NATIONAL INSTRUMENTS ®
The Software is the Instrument ®

LabWindows®/CVI
Run-Time Libraries for HP-UX
Version 4.0.1

Welcome to LabWindows/CVI

These release notes will help you begin using the LabWindows/CVI Run-time Libraries for HP-UX, version 4.0.1. They contain a description of system requirements, installation instructions, compiling/linking instructions, and updated information.

Note: *These release notes supplement the documentation for version 4.0.1 of either LabWindows/CVI for Windows or LabWindows/CVI for Solaris.*

Contents

_____

Installing LabWindows/CVI

Before getting to work on your instrument control applications, you must install the LabWindows/CVI Run-time Libraries for HP-UX. The LabWindows/CVI setup program does this for you in a process that lasts approximately five minutes.

Minimum System Requirements for HP-UX

To use the LabWindows/CVI Run-time Libraries for HP-UX, you must have the following:

- HP-UX 700-Series Workstation

- At least 24 MB RAM

- At least 8 MB free hard disk space prior to installation

- HP-UX 9.05 or greater operating system

Installation Instructions for LabWindows/CVI

To install the LabWindows/CVI Run-time Libraries for HP-UX, you must be able to mount a CD-ROM drive on your computer. The drive may be physically attached to your computer, or it may be attached to another computer that permits you to mount the drive remotely. In either case, you need to have root privileges to mount the CD-ROM drive.

Note: *You must purchase a multiple-user license from National Instruments to use the LabWindows/CVI Run-time Libraries for HP-UX over a network.*

1. Log in as root or use the `su` command to get root privileges.

2. Place the LabWindows/CVI CD in your CD-ROM drive. Make sure your CD-ROM is mounted.

3. Run the installation program.

   ```
   myhpux# cd /cdrom/cvi
   myhpux# ./INSTALL
   ```

   Note: *The preceding example assumes that the CD-ROM is mounted in a directory called* `/cdrom`. *Be sure to use the correct path name for your CD-ROM.*

The installation program displays the following installation options.

- LabWindows/CVI Full Install (Run-time Libraries, Help and Samples)

- LabWindows/CVI Run-time Libraries

- LabWindows/CVI Online Help

- LabWindows/CVI Sample Programs

The installation program prompts you to name the directory that will contain the LabWindows/CVI files. This is the directory into which the installation program copies the LabWindows/CVI files from the CD-ROM. All LabWindows/CVI users must have read permissions for this directory. If you plan to run LabWindows/CVI over a network, all machines that run LabWindows/CVI must be able to read the installation directory.

When installing the Run-time Libraries, the installation program places symbolic links to the LabWindows/CVI shared libraries in `/usr/lib`.

Installation Instructions for the VISA Library

LabWindows/CVI requires the VISA library to support the new instrument driver standard accepted by the VXI*plug&play* Systems Alliance. (VISA stands for *Virtual Instrument Software Architecture*.) You should install VISA if you plan to use GPIB or if you plan to use VXI instrument drivers through GPIB.

Installing the VISA Library for HP-UX 9

1. Log in as root or use the `su` command to get root privileges.

2. Place the LabWindows/CVI CD in your CD-ROM drive. Make sure your CD-ROM is mounted.

3. Run the installation program.

   ```
   myhpux# cd /cdrom/visa/hp9
   myhpux# ./INSTALL
   ```

   Note: *The preceding example assumes that the CD-ROM is mounted in a directory called* `/cdrom`. *Be sure to use the correct path name for your CD-ROM.*

Installing the VISA Library for HP-UX 10

Contact National Instruments for more information on getting the NI-VISA library for HP-UX version 10.

What Is Installed by the Setup Programs

The setup program for LabWindows/CVI for HP-UX installs the LabWindows/CVI Run-time Libraries and related files on your hard disk in the subdirectories listed in the following table. The installation includes sample programs that illustrate many of the features in LabWindows/CVI.

Table 1. LabWindows/CVI Subdirectories

Directory Name	Contents
bin	LabWindows/CVI Run-time Library support files
fonts	Font files
include	#include files associated with the run-time libraries
instr	Sample instrument modules
hyperhelp	HyperHelp Viewer files
samples	Source code for sample programs
toolslib	ECHO controls and utility instrument drivers

The message file (msgrt*n*.txt where *n* is the version number of the Run-time Engine) is a text file containing the error messages displayed by the Run-time Engine. The message file is in the LabWindows/CVI /bin/ subdirectory. You can translate the message file into other languages. Refer to the *Configuring the Run-Time Engine* section in chapter 7, *Creating and Distributing Standalone Executables and DLLs*, of the *LabWindows/CVI Programmer Reference Manual* for information on translating the message file.

The LabWindows/CVI CD contains instrument drivers for over 400 instruments. Each instrument driver includes C source code, a LabWindows/CVI function panel file, and documentation. The installation program does not install the complete set of instrument drivers; you can manually install the ones you need. The file README.inst gives more information about installing and using the drivers.

Installing on a Network

Contact National Instruments for licensing information if you plan to use the LabWindows/CVI Run-time Libraries on a network.

Using LabWindows/CVI on HP-UX

LabWindows/CVI Run-Time Libraries

The LabWindows/CVI Run-time Libraries for HP-UX are a set of libraries that enable LabWindows/CVI users to port existing applications from the Windows and UNIX platforms to the HP-UX platform. Refer to the following table for a list of LabWindows/CVI libraries supported under HP-UX.

Table 2. LabWindows/CVI Libraries Supported under HP-UX

Library name	Name and location of Library file
Advanced Analysis	`cvidir`/bin/libanalysis.a (symbolic link created in `/usr/lib`)
GPIB	`cvidir`/bin/gpib.o
RS-232	`cvidir`/libcvi.sl (symbolic link created in `/usr/lib`)
TCP	`cvidir`/libcvi.sl (symbolic link created in `/usr/lib`)
User Interface	`cvidir`/libcvi.sl (symbolic link created in `/usr/lib`)
Utility	`cvidir`/libcvi.sl (symbolic link created in `/usr/lib`)
VISA	`/usr/vxipnp/hpux/bin/libvisa.sl` (default) (NI-VISA is installed separately.)

Note: *When you create an executable with a compiler under HP-UX, you must use the ANSI C library of the operating system.*

The LabWindows/CVI Run-time Libraries for HP-UX are not multithread safe.

The VISA library is installed separately and only supports GPIB, serial, and GPIB-VXI/c.

Compiling and Linking under HP-UX

Compatible External Compilers

Under HP-UX, the LabWindows/CVI Run-time libraries are compatible with the following ANSI C compilers.

- GNU C Compiler (`gcc`)

- HP-UX C Compiler (`cc`)

Include Files

The header files for the LabWindows/CVI Run-time Libraries are located in the LabWindows/CVI `/include/` subdirectory.

The LabWindows/CVI Run-time Libraries predefine the `_NI_hpux_` macro to help you write platform-dependent code. The `_NI_hpux_` macro is defined when you use the LabWindows/CVI Run-time Libraries header files and when a compiler defines either `__hp9000s700` or `__hppa__`.

Exporting Symbols

The LabWindows/CVI Libraries require all callback functions to be exported from an executable so that functions, such as `LoadPanel`, can resolve symbol names at run time. The suggested way to export functions is to add the `-Wl,-E` option (no spaces) to your make file so that all symbols are exported when an executable is created.

Example Make Files for an Executable

An example `Makefile` is shown below. The example application uses the CVI, GPIB, VISA, and Analysis libraries, and the toolslib `toolbox` instrument driver.

```
# Use these for cc
CC=cc
ANSIFLAG=-Ae
# Use these for gcc
#CC=gcc
#ANSIFLAG=-ansi
CVIDIR=/applications/cvihpuxdir
VISADIR=/usr/vxipnp/hpux
CFLAGS=$(ANSIFLAG) -Wl,-E -D_NI_hpux_ -I$(CVIDIR)/include \
                                    -I$(CVIDIR)/toolslib/toolbox \
                                    -I$(VISADIR)/include
LDFLAGS=-L$(VISADIR)/bin -lcvi -lanalysis -lvisa -lm
all: graph
graph: graph.c graph.h
  $(CC) $(CFLAGS) -o $@ graph.c $(CVIDIR)/bin/gpib.o \
                                $(CVIDIR)/toolslib/toolbox/toolbox.o \
                                $(LDFLAGS)
```

Note: *In the preceding example, symbolic links for the LabWindows/CVI and the Analysis libraries have been created in the* /usr/lib *directory. If these symbolic links do not exist, please refer to the* man *page for the* ld *(link editor) for information on additional required link options.*

Creating Shared Libraries under HP-UX

Follow these guidelines when you create a shared library under HP-UX.

- Create "position independent code" when compiling your source code into object modules as follows: either use the +z or +Z options when compiling with cc, or use the -fpic or -fPIC options when compiling with gcc. Refer to the man page of the compiler for more details on which option to use.

- Use the -b option to create a shared library when using ld (link editor).

Consider the following example make file for a shared library.

```
# Use these for cc
CC=cc
ANSIFLAG=-Ae
PIC=+z
# Use these for gcc
#CC=gcc
#ANSIFLAG=-ansi
#PIC=-fpic
CVIDIR=/applications/cvihpuxdir
CFLAGS=$(ANSIFLAG) $(PIC) -c -I$(CVIDIR)/include
LDFLAGS=-lcvi -lanalysis -lm -lc
all: libmodule.sl
libmodule.sl: mod1.o mod2.o
  ld -b -o libmodule.sl mod1.o mod2.c $(LDFLAGS)
mod1.o: source1.c source1.h
  $(CC) $(CFLAGS) -o $@ source1.c
mod2.o: source2.c source2.h
  $(CC) $(CFLAGS) -o $@ source2.c
```

Note: *In the preceding example, symbolic links for the LabWindows/CVI and the Analysis libraries have been created in the* /usr/lib *directory. If these symbolic links do not exist, please refer to the* man *page for the* ld *(link editor) for information on additional required link options.*

Main Function Must Call InitCVIRTE

If your program calls any functions from the LabWindows/CVI Run-time Libraries, you must call InitCVIRTE to initialize the libraries from the executable. Failure to call InitCVIRTE may cause unpredictable application failures.

The InitCVIRTE function takes three parameters. The first and third parameters to this function should always be 0 for HP-UX applications. The second should have the same value as the second parameter passed to your main function. InitCVIRTE returns 0 upon failure.

If your program does not call `InitCVIRTE`, your executable will not work. For this reason, it is recommended that you always include source code similar to the following example in your program.

```
int main(int argc, char *argv[])
{
    if (InitCVIRTE(0, argv, 0) == 0) {
        return -1;      /* Failed to initialize */
    }
    /* your program code here */
}
```

If you pass NULL for the second argument to `InitCVIRTE` then your program may still work, but has the following limitations.

- Your executable cannot accept the `-display` command line argument. As a result, you cannot specify an X display on the command line for your program to use. Your executable can accept the `DISPLAY` environment variable to specify a different X display.

- `LoadPanel`, `LoadExternalModule`, `DisplayImageFile`, `SavePanelState`, `RecallPanelState`, and other functions that normally use the directory of the executable to search for files use the current working directory instead. If you run the executable from a directory other than the one that contains your executable, some of these functions may fail to find files.

Using the Stub VXI or GPIB Libraries

Some VXI instrument drivers use both GPIB and VXI functions. Because HP-UX does not support the VXI Library, you need to include a stub file for the VXI library to compile these instrument drivers. The following stub library files have been installed into the LabWindows/CVI `/bin/` subdirectory.

- `vxistub.o`

- `gpibstub.o`

Sample Programs for HP-UX

The `/samples/` subdirectory contains sample programs for the supported LabWindows/CVI Run-time Libraries for HP-UX. Each sample subdirectory contains a predefined make file, `Makefile`.

Viewing .uir Files on HP-UX

The UIRVIEW sample program located in the `/samples/apps/uirview` subdirectory allows you to display `.uir` file panels. You can use this sample application to find out the `.uir` constants and callback function names in a `.uir` file that was created under Windows or UNIX.

Porting LabWindows/CVI Programs to HP-UX

Adhere to the following guidelines when you port a LabWindows/CVI project from Windows or Solaris to HP-UX.

- Add all source and header files used in your project to your make file.

- Recompile all instrument drivers used in your project. Add the instrument driver's source and header files to your make file, or recompile the instrument driver into an object module that is compatible with HP-UX.

- Recompile all object modules loaded using `LoadExternalModule` into shared libraries (`.sl` files) for HP-UX. For more information on creating shared libraries, see the section *Creating Shared Libraries under HP-UX* in these release notes.

- User Interface Resource (`.uir`) files do not require any changes.

  Note: *You may need to adjust some controls or fonts sizes when you port a* `.uir` *file from the Windows platform.*

For more information on building multiplatform applications, see Chapter 6, *Building Multiplatform Applications*, in the *LabWindows/CVI Programmer Reference Manual*.

Using LoadExternalModule under HP-UX

Under HP-UX the `LoadExternalModule` and `LoadExternalModuleEx` functions can load only shared libraries (`.sl` files). LabWindows/CVI does not implement the loading of static objects and libraries, `.o` and `.a`. For more information on creating shared libraries, see the section *Creating Shared Libraries under HP-UX* in these release notes.

Setting Configuration Options for Your Application

You set application configuration options under UNIX in the `.Xdefaults` file. The options do not take effect until you restart your X server, or until you load your `.Xdefaults` file using the `xrdb` command.

A configuration string is associated with each option, as in the following example.

```
myapp.editorFont:
myapp.tmpdir: /home/myhomedir/mytmpdir
```

An unused configuration string is not required in the `.Xdefaults` file. Leaving the space to the right of the colon (:) empty is equivalent to not including the configuration string.

You must specify an absolute path name for configuration strings that take a directory name.

Option Descriptions

You have the following options when configuring the LabWindows/CVI Run-time Libraries for your application.

activate

When this option is set to `True` (`myapp.activate: True`) the input focus is set to a window when the user brings that window to the front.

The default value is `False`. Use this option only when `ClickToType` is active.

appFont

This option specifies the font NIApp. The default setting is `myapp.appFont: adobe-helvetica`.

cvidir (now obsolete)

The `cvidir` option used by LabWindows/CVI version 3.1 is obsolete. You must install the support files for the LabWindows/CVI Run-time Engine into the same directory tree as the LabWindows/CVI Run-time Engine.

dialogfont

This option specifies the font NIDialog, which LabWindows/CVI uses in dialog boxes and built-in pop-up panels. The default setting is `myapp.dialogFont: adobe-helvetica`.

editorFont

This option specifies the font NIEditor, which LabWindows/CVI uses in the Standard I/O window. The default setting is `myapp.editorFont: adobe-courier`.

menuFont

This option specifies the font NIMenu, which LabWindows/CVI uses in menus. The default setting is `myapp.menuFont: adobe-helvetica`.

tmpdir

`tmpdir` sets the location for temporary files.

Distributing Standalone Executables under HP-UX

To package your standalone programs for distribution, you can use the UNIX scripts in the `misc/bin` directory of the LabWindows/CVI installation directory.

To distribute executable programs under HP-UX, complete the following steps.

1. If your program loads `.uir` files with `LoadPanel` or loads external modules with `LoadExternalModule`, use caution when you specify the file names in calls to these functions. If you use a relative path, the path is relative to the directory containing the executable.

2. Create a directory containing your executable program and associated files. Structure the directory exactly as you want it to appear after installation. Test your program by running it from that directory.

3. Use the shell script `makedist` in the `misc/bin` directory to create a distribution package. This script creates a compressed `tar` file that contains the directory you created in step 2

and a copy of the LabWindows/CVI Run-time Engine, which is required to run your executable.

4. Make a copy of the installation script `INSTALL.sample` in the `misc/bin` directory and customize it using the information provided by `makedist`. This installation script unpacks a distribution package, creating a directory like the one you created in step 2, and then installs the LabWindows/CVI Run-time Engine. The installation script can install from floppy disks or from the current directory.

General Information

HyperHelp Notes

LabWindows/CVI uses the HyperHelp viewer from Bristol Technology, Inc. to provide online help. The HyperHelp viewer is distributed with LabWindows/CVI and is installed in the `hyperhelp` subdirectory of the installation directory.

If you already have HyperHelp on your system, then you may not need the contents of the `hyperhelp` directory. However, you must set certain environment variables so that LabWindows/CVI works properly with your copy of HyperHelp. In particular, you must set the variables listed in the following table before running LabWindows/CVI.

Table 3. Setting Variables for Viewing LabWindows/CVI Online Help

Variable	Description	Setting
HHPATH	Location of HyperHelp executable.	hyperhelp
HOHPATH	Location of the HelpOnHelp file.	hoh.hlp
XPPATH	Print files directory path.	Xp

HyperHelp might fail to start if your environment is not configured to find certain system shared libraries (such as `libXmu.so`). You can avoid this problem by setting the environment variable `LD_LIBRARY_PATH` to `$OPENWINHOME/lib` or to another directory that contains the system-shared libraries.

Using GPIB and LabWindows/CVI on Your HP-UX Workstation

The following section specifies the required or suggested NI-488.2M driver software to use with LabWindows/CVI for HP-UX. The NI-488.2M driver software is included with your GPIB hardware. Contact National Instruments to obtain the latest version.

To use the following GPIB hardware with LabWindows/CVI, you must install NI-488.2M for HP-UX. The NI-488.2M software works with the following hardware.

- GPIB-HP700-EISA

- GPIB-ENET

- AT-GPIB/TNT

- EISA-GPIB

GPIB Portability

If you want your applications to be portable, do not monitor GPIB errors by checking for negative return values. Instead, check for the ERR bit in the return value. For example, the following statement is not portable because it checks for a negative return value:

```
if (ibwrt(bd[instrID], buf, cnt) < 0)
     PREFIX_err = 230;
```

The following revised statement is portable because it checks for the ERR bit in the return value:

```
if (ibwrt(bd[instrID], buf, cnt) & ERR)
     PREFIX_err = 230;
```

Under Windows 3.*x*, the return value is a 16-bit value and is negative when the ERR bit is set. On other systems, such as UNIX, the return value is a 32-bit value and is never negative.

Asynchronous Signal Handling

The SIGPOLL (SIGIO) and SIGPIPE signals have special meaning in the LabWindows/CVI Run-time Libraries. The LabWindows/CVI Libraries catch these two signals as part of the TCP library. If your programs catch either of these signals, you should call the previously installed handler, if any.

Limitations for Clipboard Function under HP-UX

The following new clipboard functions only use the LabWindows/CVI internal clipboard. These functions do not affect the UNIX system clipboard.

```
ClipboardGetText
ClipboardSetText
ClipboardGetBitmap
ClipboardPutBitmap
```

ComSetEscape Function Implemented under HP-UX

The RS-232 Library function, ComSetEscape, was implemented in HP-UX to direct the specified com port to carry out an extended function such as clearing or setting the RTS signal line. This function was previously supported only under Windows.

Setting LabWindows/CVI to Use Fewer CPU Resources

If you want LabWindows/CVI to use fewer CPU resources so that other applications receive more processor time, you should use the function `SetSleepPolicy`. This function sets the degree your program "sleeps" when checking for events. The setting that is optimal for your program depends on the operating system you are using and the other applications you are running. If you think you may need to make an adjustment, try the different settings listed in the following table and choose the setting that best matches your needs.

Table 4. Sleep Settings for LabWindows/CVI

Setting	Description
`VAL_SLEEP_NONE`	Never sleep (default).
`VAL_SLEEP_SOME`	Sleep for a small period.
`VAL_SLEEP_MORE`	Sleep for a longer period.