

Telephone and Fax Support

National Instruments has branch offices all over the world. Use the following list to find the technical support number for your country. If there is no National Instruments office in your country, contact the source from which you purchased your software to obtain support.

Country	Telephone	Fax
Australia	03 9879 5166	03 9879 6277
Austria	0662 45 79 90 0	0662 45 79 90 19
Belgium	02 757 00 20	02 757 03 11
Brazil	011 284 5011	011 288 8528
Canada (Ontario)	905 694 0085	905 785 0086
Canada (Québec)	514 694 8521	514 694 4399
Denmark	45 76 26 00	45 76 26 02
Finland	09 725 725 11	09 725 725 55
France	0 1 48 14 24 24	0 1 48 14 24 14
Germany	089 741 31 30	089 714 60 35
Hong Kong	2645 3186	2686 8505
India	91805275406	91805275410
Israel	03 6120092	03 6120095
Italy	02 413091	02 4139215
Japan	03 5472 2970	03 5472 2977
Korea	02 596 7456	02 596 7455
Mexico (D.F.)	5 280 7625	5 520 3282
Mexico (Monterrey)	8 357 7695	8 365 8543
Netherlands	0348 433466	0348 430673
Norway	32 84 84 00	32 84 86 00
Singapore	2265886	2265887
Spain (Madrid)	91 640 0085	91 640 0533
Spain (Barcelona)	93 582 0251	93 582 4370
Sweden	08 587 895 00	08 730 43 70
Switzerland	056 200 51 51	056 200 51 55
Taiwan	02 2377 1200	02 2737 4644
United Kingdom	01635 523545	01635 523154
United States	512 795 8248	512 794 5678

Index