BRIDGEVIEW™

Version 2.0

These upgrade notes describe the process of upgrading to BridgeVIEW 2.0. You can find additional information in the readme file, `bvreadme.doc`, located in your BridgeVIEW directory.

For more information about features, refer to the *BridgeVIEW User Manual* and the *G Programming Reference Manual*. BridgeVIEW also offers extensive online documentation, which you can access by choosing **Help»Online Reference...**.

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

Contents

Moving Your Application to BridgeVIEW 2.0 2
- Converting Your SCF File 2
- Compatibility between Versions 1.*x* and 2.0 2
- Converting VIs 3
- VI Control VIs 3
- Upgrading Toolkits 4
- Converting Boolean Data to and from BridgeVIEW 1.*x* 4

New Features 6
- Networking 6
  - Configuring the Server 6
  - Configuring the Client 7
  - Security 8
- Utility Enhancements 8
  - Tag Configuration Editor 8
  - Tag Monitor 10
- HMI G Wizard Enhancements 10
  - Creating, Editing, and Copying Tags from the HMI G Wizard 10
  - Changes to the HMI G Wizard Templates 11

BridgeVIEW™, CVI™, HiQ™, LabVIEW™, natinst.com™, National Instruments™, NI-DAQ™, SCXI™ are trademarks of National Instruments Corporation. Product names referenced in this document are trademarks or trade names of their respective companies.

 © Copyright 1998 National Instruments Corp. All rights reserved.

Servers 12
Server Browser 12
Remote OPC Servers 12
DAQ OPC Server 13
HMI Development Environment (G) 19
Multithreading 19
ActiveX Automation 20
Analysis 21
Data Acquisition 22
Translation Tools 22
General Interface 24
Changes and Bug Fixes 28
SCADA Fixes 28
G Fixes 29
Changes 30

Moving Your Application to BridgeVIEW 2.0

This section provides information about porting your BridgeVIEW 1.*x* applications to BridgeVIEW 2.0.

Converting Your SCF File

BridgeVIEW 2.0 automatically converts your BridgeVIEW 1.*x* `.scf` files. When you open an `.scf` file saved in BridgeVIEW 1.*x*, a dialog box tells you that your `.scf` file needs to be converted. Selecting **OK** brings up another dialog box, where you can select the name and location of the new `.scf` file.

Compatibility between Versions 1.*x* and 2.0

Compatibility VIs for New Server Functionality—BridgeVIEW now 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 2.0 includes Compatibility VIs for the VI Control VIs existing in previous versions. For information about how to implement the functionality from the VI Control VIs using the new server functions, open each VI Control VI and analyze the implementation of the VI Server feature. You can copy this code to your new BridgeVIEW applications.

Compatibility VIs for ActiveX Functions—In BridgeVIEW 2.0, the ActiveX functionality has expanded. The functions are more generic because BridgeVIEW now can act as an ActiveX server as well as a client. Compatibility VIs are provided for the ActiveX functions existing in previous versions. For more information about the new ActiveX

functionality, refer to the *ActiveX Automation* section in the *HMI Development Environment (G)* section later in this document.

Converting VIs

Upgrading BridgeVIEW 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 loads 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. In addition, BridgeVIEW loads into memory the components of all subVIs needing conversion.

You can estimate the amount of memory required to convert VIs by totalling the amount of memory your VIs and all of their subVIs occupy on disk. If these VIs are in VI libraries, add approximately 30 percent of the VI library size because the VIs are compressed. The conversion process might require at least that much memory and an additional 3 MB of memory to run BridgeVIEW.

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 loading and saving subVIs in the lower levels of the hierarchy. Then you can progress to the higher levels of the hierarchy. You also can choose **File»Mass Compile** to convert a directory of VIs. However, this option converts 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 if you opened the high-level VI first.

You can monitor your memory usage with the **Help»About BridgeVIEW...** option, which summarizes the amount of memory you have used.

VI Control VIs

The VI Control VIs (`vi.lib\utility\victl.llb`) have been removed from the default menu palette set and now exist as compatibility VIs. Their functionality has been subsumed by the new VI Server functions (Open VI Reference, Call By Reference, Property Node, and Invoke Node). You can find these functions in the new **Functions»Application Control** palette.

Some of the error codes passed from the VI Control VIs have changed in BridgeVIEW 2.0. In previous versions of BridgeVIEW, the VI Control VIs passed the error codes `7` and `1000`. The VI Control VIs in BridgeVIEW 2.0 pass the correct error codes, `1004` and `1003`, respectively. If a VI built in a

previous version of BridgeVIEW checks for these specific error codes, you must make modifications so the VI works in BridgeVIEW 2.0.

Upgrading Toolkits

Most existing toolkits function with BridgeVIEW 2.0 without problems. However, you need to move the VIs so they appear in the menus. With few exceptions, you can use the previous version of the following toolkits with BridgeVIEW 2.0:

- **Picture Control Toolkit for G**—You can use the Picture Control Toolkit 1.0 with BridgeVIEW 2.0 with the exception of the Draw 1-bit Pixmap VI. You can find an updated version of this VI in the `support\bridgeview\toolkits\pictctrl` folder. The Picture Control Toolkit is being updated to include this correction, and the upgrade is free to existing users.
- **Internet Developers Toolkit for G**—You can use the Internet Developers Toolkit 4.1 with BridgeVIEW 2.0, but you must delete `printvi.llb`, located in the `user.lib\internet\image` directory. The Internet Developers Toolkit is being updated to version 2.0 to include this correction, and the upgrade is free to existing users.

The following toolkits do not install VIs in a location that causes them to appear in the palettes. These toolkits are being updated to version 2.0. You can use the existing toolkits by moving VIs to `vi.lib\addons` or `user.lib`. Alternatively, you can choose **Edit»Edit Control and Function Palettes** and add them to the palette of your choice.

- Picture Control Toolkit 1.0
- Statistical Process Control Toolkit 1.0
- Proportional-Integral-Derivative Toolkit 1.0

Converting Boolean Data to and from BridgeVIEW 1.*x*

The format in which BridgeVIEW 2.0 stores Boolean data has changed since BridgeVIEW 1.*x*. BridgeVIEW 1.*x* stores Boolean data in two bytes unless the data is in an array, in which case BridgeVIEW 1.*x* stores each Boolean element in a single bit. BridgeVIEW 2.0 stores a Boolean value in a single byte, regardless of whether it is in an array. This change enables more block diagram functions to support arrays of Booleans and makes the behavior of these arrays more consistent with the behavior of arrays of numbers. The new Boolean data format affects data manipulation in code interface nodes (CINs), but BridgeVIEW 2.0 provides compatibility for existing CINs.

When you open a datalog file created in a previous version of BridgeVIEW, BridgeVIEW 2.0 prompts you to convert the file to the BridgeVIEW 2.0 format. If you choose to convert it, BridgeVIEW replaces the datalog file with data converted to the new format. If you choose not to convert the file, BridgeVIEW 2.0 returns an error and does not open the file.

If you write binary data including one or more Booleans to a file in BridgeVIEW 1.*x*, its format is different than if you write the same data in BridgeVIEW 2.0. BridgeVIEW 2.0 provides a mechanism for reading binary data written in BridgeVIEW 1.*x* and writing binary data BridgeVIEW 1.*x* can read. Five functions (Write File, Read File, Type Cast, Flatten To String, and Unflatten From String) have a **Convert 4.x Data** pop-up menu option. If you select this option, the function treats binary data as if it were written for BridgeVIEW 1.*x*. To produce data formatted for BridgeVIEW 1.*x*, use the Write File, Flatten to String, or Type Cast function. To read data formatted for BridgeVIEW 1.*x*, use the Read File, Unflatten From String, or Type Cast function. When you select the **Convert 4.x Data** option, BridgeVIEW 2.0 draws a red 1.*x* on the function to indicate it is converting data to or from BridgeVIEW 1.*x* format. To stop this conversion of data, deselect the **Convert 4.x Data** pop-up menu option.

☞ **Note** ***The* 4.x *in the* Convert 4.x Data *option refers to the version of G, and not the version of BridgeVIEW. BridgeVIEW 2.0 uses G version 5.0.***

If you have several data files with Boolean values, you can create a VI that opens these files, then writes the data to a new data file BridgeVIEW 2.0 recognizes.

In BridgeVIEW 2.0, when you load a VI last saved in a previous version of BridgeVIEW, BridgeVIEW 2.0 automatically sets the **Convert 4.x Data** attribute on the Write File, Read File, Type Cast, Flatten To String, and Unflatten From String functions. Any VIs created in a previous version of BridgeVIEW continue to function as before. When you decide your VIs need to use the new BridgeVIEW 2.0 Boolean data format, deselect the **Convert 4.x Data** attribute on each of the functions listed above. Typically, if your VIs do not need to manipulate files containing Boolean data written in a previous version of BridgeVIEW or send or receive data containing Booleans to or from VIs running in a previous version of BridgeVIEW, you should use the new BridgeVIEW 2.0 Boolean data format. Support for the previous Boolean data format might be discontinued in future versions of BridgeVIEW.

New Features

The following sections describe the new features in BridgeVIEW 2.0, including networking, utility enhancements, HMI G Wizard enhancements, servers, and the HMI development environment.

Networking

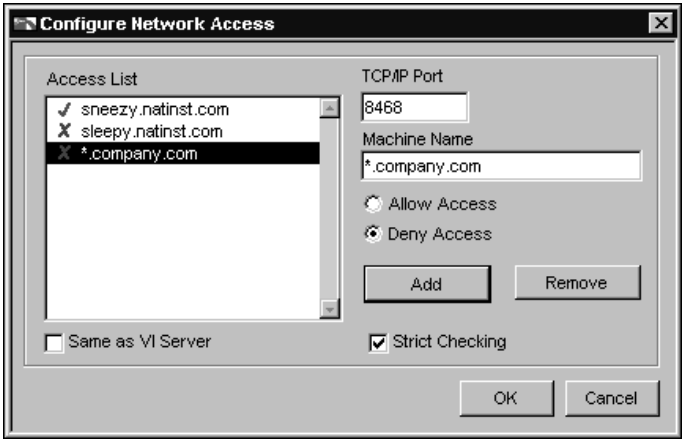
BridgeVIEW 2.0 makes it easy to create distributed applications by providing Engine to Engine connectivity between multiple clients and servers. A BridgeVIEW server machine is a computer that allows its tags to be accessed by other machines on the network. The server machine may or may not have an HMI running on it. A BridgeVIEW client machine is a computer that gets its data from tags on one or more BridgeVIEW server machines.

Providing Engine to Engine connectivity requires configuring the server and client machines, as described in the following sections.

Configuring the Server

To configure the server machine, you must specify a list of clients which are allowed network access and configure the `.scf` file so it is accessible by any network client.

To specify a list of allowed clients, select **Project»Configure Network Access...**, which brings up the following dialog box.



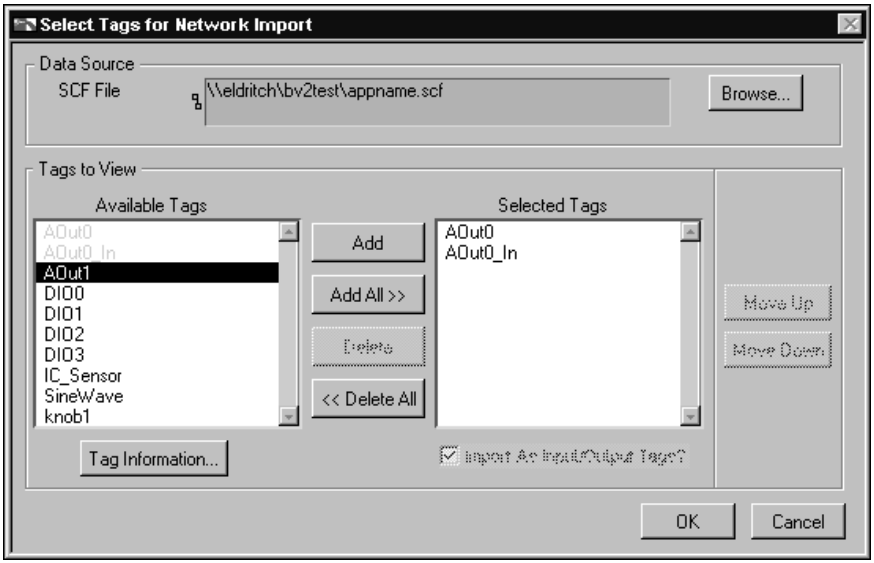
Enter the machine name or IP address of the desired clients in the Machine Name field, choose whether to allow or deny access, and click **Add** to add them to the Access List. The Access List specifies all clients which are allowed or denied access for the server machine. Clients that are allowed access are marked with a checkmark, while clients denied access are marked with an **X**.

Note ***When entering the desired clients in the Machine Name field, wild cards are supported, as illustrated in the above figure.***

To configure the `.scf` file so that it is accessible by a network client, select **Configure»Allow Network Access** in the Tag Configuration Editor. Also, make sure that the folder with the `.scf` file is shared over the network with the clients.

Configuring the Client

Configuring the client machine involves adding tags from one or more server machines to the client's `.scf` file. To add tags from a given server machine, select **File»Import Network Tags** in the Tag Configuration Editor. A dialog box pops up and prompts you for the network path to the servers `.scf` file. Then, another dialog box, shown in the following figure, pops up so you can select tags from the server machine for inclusion in the clients `.scf` file.



Select **OK** to configure the selected tags as network tags on the client machine.

Security

The Edit User Accounts dialog box, invoked from **Project»Security**, now allows you to import or export user account information. Do this by selecting **File»Import** or **File»Export**. There are two file formats supported, binary and text. While binary files contain all account information, including encrypted passwords, text files contain only User Names, Access Levels and Privileges. You can also browse a networked machine to locate a file by selecting **File»Import» Network BridgeVIEW…**.

Utility Enhancements

The following sections describe the utility enhancements made to the Tag Configuration Editor and the Tag Monitor.

Tag Configuration Editor

The following sections describe changes to the Tag Configuration Editor.

Configuring Network Tags

BridgeVIEW 2.0 supports the configuration of tags associated with a separate BridgeVIEW installation on a remote machine, which allows for client-server applications. A tag that is connected to a tag on another BridgeVIEW Engine is called a *network tag*. On the BridgeVIEW server, all tags in an `.scf` file can be networked by selecting **Allow Network Access** from the configuration menu of the Tag Configuration Editor and saving the `.scf` file.

To import network tags, open the Tag Configuration Editor on the BridgeVIEW client and select **File»Import Network Tags…**. A dialog box allows you to browse the network for an `.scf` file from which you can select tags to import. After selecting the tags you want to import, save the `.scf` file on the BridgeVIEW client and start the BridgeVIEW Engine.

Changing Tag Attributes Dynamically from the Tag Configuration Editor

In BridgeVIEW 1.*x*, the Tag Configuration Editor performed tag configuration as an offline task, so changing a tag attribute from the Tag Configuration Editor did not affect the BridgeVIEW Engine until the Engine was restarted. BridgeVIEW 2.0 supports online reconfiguration from the Tag Configuration Editor for several tag attributes.

Tag attributes are classified as either static or dynamic. Static tag attributes are configured offline and require the BridgeVIEW Engine to restart. Dynamic attributes can be configured online using the Tag Configuration

Editor or the Tag Attribute VIs, and thus do not require you to restart the Engine before your changes take effect.

If you make changes to the `.scf` file and save the file while the BridgeVIEW Engine is running, a dialog box appears asking if you want to update the Engine. If you select **Yes** and have changed any static attributes, the Engine is restarted. If you changed only dynamic tag attributes, the Engine is updated without restarting.

For a list of dynamic tag attributes, go to the block diagram of any VI and select the Tag Attribute Ring, found on the **Functions»Tag Attributes** palette, and place it on the block diagram. Click on the Tag Attribute Ring you just placed on the diagram to display a list of all the dynamic tag attributes. All other attributes you can configure in the Tag configuration Editor are static attributes.

Invoking the Tag Configuration Editor from the Tag Control

The pop-up menu for a Tag constant, control, or indicator has additional options to create, edit or copy a tag. Selecting one of those options from a user application causes the Tag Configuration dialog box to open. If the user selects **OK**, control returns to the Tag Configuration Editor, which automatically saves the new or changed tag to the `.scf` file. The change is then reflected in the tag constant, control, or indicator automatically. If the user cancels out of the Tag Configuration dialog box, no change occurs.

Appending Tags to an SCF File

You can import tags from a spreadsheet into an existing `.scf` file. If you have an `.scf` file open in the Tag Configuration Editor, the action of importing automatically adds the imported tags to the open `.scf` file.

You also can generate tags automatically (via the SCF Wizard) for a server and append it to an existing `.scf` file. The **Select Servers** dialog box that pops up when you invoke the SCF Wizard has a checkbox labeled **Append Tags to SCF?**, which is turned on by default. Disabling this checkbox creates a new configuration file which is added to the automatically generated tags.

Multiple Tag Editing

With BridgeVIEW 2.0, you can edit, copy, and delete multiple tags. To select multiple tags, hold down the <Shift> key while clicking additional tag names.

☞ **Note** ***The most efficient method for simultaneously making the same type of change for several tags is still with an external spreadsheet application.***

To edit multiple tags, select a set of tags and click the **Edit Tag(s)…** button. Select the **Edit Next Tag** button to advance to the next tag in the list. Click **OK** to close the dialog box.

To copy multiple tags, select a set of tags and click the **Copy Tag(s)…** button. This brings up the Tag Configuration dialog box with a new tag name, which is the copy of the first tag in the list of selected tags. Selecting **Copy Next Tag** takes you to the Tag Configuration dialog box for the copy of the second tag on the list, and so forth. Click **Copy Next Tag** until you get to the last tag on the list. Select **OK** to complete the operation.

To delete multiple tags, select them and click the **Delete Tag(s)…** button. This makes a trash can symbol appear next to the tag names. If you select a tag that has been marked for deletion, the **Delete** button changes to **Undelete**. You cannot edit or copy tags marked for deletion, but you can undelete them to their previous known states. All tags marked for deletion are thrown out when you save the `.scf` file. A confirmation dialog box with a list of tags to be deleted pops up before the file is saved.

Tag Monitor

The Tag Monitor contains a new feature that allows you to select the value of any Input/Output or Output-only tag and write values from within the Tag Monitor utility. A dialog box pops up when you select the value field of the tag to write to in the Display Table, and allows you to specify the new value. This new value is also written to the Engine and updated in the Real Time Database, and to your HMI as well, as long as it is running.

HMI G Wizard Enhancements

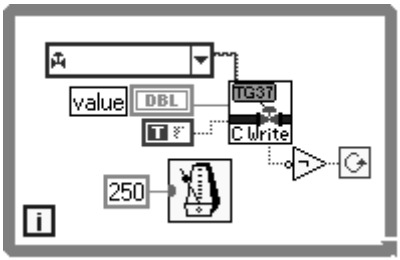
The following sections describe enhancements made to the HMI G Wizard.

Creating, Editing, and Copying Tags from the HMI G Wizard

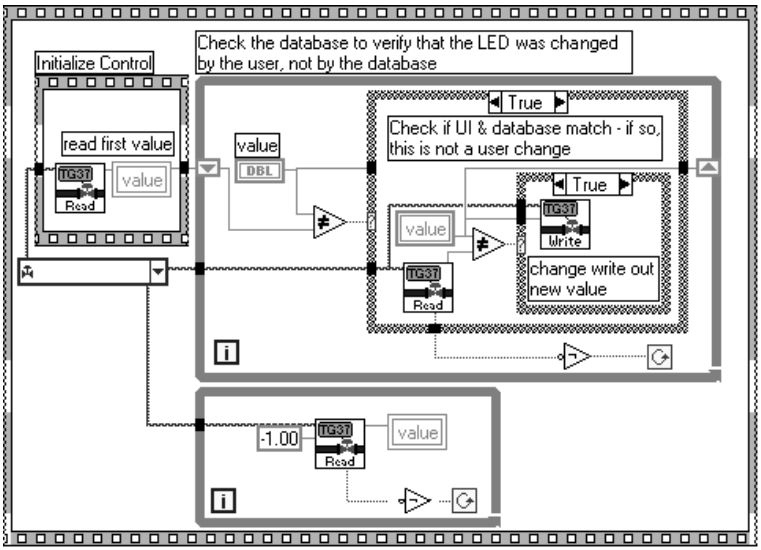
From the HMI G Wizard, you can right-click on the tag selection list and select **Copy Tag…**, **Edit Tag…**, or **Create Tag…**, to edit or copy the selected tag, or create a new one using the Tag Configuration Editor. When you select **OK**, the newly created or edited tag is automatically saved to the current `.scf` file, and you are returned to the HMI G Wizard.

Changes to the HMI G Wizard Templates

All HMI G Wizard templates which used controls for tag values in previous versions of BridgeVIEW now use control-indicators. The current value in the tag control now updates to reflect changes to the tag value. Previously, the value in the tag control was static, depending only on the user's current input value. For example, the previous version of the basic analog output template was:



The new version, with code for a control-indicator, is:



With this change, a single front panel object can both control and display the value of a given tag.

Servers

This section provides information on the Server Browser, Remote OPC Server, and the DAQ OPC Server.

Server Browser

You can view the OPC Server Items and their attributes using the Server Browser utility. Launch the Server Browser by selecting **Projects»Server Tools»Server Browser…** or pressing the **Server Browser…** button on the Engine Manager display.

When an OPC server is selected in the Servers list, you can press the **View Server Information…** button to bring up the View Server Information dialog box for OPC Servers.

The View Server Information dialog box displays general information about the OPC Server read from your local system registry, and if the OPC server supports the Server Browse Address Space interface, also displays the items available from the server and their attributes. If the OPC Server does not support this interface, the **No Items Found** checkbox and the item table appear dimmed.

Remote OPC Servers

You can use the Server Browser to configure BridgeVIEW to access OPC Servers on other machines on your network. Use this utility to select remote OPC Servers of interest, and add them to the BridgeVIEW server list.

To view the OPC Servers available on other machines on your network, press the **Browse Network OPC Servers…** button on the Server Browser.

Use the Browse OPC Servers on Network dialog box to view the OPC servers registered on other machines on your network. The Registered Remote OPC Servers list shows which remote servers have been added to the BridgeVIEW servers list. To use the server from a remote machine, use the network tree control to select one of the OPC servers shown on that machine, and press the **Add Server** >> button. The information for the remote OPC server is now stored in your local machine registry, and the server appears in your BridgeVIEW servers list with the server name format of `(`*`machine name`*`)programID`. BridgeVIEW runs the server on the remote machine when you configure a tag to use that server.

To remove one or more remote OPC server from the BridgeVIEW server list, select the servers and press the **Remove Server(s)** button.

DAQ OPC Server

The DAQ OPC Server is available with NI-DAQ 6.*x*. Your BridgeVIEW 2.0 CD contains NI-DAQ 6.1, which has a version of the DAQ OPC Server that supports most OPC features, including address space browsing. The DAQ OPC Server is intended as a replacement for the NI-DAQ Server that was available with earlier versions of BridgeVIEW. The BridgeVIEW 2.0 CD also contains a previous version of the NI-DAQ Server, so users do not need to upgrade to the DAQ OPC Server immediately.

Note ***The NI-DAQ Server will not be made available in later versions of BridgeVIEW.***

With the DAQ OPC Server, you now use the NI-DAQ Configuration utility and NI-DAQ Channel Wizard to do all configuration. The DAQ OPC Server does not use the NI-DAQ Server Configuration Utility and the earlier version of the NI-DAQ Channel Wizard used by the NI-DAQ Server.

Porting Your DAQ Application to the DAQ OPC Server

Complete the following steps to migrate to the DAQ OPC Server from the previous NI-DAQ Server.

1. Use the CCF to DAQ Conversion Wizard to import channels defined in a `.ccf` file into your NI-DAQ configuration file so NI-DAQ can recognize your channels. This conversion allows you to use these channels from any DAQ VI, or select them with the DAQ OPC Server.
2. Convert your `.scf` files that use the NI-DAQ Server to use the DAQ OPC Server instead. These steps are outlined in the *Changing Your SCF File for DAQ Applications* section later in these upgrade notes.
3. There is no utility that replaces the NI-DAQ Server Configuration utility. Instead, the NI-DAQ Configuration utility handles some of the settings, and the Tag Configuration Editor handles others, such as the creation of groups and group update rates.

CCF to DAQ Conversion Wizard

BridgeVIEW no longer requires a separate Channel Configuration File (`.ccf`). Instead, the information it contained is now stored in the NI-DAQ configuration file (`.daq`). The CCF to DAQ Conversion Wizard is provided to convert the channel configuration data to the new format.

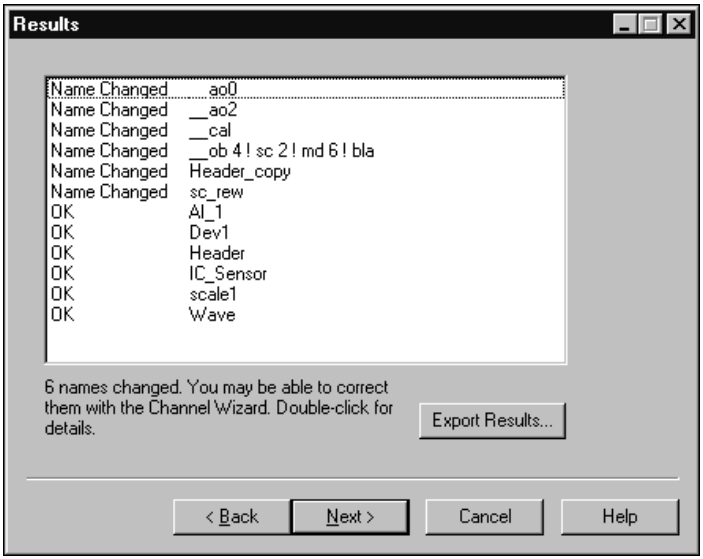
Note ***Before you convert to the new format, NI-DAQ must be configured to the physical configuration of your system.***

To run the wizard, double-click on the icon for the wizard executable and follow the on-screen directions.

The first page you encounter is the **Welcome!** page. Click the **Next** button to continue.

The second page you encounter is the **Source Files** page. You can select your old Channel Configuration File (`.ccf`) and the new NI-DAQ configuration file (`.daq`) from this page. By default, the Wizard selects the currently active `.daq` file. You might change this if you want. To browse for either file, click on the appropriate **Browse** button. This brings up a standard Open File dialog box. The Wizard does not let you continue to the next page if it cannot find one of the files you specify. When you have selected valid `.ccf` and `.daq` files, click the **Next** button to begin the conversion.

The third page you encounter is the **Results** page.



The list box at the top of the **Results** page shows the status of the conversion of each channel. Errors are listed first, followed by items whose names have been changed because of name conflicts, followed by items that were successfully converted exactly as described in the `.ccf`. If there is an error, you can obtain further details about the error by double-clicking on the row that contains the error.

Save the output list to a text file by clicking the **Export Results…** button. This brings up a standard Save File dialog box. The saved file is tab-delimited text with three columns. The first column contains the status indicator (`OK/Error/Name Changed`). The second contains the name of the channel or custom scale. The third contains error details, as shown when you double-click a row with an error.

A text string appears under the list box, summarizing errors and name changes. Names must be unique within the `.daq` file, and must follow the following rules:

- Channel names cannot include commas, semi-colons, or colons, and cannot begin with a single underscore. The list below contains keywords that are not allowed as channel names when alone or followed only by numbers, periods, and/or spaces.

ai	gpctr	amux	trigger	cjtemp	ob
ao	ctr	am	trig	dtemp	ch
di	rtsi	external	cal	mtemp	sc
do	pfi	ext	calgnd	shunt	scxi
0	1	2	3	4	5
6	7	8	9		

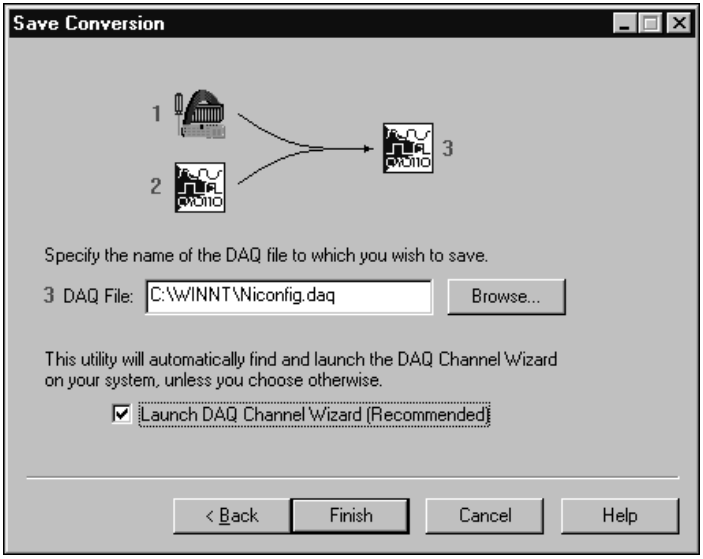
- In addition, channel names cannot begin with the following character sequences (where *x*, *y*, and *z* represent numbers).

`ob` *x* `! sc` *y* `! md` *z* `!`	`ob` *x* `sc` *y* `md` *z*
`ob` *x* `! scxi` *y* `! md` *z* `!`	`ob` *x* `scxi` *y* `md` *z*
`sc` *y* `! md` *z* `!`	`sc` *y* `md` *z*
`scxi` *y* `! md` *z* `!`	`scxi` *y* `md` *z*

- Commas, semi-colons, or colons are replaced with a single underscore. Other rule violations are corrected by pre-pending the name with a double underscore. Finally, duplicate names are corrected by appending `_copy`, `_copy2`, `_copy3`, and so on.

Click the **Next** button to continue to the next page.

The final page is the **Save Conversion** page, shown below.



You can select a different file to save your results. You might do this if you are unsure of the conversions made, or converting multiple `.ccf` files that you want to keep distinct.

To browse for the file to save, click the **Browse…** button. This brings up a standard Save File dialog box. The file is not saved until you click the **Finish** button.

By default, the CCF to DAQ Conversion Wizard launches the DAQ Channel Wizard upon completion. The Channel Wizard performs a final validation check on the converted data. At this point, you can correct any errors and rename any items whose names have changed. If there are no errors or name changes, it is still recommended that you launch the Channel Wizard for thorough validation of the data. You can choose to skip this step by de-selecting the **Launch DAQ Channel Wizard** checkbox.

Click the **Finish** button to commit the changes and optionally launch the DAQ Channel Wizard.

Changing Your SCF File for DAQ Applications

To use the DAQ OPC Server instead of the NI-DAQ Server, you must change your BridgeVIEW Tag Configuration (`.scf`) file to refer to the National Instruments DAQ OPC server. You also might need to modify the I/O group information in the `.scf` file. You can complete these steps using the export and import capabilities of the Tag Configuration Editor.

Complete the following steps to convert your `.scf` file to use the DAQ OPC Server.

1. **Export your .scf file**—Launch BridgeVIEW, and launch the Tag Configuration Editor. Export your `.scf` file that uses the NI-DAQ Server by selecting **File»Export…**. In the **Select Tag Fields for Export** dialog box, all tag files should be in the **Fields to Export** list box. Click **OK**, and select the `.txt` file name to export into.
2. **Edit your exported file using Excel**—Open your exported `.txt` file in Excel or some other spreadsheet program. In Excel, look for the Server column, which has `NI-DAQ Server` listed for each tag using the NI-DAQ Server. Replace each occurrence of `NI-DAQ Server` in this column with `National Instruments.DaqOpc`.

   The `Item` column contains the names for the different DAQ channels you are using. These item names are normally the same because you converted your `.ccf` file using the CCF to DAQ Conversion Utility. If in the process of converting channels with the CCF to DAQ Conversion utility some channels had to be renamed because of NI-DAQ channel naming rules, you can update the item names here to match the new channel names.

   The `I/O Group Rate` column contains the rate in seconds at which the OPC server updates the group. If you want to have different I/O groups at different rates, you can make those changes in this column. You also must change the `I/O Group` column to have a different name for each different I/O group rate. You can also make these changes in the Tag Configuration Editor.

Note ***The*** `I/O Group Rate` ***column might be set to*** `100` ***by default. You can correct this to a more reasonable number, such as 1 second.***

   Save your edited Excel file in tab-delimited text format by selecting **Save As…** and selecting **Tab Delimited Text** as the file type. Exit Excel.

3. **Import your edited text file**—Return to the Tag Configuration Editor and select **File»New** to create a new `.scf` file. Select **File»Import** and select the `.txt` file you saved from Excel. The Tag Configuration Editor launches the NI-DAQ OPC Server and validates the tags that are imported.
4. **Save your new SCF file**—The new tag configuration operates like the previous one, except that you are now using the DAQ OPC server instead of the NI-DAQ Server.

Other DAQ Configuration Settings

With the NI-DAQ Server, the NI-DAQ Server Configuration Utility was used to handle defining different groups and rates. This is all handled from the Tag Configuration Editor now. When you edit a tag, you can create one or more I/O groups for each server that determines the update rate for the tag. So when using the DAQ OPC Server, you can create as many I/O groups as you need for different rates, and assign each tag associated with the DAQ OPC Server to the appropriate I/O group based on rates. You also can configure the deadband for each I/O group.

The NI-DAQ Server Configuration Utility allowed you to specify a **Base AI Sample Rate**. This normally was set to the maximum of the different analog input group rates automatically, but could be entered by the user. In addition only one digital input group was supported, and you entered the **DI Sample Rate** on the Digital Channels tab. All rates are now determined via the Tag Configuration Editor I/O Group Rate configurations. You can create as many different I/O groups as you want, and different channel types can be mixed in the same I/O groups.

Some other settings were configured using the NI-DAQ Server Configuration Utility with the AI Restart Interval. This parameter determined how often the NI-DAQ Server would stop to read calibration channels such as the cold junction compensation channels for thermocouples. Now this parameter is set up on a per device basis and is configured in the NI-DAQ Configuration Utility. Launch the NI-DAQ Configuration Utility and double-click each device of interest in the **DAQ Devices** tab. In the **Configuring Device** dialog box, select the **OPC** tab and enter the **AI Recalibration Period** in seconds. This feature is disabled by default.

Unsupported Features in the DAQ OPC Server

The NI-DAQ Server included several analog input averaging options. This is not currently an option with the DAQ OPC Server.

Another feature that was supported by the NI-DAQ Server Channel Wizard was scaling based on user-defined expressions. This feature is currently not supported by the DAQ OPC Server. You need to convert any user-defined expressions into linear, polynomial, or table lookup forms for scaling to be performed by NI-DAQ.

☞ **Note** ***If you use either of these features and strongly feel that they should be included in the future, please let the BridgeVIEW team know.***

HMI Development Environment (G)

The HMI development environment (G) for BridgeVIEW 2.0 includes new multithreading and ActiveX Automation functionality, data acquisition features, translation tools, a VI Server interface, multi-step undo and redo, as well as other improvements. The *Multithreading*, *ActiveX Automation*, *Analysis*, *Data Acquisition*, *Translation Tools*, and *General Interface* sections of these Upgrade Notes describe the features added and the significant changes made between versions 1.*x* and 2.0.

To help you learn more about BridgeVIEW, there is extensive online documentation available, which you can access by selecting **Help**»**Online Reference...**.

Multithreading

BridgeVIEW 2.0 incorporates multithreading technology, enabling different parts of your application to run independently. This capability can resolve performance conflicts between user interface and data acquisition, for example. Using multithreading, you can take full advantage of computers with multiple processors.

In previous versions of BridgeVIEW, you can execute several VIs simultaneously and still respond to user input from the mouse or keyboard. To provide this capability, the execution system uses *cooperative multitasking*—each of the different activities processes one at a time and in turn. Although cooperative multitasking works well, processor time does not distribute evenly to each activity, and one activity can prevent the execution of others.

With multithreading in BridgeVIEW 2.0, the operating system can preempt a thread of execution to give processor time to another thread. Therefore, processor time is shared more evenly among the threads.

You need no special programming to use multithreading. For advanced multithreaded programming, you can use utilities such as semaphores, synchronization, and message queues.

Multithreaded Performance Profiler—The performance profiler monitors multithreaded and multiprocessor applications. You access the performance profiler by choosing **Project»Show Profile Window**. If you are executing on a multiprocessor computer, the displayed timings add the time spent on each processor. If many parallel activities exist, the times approach the number of processors multiplied by the execution time.

Multithreading and Call Library Node—On a multithreaded operating system, you can make multiple calls to a dynamic link library (DLL) or a shared library simultaneously. The library must be reentrant or use other protection techniques before it is marked as reentrant.

Multithreading and Code Interface Node—By default, code interface nodes (CINs) written prior to BridgeVIEW 2.0 run in a single thread, the user interface thread. When you change a CIN to be reentrant (execute in multiple threads), more than one execution thread can call the CIN at the same time.

Color of Code Interface and Call Library Function Nodes—In BridgeVIEW 2.0, 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.

Synchronous Display on Indicators—With multithreading, the user interface is decoupled from the diagrams that create the data to display. As a result, the user interface might discard some data if it already has been replaced by new data to draw. If you need to see all the data on your front panel, you can choose **Synchronous Display** from the pop-up menu of front panel controls. With this option selected, the diagram waits for any previous data to be drawn before sending new data to the front panel.

Synchronization VIs—You can synchronize tasks executing in parallel by using the Synchronization VIs. You also can use these VIs to pass data between parallel tasks. You access the **Synchronization** palette by choosing **Functions»Advanced»Synchronization**. This palette consists of five subpalettes containing the Notification VIs, Queue VIs, Rendezvous VIs, Semaphore VIs, and Occurrence Functions.

ActiveX Automation

BridgeVIEW 2.0 incorporates enhanced ActiveX (OLE) Automation functionality, including the ActiveX server, ActiveX Container, and an improved ActiveX client interface.

ActiveX Automation Server—Using BridgeVIEW as an ActiveX Automation server, other ActiveX-enabled applications (such as LabWindows/CVI, Excel, Visual Basic, and so on) can control BridgeVIEW. These applications can request properties and methods from BridgeVIEW and individual VIs to call VIs and pass them data, for example.

ActiveX Front Panel Objects—BridgeVIEW 2.0 front panels include a new control subpalette, the **ActiveX** subpalette, which includes two ActiveX objects: ActiveX Container and ActiveX Variant. With these new objects, you can take advantage of the ActiveX Container capability and enhance the interactions between BridgeVIEW and other applications.

- **ActiveX Container**—Use the ActiveX Container to embed ActiveX controls on BridgeVIEW front panels. You can display programmatic changes in the container on the front panel. For example, you can embed a web browser or a calendar.
- **ActiveX Variant Control and Indicator**—Use the ActiveX Variant to pass ActiveX Variant data into BridgeVIEW, which enhances ActiveX client functionality. Use this front panel object when ActiveX Variant data is converted to data BridgeVIEW can display.

ActiveX Client Automation—BridgeVIEW 2.0 provides updated ActiveX Automation Client functions, which you can use to control other ActiveX-enabled applications and ActiveX controls. To implement BridgeVIEW as an ActiveX client, use the following new functions: Automation Open, Automation Close, Invoke Node, and Property Node.

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 including 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.

Data Format—The compatibility VIs for the BridgeVIEW 1.*x* Automation functions require you to pass flattened data in the BridgeVIEW 1.*x* format. BridgeVIEW 2.0 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. For more information, see the section, *Converting Boolean Data to and from BridgeVIEW 1.x*, earlier in this document.

Note ***The* 4.x *in the* Convert 4.x Data *option refers to the version of G, and not the version of BridgeVIEW. BridgeVIEW 2.0 uses G version 5.0.***

Analysis

Signal Generator by Duration VI—The Signal Generator by Duration VI has been added to the **Analysis»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.

Data Acquisition

NI-DAQ 6.1—BridgeVIEW 2.0 ships with NI-DAQ 6.1, which supports new DAQ hardware devices on Windows 95/NT and adds several new library VIs. For more information, see the NI-DAQ `README` file.

NI-DAQ Remote Device Access Support— BridgeVIEW 2.0 supports NI-DAQ Remote Device Access, which enables you to acquire data from DAQ boards in remote computers over a network. You do not need to change your DAQ applications to support this feature.

New DAQ VIs—The following DAQ VIs have been added to the **Data Acquisition»Calibration and Configuration** palette: DSA Calibrate, Get DAQ Channel Names, Get Channel Information, and Get Scale Information.

New DAQ Examples—This release includes many new data acquisition examples, which are divided into categories to illustrate techniques, and more complete examples called *solutions*. The technique examples are based on the DAQ examples in previous releases, but BridgeVIEW 2.0 includes many new and improved examples for analog input, analog output, digital input/output, counter/timer, and more.

Solution Wizard for DAQ and Instrument I/O—You can use the Solution Wizard to generate data acquisition VIs based on user-defined parameters. You specify solutions either from a list of common solutions or by defining the desired DAQ components. The number of these solutions has increased since BridgeVIEW 1.*x*. To find the example you need, run the Solution Wizard, which you access from the BridgeVIEW dialog box or from the **Project** menu. You can find these examples by browsing the folders and VI libraries in `examples\daq`. To locate your solution, run the DAQ Solution Wizard and choose **Solutions Gallery**. You can find these examples in VI libraries in the `examples\G examples\daq\solution` directory.

Translation Tools

New tools included in BridgeVIEW 2.0 simplify the translation of text in the user interface of VIs into different languages.

Switching between Languages—In conjunction with the VI server, you programmatically can switch between two languages, such as English and Japanese. For more information about the VI Server, see the *VI Server Capabilities* item in the following section, *General Interface*.

Importing and Exporting VI Strings—The VI string export and import tool writes all the localizable strings contained in the front panel of a VI to the VI string file, a tagged text file. You can localize the following strings: VI name and description, object caption labels, free labels, default data (string, table, path, and array default data), and private data (list-box item names, graph plot names, graph cursor names, and table, row, and column headers).

Editing VI Window Titles—In previous versions, the VI window title is the same as the VI file name. Now you can customize the VI window title so it is different from and more descriptive than the VI file name. This feature is important for localized VIs because you easily can translate the VI window title to the local language; file system naming constraints do not govern the window title and the VI still is recognized by the VIs that call it.

Period and Comma Decimal Separators—In BridgeVIEW 2.0, you can control the formatting of the decimal point precisely. This feature is useful, for example, in countries in which a comma is used for the decimal point when formatting strings for instruments that require a period decimal point.

In previous versions of BridgeVIEW, you can choose to use the decimal separator or period (`.`) of the system from the **Front Panel** section in the **Edit»Preferences…** dialog box. When previous versions of BridgeVIEW use the system decimal separator and the system uses a comma (`,`), the comma is not recognized as a decimal separator by the instruments when numbers are converted to strings for instrument communication.

In BridgeVIEW 2.0, you can force a period as a decimal separator when converting numbers to strings, and vice versa, using the following functions: To Engineering, To Fractional, To Exponential, From Exponential/Fract/Eng, Format Into String, and Scan From String.

Front Panel Caption Labels—Front panel objects can have caption labels. The caption does not affect the name of the object and you can use it as a more descriptive name of the object. You also can show, hide, and change the caption programmatically with attribute nodes. Changes to the caption do not cause the VI or its callers to be recompiled, so you can localize captions without affecting the code of the VI or that of its callers.

Format Date/Time String Function—You can display the date and time in a format you specify.

General Interface

Undo and Redo—BridgeVIEW 2.0 incorporates multi-step undo and redo, which simplify the correction of mistakes made while editing. You can undo an action immediately after you have performed it, and once you undo an action you can redo it. Set the number of actions that you can undo or redo in **Preferences»Block Diagram»Maximum undo steps per VI**.

VI Server Capabilities—BridgeVIEW 2.0 now exports many of its capabilities to other applications through a new set of features collectively referred to as the *VI Server*. You can use the VI Server interface to control VIs and the BridgeVIEW application programmatically. You can load VIs dynamically, get and set attributes of those VIs, print them, save them, and so on. This feature automates routine tasks that previously were done manually.

You also can execute the server across a heterogeneous network. With the VI Server, you can set properties and invoke operations on applications written in BridgeVIEW and on VIs located on the local computer or anywhere on your TCP/IP network. With new diagram functions, you can access these capabilities within BridgeVIEW. These capabilities subsume the functionality of the VI Control VIs, which have been removed from `vi.lib\utility\victl.llb`.

Printing and Exporting VI Documentation to an RTF or HTML File—BridgeVIEW 2.0 simplifies web and help publishing for your VI documentation. In previous versions of BridgeVIEW, you are limited to printing your control and VI descriptions to a printer or a text file. With BridgeVIEW 2.0, you can print or export these descriptions to formatted file formats such as Rich Text Format (RTF) and Hypertext Markup Language (HTML). You can import RTF files into most document-publishing software or use them as the source for Help files. You use the HTML format for online documents, particularly those you intend to publish on the World Wide Web. The Print to RTF/HTML feature supports graphics in uncompressed graphics interchange format (GIF).

Custom Menus—You can create custom menus for applications built with BridgeVIEW 2.0. VIs can override existing BridgeVIEW menus and install and respond to their own menu items. For every VI you build, you can customize menus in two steps: creating the menus and responding to menu selections. The custom menus are installed only when the VI is running.

Case Structure Enhancements—The enhanced Case structure in BridgeVIEW 2.0 simplifies programming for common situations, such as state machines and handling menu selections. You now can set any frame as the default case and map multiple cases to a single frame. Additionally, you can wire strings directly to the case selector with no need for parsing.

Case structures perform a certain action based on a particular value, which is called the *selector*. With Case structures in BridgeVIEW 2.0, you can specify ranges of selector values, include negative integers as selectors, specify a default case (or action), sort cases based on the first selector value, and convert the input selector to the selector values listed in the Case structure. The enhancements made to the BridgeVIEW 2.0 Case structure do not break your existing VIs. The new options for non-Boolean cases are **Rearrange Cases...** and **Make This Case Default**.

Password-Protected VIs—You now can protect VIs with a password, which prevents viewing and editing inadvertently. By keeping the block diagram, this feature allows platform and version changes.

Configuration File VIs—The Configuration File VIs provide tools for reading from and writing to a platform-independent configuration file similar in format to a Windows initialization (`.ini`) file.

Dragging and Dropping VI Icons—BridgeVIEW 2.0 simplifies the creation of VI icons. By selecting 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.

Call Library Parameter Type Enhancements—BridgeVIEW 2.0 includes one new return type and one new parameter type. The new string return option enables functions to return a C or Pascal string. You can use the **Any BridgeVIEW type** input parameter to pass arbitrary G data types to DLLs.

Max & Min and In Range Functions Enhancements—BridgeVIEW 2.0 includes two enhanced Comparison functions: Max & Min and In Range. In previous versions of BridgeVIEW, the Max & Min and In Range functions only compare arrays and clusters as a group of elements (aggregates). In BridgeVIEW 2.0, the Max & Min and In Range functions also can compare individual elements of arrays and clusters.

Boolean Data Storage Format—BridgeVIEW 2.0 stores Boolean data in a single byte, regardless of whether it is an array. For more information, see the *Converting Boolean Data to and from BridgeVIEW 1.x* section earlier in this document.

New Preferences Options—BridgeVIEW 2.0 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.

Reorder Commands—The reorder commands (**Move Forward**, **Move Backward**, **Move to Front**, **Move to Back**) have moved from the **Edit** menu to the Reorder ring on the far-right corner of the toolbar.

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

Icon and Text Palettes—You can display the **Controls** or **Functions** palette using icons, text, or a combination of both.

Icon Editor—In the Icon Editor, you can select an area of an icon for moving, copying, or deleting. You also can use the **Edit** menu to cut, copy, and paste images from and to the icon. When you paste an image and a portion of the icon is selected, the image is resized to fit into the selection. The **Undo** button has been removed from the Icon Editor, but you can undo an action by choosing **Edit»Undo** or <Ctrl-Z>.

Seconds to Date/Time Function Change for Windows—The value of the DST element of the **date time rec** cluster returned by the Seconds to Date/Time function is not set correctly. Instead of returning whether the **date time rec** cluster has been modified for daylight saving time, it returns whether your current system settings are set to account for daylight saving time.

Save Command—The **Save** command, which you use by choosing **File»Save**, now is enabled always, regardless of whether you have made changes.

Disabling Items in Ring Controls—You can disable items in ring controls while you are editing or running a VI.

Create Constant, Control, and Indicator Improvements—When you create a constant, control, or indicator for a VI by popping up on a subVI terminal, BridgeVIEW 2.0 copies the appropriate control or indicator from the subVI, which preserves its appearance, description, and so on. If the terminal corresponds to a type definition, you create a type definition constant, control, or indicator. If the type definition is changed later, the control, indicator, or constant is updated.

Scan String for Tokens Function—Use this function to scan an input string starting at a specified offset and returning the next token found. A *token* is a substring of the input string, which is surrounded by specified delimiters or matches an element in the operators array. Typically, tokens represent individual keywords, numeric values, or operators found when parsing a configuration file or other text-based data format.

New Modes for the File Dialog Function—The File Dialog Function displays a file dialog box so you can select an existing file or directory or select the location for a new file or directory. In BridgeVIEW 2.0, the following three new **select modes** have been added so you can open any file stored in VI libraries (LLBs): select an existing file in an LLB, select a new file in an LLB, and select an existing or new file in an LLB.

TCP/IP Functions—The following TCP/IP VIs are now functions in BridgeVIEW 2.0: TCP Open Connection, TCP Create Listener, TCP Wait on Listener, TCP Write, TCP Read, and TCP Close Connection.

TCP Read now has four operating modes. A TCP Read with zero timeout in standard (default) mode now reports a timeout error if no bytes are present.

Advanced Palette Changes—New VIs for VI Server functionality and programmatic control of window menu bars have been added to the **Advanced** palette. The VI Control VIs have been replaced by the VI Server functions, which contain the functionality of the VI Control VIs. The following palettes and functions have been moved from the **Advanced** palette to the **Application Control** palette: **Printing** palette, **Help** palette, Quit BridgeVIEW function, Exit function, and Call Chain function.

File Manager Tool—BridgeVIEW 2.0 includes the File Manager Tool, which you can access by choosing **Project»File Manager**. Use this tool to organize files in VI libraries (LLBs) and directories.

Changes and Bug Fixes

This section describes SCADA-specific and G-specific changes and bug fixes in BridgeVIEW.

SCADA Fixes

- In BridgeVIEW 1.1, choosing to print from the Historical Trend Viewer resulted in an error message. This has been fixed in BridgeVIEW 2.0.
- In BridgeVIEW 1.1, invoking the Profiler would incorrectly return a `Not enough memory error`. The Profiler can be invoked and used without error in BridgeVIEW 2.0.
- A bug in the `BVRTDatabase.dll` which caused BridgeVIEW 1.1 to hang when a system was low on memory has been fixed in BridgeVIEW 2.0.
- If a VI is running in BridgeVIEW 1.1, the VI did not pop to the front if a button associated with a panel in the background was pressed. This has been fixed in BridgeVIEW 2.0.
- In BridgeVIEW 1.1, if a tag was created manually by selecting **Create Analog Tag** in the Tag Configuration Editor, the unit and scaling information was not pulled from the `.ccdb` file. This is fixed in BridgeVIEW 2.0.
- If you have multiple copies of BridgeVIEW on your computer and try installing the VI Server Toolkit, it will now allow you to select which BridgeVIEW to install under.
- If you specified scaling for an RTD in the Channel Wizard in BridgeVIEW 1.1, the scaling information was ignored and the raw values were propagated to the Engine. This works properly in the NI-DAQ server that ships with BridgeVIEW 2.0. Notice this is different from the DAQ OPC Server that also ships with BridgeVIEW 2.0.

G Fixes

The following list describes bug fixes in G language component of BridgeVIEW 2.0.

- Wiring an enumeration to the string to number functions (From Decimal, for example) no longer causes BridgeVIEW to report an error message and quit.
- When converting CVI Function Panels to BridgeVIEW, if the Function Panel file contains errors, BridgeVIEW no longer reports an error message and quits.
- Knobs with hidden digital displays set to a logarithmic scale no longer crash if you use the arrow keys to increment or decrement them.

The following list describes general improvements in BridgeVIEW 2.0.

- If an indexing output tunnel on a loop is not wired on the outside of the loop, BridgeVIEW no longer allocates storage for the array.
- The To Hexadecimal function, which previously returned eight characters, now returns only four characters for 16-bit integer and unsigned 16-bit integer data and two characters for 8-bit integer and unsigned 8-bit integer data.
- BridgeVIEW 1.*x* introduced a new behavior for type coercion of integers to enumerations. BridgeVIEW 2.0 converts signed integers according to new rules. Unsigned integers are converted with the standard BridgeVIEW rules. In particular, signed negative numbers now coerce to the first item in the enumeration.
- You now can use Sort 1D Array for paths and pictures.
- You now can assign Formula Node inputs without a corresponding output terminal.
- Unsigned numbers wired to the selector of a Case structure now coerce as if the selector also is unsigned.
- You now can use underscore in variable names in the Formula Node.
- When the Call Library Function cannot find its corresponding DLL or shared library, BridgeVIEW 2.0 reports errors that better describe the problem.
- String controls and indicators in hexadecimal display mode now print with a correct font size.
- If you attempt to read a large number of items in a byte stream file, BridgeVIEW no longer reports memory full.
- If you open the pop-up menu of a subVI and choose **Create Control** on an input or output that is a typedef in the subVI, BridgeVIEW now creates a copy of the typedef.

- You now can convert Julian dates correctly with the Date/Time To Seconds function.
- If the preference file cannot be edited (for example, it is on a read-only server or a CD-ROM), you now can change palette menu sets.
- You can use Array Max Min with an array beginning with a NaN value.
- You now can change the serial port buffer size with VISA.
- You now can use To Upper Case and To Lower Case with languages other than English.
- BridgeVIEW 2.0 corrects problems with hidden list boxes that draw in previous versions.
- When printing, all frames of a sequence now print.
- Extremely large printouts (for example, diagrams with hundreds of frames of a sequence) now print correctly.
- BridgeVIEW 2.0 corrects various page layout problems when printing.

Changes

The following list describes major changes in G language component of BridgeVIEW 2.0.

- If you run code interface nodes (CINs) and call library nodes from previous versions of BridgeVIEW in BridgeVIEW 2.0, the CINs and call library nodes run in the user interface thread. Most VIs included with BridgeVIEW that use CINs, such as the Analysis VIs, run using multiple threads if the operating system supports multithreading.
- Priority categories for parallel tasks have changed.
- With BridgeVIEW 2.0, you no longer need to provide default subroutines for CINAbort, CINDispose, CINInit, CINLoad, CINSave, CINUnload, or the new CINProperties function.

  If a user supplies those functions, those versions are used, but otherwise, they link to default functions with the return values described in the following table.

Table 1. Return Values

Function Name	Return Value
CINAbort	`noErr`
CINDispose	`noErr`
CINInit	`noErr`
CINLoad	`noErr`
CINSave	`noErr`

Table 1. Return Values (Continued)

Function Name	Return Value
CINUnload	`noErr`
CINProperties	`mgNotSupported`

- In BridgeVIEW 2.0, the storage of Booleans and Boolean arrays has changed. The storage format change affects the Read File, Type Cast, and Unflatten String functions. BridgeVIEW 2.0 stores Boolean data in a single byte, regardless of whether it is an array. For more information, see *Converting Boolean Data to and from BridgeVIEW 1.x* section earlier in this document.

321983A-01 May98