

# PROFESSIONAL G DEVELOPERS TOOLKIT

## Version 5.0

These release notes describe system requirements, give installation instructions, and contain updated information for the *Professional G Developers Toolkit Reference Manual*.

## Contents

---

Required System Configuration.....	2
Installation and Configuration .....	2
Installation .....	2
Configuration.....	3
Changes to the Professional Developers Tools.....	3
New Tools .....	3
Compare VIs .....	3
Compare Hierarchy .....	4
Source Code Control»Compare Files .....	4
Documentation Tool .....	4
Changes to Existing Tools.....	4
Source Code Control.....	4
File Manager Tool.....	5
Manual Clarifications and Additions.....	6
VI Comparison .....	6
Compare Hierarchies.....	6
Compare VIs (Graphical Differencing) .....	10
Source Code Control»Compare Files .....	11
Documentation Tool .....	12
Source Code Control Tools .....	14
QuickStart Guide to Using the SCC Tools .....	14
Support for Third-Party Tools.....	16
ClearCase Support.....	17
Built-in System Configuration .....	20
Managing Files with the Same Name.....	21
SCC File Wizard .....	21
Frequently Asked Questions about the SCC Tools.....	22

# Required System Configuration

---

The Professional G Developers Toolkit 5.0 works with LabVIEW 5.0 or later and BridgeVIEW 2.0 or later. If you have an older version of LabVIEW or BridgeVIEW, you must upgrade to use this version of the toolkit.

This toolkit is available for all platforms except Windows 3.1. Windows 3.1 is not supported because the Source Code Control (SCC) tools require that virtual instruments (VIs) be stored in individual files rather than in libraries (LLBs). You can use the File Manager tool to convert LLBs to directories.



## Note

*Although you cannot use this toolkit under Windows 3.1, you still can develop for customers who need Windows 3.1 support. You can use Windows 95 or NT as your development platform and save your VIs in individual files. When you need to send software to a Windows 3.1 user, save your VIs in LLBs.*

# Installation and Configuration

---

You can install the Professional G Developers Toolkit on the following platforms:

- Windows 95 and NT
- Macintosh and Power Macintosh
- Solaris for Sun SPARCstation
- HP-UX



## Note

*Some virus detection programs might interfere with the installer program. Turn off the automatic virus checker before running the installer. After installation, check your hard disk for viruses and turn on the automatic virus checker.*

## Installation

The Professional G Developers Toolkit CD contains the installer programs for all of the platforms listed above. Run the installer program for your platform. The following list includes the names of the setup files and the directories in which they are located:

- **(Windows 95 and NT)** `setup.exe` in the `win95_nt` directory
- **(68K Macintosh)** ProDev 68K Installer in the 68K Macintosh folder
- **(Power Macintosh)** ProDev PM Installer in the Power Macintosh folder
- **(Solaris1)** `install` in the `solaris1` directory

- **(HP-UX)** `INSTALL` in the `HPUX` directory
- **(Solaris2)** Use `pkgadd -d .` in the `solaris2` directory

When prompted for the destination directory, select the directory in which you have BridgeVIEW or LabVIEW installed.

Also, to use the Source Code Control tools, you must decide if you want to install as a user or as an administrator. Usually only one person from a development team should be designated as the administrator. The administrator installation includes everything the user installation does, with the addition of the **Source Code Control»Administration** command that allows the administrator to set up the SCC system for use by the development team.

## Configuration

After installation is complete, the administrator must set up the Source Code Control system for the other users. All users must perform local configuration. See the [Source Code Control Tools](#) section of these release notes and the *Administration (Administrator Only)* and *Local Configuration (All Users)* sections in Chapter 11, *Source Code Control Tools*, of the *Professional G Developers Toolkit Reference Manual* for more details.

## Changes to the Professional Developers Tools

---

This sections describes the new tools and updated features for the Professional G Developers Toolkit, version 5.0.

### New Tools

The following tools have been added to the Professional G Developers Toolkit.

#### Compare VIs

You can use the **Project»Compare VIs** command to graphically compare two VIs. You can select options to control the types of differences you want to detect and view. For example, you can filter out cosmetic changes such as objects being moved or resized. When you compare the VIs, a dialog displays a summary of the differences. If you select an item from the summary, **Compare VIs** displays and highlights the differences between the two VIs.

For more information about the Compare VIs tool, see the [Compare VIs \(Graphical Differencing\)](#) section of these release notes.

## Compare Hierarchy

You can use the **Compare Hierarchy** command to compare two hierarchies of VIs. Any file with the same name in both hierarchies is compared. After the comparison completes, **Compare Hierarchy** displays a summary of the differences. You can select a set of VIs that have differences and visually compare them using the Compare VIs tool.

For more information about the Compare Hierarchy tool, see the [Compare Hierarchies](#) section of these release notes.

## Source Code Control»Compare Files

You can use the **Source Code Control»Compare Files...** command to compare files from projects under Source Code Control (SCC) with the local versions of those files.

For more information about the Compare Files tool, see the [Source Code Control»Compare Files](#) section of these release notes.

## Documentation Tool

You can use the **Project»Documentation Tool** command to create documentation for multiple VIs. You can select hierarchies of VIs or files from disk and add them to a list of files. You can use the Documentation Tool to then create documentation for those VIs in HTML, Rich Text Format (RTF), source material for online help files, or you can print the material directly to a printer.

For more information about the Documentation tool, see the [Documentation Tool](#) section of these release notes.



**Note**

*The **Project»Documentation Tool** described in this section replaces the **Print Hierarchy** tool described in the *Professional G Developers Toolkit Reference Manual*.*

## Changes to Existing Tools

### Source Code Control

The following changes have been made to Source Code Control (SCC):

- The SCC tools now support ClearCase on Solaris 2. ClearCase is a standard SCC system on UNIX. If you have ClearCase installed, the Administration dialog box lets you select ClearCase as a source code control backend. All SCC tools works with the same interface as described in the *Professional G Developers Toolkit Reference Manual*.

See the [ClearCase Support](#) section of these release notes for information about configuration ClearCase for use with the SCC tools.

- SCC now can manage files with same names located in different directories, unlike the previous version of the SCC tools. When you add a VI to SCC and a VI with the same name already exists in a different location, you are alerted, but you still can add the file.
- Edit Project now uses a wizard interface to update the list of files. The interface for project editing has been updated to make it easier to see the file list differences between your local VI hierarchy and the version under SCC. If your hierarchy differs from the version under SCC, a dialog summarizes the differences and gives you the option of using the SCC file wizard to resolve those differences. You can use the file wizard to decide whether to add files to SCC, remove files from the SCC project, or move files to reflect the location of your local hierarchy.
- The Administration dialog box has options for setting file system locking and internal locking. When you check files into SCC, they are locked to prevent modifying files without checking them out. In previous versions, this was done using the **Lock** option in the Show VI Info dialog box. In the new version of the SCC tools, the administrator has the option of using file system locking instead. When you check files in, the files are marked as read-only in the file system. When you check them out, they are changed to read-write.

## File Manager Tool

The File Manager tool has been revised and is now part of the standard LabVIEW and BridgeVIEW distributions.

The File Manager tool now has options for converting directories to VI libraries, which simplifies packaging files for distribution, particularly to systems running Windows 3.1. In addition, it now has an option to create folders or VI libraries directly.

# Manual Clarifications and Additions

---

This section is a supplement to *Professional G Developers Toolkit Reference Manual* and provides information on the enhancements to the Professional Developers tools in version 5.0.

## VI Comparison

You can use the VI comparison tools to manage different versions of VIs as you develop large applications. When you make changes to a VI, you might want to compare the VI to an older version to verify the changes you have made. Also, when multiple users are working on the same VIs, you might need to view two versions of the same files to merge changes made by different users.

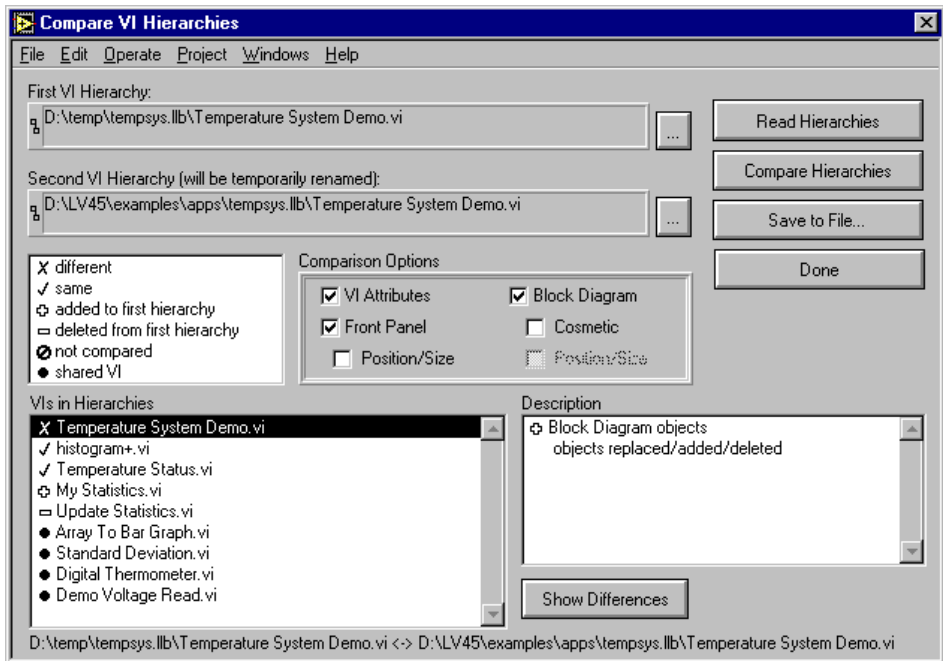
G provides several tools for comparing VIs, which are described in the following sections.

- **Compare Hierarchy**—compares two different versions of the same hierarchy of VIs
- **Compare VIs**—compares two VIs
- **Source Code Control»Compare Files**—compares the local versions of files with the versions under Source Code Control

## Compare Hierarchies

You can use the **Compare Hierarchy** command to compare two hierarchies of VIs. Any file with the same name in both hierarchies is compared. After the comparison completes, **Compare Hierarchy** displays a summary of the differences. You can select a set of VIs that have differences and visually compare them using the Compare VIs tool.

To compare two VI hierarchies, select **Project»Compare VI Hierarchies...** Use the Compare VI Hierarchies dialog box, shown in Figure 1, to select two hierarchies to compare.



**Figure 1.** Compare VI Hierarchies

Select two VI hierarchies to compare by entering the path to the top-level VI. Use the ... button to open a file dialog box to select a VI from the file system. You then can use the **Compare Hierarchies** button to compare VIs in the hierarchies. VIs with the same name but different paths are compared and categorized as being either the same or different. VIs with the same name and path are considered shared VIs. All other VIs are categorized as either added or deleted from the first VI hierarchy. The VIs and a symbol indicating how they are categorized are displayed in the **VIs in Hierarchies** listbox. As you select different VIs in the listbox, a more detailed description appears in the **Description** listbox. To view the differences on the screen, double-click an item in the listbox or use the **Show Differences** button.

The **Read Hierarchies** button categorizes the VIs in the hierarchy. VIs that need to be compared are categorized as *not compared*. You can then selectively compare individual VIs by double-clicking an item or using the **Show Differences** button.

To abort comparisons of large VI hierarchies, use the **Operate>Stop** menu item or the <Ctrl-.> (Control and the period) key combination.

## Comparison Options

You selectively can find differences in the front panel and block diagram of a VI as well as VI attributes such as settings in VI Setup. For the block diagram, you can choose to ignore cosmetic differences. A *cosmetic change* is any change that does not affect the execution of the block diagram. Furthermore, you can choose to ignore the position and size changes in front panel and block diagram objects. Note that position and size changes include movement of an object from front to back and vice versa.

## Show Differences

Differences are shown by tiling the front panels and block diagrams of the two VIs in the **Differences** window.

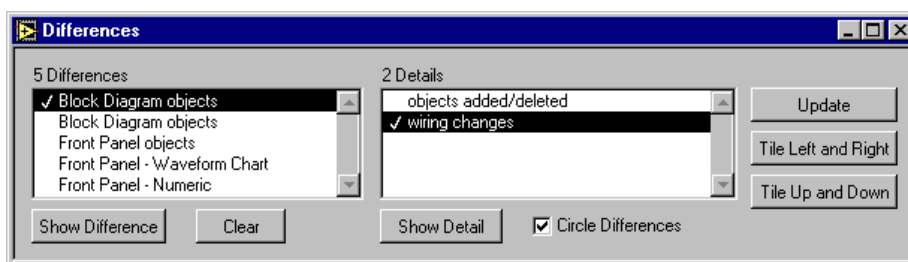
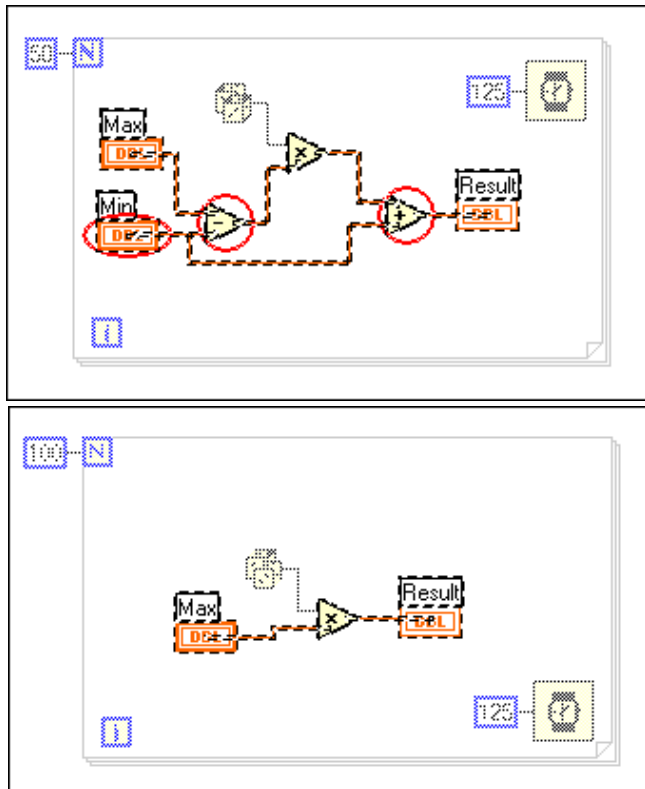


Figure 2. Differences Window

The **Differences** window includes a list of differences and details of the selected difference. To highlight a difference, double-click an item in the differences list, or select an item and click the **Show Difference** button. To highlight a detail, either double-click an item in the details list or select an item and click the **Show Detail** button. A check mark indicates the items you selected. You also can use the **Tile Left and Right** and the **Tile Up and Down** buttons to tile the windows of the two VIs you are comparing. Use the **Clear** button to clear the differences list. If you have made edits after a comparison, some of the differences might be stale; therefore, use the **Update** button to again compare the two VIs and update the differences list. You also can show the **Differences** window by selecting **Project»Show Differences**.

When you highlight a difference, objects that are part of the difference are selected. The **Circle Differences** checkbox option allows you to draw a red circle around the object(s) that has changed. An example of a block diagram difference is shown in Figure 3.





**Figure 3.** Block Diagram Difference

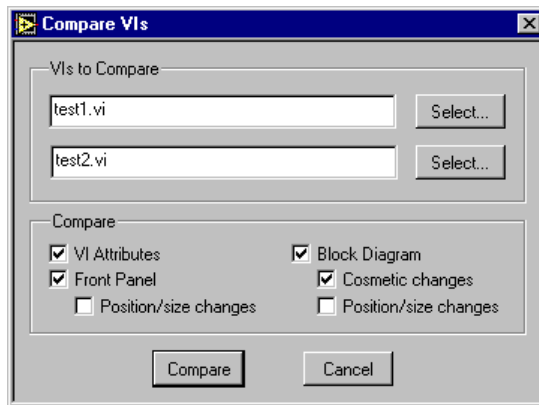
In Figure 3, the circled objects are inserted into the diagram. Objects selected but not circled are not differences. They are selected as *anchor objects* to provide reference points for the difference. Objects that appear dimmed are not part of the difference.

## Compare VIs (Graphical Differencing)

When developing applications you might have multiple versions of the same VIs. The Compare VIs feature, also called Graphical Differencing, helps you track changes in your application by comparing multiple versions of a VI. This becomes especially important as your project grows and involves more developers.

You can use the **Project»Compare VIs** command to graphically compare two VIs. You can select options to control the types of differences you want to detect and view. For example, you can filter out cosmetic changes such as objects being moved or resized. When you compare the VIs, a dialog displays a summary of the differences. If you select an item from the summary, **Compare VIs** displays and highlights the differences between the two VIs.

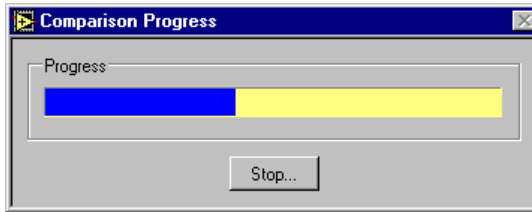
To compare two VIs, select **Project»Compare VIs...** Use the following dialog box to select two VIs to compare.



**Figure 4.** Compare VIs Dialog Box

The **Select...** button opens a dialog box to select a VI by name. You only can select VIs that are already loaded into memory.

Comparing very large VIs can be lengthy. You can cancel the comparison of two VIs through the **Comparison Progress** dialog box, shown in Figure 5.



**Figure 5.** Comparison Progress Dialog Box

The progress bar indicates the steps in the comparison algorithm, not the number of differences left to find. When the comparison is complete, the front panels and block diagrams of the two VIs are displayed in the **Differences** window, as shown in Figure 2.

## Comparison Issues

Because G cannot load two VIs with the same name, you must rename your VIs to compare them. When the **Compare VI Hierarchies** tool compares two VIs, the first VI loads as is, and the second VI is moved to the temporary directory with a `cmp .` prefix. When using the **Compare VIs** tool you must rename the appropriate VI to load the two VIs into memory.

However, renaming the VIs does not affect the name of subVIs. Because the **Compare VI Hierarchies** tool does not rename the subVIs, *the renamed VI will link to the subVIs loaded by the first VI.*

## Source Code Control»Compare Files

You can use the **Source Code Control»Compare Files...** command to compare files from projects under Source Code Control (SCC) with the local versions of those files.

To compare VIs under SCC, select **Project»Source Code Control»Compare Files....** You can use the SCC Compare Files dialog box, shown in Figure 6, to compare the project files on the master directory with the project files on the local directory.

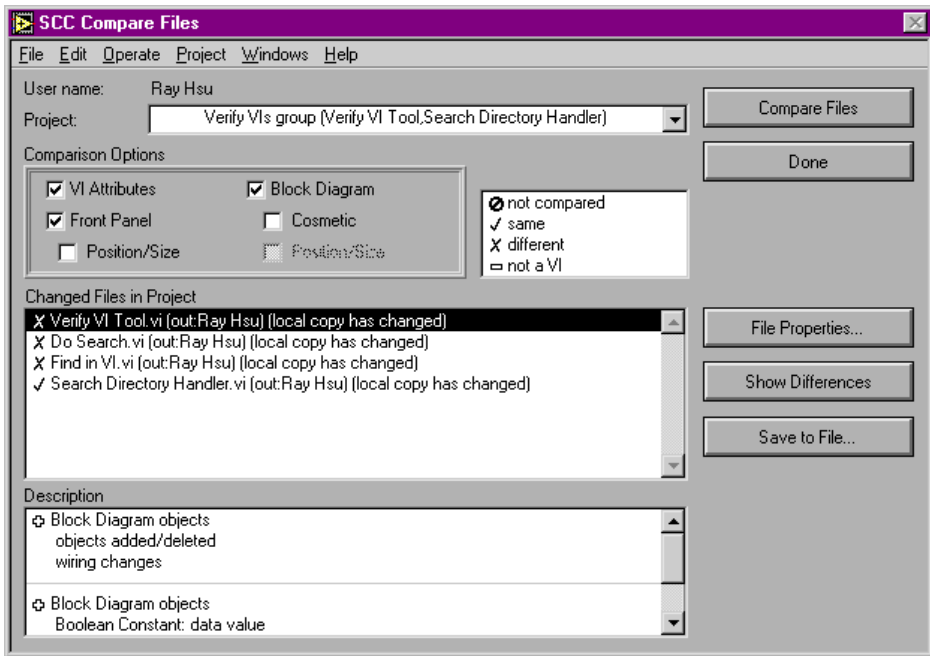


Figure 6. SCC Compare Files Dialog Box

After you select a project, the SCC Compare Files dialog box displays all changed files between the master copy and local copy. You then can use the **Compare Files** button to compare the changed files. Only VI files are compared. Other external files are not compared. This tool is similar to the **Project»Compare VI Hierarchies** tool. See the [Compare VIs \(Graphical Differencing\)](#) section for additional details.

## Documentation Tool

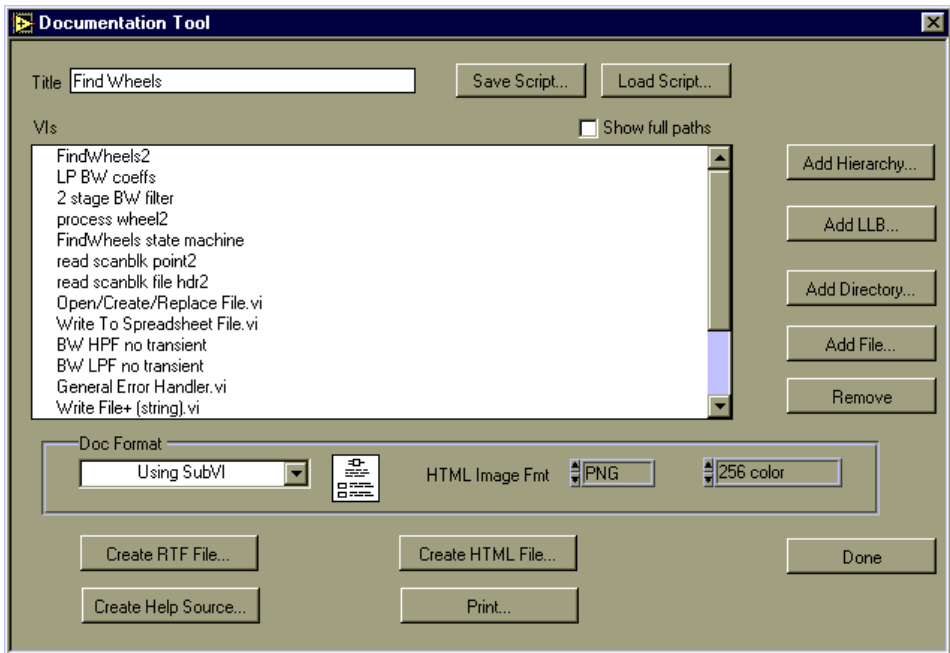


### Note

*The **Project»Documentation Tool** described in this section replaces the **Print Hierarchy** tool described in the *Professional G Developers Toolkit Reference Manual*.*

You can use the **File»Print Documentation** command to create documentation for a single VI, either by printing it directly to a printer or by creating source material for online help, web pages, or word processors.

You can use the **Project»Documentation Tool** command to create documentation for all of the VIs in your applications. The Documentation Tool dialog box is shown in Figure 7.



**Figure 7.** Documentation Tool Dialog Box

Use the **Add Hierarchy...**, **Add LLB...**, **Add Directory...**, and **Add File...** buttons on the right side of the dialog box to add VI hierarchies from memory, add all of the files in a directory or LLB, or add individual files from disk. The files are listed in the listbox in the dialog box after they are added. The order of the files controls the order that the VIs appear in the resulting documentation. You can reorder the VIs by selecting files from the list and dragging them to the desired location. If the paths are too long to view, you can deselect the **Show Full Paths** checkbox at the top of the dialog box.

When you have the files in the desired order, select one of the following buttons to create the documentation:

- **HTML**—With HTML, you can make your documentation accessible from the World Wide Web. Images are written to external files in .PNG, .JPEG, or .GIF format based upon the format you choose in the Documentation Tool dialog box. See the *Printing/Exporting Control and VI Descriptions to an RTF or HTML File* section in Chapter 5, *Printing and Documenting VIs*, in the *G Programming Reference Manual* for more information about these formats.
- **Rich Text Format (RTF)**—RTF is a standard format that is supported by a number of word processors, including Microsoft Word.

- **Online Help Source**—This produces an RTF file that is formatted for online help (.hlp) files. All images are written out as external bitmaps, and a help project and table of contents are created. These files can then be compiled using a help compiler. See the *Creating Your Own Help Files* section in Chapter 5, *Printing and Documenting VIs*, in the *G Programming Reference Manual* for information about compilers for different platforms.
- **Printer**—Select this item to print documentation to a printer.

You can use the **Save** button at the bottom of the dialog box to save the settings and list of files you create to a file. If you select the **Load** button, you can restore the settings from the file you created using the **Save** button.

## Source Code Control Tools

The Source Code Control (SCC) tools make it easier to share work and coordinate access to VIs. They provide a complete system for controlling, accessing, and documenting the different revisions of VIs that make up your projects. These tools are described in detail in the *Professional G Developers Toolkit Reference Manual*. The following sections describe how to get started quickly with these tools and also describe several enhancements to these tools that are not described in the manual.

### QuickStart Guide to Using the SCC Tools

Following is a brief summary of the steps for configuring and using the SCC tools. For more information on any of these steps, see the *Professional G Developers Toolkit Reference Manual*.

1. **Administrator sets up system**—At least one user in your workgroup should be selected to administer the SCC system. This user should select the Administration option when installing the Professional Developers Toolkit.

The administrator is responsible for initializing the SCC system and setting up system-wide preferences. You do this from the **Project»Source Code Control»Administration** dialog box. When you set up the SCC system there are a couple of decisions you will need to make:

- **Which storage system do you want to use?** The SCC tools can manage files or they can integrate with a third-party tool. See the [Support for Third-Party Tools](#) and the [Frequently Asked Questions about the SCC Tools](#) sections for help in deciding which system to use.
- **Will users be developing on multiple platforms or a single platform?** This decision might help decide the storage system you will use, as well as how to configure that system. For

multiplatform development, we recommend the built-in system. For single platform development you can use either the built-in system or one of the third party SCC tools that are supported.

Once the system is up and running, there is typically not a lot of work involved unless you want to change access privileges for files or limit access for specific users.

If you are using ClearCase, refer to the [ClearCase Installation Instructions](#) section.

2. **Users configure the SCC tools to access the system**—The main decision users must make is where they will have a *working directory*. The working directory is a directory on your local system where all VI development takes place. As you retrieve files from SCC, they are retrieved to this directory and its subdirectories. You cannot add files to SCC which are outside of this directory.

Also, if you are using the Built-in system, it is important to understand the difference between the working directory, which is where you do your work, and the master directory, which is where SCC stores the latest versions of files. Users will not modify the master directory except when they check in files or create projects.

With ClearCase, your work directory is the same directory as the master directory. ClearCase uses a virtual file system model that allows multiple users to have a unique view and set of files in the same directory. See the [ClearCase Installation Instructions](#) section for detailed information.

3. **Users set up VIs for use with the SCC tools**—As mentioned above, all files that the user will work with must be in the working directory or one of its subdirectories. In addition, files that will be stored under SCC cannot be stored in LLBs. LLBs are not flexible enough for them to work well with an SCC system because they do not allow for easy access to individual files. You can use the **Project»File Manager** tool to convert your LLBs to directories of VIs.

In converting LLBs to directories, you may find that you have some files with names that cannot be managed by the file system (i.e.: if it contains a separator character such as \, /, or : ). Consequently, you might have to rename of some files and update references within your hierarchy.

If you need to support Windows 3.1, see the [Frequently Asked Questions about the SCC Tools](#) section for suggestions on the best strategy for handling development.

4. **Create SCC projects**—A project in the SCC tools corresponds to a hierarchy of VIs. To create a project, first open the top-level VI of a hierarchy. Then select Project»Source Code Control»Project and click the New Project button. The subsequent dialog box lets you select your VI hierarchy, which can then be added to SCC.

You can add other project-related files such as documentation and shared libraries by using the **Extra Files** button in this dialog box.

5. **Use SCC commands to access project files**—At this point, you can begin to work with files under SCC. All commands related to SCC are in the **Project»Source Code Control** menu. Following is a brief description of the remaining commands:
  - **Retrieve Files**—Use to copy files from SCC to your working directory.
  - **Checkout Files**—Use to checkout files to your working directory so that you can modify them.
  - **Checkin Files**—Use to copy a file from your working directory to SCC so that other users can access it.
  - **Compare Files**—Use to compare the files in your working directory with those under SCC. This command is described later in these release notes.
  - **Advanced**—Use to create reports, view the history of files under SCC, access previous versions of files, and delete files.

See the *Professional G Developers Toolkit Reference Manual* for more information about these commands. In addition, the following sections provided updated material and clarifications about using the SCC tools.

## Support for Third-Party Tools

The Source Code Control (SCC) tools are designed to work with a variety of industry standard third-party tools for managing source code. You can either use the built-in system, Visual SourceSafe for Windows, or ClearCase for Solaris 2. Support for other third-party tools might be added in the future based upon demand.

The Built-in system and Visual SourceSafe are described in the *Professional G Developers Toolkit Reference Manual*. ClearCase for Solaris 2 is described in the following section.



## ClearCase Support

ClearCase is a standard Source Code Control system on UNIX. If you have ClearCase installed, you can select ClearCase for source code control in the Administration dialog box. All Source Code Control tools work with the same interface as described in the *Professional G Developers Toolkit Reference Manual*.

## ClearCase Installation Instructions

The Professional Developers version provides the interface for using ClearCase in the G development environment. The administrator must create a ClearCase Versioned Object Base (VOB) and an associated storage space outside of G before users can use source code control with ClearCase. When the administrator sets up the source code control system from inside G, he or she must provide the VOB directory as the master directory. Because ClearCase uses a virtual file system, users also must use this VOB directory as their local directory.

In addition to providing the correct local directory, users must set up a ClearCase view, and make sure it is set every time they launch LabVIEW or BridgeVIEW. To emulate the model of having local copies of files, users must set up unique labels to mark current working versions of files. When working under this model, users must edit their views to show labeled files before most recent versions.

Once the administrator and users configure their ClearCase systems, ClearCase is invisibly integrated into the G source-code control interface.

### Setup—Administrator

1. Install ClearCase if it is not already on your system. Refer to the ClearCase documentation for instructions.
2. If you want to create public VOBs, either set up a registry password or locate the existing registry password.
3. Create the VOB.
  - a. Set your `umask`. For shared VOBs, the suggested `umask` setting is 2.
  - b. Create a mount point directory, such as `/vobs`.
  - c. Create or choose a directory to store the VOB database, such as `/vobstore`.
  - d. Type `cleartool mkvob -public -tag/vobs/vobname /vobstore/vobname.vbs`, where `vobs` is the mount point directory and `vobstore` is the directory where the VOB database is stored.
  - e. Enter registry password (from step 2) if creating a public VOB.

- f. Enter comments, if any.
- g. Restore your `umask` to its original setting.
4. Mount the VOB. For example, type  

```
cleartool mount /vobs/vobname
```
5. Provide the name of the VOB (for example, `/vobs/vobname`) to users.
6. For ClearCase, you must specify that VIs are binary files and decide if you want to compress files. For each file extension that your VIs use, you must define an association of these extensions with their file type. For most G applications, this includes VIs (`.vi`), controls (`.ctl`), and globals (`.glb`). If you use any other extensions, you must define file types for those as well.
  - a. Choose a file type for VIs, such as `file`, `compressed_file`, or `binary_delta_file`. You also can define your own file type and type manager (see the ClearCase documentation). National Instruments recommends `compressed_file` because `binary_delta_file` storage does not work as well.
  - b. You specify this association in a *magic file* in one of the paths in the *magic path*. See ClearCase documentation to determine where the magic path is.
  - c. Somewhere in the search path for magic files, create a file with a `.magic` suffix.
  - d. Add the line: `compressed_file : -name "*.vi" ;`  
Replace `compressed_file` with your chosen file type.
7. Set up a view. Instructions for setting up a view are in the [Setup—User](#) section.
8. Set the view, and launch LabVIEW or BridgeVIEW.
9. Select **Project»Source Code Control»Administration**.
10. Select ClearCase as the source code control system.
11. The ClearCase configuration window opens.
12. Set VOB directory (e.g. `/vobs/vobname`).
13. Click **OK** in configuration window.
14. Click **OK** in administration window.

## Additional Administrator Setup—Optional

You must complete these steps if you want to add VIs to ClearCase.

1. Complete local configuration (See the [Setup—User](#) section.)
2. Set your `umask` to 2 to give other users in your group access to any new directories you create.
3. Set a view and launch LabVIEW or BridgeVIEW.
4. Copy files to the VOB directory. Use **Project»File Manager** or copy them using UNIX commands. The file manager tools is useful because it can help to convert LLBs to VIs. (LLBs are not supported by the SCC tools).
5. Open the top-level VI for a new project.
6. Create a new project, using **Project»Source Code Control»Project**.
7. Select **New Project**.
8. Select the top-level VI from the pop-up menu near the top of the dialog box. It will automatically assign a name for the project, although you can change it if you want to. Click **Save**.
9. The VI hierarchy and its subVIs are now in source code control. Repeat this process for additional hierarchies.

## Setup—User

1. Find out the VOB directory name from the administrator(s). For example, `/vobs/vobname`.
2. Create a view.
  - a. Set `umask` (typically 2 for shared files). See ClearCase documentation for more details.

```
cleartool mkview -tag viewname viewstorage  
(for example, cleartool mkview -tag george  
/viewstore/george.vws)  
(each user should have his own view)
```
  - b. Restore `umask` to original value.
3. Set your view. For example, `cleartool setview viewname`.
4. Choose a label to mark current working versions of files. Your userid in all uppercase letters is a good choice. (Your label must be unique among all who will access a particular VOB).
5. Edit your view to show versions with this label.
  - a. Edit your view. With your view set, execute `cleartool eds`.
  - b. Using the text editor (launched by `cleartool`), add the line:  
`element * LABELNAME`, replacing `LABELNAME` with your chosen label to the view configuration spec just after the line  
`element * CHECKEDOUT`.

6. Launch LabVIEW or BridgeVIEW. You must set the view every time you launch LabVIEW or BridgeVIEW.
7. Select **Project»Source Code Control»Local Configuration**.
8. Choose ClearCase as your source code control system.
9. Enter the VOB directory name given to you by the administrator in step 1.
10. Enter the label name from step 4.
11. Click **OK**.
12. Enter your local working directory, which must be the same as the VOB directory.
13. Click **OK**.
14. To freeze your working set of VIs so that changes made by other users do not affect your work until you retrieve them, select **Project»Source Code Control»Retrieve Files** and retrieve the files you want to freeze. When you select **Retrieve Files**, G indicates if the files in your view are out of date and if retrieving files will update your view by showing a snapshot of the latest versions of those files.

## Built-in System Configuration

If you select the built-in system from the Administrator dialog box, you have a set of options for deciding how files are locked for each user. Locking prevents users from accidentally modifying files without first checking them out. The Built-in Administration dialog box gives you the option of using file system locking or internal locking.

- **File System Locking**—For many sites, this is the appropriate selection. With File System Locking, as you check files in they are marked as read-only in the file system. As you check them out, they are changed to read-write. This works well for most developers.
- **Internal Locking**—With this option, as you check files in they are locked by LabVIEW or BridgeVIEW. If you open the Show VI Info dialog box, you will see that the lock option is turned on. When you check files out they are automatically unlocked. This is the best choice for multiplatform development (i.e.: Windows, Macintosh, and/or UNIX development). When you bring a VI from another platform, it must be recompiled and saved. With internal locking, you can actually save the VI. The internal locking just prevents you from making edits to a VI. Note that the SCC tools can distinguish between real changes and simple modifications such as a recompile that occurs when loading the VI.

In addition to these locking options, you must select a master directory, which is where the Source Code Control tools store the files you add to SCC. It's important that all users have access to this directory. If you want users to be able to modify files, then they need to have read write access to this directory.

In addition, under Windows 95/NT, each user should connect to the volume containing this directory and map that volume as a Network drive (i.e.: as D:\ rather than as a UNC path of the form \\machine\volume). UNC filename support varies slightly between Windows 95/NT, and consequently the SCC tools do not support UNC filenames.

## Managing Files with the Same Name

The manual incorrectly indicates that you cannot manage files that have the same name under SCC. In general, you should be cautious when working with files with the same name because it is easy to accidentally link the wrong version of a file to a hierarchy. The SCC tools detect if you accidentally link the wrong version of a file when you edit a project using the **Project»Source Code Control»Project** dialog box. You have the choice to avoid changing the project by moving the existing file in SCC to the new location, or adding the new file as file with the same name in a different location.

## SCC File Wizard

When you edit a project using **Project»Source Code Control»Project**, the SCC tools compare your hierarchy to the version that is already under SCC. If there are differences in the list of files, a File Wizard walks you through the process of updating SCC to reflect local changes.

This wizard first presents a dialog box summarizing the differences between the local version of your hierarchy and the version that is under SCC. This wizard detects whether one hierarchy has files that the other does not, and whether files are in different locations in the local hierarchy from the hierarchy under SCC. If you choose to edit the project, a series of dialog boxes describe these differences in more detail and give you the option to change the project or ignore the difference. No changes are made unless you select **Save** in the final dialog box, which summarizes the new list of files.

## Frequently Asked Questions about the SCC Tools

### How do I decide between the Built-in system, SourceSafe, and ClearCase?

In many cases, the decision to use one of the third-party tools (SourceSafe for Win95/NT or ClearCase for Solaris 2) might be based on the fact that your company has standardized on one of these tools. If not, you must decide what strategy you want to use for managing your files.

Fortunately, the SCC user interface in LabVIEW and BridgeVIEW is the same regardless of which storage system you select. The main difference between the three systems are cost, control capabilities, and platform support.

The built-in system is available on all platforms, and does not have any additional costs associated with it. It has most of the functionality of the other systems. One key difference is that it works by copying files to and from a shared network directory that all users in a development group can access. It currently does not provide tools for limiting user access to files. If you need to control the degree of access various users have, such as read-only and read-write, you can get some of this functionality using file system permissions. However, this is not as powerful as the control features found in third party packages.

SourceSafe for Windows 95/NT can be used transparently as part of the SCC tools. It provides a greater degree of user access control, allowing you to create accounts with passwords and to assign access levels on a per file basis. The principal disadvantages are that it is not currently supported by the SCC tools on either Macintosh or UNIX, and there is an additional licensing cost from Microsoft, the developers of SourceSafe.

ClearCase for Solaris 2 has many of the same advantages and disadvantages as SourceSafe when compared to the built-in system. It has a high degree of user access control, and works well when security is a primary issue. As with SourceSafe, it involves an additional licensing cost from Atria, the developers of ClearCase. While a version of ClearCase is available for Windows NT, it is currently not supported by the Pro SCC tools.

### How do I manage multiple hierarchies of VIs?

A project corresponds to a single VI hierarchy. This relationship makes it possible for the Project dialog box to scan your local hierarchy for changes and verify that it is consistent with the project under Source Code Control.

If you are using server functionality to dynamically call subVIs, your application may consist of multiple hierarchies. You can manage systems

of multiple hierarchies by creating individual projects for each set of VIs that your top-level VIs call dynamically. You can then create a *project group* to make it easier to access files from any of the projects used by your application. A project group is a collection of projects. If you select a project group in the Checkout, Checkin, or Retrieve dialog boxes, you can work on any file that is in any of the VI hierarchies used by your application. You can create project groups by selecting **Source Code Control»Project**.

## **Why isn't Windows 3.1 supported?**

### **What if I need Windows 3.1 support?**

Windows 3.1 is not supported because the SCC tools manage VIs as individual files and the Windows 3.1 8+3 character naming limitation makes this impractical. While LLBs make it possible to use longer names, they do not provide the level of transparent access to files that is needed for the Pro SCC tools.

While you cannot use the SCC tools on Windows 3.1, National Instruments understands that customers need Windows 3.1 support. As a developer, you might need to deploy applications to customers that are using systems with Windows 3.1. In that case, we suggest you do your development on Windows 95 or NT. On those platforms you can use individual files for VIs. When you are ready to distribute to Windows 3.1 users, you can save copies of your VIs into LLBs. An easy way for to convert your files to or from LLBs is to use the File Manager dialog box, which has buttons to convert between directories and LLBs.

### **In addition to managing VIs, can I use the SCC tools to manage other project files such as Word documents, text files, CIN source files, and so on?**

Yes. When you edit a project, you will see an **Extra Files** button. Use this button to add non-VI files to your project.

### **How do I access previous versions of files?**

If you are using SourceSafe, you can access previous versions of files from the SourceSafe Explorer.

The **Project»Source Code Control»Advanced** dialog box has a button for accessing a file history. With the built-in system and with ClearCase, you can use this dialog box to retrieve previous versions of files. Note that the administrator can configure the number of previous versions of files to maintain, so you might not have access to all previous versions of a file.

You also can use this dialog box to label a version of a file with a short name such as BETA, making it easy to retrieve it later. Also, labeled versions are not automatically deleted based upon the administration option just described. Instead, you must delete them manually if you no longer want the files.

If you want to label multiple files or retrieve all files with the same label, you can use the System History option from the Advanced dialog box.



321889A-01

Feb98