NI-VXI™ Software Manual for the VXIpc™-850 Series

Internet Support

GPIB: gpib.support@natinst.com
DAQ: daq.support@natinst.com
VXI: vxi.support@natinst.com
LabVIEW: lv.support@natinst.com
LabWindows: lw.support@natinst.com
HiQ: hiq.support@natinst.com
VISA: visa.support@natinst.com

E-mail: info@natinst.com
FTP Site: ftp.natinst.com
Web Address: http://www.natinst.com

Bulletin Board Support

BBS United States: (512) 794-5422 or (800) 327-3077
BBS United Kingdom: 01635 551422
BBS France: 1 48 65 15 59

FaxBack Support

(512) 418-1111

Telephone Support (U.S.)

Tel: (512) 795-8248
Fax: (512) 794-5678

International Offices

Australia 03 9 879 9422, Austria 0662 45 79 90 0, Belgium 02 757 00 20,
Canada (Ontario) 519 622 9310, Canada (Québec) 514 694 8521, Denmark 45 76 26 00,
Finland 90 527 2321, France 1 48 14 24 24, Germany 089 741 31 30, Hong Kong 2645 3186,
Italy 02 413091, Japan 03 5472 2970, Korea 02 596 7456, Mexico 95 800 010 0793,
Netherlands 0348 433466, Norway 32 84 84 00, Singapore 2265886, Spain 91 640 0085,
Sweden 08 730 49 70, Switzerland 056 20 51 51, Taiwan 02 377 1200, U.K. 01635 523545

National Instruments Corporate Headquarters

6504 Bridge Point Parkway   Austin, TX 78730-5039   Tel: (512) 794-0100

Important Information

Warranty

Copyright

Trademarks

WARNING REGARDING MEDICAL AND CLINICAL USE OF NATIONAL INSTRUMENTS PRODUCTS

About This Manual

Chapter 1
Introduction

Chapter 2
NI-VXI Software Installation

Chapter 3
NI-VXI Configuration Utility

Chapter 4
Using the NI-VXI Software

Appendix A
NI-VXI Software Overview

Appendix B
Common Questions

Appendix C
Customer Communication

Glossary

Index

Figures

This manual contains instructions for installing and configuring the NI-VXI bus interface software for Microsoft operating systems and the VXIpc-850 Series. The NI-VXI software works with all models of the Model 850 Series of the National Instruments VXIpc-850 embedded computers. The NI-VXI software is fully VXI*plug&play* compliant.

Microsoft operating systems include DOS, Windows 3.1, Windows NT, and Windows 95. This manual uses the term *Window 95/NT/3.1* when information applies to all three Windows operating systems.

Organization of This Manual

This manual is organized as follows:

- Chapter 1, *Introduction*, describes the NI-VXI software for the VXIpc-850 Series, lists what you need to get started, includes a brief description of the NI-VXI software, and lists optional software.

- Chapter 2, *NI-VXI Software Installation*, contains the instructions to install the NI-VXI software. Because the software is already installed on your VXIpc-850, the information in this chapter is useful only if you are reinstalling the software.

- Chapter 3, *NI-VXI Configuration Utility*, contains instructions for using the VXI Resource Editor utility of the NI-VXI software to configure the VXIpc-850 Series embedded computer.

- Chapter 4, *Using the NI-VXI Software*, discusses programming information for you to consider when developing applications that use the NI-VXI driver.

- Appendix A, *NI-VXI Software Overview*, lists and describes the main programs and files that make up the NI-VXI software.

- Appendix B, *Common Questions*, addresses common questions you may have about using the NI-VXI bus interface software on the VXIpc-850 platform.

- Appendix C, *Customer Communication*, contains forms you can use to request help from National Instruments or to comment on our products and manuals.

- The *Glossary* contains an alphabetical list and description of terms used in this manual, including abbreviations, acronyms, metric prefixes, mnemonics, and symbols.

- The *Index* contains an alphabetical list of key terms and topics used in this manual, including the page where you can find each one.

Conventions Used in This Manual

The following conventions are used in this manual:

bold Bold text denotes menus, menu items, dialog box buttons or options, or LEDs.

bold italic Bold italic text denotes a note, caution, or warning.

italic Italic text denotes emphasis, a cross reference, or an introduction to a key concept.

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

monospace Text in this font denotes text or characters that are to be literally input from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of paths, directories, programs, subprograms, subroutines, functions, and extensions, and for statements and comments taken from program code.

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

♦ The ♦ symbol indicates that the text following it applies only to a specific operating system.

<>   Angle brackets enclose the name of a key on the keyboard–for example, <PageDown>.

Abbreviations, acronyms, metric prefixes, mnemonics, symbols, and terms are listed in the *Glossary*.

How to Use This Documentation Set

Begin by reading the *Getting Started with Your VXIpc-850* for basic instructions on installing and configuring the hardware and software. This is a brief quick start manual that describes how to get started with your kit using the default hardware and software settings.

The *VXIpc-850 Series User Manual* contains more details about changing the installation or configuration from the defaults, and using the hardware.

The *VXIpc-850 Peripherals User Manual* contains in-depth information about various peripherals on the VXIpc-850.

This manual, the *NI-VXI Software Manual for the VXIpc-850 Series*, contains more details about changing the NI-VXI software installation or configuration from the defaults, and using the NI-VXI software on the VXIpc-850.

When you are familiar with the material in this manual, you can begin to use the *NI-VXI Software Reference Manual for C*. Chapter 1, *Introduction to VXI*, and Chapter 2, *Introduction to the NI-VXI Functions,* present the concepts of VXI and prepare you for detailed explanations of the NI-VXI functions. Study the descriptions of each function given in Chapters 3 through 13 to fully understand the purpose and syntax of each function.

Refer to the *NI-VXI Graphical Utilities Reference Manual* and the *NI-VXI Text Utilities Reference Manual* to learn more about the NI-VXI utilities.

Related Documentation

The following documents contain information that you may find helpful as you read this manual:

- ANSI/IEEE Standard 1014-1987, *IEEE Standard for a Versatile Backplane Bus: VMEbus*

- ANSI/IEEE Standard 1155-1993, *IEEE VMEbus Extensions for Instrumentation: VXIbus*

- ANSI/VITA 1-1994, *VME64*

- VXI-6, *VXIbus Mainframe Extender Specification*, Rev. 1.0, VXIbus Consortium

Customer Communication

National Instruments wants to receive your comments on our products and manuals. We are interested in the applications you develop with our products, and we want to help if you have problems with them. To make it easy for you to contact us, this manual contains comment and configuration forms for you to complete. These forms are in Appendix C, *Customer Communication*, at the end of this manual.

Introduction
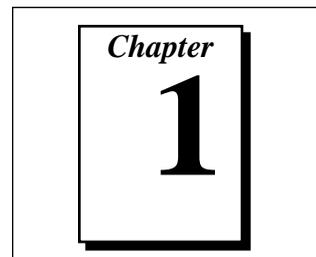
This chapter describes the NI-VXI software for the VXIpc-850 Series, lists what you need to get started, includes a brief description of the NI-VXI software, and lists optional software. You should already have installed and configured the VXIpc-850 embedded computer according to the instructions in the *VXIpc-850 Series User Manual*.

The VXIpc-850, a Pentium-based, C-size, embedded computer based on the Peripheral Component Interface (PCI) bus and Industry Standard Architecture (ISA), is a high-performance, easy-to-use platform for controlling VXIbus systems, featuring complete VXI functionality through interactive utilities and function calls for C and BASIC. In addition, the VXIpc-850 has an IEEE 488 interface that is compatible with the NI-488.2 architecture.

What You Need to Get Started

❑ VXIpc-850 Series embedded computer and accessories

❑ VXIbus mainframe

The following items are already installed on your VXIpc-850 computer. They are also included on disk in case you need to reinstall your software.

❑ LabWindows®/CVI Run-Time Engine
   (for Windows 95/NT/3.1 users)

❑ NI-VXI software media for the VXIpc-850 Series

Software Description

The NI-VXI bus interface software for the VXIpc-850 Series includes a VXI Resource Manager, graphical and text-based versions of an interactive VXI resource editor program, a comprehensive library of software routines for VXI programming, and an interactive control program. You can use this software to seamlessly control multiple-mainframe configurations and have software compatibility across a variety of VXI controller platforms.

Software Configurations

There are four software configurations described in this manual:

- NI-VXI for DOS/Windows 3.1—you can use this version of the software to develop and run 16-bit DOS/Windows 3.1 applications. You can also use this software under Windows 95 if you intend to use 16-bit applications only.

- NI-VXI Upgrade for Windows 95—this is a compatibility release that extends your NI-VXI for DOS/Windows 3.1 to allow 32-bit applications running in Windows 95 to use the 16-bit driver. In this configuration you can run both 16-bit and 32-bit applications; however, the core of the driver is 16-bit.

- NI-VXI for Windows 95—this is a fully 32-bit native Plug and Play driver for Windows 95. You can run *only* 32-bit applications with this driver. You cannot use this driver in conjunction with either NI-VXI for DOS/Windows 3.1 or the NI-VXI Upgrade for Windows 95 to run 16-bit applications. Applications developed using this driver will run with NI-VXI for Windows NT without the need to recompile.

- NI-VXI for Windows NT—this is a 32-bit driver designed for Windows NT. You can use this version to develop and run 32-bit applications for Windows 95/NT.

Optional Software

Your VXIpc-850 kit includes the NI-VXI bus interface software. In addition, you can use the National Instruments LabVIEW and LabWindows/CVI application programs and instrument drivers to ease your programming task. These standardized programs match the modular virtual instrument capability of VXI and can reduce your VXI/VMEbus software development time. These programs are fully VXI*plug&play* compliant and feature extensive libraries of VXI instrument drivers written to take full advantage of direct VXI control.

LabVIEW is a complete programming environment that departs from the sequential nature of traditional programming languages and features a graphical programming environment.

LabWindows/CVI is an interactive C development environment for building test and measurement and instrument control systems. It includes interactive code-generation tools and a graphical editor for building custom user interfaces.

LabVIEW and LabWindows/CVI include all the tools needed for instrument control, data acquisition, analysis, and presentation. When you order the LabVIEW VXI Development System for Windows 95/NT/3.1 or the LabWindows/CVI VXI Development System for Windows 95/NT/3.1, you also get more than 500 complete instrument drivers, which are modular, source-code programs that handle the communication with your instrument to speed your application development.

NI-VXI Software Installation
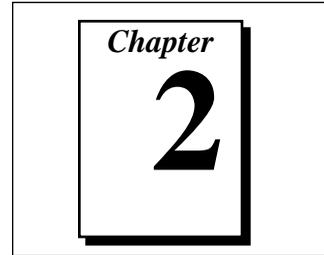
This chapter contains the instructions to install the NI-VXI software.

☞ **Note:** *Because the software is already installed on your VXIpc-850, the information in this chapter is useful only if you are reinstalling the software.*

There are two methods for installing your NI-VXI software: the Windows Setup program, and a DOS INSTALL program. You can use the Windows Setup program to install NI-VXI for Windows 95/NT/3.1 and DOS. The DOS INSTALL program can install only the NI-VXI software for DOS. Refer to the section appropriate for the Microsoft operating system you are using.

Be sure you have up to 5 MB of free space available to accommodate the LabWindows/CVI Run-Time Engine and the NI-VXI software.

☞ **Note:** *If you are using NI-VXI for Windows 95/NT/3.1, you must install the LabWindows/CVI Run-Time Engine first. You do not need the LabWindows/CVI Run-Time Engine if you are using the DOS version of the software.*

Using the Windows Setup Program (Windows 3.1)

You can use the Setup program that came with your NI-VXI software to install the entire NI-VXI software package, a software update, or to reinstall software in the event that your files were accidentally erased. Follow these steps to install all or part of the NI-VXI software.

1. Installing the LabWindows/CVI Run-Time Engine

Some of the NI-VXI utilities use the LabWindows/CVI Run-Time Engine to present a graphical user interface. Even if you already have LabWindows/CVI on your computer, you should install the LabWindows/CVI Run-Time Engine for NI-VXI before installing the NI-VXI software.

1. Insert the disk labeled *LabWindows/CVI Run-Time Engine for Windows 3.x.*

2. Run SETUP.EXE from the Windows Program Manager's **File** menu.

3. Accept the default destination directory as prompted, or choose another directory in which to install the LabWindows/CVI Run-Time Engine.

2. Installing the NI-VXI Software

This section describes how to install the NI-VXI software for the VXIpc-850 and Windows 3.1. Please carefully read these directions along with any messages on the screen before making your selections.

You can quit the Setup program at any time by pressing the **Cancel** button. If you do not have a mouse, pressing the <Esc> key performs the same action.

Setup is an interactive, self-guiding program that installs the NI-VXI software and configures your system to use the NI-VXI software with the VXIpc-850. Follow these steps to perform the installation.

1. Insert the disk labeled *NI-VXI for VXIpc-850 for DOS/Windows.*

2. Select **Run...** from the Windows Program Manager's **File** menu and enter the following code, where *X* is your floppy drive (usually A or B).

   X:\setup.exe

   and press <Enter>.

3. Click on the **Next** option at the **Welcome** screen to start the installation.

4. Select the target directory for your installation. Although Setup prompts you to accept a default directory, you can select a different location by clicking on **Browse...**.

   • If you do not already have NI-VXI on your computer, Setup prompts you to select the default directory of C:\NIVXI.

- Setup detects if you already have a previous installation of
  NI-VXI on your computer, and prompts you to either
  overwrite the previous version or install the new version in a
  different directory. Only one copy of NI-VXI can be active on
  your computer at any time. If you install the new version of
  NI-VXI into a different directory, the old installation is
  disabled.

Caution: *If you have a previous version of the NI-VXI software installed, Setup does not automatically create a backup of the software files. If there are files you want to preserve, you should exit the installation program now and make a backup before continuing.*

5. Select the type of installation you want.

   - **Typical** installation includes Windows and DOS drivers,
     utilities, and development files for all supported compilers.

   - **Compact** installation includes only the driver and utilities
     necessary to run applications written with NI-VXI.

   - **Custom** installation lets you select the target operating system
     and development environments according to your needs.

Note: *If you install the driver files, you must include the* **Common Windows/DOS Driver Files.** *Similarly, when installing development files, include* **Common Development Files** *in your custom installation.*

6. Select the name of the Program Manager folder that will contain
   icons for the NI-VXI software for VXIpc-850.

   At this point Setup copies the necessary files to your hard drive
   and creates Program Manager icons.

7. Your AUTOEXEC.BAT file needs to be modified to use NI-VXI.
   You can either let Setup modify the file, create a different file, or
   do nothing. If you decide to not let Setup modify your
   AUTOEXEC.BAT file, you should make the necessary changes
   manually. Refer to the following section, *Modifying the
   AUTOEXEC.BAT File*, for complete details.

8. When the installation process completes, you must exit Windows
   and reboot your computer for the changes to take effect. You can
   let the Setup program reboot your computer by selecting the
   Restart computer option on the last screen. Click on **Finish** to
   end the installation.

9. You can now use VXIEDIT to configure the hardware in your VXI
   system. Continue with Chapter 3, *NI-VXI Configuration Utility*, for
   instructions on using the configuration editors in VXIEDIT.

Modifying the AUTOEXEC.BAT File

If you choose to let Setup modify your `AUTOEXEC.BAT` file, it updates the setting of environment variables `PATH`, `LIB`, and `INCLUDE` to include the relevant subdirectories of the NI-VXI directory. When Setup modifies the file, it saves the old file as `AUTOEXEC.OLD` in the NI-VXI directory. The previously specified directories in `PATH`, `LIB`, and `INCLUDE` remain unchanged. Setup also adds a new environment variable `NIVXIPATH`, and appends a command line to execute `VXIINIT.EXE` automatically.

If you choose not to let Setup modify your `AUTOEXEC.BAT` file, it creates a file called `AUTOEXEC.VXI` in the NI-VXI directory. Refer to the `AUTOEXEC.VXI` file for suggestions on how to change the following lines manually.

*   The `PATH` variable should include the full path to the subdirectory where the NI-VXI utilities and `NIVXI.DLL` are located, in addition to whatever other directories you have already specified in `PATH`. The path must be specified so that Windows can locate the executable code when the library needs to be loaded. Normally, these files reside in the root of the NI-VXI directory, and also the `WIN` subdirectory.

*   The `LIB` variable should include the full path to the subdirectories that contain the C libraries for the compiler you choose to install.

*   The `INCLUDE` variable should include the full path to the subdirectory that contains the NI-VXI include files. By default, the include files reside in the `INCLUDE` subdirectory of the NI-VXI directory.

*   The `NIVXIPATH` variable should contain the full path to the NI-VXI directory.

Modifications to the SYSTEM.INI File

Setup adds a line to the `[386Enh]` section to load two device drivers that NI-VXI needs for memory management. The files, which must be loaded at Windows startup, are called `NIVXIPHM.386` and `DAQMEM32.386`. These two files are normally located in the `WIN` subdirectory of the NI-VXI directory as shown below. Setup also adds a line that keeps Windows from using the VXI space for its own purposes.

```
[386Enh]
DEVICE= <NI-VXI directory>\WIN\NIVXIPHM.386
DEVICE= <NI-VXI directory>\WIN\DAQMEM32.386
EMMexclude= A000-EFFF

[NIVXI]
NIVXIPATH= <NI-VXI directory>
Load Data= 6bytes
```

When Setup modifies the file, it saves the old file as SYSTEM.OLD
in the NI-VXI directory.

Modifications to the WIN.INI File

Setup adds the following lines to the WIN.INI file.

```
[NIVXI]
NIVXIPATH= <NI-VXI directory>
```

In this situation, the NIVXIPATH variable is used by the NIVXI.DLL
dynamic link library in addition to the application programs that come
with NI-VXI (for example, RESMAN.EXE) to locate the NI-VXI
configuration and help files.

When Setup modifies the file, it saves the old file as WIN.OLD in the
NI-VXI directory.

Completing the Software Installation

After you execute Setup, you should exit Windows and reboot your
machine to make your system aware of the NI-VXI directory.

After the NI-VXI software is installed, run VXIINIT.EXE and
then RESMAN.EXE. You need to run VXIINIT to initialize the
VXIpc-850 before you perform any VXI operations and after each
computer reset. RESMAN is the National Instruments Resource Manager,
which you must run every time the chassis power is cycled so that your
application can access devices in the VXI chassis.

After you run VXIINIT and RESMAN, you are ready to use the
NI-VXI Resource Editor program VXIEDIT to interactively configure
the hardware in your system. Continue with Chapter 3, *NI-VXI
Configuration Utility*, for instructions on using the configuration editors
in VXIEDIT.

Using the Windows Setup Program (Windows 95)

You can use the Setup program that came with your NI-VXI software to install the entire NI-VXI software package, a software update, or to reinstall software in the event that your files were accidentally erased. Follow these steps to install all or part of the NI-VXI software.

1. System Preparation

If you are currently using either the NI-VXI software for DOS/Windows 3.1 or the NI-VXI Windows 95 upgrade, Setup will remove it before installing the new software. You cannot have both the 16-bit and the 32-bit versions of NI-VXI installed at the same time.

☞ **Note:** *If you plan to run both 16-bit and 32-bit applications, you should use the NI-VXI Windows 95 upgrade version instead.*

If you have been using your VXIpc-850 under Windows 95 with either the NI-VXI software for DOS/Windows 3.1 or the NI-VXI Windows 95 upgrade, you need to remove the Plug and Play information from the Windows 95 Device Manager before installing the new NI-VXI software.

Follow these steps to remove the VXIpc-850 information.

1. Double-click on the **System** icon under **Start**»**Settings**»**Control Panel**.

2. Select the **Device Manager** tab from the **System Properties** dialog that appears.

3. Click on the **View devices by type** button and double-click on the **Other Devices** icon.

4. Select the VXIpc-850 from the list of devices under **Other Devices**. It will appear under the name **PCI Card** and will have a circled exclamation point through the **?** (question mark) icon.

5. Click on the **Remove** button.

6. Click **OK** to exit the Device Manager after removing the device information.

2. Installing the LabWindows/CVI Run-Time System

Some of the NI-VXI utilities use the LabWindows/CVI Run-Time Engine to present a graphical user interface. Even if you already have LabWindows/CVI on your computer, you should install the LabWindows/CVI Run-Time Engine before installing the NI-VXI software.

1. Insert the disk labeled *LabWindows/CVI Run-Time Engine for Windows 95/NT.*

2. Select **Run...** from the **Start** menu and run SETUP.EXE.

3. Accept the default destination directory as prompted, or choose another directory in which to install the LabWindows/CVI Run-Time Engine.

3. Installing the NI-VXI Software

This section describes how to install the NI-VXI software for the VXIpc-850 and Windows 95. Please carefully read these directions along with any messages on the screen before making your selections.

You can quit the Setup program at any time by pressing the **Cancel** button.

Setup is an interactive, self-guiding program that installs the NI-VXI software and configures your system to use the NI-VXI software with the VXIpc-850.

Follow these steps to perform the installation.

1. Insert the disk labeled *NI-VXI for VXIpc-850 for Windows 95*.

2. Select **Run...** from the **Start** menu and enter the following text, where *X* is your floppy drive (usually A or B).

    X:\setup.exe

    and press <Enter>.

3. Click on the **Next** button at the **Welcome** screen to start the installation.

☞ **Note:**   *If Setup detects a DOS/Windows 3.1 or Windows 95 upgrade version, it will warn you that this version will be deleted.*

4. Select the target directory for your installation. Although Setup prompts you to accept the C:\NIVXI directory by default, you can select a different location by clicking on **Browse...**.

⬖　**Caution:**　　*If you have a previous version of the NI-VXI software installed, Setup can convert the configuration settings to the new format. However, manufacturer and model name files will not be preserved. If you want to preserve these files, you should exit the installation program now and make a backup before continuing.*

5. Select the components you want to install.

   • **NI-VXI for Windows 95** includes only the driver and utilities needed to configure and use your VXI or VME system.

   • **Microsoft C/C++ Development Files** includes the components necessary to develop applications using the Microsoft Visual C++ compiler. Please read the *Modifying the Environment* section if you choose to install these files.

   • **Borland C/C++ Development Files** includes the components necessary to develop applications using the Borland C/C++ compiler. Please read the *Modifying the Environment* section if you choose to install these files.

6. Select the name of the program folder that will contain icons for the NI-VXI software for VXIpc-850.

   At this point Setup copies the necessary files to your hard drive and creates program icons.

7. When the installation process completes, you must exit Windows and reboot your computer for the changes to take effect. You can let the Setup program reboot your computer by selecting the **Restart computer** option on the last screen. Click on **Finish** to end the installation.

8. If you backed up the manufacturer and model name files, you should restore them to the TBL subdirectory of your NI-VXI directory before running VXIEDIT.

9. You can now use VXIEDIT to configure the hardware in your VXI system. Continue with Chapter 3, *NI-VXI Configuration Utility*, for instructions on using the configuration editors in VXIEDIT.

Modifying the Environment

The installer does not modify the environment variables under Windows 95. No changes are necessary. However, you may want to add the NI-VXI directory to your LIB and INCLUDE paths if you use makefiles to build your projects, and to the PATH variable if you plan to run the VXI utilities from the command line.

Completing the Software Installation

After you execute Setup, you should restart Windows 95 to make your system load the NI-VXI driver.

After the NI-VXI software is installed, run RESMAN.EXE, which is the National Instruments Resource Manager. You must run RESMAN every time the chassis power is cycled so that your application can access devices in the VXI/VME chassis. Notice that because Windows 95 supports the plug and play architecture, you do *not* need to run VXIINIT.EXE before you do any VXI operation.

After you run RESMAN, you are ready to use the NI-VXI Resource Editor program VXIEDIT to interactively configure the hardware in your system. Continue with Chapter 3, *NI-VXI Configuration Utility*, for instructions on using the configuration editors in VXIEDIT.

Using the Windows Setup Program (Windows NT)

You can use the Setup program that came with your NI-VXI software to install the entire NI-VXI software package, a software update, or to reinstall software in the event that your files were accidentally erased. Follow these steps to install all or part of the NI-VXI software.

1. Installing the LabWindows/CVI Run-Time System

Some of the NI-VXI utilities use the LabWindows/CVI Run-Time Engine to present a graphical user interface. Even if you already have LabWindows/CVI on your computer, you should install the LabWindows/CVI Run-Time Engine before installing the NI-VXI software.

1. Insert the disk labeled *LabWindows/CVI Run-Time Engine for Windows 95/NT*.

2. Select **Run...** from the **Start** menu on the taskbar or from the Program Manager **File** menu and run SETUP.EXE.

3. Accept the default destination directory as prompted, or choose another directory in which to install the LabWindows/CVI Run-Time Engine.

2. Installing the NI-VXI Software

This section describes how to install the NI-VXI software for the VXIpc-850 and Windows NT. Please carefully read these directions along with any messages on the screen before making your selections.

You can quit the Setup program at any time by pressing the **Cancel** button.

Setup is an interactive, self-guiding program that installs the NI-VXI software and configures your system to use the NI-VXI software with the VXIpc-850. Follow these steps to perform the installation.

1. Insert the disk labeled *NI-VXI for VXIpc-850 for Windows NT*.

2. Select **Run...** from the **Start** menu on the taskbar or from the Program Manager **File** menu and enter the following text, where *X* is your floppy drive (usually A or B).

   `X:\setup.exe`

   and press <Enter>.

3. Click on the **Next** button at the **Welcome** screen to start the installation.

4. Select the target directory for your installation. Although Setup prompts you to accept the `C:\NIVXI` directory by default, you can select a different location by clicking on **Browse...**.

Caution: *If you have a previous version of the NI-VXI software installed, Setup does not automatically create a backup of the software files. If there are files you want to preserve, you should exit the installation program now and make a backup before continuing.*

5. Select the components you want to install.

   • **NI-VXI for Windows NT** includes only the driver and utilities needed to configure and use your VXI or VME system.

   • **Microsoft C/C++ Development Files** includes the components necessary to develop applications using the Microsoft Visual C++ compiler. Please read the *Modifying the Environment* section if you choose to install these files.

- • **Borland C/C++ Development Files** includes the components necessary to develop applications using the Borland C/C++ compiler. Please read the *Modifying the Environment* section if you choose to install these files.

6. Select the name of the program folder that will contain icons for the NI-VXI software for VXIpc-850.

   At this point Setup copies the necessary files to your hard drive and creates program icons.

7. When the installation process completes, you must exit Windows and reboot your computer for the changes to take effect. You can let the Setup program reboot your computer by selecting the **Restart computer** option on the last screen. Click on **Finish** to end the installation.

8. If you backed up the manufacturer and model name files, you should restore them to the TBL subdirectory of your NI-VXI directory before running VXIEDIT.

9. You can now use VXIEDIT to configure the hardware in your VXI system. Continue with Chapter 3, *NI-VXI Configuration Utility*, for instructions on using the configuration editors in VXIEDIT.

Modifying the Environment

The installer does not modify the environment variables under Windows NT. No changes are necessary. However, you may want to add the NI-VXI directory to your LIB and INCLUDE paths if you use makefiles to build your projects, and to the PATH variable if you plan to run the VXI utilities from the command line.

Completing the Software Installation

After you execute Setup, you should restart Windows NT to make your system load the NI-VXI driver.

After the NI-VXI software is installed, run VXIINIT.EXE and then RESMAN.EXE. You need to run VXIINIT to initialize the VXIpc-850 before you perform any VXI operations and after each computer reset. RESMAN is the National Instruments Resource Manager, which you must run every time the chassis power is cycled so that your application can access devices in the VXI/VME chassis.

After you run VXIINIT and RESMAN, you are ready to use the NI-VXI Resource Editor program VXIEDIT to interactively configure the hardware in your system. Continue with Chapter 3, *NI-VXI*

Configuration Utility, for instructions on using the configuration editors in VXIEDIT.

Using the DOS INSTALL Program

If you do not have any Windows operating system installed on your machine, you can install the NI-VXI software for DOS with the INSTALL program. This program functions similarly to the Windows Setup program. You do not need to install the LabWindows/CVI Run-Time Engine.

Running INSTALL

Run INSTALL.EXE. This program copies the necessary files to your hard drive and performs the necessary steps to install the NI-VXI software for DOS to your system.

Caution: *In its default configuration, the VXIpc-850 requests memory space for use by the NI-VXI driver above the 1 MB DOS boundary. To run your NI-VXI software for the VXIpc-850 in DOS you must reconfigure the VXIpc-850 to request memory below the 1 MB DOS boundary.*

To change the requested memory space you need to use the VXIEDIT program described in Chapter 3, *NI-VXI Configuration Utility*. Use the **VXIpc Configuration Editor** and access the **Bus Configuration Editor** menu. Change the user and driver windows to be below the 1 MB boundary by selecting the **Yes** option. Please refer to the *User Window and Driver Window* section of Chapter 3 for more details.

NI-VXI Configuration Utility

This chapter contains instructions for using the VXI Resource Editor utility of the NI-VXI software to configure the VXIpc-850 Series embedded computer.

VXIEDIT.EXE is the VXI resource editor program that you use to configure the system and to edit the manufacturer name and ID numbers, the model names of VXI and non-VXI devices in the system, and the system interrupt configuration information. This program also displays the system configuration information generated by the Resource Manager.

The displays shown in this section are from the Windows 95 version of VXIEDIT. The Windows NT, Windows 3.1, and DOS versions of VXIEDIT have the same organization and functionality as the Windows 95 version although the displays may look slightly different. The descriptions and instructions in this chapter apply to all versions of VXIEDIT.

☞ **Note:** *A text-based version,* VXITEDIT*, is also available as an alternative. You can run* VXITEDIT *in either Windows 95/NT/3.1 or DOS. Although this chapter focuses only on the graphical* VXIEDIT *program, the two programs are functionally equivalent. For information on* VXITEDIT*, refer to the NI-VXI Text Utilities Reference Manual.*

Running the VXIEDIT Configuration Utility

To run VXIEDIT in Windows 95/NT/3.1, double-click the **VXIEDIT** (Windows) icon in the NI-VXI group. To run VXIEDIT in DOS, type VXIEDIT at the DOS command prompt. In DOS, you can run VXIEDIT from any directory, but make sure that both the PATH and NIVXIPATH environment variables have the destination directory of the NI-VXI software added to it. Under Windows 3.1 and DOS, NIVXIPATH is used by the application to find the different configuration files (*.CFG), table

files (*.TBL), and help files (*.HLP) during its execution. The default pathname used by the program if NIVXIPATH is not set is C:\NIVXI. Under Windows 95/NT, NI-VXI uses the system registry for all the configuration information.

Most of the features on the VXIpc-850 are configurable through software, using VXIEDIT, rather than through hardware switches or jumpers on the board itself. In addition, the VXIEDIT utility can override some of the hardware settings.

Figure 3-1 shows the main menu of the VXIEDIT resource editor.

Figure 3-1. VXIEDIT Main Screen

The rest of this chapter describes only the features of the **VXIpc Configuration Editor**. For instructions on using the other editors, refer to your software utility reference manual, either the *NI-VXI Graphical Utilities Reference Manual* or the *NI-VXI Text Utilities Reference Manual*.

VXIpc Configuration Editor

Figure 3-2 shows the opening screen of the **VXIpc Configuration Editor**. Notice that the screen displays the serial number and hardware revision of the VXIpc-850 board in addition to several configuration options.

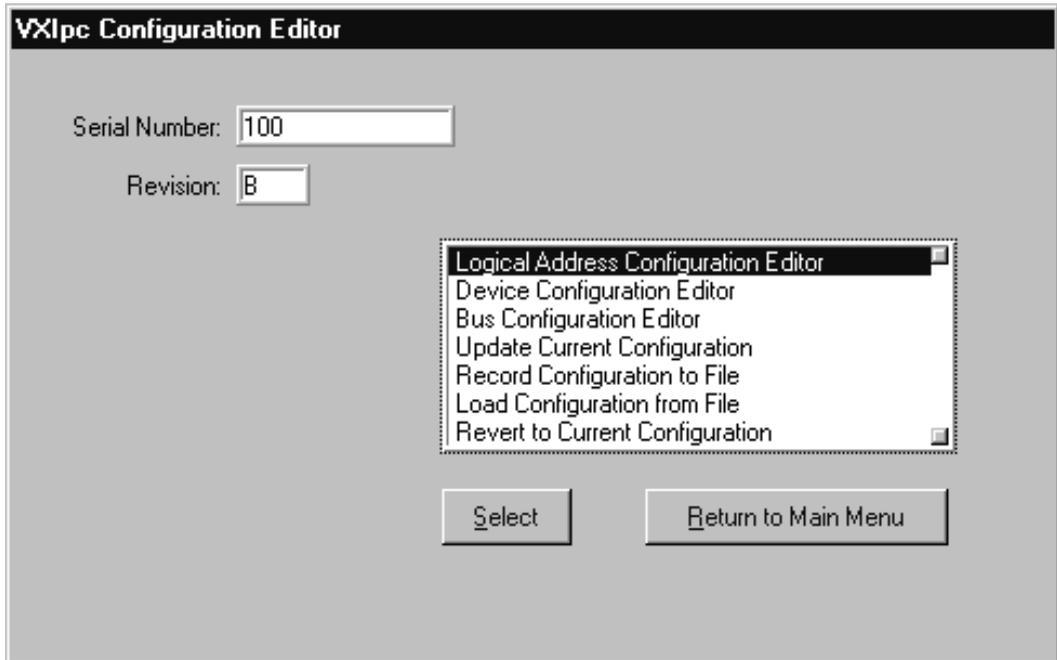**Figure 3-2.** VXIpc Configuration Editor

The first three options under the **VXIpc Configuration Editor** are:

- **Logical Address Configuration Editor**
- **Device Configuration Editor**
- **Bus Configuration Editor**

When making changes to the VXIpc-850 through these editors, remember that the changes do not take effect until you commit them by selecting the **Update Current Configuration** option.

♦   **DOS users**—If the VXIpc-850 driver window is located above 1 MB, the program will prompt you for a memory location below 1 MB. The memory location is a 32 KB window that allows VXIEDIT to access the VXIpc-850 registers. This is due to a DOS limitation that restricts DOS programs from accessing memory above 1 MB.

Before proceeding to a description of each field in these editors, review the remaining four options of the **VXIpc Configuration Editor**. These options directly relate to how you can use the changes you make using the configuration editors, which are described after the options.

Update Current Configuration

Use this option to write the configuration settings to the VXIpc-850 EEPROM and files used by NI-VXI. It configures the VXIpc-850 to be consistent with the configuration EEPROM. Notice that some of the configuration settings cannot take effect until you reset the machine, either by using the reset button or by turning the power off and on again.

☞   **Note:**   *You will not be able to use the <Control-Alt-Del> keystroke combination or the Windows* **Restart** *command to perform this reset. Instead you must perform a hardware reset as described above.*

Record Configuration to File

With this option you can save your configuration settings to a file. Notice that this option does *not* write the configuration settings to the VXIpc-850 configuration EEPROM.

If you want to update the VXIpc-850 configuration settings, use the **Update Current Configuration** option instead.

Load Configuration from File

You can use this option to load your configuration settings from a file. This action only updates the configuration settings in your editor. This does *not* write the configuration settings to the VXIpc-850 configuration EEPROM. To update the configuration use the **Update Current Configuration** option for the changes to take effect.

Revert to Current Configuration

If you made changes to the configuration settings without committing those changes (writing to configuration EEPROM), you can revert the configuration settings to the values they had before you made the changes.

☞ **Note:** *You can successfully revert only if you have NOT yet selected the* **Update Current Configuration** *option.*

Logical Address Configuration Editor

Figure 3-3 shows the **Logical Address Configuration Editor**. Notice that the options are arranged into three groups—**Device Settings**, **VXI Shared Memory**, and **Resource Manager**. The following paragraphs describe the options you can select for each of the fields.
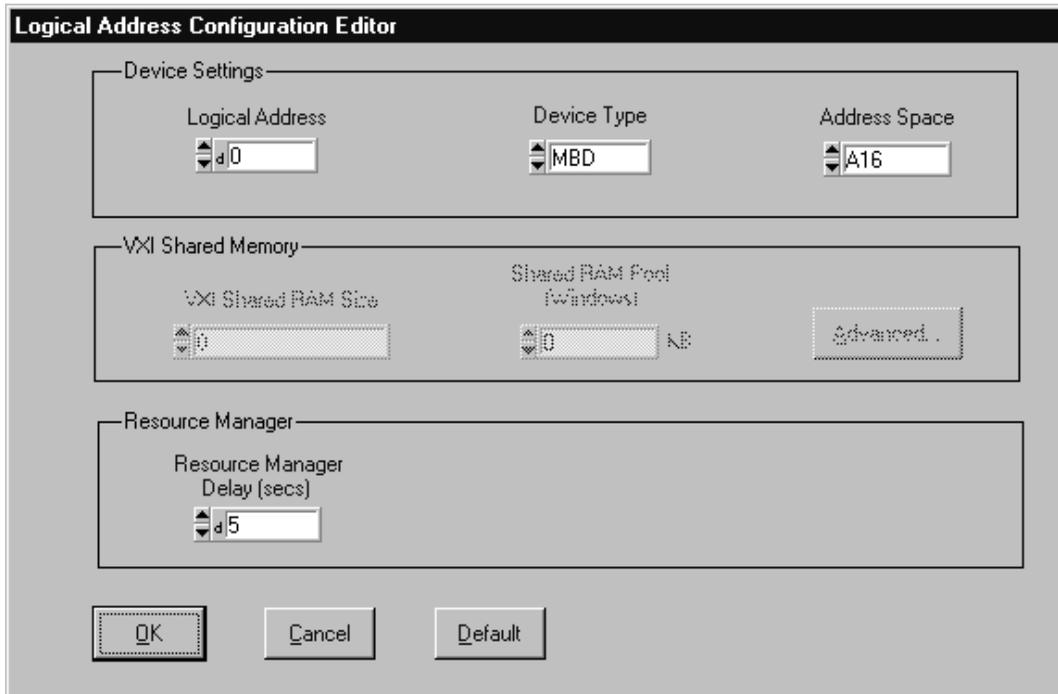

Figure 3-3. Logical Address Configuration Editor

Device Settings

The **Device Settings** group contains the controls to set the logical address, device type, and address space of the VXIpc-850.

Logical Address

This parameter sets the logical address of the VXIpc-850. The following table shows the allowable range of values and the default value.

Logical Address Range	Default Value
0 to 254	0

Device Type

This field indicates the classification of the VXIpc-850. The default value is **MBD**, designating a message-based device. The following table shows the available options.

Classification	Setting
Extended Device	**EXT**
Message-Based Device	**MBD**
Register-Based Device	**RBD**

The device type affects only the contents of the **Device Class** field in the Device Type register. The functionality of the other registers does not change.

Address Space

This field indicates the addressing mode(s) of the device's operational registers. The VXIpc-850 can be configured in one of three ways. The default addressing mode is for **A16** space only. Your other options are **A16/A24** and **A16/A32**.

Notice that the options under the **VXI Shared Memory** group are disabled when the **Address Space** control is set to **A16**, as shown in Figure 3-3. Only if you select **A16/A24** or **A16/A32** are the following controls relevant:

• **VXI Shared RAM Size**

• **Shared RAM Pool (Windows)**

• **Advanced**

– **Lower Half Window Byte Order**

– **Upper Half Window Byte Order**

– **Upper/Lower Half Map To. . .**

VXI Shared Memory

The **VXI Shared Memory** group contains the controls to set the VXI shared RAM size and the shared RAM pool. The **Advanced** button leads to additional options that configure the upper and lower half of the shared RAM area.

☞ **Note:**   *When the* **Address Space** *field is in the default setting of A16 only, all of the options in this group are disabled because they are irrelevant.*

VXI Shared RAM Size

This field indicates the amount of RAM (in bytes) that are shared in either A24 or A32 space. This determines the *total* shared RAM size. Setting this field to **All System RAM** will detect how much memory you have installed in your VXIpc-850 and request the same amount of A24 or A32 space.

☞ **Note:**   *If you have more than 8 MB installed in your VXIpc-850, the* **All System RAM** *setting is allowed only if you use A32 memory space.*

Shared RAM Pool (Windows)

This field indicates the size of memory in kilobytes that is allocated on Windows startup. This is physically contiguous memory that can be dual-ported on the VXIbus.

The shared RAM pool is used by the VXImemAlloc() function calls from both Windows applications and DOS applications running under Windows. For information on the VXImemAlloc() function, refer to the *NI-VXI Software Reference Manual for C*.

If you make a change to this setting, you must restart Windows 95/NT/3.1 to enable the change.

Memory Range	Default Value
0 to 65535 KB	0 KB

> **Caution:**   *This memory pool is allocated at Windows 95/NT/3.1 startup, and Windows is denied access to this memory. Take into account the memory requirements of Windows and your applications and the amount of RAM in your system before setting this option.*

Advanced Shared RAM Settings

Click on the **Advanced** button to reach additional configuration options for the VXI shared RAM, as shown in Figure 3-4. These options are intended for more advanced users.

The VXI shared RAM is divided into two halves, or *windows*. You can select the byte order and mapping scheme for each half independently.
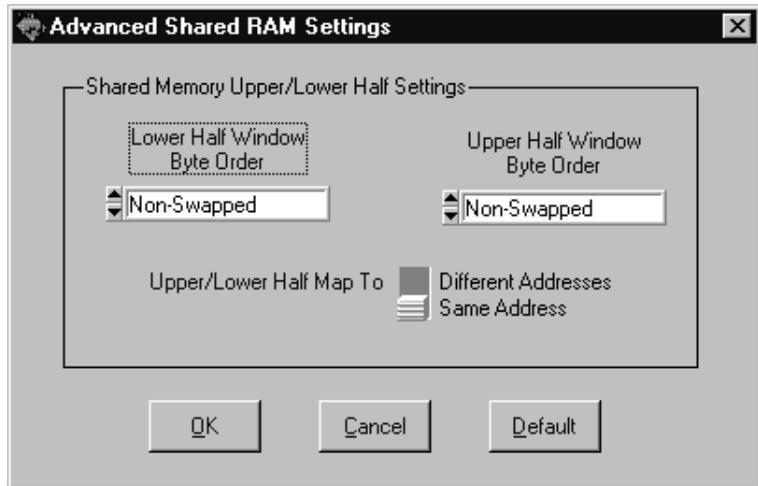

Figure 3-4. Advanced Shared RAM Settings

Upper/Lower Half Window Byte Order

This field indicates whether byte swapping should be performed for slave accesses to this half of the VXI shared RAM space. For example, if the native byte order of the shared RAM is Intel (Little Endian), and you want to present data to the VXIbus in Motorola (Big Endian) byte order, you will need to enable byte swapping. The default value is **Non-Swapped**. Choose **Swapped** to enable byte swapping.

Upper/Lower Half Map To

This switch determines if the upper/lower half windows map to the same address or different addresses in system memory.

When this option is set to **Same Address**, the buffer in system RAM can be dual-ported to the VXIbus in both Little Endian and Big Endian byte order. The **Byte Order** setting for each half window determines whether the byte order is Little Endian or Big Endian. When this option is set to **Different Addresses**, which is the default value, each half window maps to a unique local address on the VXIpc-850.

Resource Manager

The only option under the **Resource Manager** portion of the **Logical Address Configuration Editor** is the **Resource Manager Delay** control.

Resource Manager Delay

☞ **Note:** *This field is effective only when the VXIpc-850 is at its default logical address of 0. This logical address is required for the Resource Manager.*

This field specifies the time in seconds that the Resource Manager (RM) waits before accessing any other VXIbus device's A16 configuration registers.

RM Delay Range	Default Value
0 to 65535 s	5

Device Configuration Editor

Figure 3-5 shows the **Device Configuration Editor**. The following paragraphs describe the options you can select for each of the fields.
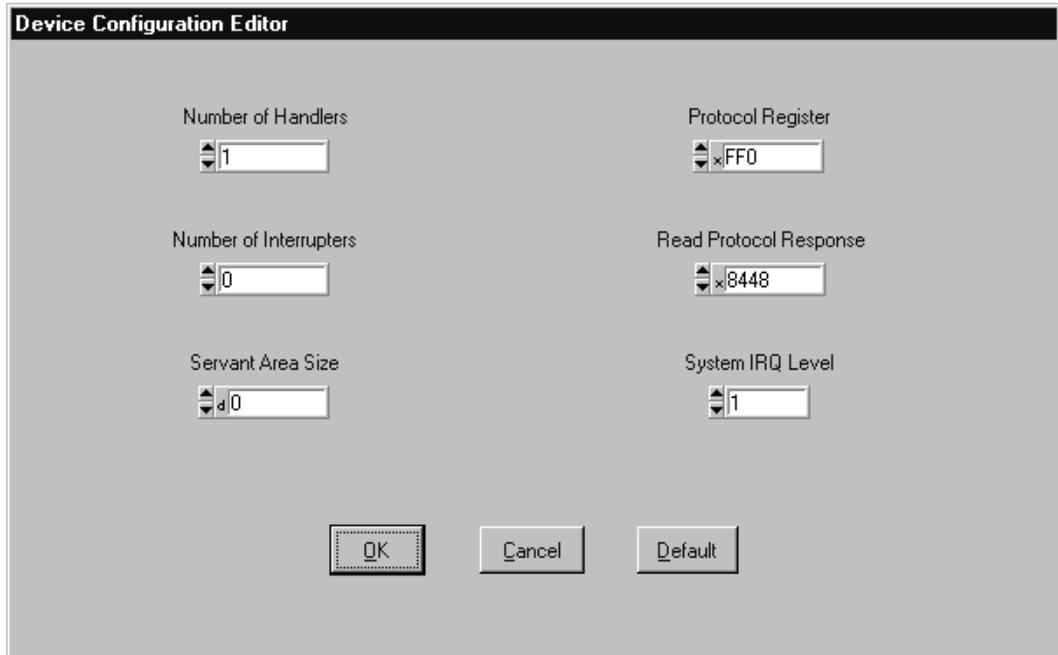

Figure 3-5. Device Configuration Editor

Number of Handlers

This field gives the number of interrupt handlers that the VXIpc-850 supports.

Interrupt Handlers	Default Value
0 to 7	1

Number of Interrupters

This field gives the number of interrupters that the VXIpc-850 supports.

Interrupters	Default Value
0 to 7	0

Servant Area Size

This field designates the servant area size, which is supplied to the Resource Manager in response to the *Read Servant Area* command (if the VXIpc-850 is *not* the Resource Manager in your system). The servant area size is an 8-bit value (0 through 255) that indicates the servant area. The servant area begins at the logical address following the VXIpc-850 logical address, and includes *N* contiguous logical addresses, where *N* is the value of the servant area size. This field is meaningful only when the VXIpc-850 is configured as a message-based device.

Servant Area Range	Default Value
0 to 255	0

☞ **Note:** ***If the VXIpc-850 is the Resource Manager (Logical Address 0), this setting is irrelevant.***

Protocol Register

This field specifies the contents of the Protocol register, indicating which protocols the device supports. This field is meaningful only when the VXIpc-850 is configured as a message-based device. The default value is 0x0FF0 (Commander, Signal Register, Master).

Read Protocol Response

This field specifies the response value to a *Read Protocol* command received by the VXIpc-850 from the Resource Manager (if the VXIpc-850 is *not* the Resource Manager in your system). This field is meaningful only when the VXIpc-850 is configured as a message-based device. The default value is 0x8448 (Response Generation, Event

Generation, Programmable Handler, Word Serial Trigger, Instrument, Extended Longword Serial, Longword Serial).

System IRQ Level

Remote controllers can report events such as triggers and DMA to the VXIpc-850 through a VXI IRQ line. This field selects which VXI IRQ level the remote controllers should use to report such events.

Interrupt Request Levels	Default Value
1 to 7	1

☞ **Note:** *The system IRQ level cannot be disabled and the VXIpc-850 will always acknowledge it automatically when it is the Resource Manager.*

Bus Configuration Editor

Figure 3-6 shows the **Bus Configuration Editor**. The following paragraphs describe the options you can select for each of the fields.
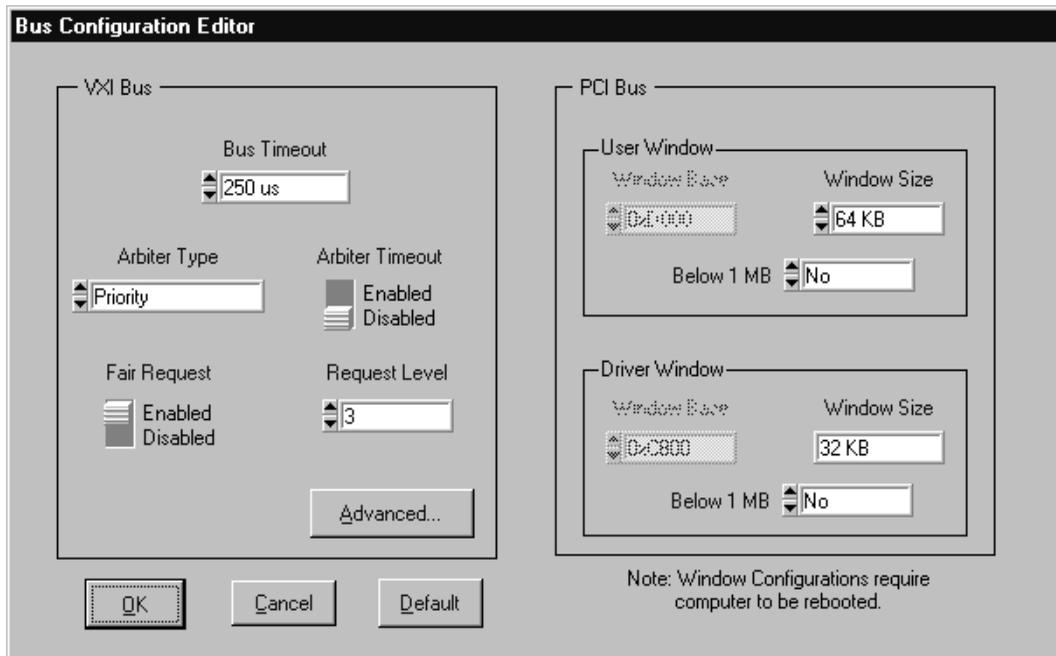

Figure 3-6. Bus Configuration Editor

VXI Bus

The following paragraphs describe the options for the **VXI Bus** portion of this editor.

VXI Bus Timeout

The Bus Timeout (BTO) is a watchdog timer for transfers on the VXIbus. After the specified amount of time has elapsed, the BTO circuitry terminates a VXIbus cycle if no slave has responded. This control is applicable only if the VXIpc-850 you are configuring is a VXI Slot 0 device. You should disable the BTO of any other non-Slot 0 devices residing in the mainframe.

The lowest value in the allowable range is 15 µs and the highest is 256 ms. The default value is 250 µs.

Arbiter Type

You can use the **Arbiter Type** feature to configure the VXIpc-850 as either a Priority or Round Robin VMEbus arbiter. This control is applicable only if the VXIpc-850 you are configuring is a VXI Slot 0 device. The default value is **Priority**.

When configured for **Priority** arbitration, the VXIpc-850 grants the bus to the highest pending bus request level. In **Round Robin** arbitration mode, the VXIpc-850 grants the bus to the next highest bus request level after the level of the previous bus owner. This effectively gives the same priority to each bus request level. Refer to the VMEbus specification for more information on the different types of arbiters.

Arbiter Timeout

An arbitration timeout feature is available on the VXIpc-850 when it is acting as the VMEbus arbiter. This feature applies only to a VXI Slot 0 VXIpc-850. The default value is **Disable**.

The timer begins when the arbiter circuit on the VXIpc-850 drives one of the BGOUT lines on the backplane. If no device takes over the bus within the timeout limit, the BGOUT is removed and the bus is either idle or granted to another requester.

Fair Request

The VXIpc-850 is always a Release On Request requester. However, you can configure whether the VXIpc-850 acts as either a fair or unfair requester on the VXIbus. The default value for this control is **Enable**, signifying a fair requester. For more information on the different types of requesters, refer to the VMEbus specification.

Request Level

The VXIpc-850 uses one of the four VXIbus request levels (0 to 3) to request use of the VXI Data Transfer Bus (DTB). The VXIpc-850 requests use of the DTB whenever a local cycle maps into a VXIbus cycle.

The VXIpc-850 uses VXIbus request level 3 in its factory-default setting, as required by the VXIbus specification. This is suitable for most VXIbus systems. However, you can change the VXIpc-850 to use any of the other three request levels (0, 1, or 2) by changing the setting of the **Request Level** control. You may want to change request levels to change the priority of the VXIpc-850 request signal. For more information, refer to the VMEbus specification.

Advanced VXI Bus Settings

Click on the **Advanced** button to reach additional configuration options for the VXI bus, as shown in Figure 3-7. These options are intended for more advanced users.
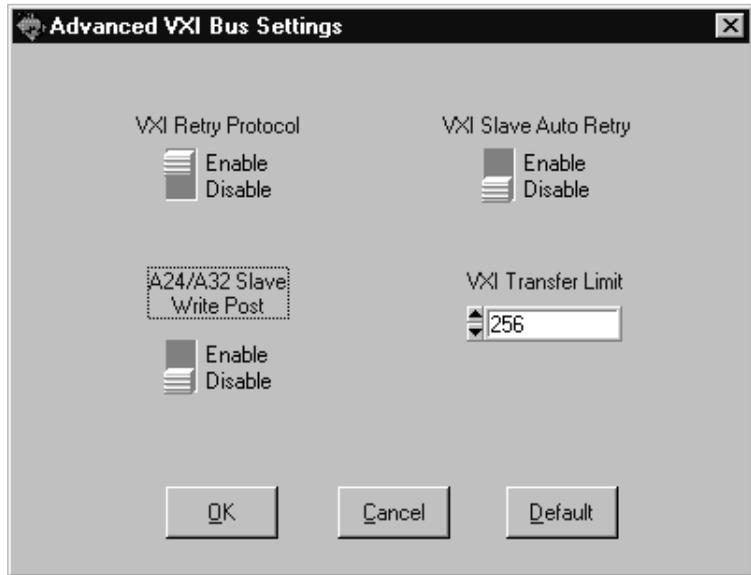
Figure 3-7. Advanced VXI Bus Settings

A24/A32 Slave Write Post

The VXIpc-850 can increase performance with its capability to post write cycles from the VXIbus. Write cycles should be posted only to addresses that cannot return a BERR signal, because the BERR will not be reported to the originating master. The options are **Enable** and **Disable**. By default, this option is disabled.

The **A24/A32 Slave Write Post** control affects write cycles to the VXIpc-850 via its requested memory space from the VXIbus. When this option is enabled, the VXIpc-850 will complete a VXIbus write cycle before writing the data from the cycle to the local destination on the VXIpc-850.

VXI Retry Protocol

Setting this control to **Enable** causes the VXIpc-850 to recognize and send the VXIbus retry protocol. Otherwise a retry is mapped to a bus error response. By default this option is disabled.

VXI Slave Auto Retry

The VXIpc-850 has an automatic retry feature for cycles that map from the VXIbus to the PCI bus on the VXIpc-850. You can select **Enable** or **Disable** for this control. By default this option is enabled.

Normally, when a cycle maps from the VXIbus to the PCI bus, any retry response received on the PCI bus is passed to the VXIbus. When the **VXI Slave Auto Retry** feature is enabled, the VXIpc-850 automatically retries any PCI cycle when the PCI host responds to a cycle with a retry. The VXIpc-850 automatically continues to retry the PCI cycle until it receives either a Disconnect or Target-Abort response, which it then passes to the VXIbus. This behavior is the default because many VXIbus masters do not support VXI retries. If the VXIbus master does support retries, it may be beneficial to disable **VXI Slave Auto Retry**. With this feature disabled, you can lower the value of the **Bus Timeout** because there will be no delay due to the inward cycles being retried.

☞ **Note:** *The VXIpc-850 has a limit on the number of automatic retries it will perform on any one cycle. If the limit is exceeded and the VXIpc-850 receives another retry, it will pass a retry or BERR (depending on whether the* **VXI Retry Protocol** *control is set to* **Enable***) to the VXIbus even though* **VXI Slave Auto Retry** *is enabled.*

VXI Transfer Limit

You can use this feature to control how many data transfers the VXIpc-850 will perform on the VXIbus before releasing it to another master device that is requesting use of the bus.

The available options you can choose from are **16**, **64**, and **256** transfers. If you do not want the VXIpc-850 to hold the VXIbus long enough to perform 256 transfers (the default value), you can use this control to select a smaller value.

PCI Bus

Your VXIpc-850 uses the MITE custom ASIC to interface to the VXIbus. The MITE chip is located on the PCI bus on the VXIpc-850.

The following paragraphs describe the options for the **PCI Bus** portion of the **Bus Configuration Editor**. You can use these options to modify settings relating to how much PCI address space the MITE chip will use and where this space will be located.

User Window and Driver Window

The VXIpc-850 driver requires the use of two PCI windows: a user window and a driver window. Calling the `MapVXIAddress()` function allocates regions of the user window to your application. `VXIpeek()` and `VXIpoke()` accesses are performed through this window. NI-VXI uses the driver window to perform high-level functions such as `VXIin()` and `VXIout()`, and to access MITE registers on the VXIpc-850.

The windows are mapped to PCI base address registers and determine the amount of PCI memory space the VXIpc-850 requests from the PCI system during initialization. You can set the window base, window size and whether the window resides above or below the 1 MB address space boundary.

☞ **Note:** **(To DOS Users)** *If you are intending to use DOS applications, note that DOS restricts programs from accessing memory above 1 MB. Verify that the* **Below 1 MB** *controls for both the user window and the driver window are set to* **Yes.**

Window Base

This field allows you to select a region of memory below 1 MB for the user or driver windows. Use this field to assign a new address for these windows. You should choose a region that does not conflict with any other devices you have installed in your system.

☞ **Note:** *This option is not available under Windows 95/NT. However, under Windows 95 you can change the base of the driver/user window by using the* **Resources** *page in the* **VXIpc-850 Properties** *dialog in the Device Manager.*

Notice that you can select the **Window Base** control only if the **Yes** option is active for the **Below 1 MB** control.

Window Size

The amount of space you can allocate for the user window is system dependent. By using the up/down arrow of this parameter, you can select the size of the user window (minimum of 4 KB, maximum of 2 GB). The more you increase the size of the user window, the larger the window you can map in `MapVXIAddress()`.

You can also disable this option. Disabling the user window causes the VXIpc-850 to request the minimum amount of address space on the PCI bus. With the window disabled, you will be unable to perform any low-level function calls such as `VXIpeek()`, `VXIpoke()`, and `MapVXIAddress()`. For example, on DOS systems, you may not be able to request more than the 32 KB driver window if the address space between 640 KB and 1 MB is being used by other devices in your computer.

The default setting for the user window is set at 64 KB. It is recommended to have a user window of at least this value. If you are going to be initiating transfers to a wide variety of addresses in both A24 and A32, you should increase the size of the user window.

The size of the driver window, however, is system defined and is not user configurable.

Below 1 MB

This field determines whether the user or driver window is required to be located below the 1 MB address boundary (a DOS limitation). By default the user and driver windows are above the 1 MB boundary, as denoted by the **No** option for this control. Unless you are running in DOS, you can keep the windows above the 1 MB boundary, and it is recommended that you keep the default settings.

Using the NI-VXI Software
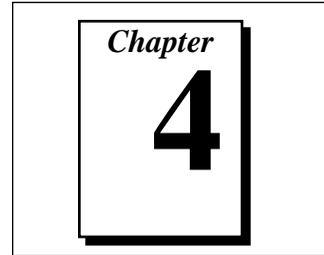
This chapter discusses programming information for you to consider when developing applications that use the NI-VXI driver.

After installing the driver software, you can begin to develop your VXI application software. Be sure to check the README.DOC file for the latest application development notes.

♦ **DOS, Windows NT, and Windows 3.1 users**—You must run the VXIINIT initialization program before performing any VXI operations and after each computer reset.

You must also run RESMAN each time the chassis power is cycled so that your application can access devices in the VXI chassis.

Refer to the *NI-VXI Software Reference Manual for C* for an overview of NI-VXI and detailed descriptions of the NI-VXI functions.

Interactive Control of NI-VXI

The easiest way to learn how to communicate with your instruments is by controlling them interactively. Use the VXI interactive control utility (VIC) to write to and read from your instruments. This utility displays the status of your VXI transactions and informs you of any errors that occur.

Refer to the *NI-VXI Graphical Utilities Reference Manual* for instructions on how to use VIC and to learn about its features. If you are using VICTEXT refer to the *NI-VXI Text Utilities Reference Manual* for information.

Warning: *DOS and Windows 3.1 users—If* `NIVXI.DLL` *is loaded in memory, do not attempt to execute* `VIC` *or any DOS program that uses the NI-VXI library in a DOS shell. Conflicts occur if both the DOS and Windows NI-VXI drivers are active at the same time, and may cause a system failure. To guard against this conflict, the safest approach is to always exit Windows before attempting to execute any DOS program that uses the NI-VXI Library, including* `VIC`*. You can execute* `VIC` *from a Windows DOS shell, however, if you ensure that no other Windows application that uses the* `NIVXI.DLL` *file is executing.*

Note: *When compiling NI-VXI applications, you must define one of these macros in your makefile/project:*

- `VXIWIN` *(if you are developing a Windows 3.1 application)*
- `VXIDOS` *(if you are developing a DOS application)*
- `VXINT` *(if you are developing a Windows 95/NT application)*

Refer to the example programs on your software diskettes for details.

Example Programs

The `EXAMPLES` subdirectory contains various example programs along with a makefile that show how to use various functions in the NI-VXI software and how to develop application programs using these functions. Make certain that the environment variables `LIB` and `INCLUDE` are set correctly as described in Chapter 2, *NI-VXI Software Installation*. Also refer to the *NI-VXI Software Reference Manual for C* for additional examples.

Programming Considerations

The following paragraphs contain information for you to consider when developing DOS or Windows 95/NT/3.1 applications that use the NI-VXI bus interface software. This information applies to all four Microsoft operating systems unless otherwise noted.

Memory Model (DOS or Windows 3.1 Only)

The NI-VXI libraries were compiled using the large memory model. All DOS applications must also be compiled for the large memory model. However, Windows 3.1 application programs that link with the NI-VXI library can also use the medium, compact, or small memory models. Because of this ability to use different memory models for

your application, you not only can take advantage of the efficiency inherent in small memory model programs, but also run multiple instances of the application as well. (Normally, you cannot run multiple instances of an application if it is a large memory model application.)

Multiple Applications Using the NI-VXI Library

Multiple-application support is another feature in the NI-VXI library. You can have several applications that use the NI-VXI library running simultaneously in Windows 95/NT/3.1. In addition, you can have multiple instances of the same application that uses the NI-VXI library running simultaneously. The NI-VXI functions perform in the same manner whether you have only one application that uses the NI-VXI library or several applications (or several instances of an application) all using the NI-VXI library.

However, you do need to be careful in certain cases as described in the following section.

Low-Level Access Functions

The memory windows used to access the VXIbus are a limited resource. You should follow the protocol of calling the `MapVXIAddress()` function with Access Only mode first before attempting to perform low-level VXIbus access with `VXIpeek()` or `VXIpoke()`. Your application should always call the `UnMapVXIAddress()` function immediately after the accesses are complete so that you free up the memory window for other applications.

The function `MapVXIAddress()` returns a pointer for use with low-level access functions. It is strongly recommended to use the `VXIpeek()` and `VXIpoke()` macros to access the memory instead of directly dereferencing the pointer. Using these macros makes the NI-VXI software more portable between platforms, because some platforms (such as the AT-MXI) require checking for retries, which can be handled through the macros. Directly dereferencing the pointers does not give you any speed benefit because the macros reduce to pointer dereferences at compile time for the VXIpc-850. Refer to the *Compiling Your C Program* section later in this chapter for more information on portability issues, and to your NI-VXI software reference manual for more information on low-level VXIbus access functions.

Setting User Handlers (DOS or Windows 3.1 Only)

You can set a user handler that will be invoked when certain conditions occur, such as VXI signals and triggers, by using functions such as `SetSignalHandler()` in the NI-VXI library. However, setting a new user handler replaces the existing handler, meaning that the existing handler will no longer be invoked when the condition occurs. The following example illustrates the point.

```
SetSignalHandler(5, mySignalHandler1)

/* mySignalHandler1 is now the handler for VXI
signals from logical address 5. */

SetSignalHandler(5, mySignalHandler2)

/* mySignalHandler1 is replaced by mySignalHandler2
as the new handler for VXI signals from logical
address 5. */
```

Caution: **DOS and Windows 3.1 users**—*Avoid running multiple applications that indiscriminately change user handlers, or try to make sure that when the applications do change the user handlers, they do so in a coordinated manner. Setting new handlers while using multiple applications runs the risk of one application inadvertently overwriting the handler that another application had set up, causing disruptions and incorrect behavior.*

Local Resource Access Functions

By using `VXIEDIT` or `VXITEDIT`, you can set up the VXIpc-850 to share the system memory on the VXIpc-850 motherboard with the VXI system. Refer to the *NI-VXI Graphical Utilities Reference Manual* or the *NI-VXI Text Utilities Reference Manual* for more information on setting these parameters.

Notice that sharing the system memory with the VXI system does not mean that the entire range of shared system memory is available to be used for VXI transfers. You need to be cautious in specifying the portion of memory you want to share, as some areas are already used for other purposes.

Warning: *Use* `VXImemAlloc()` *to allocate a buffer in the system memory that is reserved for your use only. Using any range of addresses that was not returned from* `VXImemAlloc()` *to receive data may cause your computer to crash or behave incorrectly.*

Another factor to consider is that although you may have selected to share 8 MB, you must also inform Windows 95/NT/3.1 how much memory to set aside for possible VXImemAlloc() calls. You can use the **Shared Memory Pool (Windows)** option in VXIEDIT and VXITEDIT for this purpose. But remember that the memory you put into this pool is no longer available to Windows. If this setting is too large, you may experience memory limitation problems when you run Windows. Also remember that changes in the size of the pool do not take effect until the next time you start Windows.

System Configuration Functions

The System Configuration functions provide the lowest-level initialization of your NI-VXI software and VXI controller. You must use the InitVXIlibrary() function at the start of each application and the CloseVXIlibrary() function at the end of each application.

Compiling Your C Program

You can use the sample programs included with the NI-VXI software as a starting point to develop your own C program that uses NI-VXI functions. First, look over and compile the sample program using the makefile provided to get familiar with how the functions operate. The example program is broken into multiple files, and each file shows how to use different groups of functions. You can then modify the sample program to try out different aspects of the NI-VXI software.

♦ **Windows 3.1 users**—The sample Windows 3.1 program for the Microsoft C compiler is in the \nivxi\win\msc\examples directory, and the sample Windows program for the Borland C compiler is in the \nivxi\win\borlandc\examples directory.

♦ **Windows 95/NT users**—The sample Windows 95/NT program for the Microsoft C compiler is in the \nivxi\win32\msc\examples directory, and the sample Windows 95/NT program for the Borland C compiler is in the \nivxi\win32\borlandc\examples directory.

♦ **DOS users**—The sample DOS program for the Microsoft C compiler is in the \nivxi\dos\msc\examples directory, and the sample DOS program for the Borland C compiler is in the \nivxi\dos\borlandc\examples directory.

The easiest way to compile the sample program is to use the makefile included with the NI-VXI software. If you are using the Microsoft C compiler, go to the Microsoft C sample directory and type nmake to

compile that program. If you are using the Borland C compiler, go to the Borland C sample directory and type `make -f example.mak`.

Symbols

You may need to define some symbols so that the NI-VXI library can work properly with your program. You can define the symbols using `#define` statements in the source code or you can use either the `/D` or the `-D` option in your compiler (both the Microsoft and Borland compilers support the `/D` and `-D` options). If you use `#define` statements, you must define the symbols before including the NI-VXI header file `nivxi.h`. If you use the makefiles to compile the sample program, the makefile already defined the necessary symbols.

One of the following symbols is usually required. You must define it when using the Microsoft C or Borland C compiler.

- `VXIWIN` designates the application as a Windows 3.1 application.

- `VXINT` designates the application as a Windows 95/NT application.

- `VXIDOS` designates the application as a DOS application. You can use the same NI-VXI header files to compile DOS programs by defining `VXIDOS`.

☞ **Note:** *Because LabWindows/CVI cannot be used to compile DOS programs, the correct symbol is automatically defined. You should not define* `VXIWIN` *or* `VXINT` *when using LabWindows/CVI.*

The following symbol is optional.

- `BINARY_COMPATIBLE` makes the application binary compatible with external controllers, such as the National Instruments PCI-MXI-2 external controller. Using this option may cause a slight performance degradation when using low-level VXIbus access functions.

If you define these symbols in your source code, your source code should look something like the following sample code:

```
#define VXIWIN

#define BINARY_COMPATIBLE

.

.

.

#include <nivxi.h>
```

If you define these symbols using the /D or -D compiler options, you should specify the following when invoking the compiler.

For the Microsoft C compiler:

```
/DVXIWIN /DBINARY_COMPATIBLE
```

For the Borland C compiler:

```
-DVXIWIN; BINARY_COMPATIBLE;
```

Refer to the documentation that came with your compiler package for detailed instructions about using the compiler and the various tools (linker, debugger, and so on). Your compiler documentation is an important and useful source of information for writing, compiling, and debugging C programs.

NI-VXI Software Overview
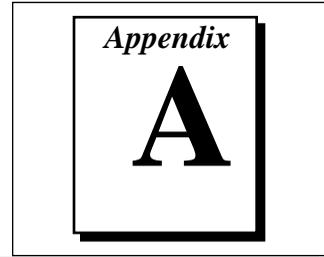
This appendix lists and describes the main programs and files that make up the NI-VXI software.

Main Programs and Files

This section lists the main programs and files of the NI-VXI software.

☞ **Note:** *Any executable not listed in this section is used by the driver and should not be executed by the user directly.*

- VXIINIT.EXE is the VXIpc-850 initialization program. You can execute VXIINIT from DOS, Windows 95/NT, the Windows 95/NT DOS shell, or the Windows 3.1 DOS shell, although the latter is not recommended. This program initializes the VXIpc-850. It may be included in the DOS batch file AUTOEXEC.BAT or the Windows Startup folder so that the VXIpc-850 is automatically initialized at startup. The configuration settings can be modified using the VXIEDIT.EXE or VXITEDIT.EXE program.

- RESMAN.EXE is the National Instruments multiple-mainframe Resource Manager. You can execute RESMAN from DOS, Windows 95/NT, the Windows 95/NT DOS shell, or the Windows 3.1 DOS shell, although the latter is not recommended. RESMAN.EXE may be executed only after VXIINIT.EXE has been run.

- VIC.EXE is an interactive control program that executes functions you enter from the keyboard. VIC helps you learn the functions, program your VXI devices, and develop and debug your application program. You can execute VIC from DOS, Windows 95/NT, the Windows 95/NT DOS shell, or the Windows 3.1 DOS shell, although the latter is not recommended. If you do use VIC from the Windows 3.1 DOS shell, you must ensure that no other Windows application that uses NI-VXI functions is executing. This

program is described in detail in the *NI-VXI Graphical Utilities Reference Manual.*

- VICTEXT.EXE is a text-based interactive control program that is functionally equivalent to VIC.EXE. You can execute VICTEXT from DOS, Windows 95/NT, the Windows 95/NT DOS shell, or the Windows 3.1 DOS shell, although the latter is not recommended. If you run VICTEXT as a Windows application, you can use it at the same time that other Windows applications that use NI-VXI functions are executing. This program is described in detail in the *NI-VXI Text Utilities Reference Manual.*

- VXIEDIT.EXE is the VXI resource editor program. You use this program to configure the system, and to edit various details such as the model names of VXI devices, the manufacturer name and ID numbers, and system interrupt configuration. This program is described in detail in the *NI-VXI Graphical Utilities Reference Manual.*

- VXITEDIT.EXE is the text-based VXI resource editor program that is functionally equivalent to VXIEDIT.EXE. You can execute VXITEDIT from DOS, Windows 95/NT, the Windows 95/NT DOS shell, or the Windows 3.1 DOS shell. This program is described in detail in the *NI-VXI Text Utilities Reference Manual.*

- README.DOC contains the latest updates and corrections to the manual when appropriate.

Header Files

The C:\NIVXI\INCLUDE directory contains the following include files for the Microsoft C and Borland C language interfaces.

- NIVXI.H is the main header file containing the C prototypes for the NI-VXI functions.

- DATASIZE.H contains data size specifications.

- BUSACC.H contains parameter and return values for the bus access functions.

- DEVINFO.H contains parameter and return values for the device information and system configuration functions.

- VXIINT.H contains parameter and return values for the interrupt and signal functions.

- SYSINT.H contains parameter and return values for the system interrupt functions.

- `TRIG.H` contains parameter and return values for the trigger functions.

- `WS.H` contains parameter and return values for the Commander and Servant Word Serial functions.

- `NIVXI.INC` is the include file for the Visual Basic for DOS language interface.

Common Questions

This appendix addresses common questions you may have about using the NI-VXI bus interface software on the VXIpc-850 platform.

How can I determine which version of the NI-VXI software I have installed?

Run the NI-VXI utility program VIC or VICTEXT. At the prompt type ver, and the utility will display the versions of VIC/VICTEXT and NI-VXI, and the latest VXIpc-850 hardware revision that this NI-VXI driver supports.

How can I determine the revision of the VXIpc-850 that my NI-VXI software supports?

Running the NI-VXI utility program VICTEXT as described above will display the versions of VICTEXT and NI-VXI, and the hardware revision of the VXIpc-850 that the NI-VXI software supports.

How can I determine the serial number and hardware revision of the VXIpc-850?

Run the NI-VXI utility program VXIEDIT. Choose the **VXIpc Configuration Editor**. The opening screen displays the serial number and hardware revision of the VXIpc-850.

Which NI-VXI utility program must I use to configure the VXIpc-850?

Use the VXI Resource Editor program, either VXIEDIT or VXITEDIT, to configure the VXIpc-850.

Which NI-VXI utility program must I use to initialize the VXIpc-850?

Use the hardware initialization program, VXIINIT, to initialize the VXIpc-850. This is required for Windows NT/3.1 or DOS, but not for Windows 95.

Which NI-VXI utility program must I use to perform startup Resource Manager operations?

Use the RESMAN program to perform startup Resource Manager operations. RESMAN uses the settings in the Configuration Editor of VXIEDIT or VXITEDIT. It initializes your VXIbus system and stores the information that it collects to the RESMAN.TBL file in the TBL subdirectory of the NIVXI directory.

What can I do to make sure that my system is up and running?

The fastest method for testing the system is to run RESMAN. This program attempts to access memory in the upper A16 address space of each device in the system. If RESMAN does not report any problems, the VXI communication system is operational.

To test individual devices, you can use the VIC or VICTEXT program to interactively issue NI-VXI functions. You can use the VXIin() and VXIout() functions or the VXIinReg() and VXIoutReg() functions to test register-based devices by programming their registers. If you have any message-based devices, you can send and receive messages with the WSwrt() and WSrd() functions. Notice that VXIinReg() and VXIoutReg() are for VXI devices only.

Finally, if you are using LabVIEW or LabWindows/CVI and you have instrument drivers for the devices in your chassis, you can use the interactive features of these programs to quickly test the functionality of the devices.

What do the LEDs on the front of the VXIpc-850 mean?

The LEDs are fully described in Appendix C, *LED Indicators*, in the *VXIpc-850 Series User Manual*.

Is something wrong with the VXIpc-850 if the red SYSFAIL and FAILED LEDs stay lit after booting the VXIpc-850?

If either the **SYSFAIL** or **FAILED** LED remains lit, perform the following steps:

1. Power off the mainframe.

2. Remove all other modules from the mainframe.

3. Make sure that the VXIpc-850 jumper settings are set correctly.

4. Make sure that the VXIpc-850 is seated properly in the mainframe.

5. Power on the mainframe and observe whether the **SYSFAIL** and **FAILED** LEDs become unlit some time before the operating system boots.

Refer also to Appendix C, *LED Indicators*, in the *VXIpc-850 Series User Manual*.

Can I access 32-bit registers in my VXIbus system from the VXIpc-850?

Yes. The VXIpc-850 uses the 32-bit PCI bus to interface to the VXIbus. In fact, the VXIbus circuitry on the VXIpc-850 also supports the new VME64 standard for D64 accesses.

What kind of signal is CLK10 and what kind of signal do I need for an external CLK10?

CLK10 is a differential ECL signal on the VXIbus backplane. However, the oscillator for the VXIpc-850 and the EXTCLK input from the front panel use TTL. Therefore, you need to supply a TTL level signal for EXTCLK. Our voltage converters convert the signal to differential ECL.

What is the accuracy of the CLK10 signal?

The CLK10 generated by the VXIpc-850 is ±100 ppm (0.01%) as per the VXIbus specification. If you need a more accurate CLK10 signal, you can use the EXTCLK input at the front of the VXIpc-850.

If I boot the computer without video, and then plug in the video, why is it in black and white?

When the computer first boots, the video chips try to synchronize with the monitor. If the monitor is not there, the video chips cannot synchronize and establish color. You need to have the monitor attached at boot time to get color.

I've installed the SCSI software. Why doesn't the VXIpc-850 recognize my SCSI device?

This problem usually occurs when SCSI has not been enabled in the BIOS. Enter your BIOS by pressing <DEL> at boot time. In the **Setup** window, click on the **Chipset** icon. Select the **On Board PCI SCSI** option and enable it.

What type of video interface is present onboard the VXIpc-850? What video drivers are included with the VXIpc-850? Can I use Super VGA with my VXIpc-850? If my application requires a special type of video display, how do I configure my VXIpc-850?

The VXIpc-850 uses the Trident Microsystems TGUI9660 chip, a combination graphics accelerator and RAMDAC. It is compatible with the Standard VGA video output MS Windows video driver, as well as

the Trident Microsystems video driver. For information on the Trident driver, refer to *VXIpc-850 Peripherals User Manual*.

If your application requires a special type of video display, you can remove the PCI video card included with your VXIpc-850. Contact National Instruments for information on how to do this.

What kind of monitor can I use with the VXIpc-850?

VXIpc-850 computers that use Super VGA video output will work only with monitors having a horizontal scan rate of at least 50 kHz and a vertical scan rate of 60 Hz.

Warning: *Make sure that your monitor meets this specification. Enabling the Super VGA option on a monitor that does not meet this specification will damage your monitor.*

What can I do if my keyboard connector does not fit into the keyboard port on the VXIpc-850?

You can plug keyboards that have a 6-pin Mini DIN PS/2 type connector directly into the VXIpc-850. You can use the keyboard adapter cable that is included with every VXIpc-850 kit to adapt the larger AT keyboard connector to the 6-pin Mini DIN connector.

How do I connect an external speaker to the VXIpc-850 to get audio capability?

A twisted-pair cable connects the front panel audio connector to the VXIpc-850 motherboard. Connect the external speaker to this front panel connector. The center pin of the connector provides the audio signal. The shield of the connector is GROUND.

How do I add RAM to the VXIpc-850? What is the maximum amount of RAM that I can have on the VXIpc-850?

To add RAM to the VXIpc-850, remove the top cover and add SIMM modules to the two empty SIMM sockets. The largest SIMM size that the VXIpc-850 socket will accept is a 32 MB SIMM (8 MB x 36). The maximum amount of RAM that can be added to a standard configuration of 16 MB is 64 MB, yielding a total of 80 MB. By replacing the 16 MB SIMMs and using four 32 MB SIMMs, you can have a total of 128 MB.

Which interrupt levels are free to be used by ISA bus boards? Which area of upper memory (adapter space) is free for use by ISA bus boards or expanded memory manager software programs?

See Appendix B, *VXIpc-850 System Resources*, in the *VXIpc-850 Series User Manual* for information on the available port I/O register space, upper memory area, interrupts, and DMA channels.

How do I install the VXIpc-850 in a slot other than Slot 0?

The VXIpc-850 automatically detects whether it is in Slot 0 of a VXIbus mainframe. You do not need to change jumper settings to install the VXIpc-850 in a slot other than Slot 0 unless you have defeated the first slot detector (FSD) circuitry by changing jumper W13.

Refer to the *VXIbus Slot 0/Non-Slot 0* section in Chapter 3, *Configuration and Installation*, in the *VXIpc-850 Series User Manual* for information on enabling and defeating the FSD circuitry.

How do I check the configuration of the memory, floppy drive, hard drive, time/date, and so on?

Follow these steps to view the BIOS setup parameters:

1. Reboot the VXIpc-850.

2. During the memory tests, press the <DEL> key.

3. After hitting the <DEL> key, you will get a graphical utility with icons that can be selected to view and change system parameters.

Refer to Chapter 4, *WinBIOS*, in the *VXIpc-850 Series User Manual* for more information about viewing or changing these settings.

How do I disable the automatic memory check when booting the VXIpc-850?

You can disable checking for memory above 1 MB. The BIOS will always check memory below 1 MB. To change the BIOS, enter the setup by pressing <DEL> as the system is booting and select the **Advanced** icon in the **Setup** window. Disable the **Above 1 MB Memory Test**, save your settings, and reboot the system.

Can I upgrade my VXIpc-850?

You can upgrade the modular CPU card to upgrade the processor. Contact National Instruments for information.

Customer Communication

For your convenience, this appendix contains forms to help you gather the information necessary to help us solve technical problems you might have as well as a form you can use to comment on the product documentation. Filling out a copy of the *Technical Support Form* before contacting National Instruments helps us help you better and faster.

National Instruments provides comprehensive technical assistance around the world. In the U.S. and Canada, applications engineers are available Monday through Friday from 8:00 a.m. to 6:00 p.m. (central time). In other countries, contact the nearest branch office. You may fax questions to us at any time.

Electronic Services

Bulletin Board Support

National Instruments has BBS and FTP sites dedicated for 24-hour support with a collection of files and documents to answer most common customer questions. From these sites, you can also download the latest instrument drivers, updates, and example programs. For recorded instructions on how to use the bulletin board and FTP services and for BBS automated information, call (512) 795-6990. You can access these services at:

United States: (512) 794-5422 or (800) 327-3077
Up to 14,400 baud, 8 data bits, 1 stop bit, no parity

United Kingdom: 01635 551422
Up to 9,600 baud, 8 data bits, 1 stop bit, no parity

France: 1 48 65 15 59
Up to 9,600 baud, 8 data bits, 1 stop bit, no parity

FTP Support

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

FaxBack Support

FaxBack is a 24-hour information retrieval system containing a library of documents on a wide range of technical information. You can access FaxBack from a touch-tone telephone at the following numbers:

(512) 418-1111

E-Mail Support (currently U.S. only)

You can submit technical support questions to the appropriate applications engineering team through e-mail at the Internet addresses listed below. Remember to include your name, address, and phone number so we can contact you with solutions and suggestions.

GPIB:	gpib.support@natinst.com
DAQ:	daq.support@natinst.com
VXI:	vxi.support@natinst.com
LabVIEW:	lv.support@natinst.com
LabWindows:	lw.support@natinst.com
HiQ:	hiq.support@natinst.com
VISA:	visa.support@natinst.com

Fax and Telephone Support

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

	Telephone	Fax
Australia	03 9 879 9422	03 9 879 9179
Austria	0662 45 79 90 0	0662 45 79 90 19
Belgium	02 757 00 20	02 757 03 11
Canada (Ontario)	519 622 9310	
Canada (Quebec)	514 694 8521	514 694 4399
Denmark	45 76 26 00	45 76 26 02
Finland	90 527 2321	90 502 2930
France	1 48 14 24 24	1 48 14 24 14
Germany	089 741 31 30	089 714 60 35
Hong Kong	2645 3186	2686 8505
Italy	02 413091	02 41309215
Japan	03 5472 2970	03 5472 2977
Korea	02 596 7456	02 596 7455
Mexico	95 800 010 0793	5 520 3282
Netherlands	0348 433466	0348 430673
Norway	32 84 84 00	32 84 86 00
Singapore	2265886	2265887
Spain	91 640 0085	91 640 0533
Sweden	08 730 49 70	08 730 43 70
Switzerland	056 200 51 51	056 200 51 55
Taiwan	02 377 1200	02 737 4644
U.K.	01635 523545	01635 523154

Technical Support Form

Photocopy this form and update it each time you make changes to your software or hardware, and use the completed copy of this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

If you are using any National Instruments hardware or software products related to this problem, include the configuration forms from their user manuals. Include additional pages if necessary.

Name _____

Company _____

Address _____

_____

Fax (____)_____ Phone (____)_____

Computer brand _____ Model _____ Processor _____

Operating system (include version number) _____

Clock Speed _____MHz   RAM _____MB   Display adapter _____

Mouse _____yes _____no   Other adapters installed _____

Hard disk capacity _____MB   Brand _____

Instruments used _____

National Instruments hardware product model _____   Revision _____

Configuration _____

National Instruments software product _____   Version _____

Configuration _____

The problem is _____

_____

_____

_____

List any error messages _____

_____

_____

_____

The following steps will reproduce the problem _____

_____

_____

Hardware and Software Configuration Form

Record the settings and revisions of your hardware and software on the line to the right of each item. Complete a new copy of this form each time you revise your software or hardware configuration, and use this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

National Instruments Products

VXIpc-850 Hardware Settings

VXIpc-850 Revision Number _____

VXIpc-850 Serial Number _____

Processor Speed _____

DRAM SIMMs Installed _____

Slot Location _____

W1 Setting: SCSI Termination _____

W2 Setting: CMOS _____

W4 Setting: LPT1 DMA _____

W13 Setting: Slot 0 Detection _____

W15 Setting: CLK10 Output Termination _____

S1 Setting: Ethernet EEPROM _____

S2 Setting: MITE Self-Configuration _____

S3 Setting: CLK10 Source _____

S4 Setting: Inverted/Non-inverted CLK10 Output _____

S5 Setting: CLK10 SMB _____

S6 Setting: CLK10 Input Termination _____

S7 Setting: External Trigger Input Termination _____

S8 Setting: GPIB IRQ Level _____

S9 Setting: MITE User/Factory Configuration _____

NI-VXI Software Settings

NI-VXI Software Version Number _____

Using `VXIEDIT` or `VXITEDIT`?_____

Logical Address _____

Device Type _____

Address Space _____

VXI Shared RAM Size _____

Shared RAM Pool (Windows) _____

Byte Order for Lower Half Window _____

Byte Order for Upper Half Window _____

Mapping Scheme for Lower and Upper Half Windows of VXI Shared RAM _____

_____

Resource Manager Delay _____

Number of Handlers _____

Number of Interrupters _____

Servant Area Size _____

Protocol Register _____

Read Protocol Response _____

System IRQ Level _____

VXI Bus Timeout _____

Arbiter Type _____

Arbiter Timeout _____

Fair Request _____

Request Level _____

A24/A32 Slave Write Post _____

VXI Retry Protocol _____

VXI Slave Auto Retry _____

VXI Transfer Limit _____

User Window Base _____

User Window Size _____

User Window Below 1 MB _____

Driver Window Base _____

Driver Window Size _____

Driver Window Below 1 MB _____

Other Products

Mainframe Make and Model _____

Microprocessor _____

Clock Frequency _____

Type of Video Board Installed _____

Operating System _____

Operating System Version _____

Operating System Mode _____

Programming Language _____

Programming Language Version _____

Other Boards in System _____

Monitor (Manufacturer, Model) _____

Mouse (Manufacturer, Model) _____

Keyboard (Manufacturer, Model) _____

Other Peripherals (Manufacturer, Model) _____

_____

_____

_____

_____

Documentation Comment Form

National Instruments encourages you to comment on the documentation supplied with our products. This information helps us provide quality products to meet your needs.

Title: NI-VXI™ Software Manual for the VXIpc™-850 Series

Edition Date: April 1996

Part Number: 321125B-01

Please comment on the completeness, clarity, and organization of the manual.

_____

_____

_____

_____

_____

_____

_____

If you find errors in the manual, please record the page numbers and describe the errors.

_____

_____

_____

_____

_____

_____

_____

Thank you for your help.

Name   _____

Title   _____

Company   _____

Address   _____

_____

Phone   (_____) _____

Mail to: Technical Publications   **Fax to:**   Technical Publications
National Instruments Corporation   National Instruments Corporation
6504 Bridge Point Parkway   (512) 794-5678
Austin, TX 78730-5039

Glossary

Prefix	Meaning	Value
n-	nano-	10^{-9}
μ-	micro-	10^{-6}
m-	milli-	10^{-3}
K-	kilo-	10^{3}
M	mega-	10^{6}
G-	giga-	10^{9}

A

A16 space   VXIbus address space equivalent to the VME 64 KB short address space. In VXI, the upper 16 KB of A16 space is allocated for use by VXI devices configuration registers. This 16 KB region is referred to as VXI configuration space.

A24 space   VXIbus address space equivalent to the VME 16 MB *standard* address space.

A32 space   VXIbus address space equivalent to the VME 4 GB *extended* address space.

address   Character code that identifies a specific location (or series of locations) in memory.

address space	A set of 2^n memory locations differentiated from other such sets in VXI/VMEbus systems by six addressing lines known as address modifiers. n is the number of address lines required to uniquely specify a byte location in a given space. Valid numbers for n are 16, 24, and 32. In VME/VXI, because there are six address modifiers, there are 64 possible address spaces.
address window	A portion of address space that can be accessed from the application program.
ANSI	American National Standards Institute

B

B	bytes
backplane	An assembly, typically a printed circuit board, with 96-pin connectors and signal paths that bus the connector pins. A C-size VXIbus system will have two sets of bused connectors called J1 and J2. A D-size VXIbus system will have three sets of bused connectors called J1, J2, and J3.
BERR*	Bus error signal
BIOS	Basic Input/Output System. BIOS functions are the fundamental level of any PC or compatible computer. BIOS functions embody the basic operations needed for successful use of the computer's hardware resources.
BTO	Bus Timeout Unit; a functional module that times the duration of each data transfer and terminates the cycle if the duration is excessive. Without the termination capability of this module, a bus master attempt to access a nonexistent slave could result in an indefinitely long wait for a slave response.
byte order	How bytes are arranged within a word or how words are arranged within a longword. Motorola ordering stores the most significant (MSB) byte or word first, followed by the least significant byte (LSB) or word. Intel ordering stores the LSB or word first, followed by the MSB or word.

C

Commander   A message-based device which is also a bus master and can control one or more Servants.

configuration registers   A set of registers through which the system can identify a module device type, model, manufacturer, address space, and memory requirements. In order to support automatic system and memory configuration, the VXIbus specification requires that all VXIbus devices have a set of such registers.

D

Data Transfer Bus   DTB; one of four buses on the VMEbus backplane. The DTB is used by a bus master to transfer binary data between itself and a slave device.

DMA   Direct Memory Access; a method by which data is transferred between devices and internal memory without intervention of the central processing unit.

DRAM   Dynamic RAM (Random Access Memory); storage that the computer must refresh at frequent intervals.

driver window   A region of address space that is decoded by the VXIpc-850 for use by the NI-VXI software.

DTB   See *Data Transfer Bus*.

E

EEPROM   Electronically Erasable Programmable Read-Only Memory

embedded controller   An intelligent CPU (controller) interface plugged directly into the VXI backplane, giving it direct access to the VXIbus. It must have all of its required VXI interface capabilities built in.

F

fair requester	A VXIbus device that will not arbitrate for the VXIbus after releasing it until it detects the bus request signal inactive. This ensures that all requesting devices will be granted use of the bus.

H

hex	Hexadecimal; the numbering system with base 16, using the digits 0 to 9 and letters A to F.

I

IEEE	Institute of Electrical and Electronics Engineers
I/O	input/output; the techniques, media, and devices used to achieve communication between machines and users.
interrupt	A means for a device to request service from another device.
interrupt handler	A VMEbus functional module that detects interrupt requests generated by interrupters and responds to those requests by requesting status and identify information.
interrupt level	The relative priority at which a device can interrupt.
IRQ*	Interrupt signal

K

KB	kilobytes of memory

L

logical address	An 8-bit number that uniquely identifies each VXIbus device in a system. It defines the A16 register address of a device, and indicates Commander and Servant relationships.

M

MB	megabytes of memory
master	A functional part of a VME/VXIbus device that initiates data transfers on the backplane. A transfer can be either a read or a write.
message-based device	An intelligent device that implements the defined VXIbus registers and communication protocols. These devices are able to use Word Serial Protocol to communicate with one another through communication registers.
MITE	A National Instruments custom ASIC, a sophisticated dual-channel DMA controller that incorporates the Synchronous MXI and VME64 protocols to achieve high-performance block transfer rates.

N

NI-VXI	The National Instruments bus interface software for VME/VXIbus systems.

P

PCI	Peripheral Component Interconnect. The PCI bus is a high-performance 32-bit or 64-bit bus with multiplexed address and data lines.

R

register-based device	A Servant-only device that supports VXIbus configuration registers. Register-based devices are typically controlled by message-based devices via device-dependent register reads and writes.
RESMAN	The name of the National Instruments Resource Manager in NI-VXI bus interface software. See *Resource Manager*.

Resource Manager	A message-based Commander located at Logical Address 0, which provides configuration management services such as address map configuration, Commander and Servant mappings, and self-test and diagnostic management.
retry	An acknowledge by a destination that signifies that the cycle did not complete and should be repeated.

S

s	seconds
Servant	A device controlled by a Commander; there are message-based and register-based Servants.
Shared Memory Protocol	A communication protocol that uses a block of memory that is accessible to both a client and a server. The memory block operates as a message buffer for communications.
slave	A functional part of a VME/VXIbus device that detects data transfer cycles initiated by a VMEbus master and responds to the transfers when the address specifies one of the device's registers.

U

user window	A region of address space reserved by the VXIpc-850 for use via the NI-VXI low-level function calls. `MapVXIAddress()` uses this address space to allocate regions for use by the `VXIpeek()` and `VXIpoke()` macros.

V

VIC or VICtext	VXI Interactive Control Program, a part of the NI-VXI bus interface software package. Used to program VXI devices, and develop and debug VXI application programs.
VME	Versa Module Eurocard or IEEE 1014

VMEbus System Controller	A device configured for installation in Slot 0 of a VXIbus mainframe or Slot 1 of a VMEbus chassis. This device is unique in the VMEbus system in that it performs the VMEbus System Controller functions, including clock sourcing and arbitration for data transfers across the backplane. Installing such a device into any other slot can damage the device, the VMEbus/VXIbus backplane, or both.
VXIbus	VMEbus Extensions for Instrumentation
VXIedit or VXItedit	VXI Resource Editor program, a part of the NI-VXI bus interface software package. Used to configure the system, edit the manufacturer name and ID numbers, edit the model names of VXI and non-VXI devices in the system, as well as the system interrupt configuration information, and display the system configuration information generated by the Resource Manager.
VXIinit	A program in the NI-VXI bus interface software package that initializes the board interrupts, shared RAM, VXI register configurations, and bus configurations.

W

Word Serial Protocol	The simplest required communication protocol supported by message-based devices in a VXIbus system. It utilizes the A16 communication registers to transfer data using a simple polling handshake method.
write posting	A mechanism that signifies that a device will immediately give a successful acknowledge to a write transfer and place the transfer in a local buffer. The device can then independently complete the write cycle to the destination.

E

e-mail support, C-2
electronic support services, C-1 to C-2
example programs
    compiling sample programs, 4-5 to 4-6
    EXAMPLES subdirectory, 4-2

F

FAILED LED, B-2
Fair Request, VXI Bus Configuration
    Editor, 3-14
fax and telephone support, C-2
FaxBack support, C-2
files and programs for NI-VXI software
    header files, A-2 to A-3
    main programs and files, A-1 to A-2
FTP support, C-1
functions
    local resource access functions,
        4-4 to 4-5
    low-level access functions, 4-3
    setting user handlers, 4-4
    system configuration functions, 4-5

H

Handlers, Number of, 3-10
header files, A-2 to A-3

I

InitVXIlibrary() function, 4-5
installation and setup, 2-1 to 2-12
    DOS INSTALL program, 2-12
    installation of VXIpc-850 Series, 2-1
    installing VXIpc-850 in slot other than
        Slot 0, B-5
    Windows 3.1, 2-1 to 2-5
        AUTOEXEC.BAT file
            modifications, 2-4
        completing software
            installation, 2-5
        LabWindows/CVI Run-Time
            Engine, 2-2

NI-VXI software, 2-2 to 2-3
SYSTEM.INI file modifications,
    2-4 to 2-5
WIN.INI file modifications, 2-5
Windows 95, 2-6 to 2-9
    completing software
        installation, 2-9
    environment modification, 2-8
    LabWindows/CVI Run-Time
        System, 2-7
    NI-VXI software installation,
        2-7 to 2-8
    system preparation, 2-6
Windows NT, 2-9 to 2-12
    completing software installation,
        2-11 to 2-12
    environment modification, 2-11
    LabWindows/CVI Run-Time
        System, 2-9
    NI-VXI software installation,
        2-10 to 2-11
Interrupters, Number of, 3-11
IRQ Level, System, 3-12
ISA board questions, B-5

K

keyboard connector, B-4

L

LabVIEW software, 1-3
LabWindows/CVI software
    description, 1-3
    installation
        Windows 3.1, 2-2
        Windows 95, 2-7
        Windows NT, 2-9
LEDs for VXIpc-850 Series, B-2 to B-3
Load Configuration from File, VXIpc
    Configuration Editor, 3-4
local resource access functions, 4-4 to 4-5

W