

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *National Instruments VXIpc-486 Model 566* at our website: ***Click HERE***

Getting Started with Your NI-VXI™ Software for the VXIpc™-486 Model 200/500 Series and Windows



March 1995 Edition

Part Number 320930A-01

© Copyright 1995 National Instruments Corporation.
All Rights Reserved.

National Instruments Corporate Headquarters
6504 Bridge Point Parkway
Austin, TX 78730-5039
(512) 794-0100
Technical support fax: (800) 328-2203
(512) 794-5678

Branch Offices:
Australia (03) 879 9422, Austria (0662) 435986, 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, Italy 02/48301892, Japan (03) 3788-1921, Mexico 95 800 010 0793,
Netherlands 03480-33466, 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. 0635 523545

Limited Warranty

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

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

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

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

Trademarks

LabVIEW®, NI-488.2™, NI-VXI™, and VXIpc™-486 are trademarks of National Instruments Corporation.

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

WARNING REGARDING MEDICAL AND CLINICAL USE OF NATIONAL INSTRUMENTS PRODUCTS

National Instruments products are not designed with components and testing intended to ensure a level of reliability suitable for use in treatment and diagnosis of humans. Applications of National Instruments products involving medical or clinical treatment can create a potential for accidental injury caused by product failure, or by errors on the part of the user or application designer. Any use or application of National Instruments products for or involving medical or clinical treatment must be performed by properly trained and qualified medical personnel, and all traditional medical safeguards, equipment, and procedures that are appropriate in the particular situation to prevent serious injury or death should always continue to be used when National Instruments products are being used. National Instruments products are NOT intended to be a substitute for any form of established process, procedure, or equipment used to monitor or safeguard human health and safety in medical or clinical treatment.

Contents

About This Manual ix
How to Use This Documentation Set ix
Organization of This Manual x
Conventions Used in This Manual xi
Related Documentation xi
Customer Communication xi

Chapter 1
Introduction 1-1
How to Use This Manual 1-2
What You Need to Get Started 1-2
Optional Software 1-3
NI-VXI Software Description 1-3
NI-488.2 Software Description 1-4

Chapter 2
NI-VXI Software Installation 2-1
Installing the Software 2-1
Running INSTALL 2-1
Modifying the AUTOEXEC.BAT File 2-3
Modifying the PROGMAN.INI File 2-3
Modifying the SYSTEM.INI File 2-4
Modifying the WIN.INI File 2-4
Completing the Software Installation 2-4

Chapter 3
NI-VXI Software Configuration 3-1
Configuring the NI-VXI Software 3-1
Running VXIEDIT from DOS or the Windows DOS Shell 3-1
Logical Address Configuration 3-3
Logical Address 3-3
Device Type 3-4
Address Space 3-4
VXI Shared RAM (Byte Order) 3-4
VXI Shared RAM (MBytes) 3-5
Shared Memory Pool (Windows) 3-6
Slave Block Transfer Mode 3-6
Slave Supervisory Access Only 3-7
Resource Manager Delay 3-7
Bus Configuration 3-8
Bus Request Level 3-8
Fair Requester Timeout 3-9
Bus Release Mechanism 3-9
Bus Arbitration 3-9

Local Bus Timeout 3-10
VXIbus Timeout 3-10
Bus Acquisition Time in Timeout 3-11
Device Configuration 3-11
SYSRESET Resets PC 3-12
Use System Sanity Timer 3-12
Servant Area Size 3-13
Protocol Register 3-13
Read Protocol Response 3-13
Number of Handlers 3-13
Number of Interrupters 3-14
Exiting VXIEDIT and Reinitializing the Hardware 3-14
Running VXITEDIT from Windows 3-14
Exiting VXITEDIT and Reinitializing the Hardware 3-14

Chapter 4
Using NI-VXI with Windows 4-1
Interactive Control of NI-VXI 4-1
Example Programs 4-2
Programming Considerations 4-2
Memory Model 4-2
Multiple Applications Using the NI-VXI Library 4-2
Low-Level Access Functions 4-3
Local Resource Access Functions 4-3
System Configuration Functions 4-3
Compiling Your C Program 4-4
Symbols 4-4

Appendix A
NI-VXI Software Overview A-1

Appendix B
Common Questions B-1

Appendix C
Troubleshooting C-1

Appendix D
Customer Communication D-1

Glossary G-1

Index I-1

Figures

Figure 3-1. VXIEDIT Main Menu (from DOS) 3-2
Figure 3-2. VXIpc-486 Logical Address Configuration Editor (from DOS) 3-3
Figure 3-3. VXIpc-486 Bus Configuration Editor (from DOS).....3-8
Figure 3-4. VXIpc-486 Device Configuration Editor (from DOS) 3-11

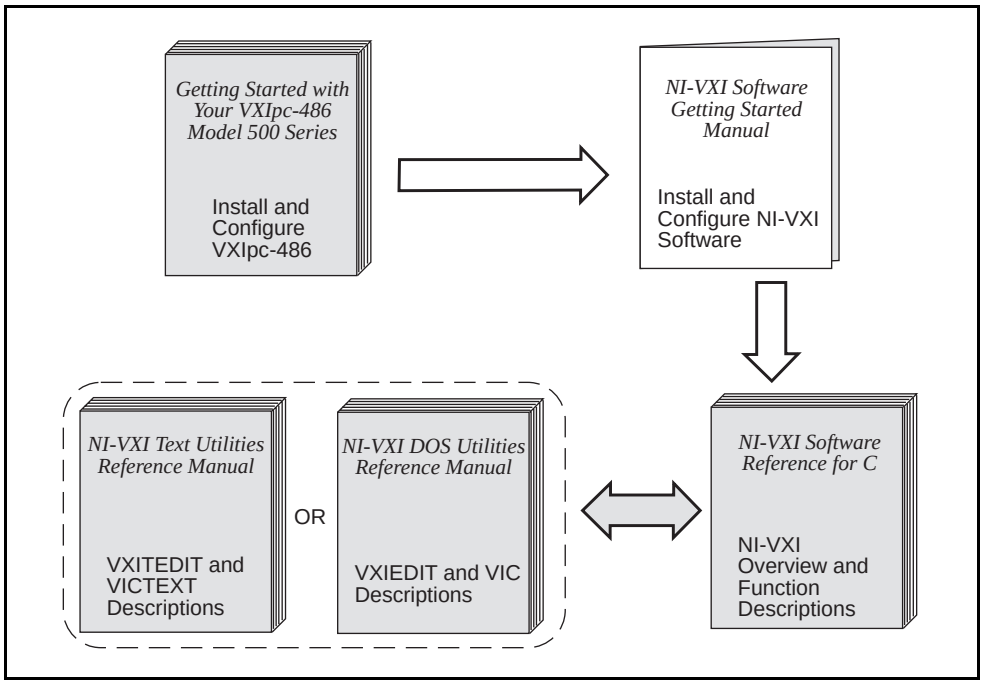
Table

Table 3-1. VXI Shared RAM Options 3-5

About This Manual

This manual contains instructions for installing and configuring the NI-VXI bus interface software for the VXIpc-486 and Windows. The NI-VXI software works with all models of the Model 200 Series and the Model 500 Series of the National Instruments VXIpc-486 embedded computers.

How to Use This Documentation Set



Begin by reading *Getting Started with Your VXIpc-486 Model 500 Series* to guide you through the installation and configuration of the hardware. If you are using a Model 200 Series VXIpc-486, you should use *Getting Started with Your Model 200 Series VXIpc-486 and the NI-VXI Software for Windows* as your hardware reference. The software configuration requires that you first complete the installation and configuration of the hardware.

Use this manual when you are ready to install and configure the NI-VXI software for Windows on the VXIpc-486. If you are using a Model 200 Series VXIpc-486, you can use either this manual or your Model 200 Series manual as a software reference.

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. Use the interactive utilities

described in the *NI-VXI DOS Utilities Reference Manual* and the *NI-VXI Text Utilities Reference Manual* to learn more about the capabilities of the NI-VXI software.

Use the *NI-488.2 User Manual for Windows* as a guide for installing, configuring, and using the NI-488.2 GPIB Talker/Listener/Controller software. Because the GPIB port on the VXIpc-486 is compatible with the National Instruments AT-GPIB plug-in board for ISA computers, we have included the manual *Getting Started with Your AT-GPIB/TNT and the NI-488.2 Software for Windows* with your VXIpc-486 kit. Refer to the *VXIpc-486 Documentation Notice: AT-GPIB and Windows Compatibility* for more information about using this documentation.

Organization of This Manual

This manual is organized as follows:

- Chapter 1, *Introduction*, is an overview of the NI-VXI software for the VXIpc-486 and Windows.
- Chapter 2, *NI-VXI Software Installation*, describes how to install the NI-VXI software on the VXIpc-486 embedded computer.
- Chapter 3, *NI-VXI Software Configuration*, contains instructions for configuring the NI-VXI software for the VXIpc-486 computer.
- Chapter 4, *Using NI-VXI with Windows*, discusses programming information for you to consider when developing applications that use the NI-VXI driver.
- Appendix A, *NI-VXI Software Overview*, lists all the programs and files located on the NI-VXI distribution diskette.
- Appendix B, *Common Questions*, answers common questions you may have when using the NI-VXI bus interface software on the VXIpc-486 platform.
- Appendix C, *Troubleshooting*, addresses system-related problems you may encounter when using the NI-VXI bus interface software on the VXIpc-486 platform.
- Appendix D, *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 acronyms, abbreviations, metric prefixes, mnemonics, and symbols.
- The *Index* contains an alphabetical list of key terms and topics 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, and dialog box options.
italic	Italic text denotes emphasis, a cross reference, or an introduction to a key concept.
`monospace`	Text in monospace font is used for the proper names of programs, subprograms, filenames, and extensions.
`bold monospace`	Bold text in this font denotes the messages and responses that the computer automatically prints to the screen.
bold italic	Bold italic text denotes a note, caution, or warning.
<>	Angle brackets enclose the name of a key on the keyboard–for example, <Esc>.
<Tab>	Key names are capitalized.

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

Related Documentation

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

- *IEEE Standard for a Versatile Backplane Bus: VMEbus*, ANSI/IEEE Standard 1014-1987
- VXI-1, *VXIbus System Specification*, Rev. 1.4, VXIbus Consortium, National Instruments part number 350083-01
- VXI-6, *VXIbus Mainframe Extender Specification*, Rev. 1.0, VXIbus Consortium, National Instruments part number 340258-01

Customer Communication

National Instruments wants to receive your comments on our products and manuals. We are interested in the applications you develop using 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 D, *Customer Communication*, at the end of this manual.

Chapter 1
Introduction

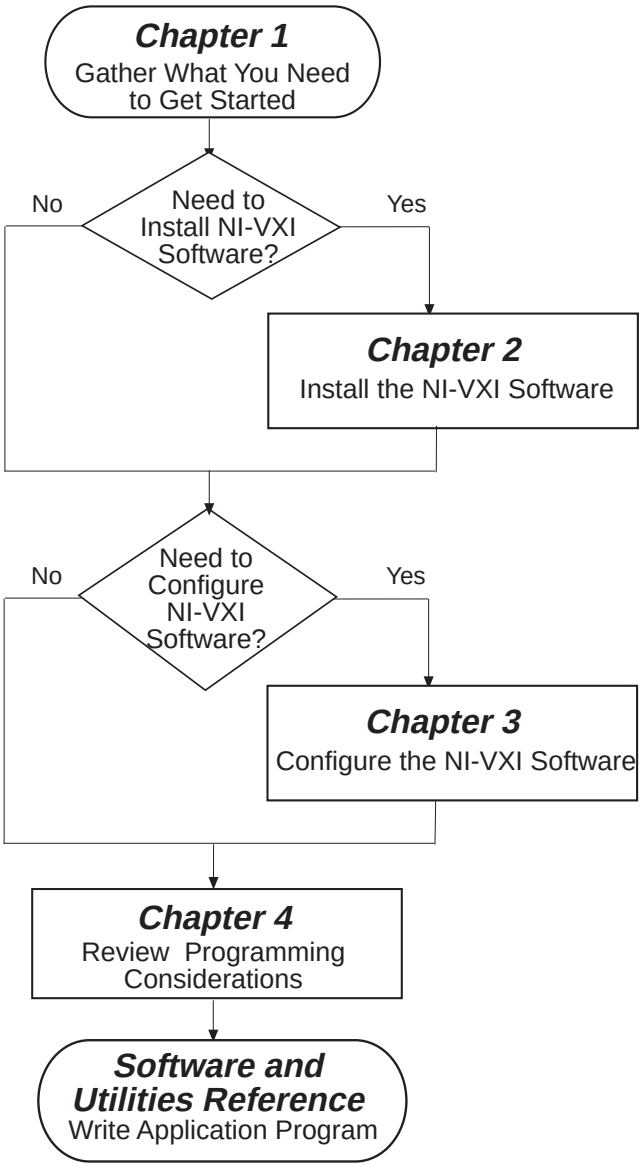
This chapter describes the NI-VXI and NI-488.2 software kits for the VXIpc-486 and Windows, lists what you need to get started, and lists optional software you can order from National Instruments. You should already have installed and configured the VXIpc-486 embedded computer according to the instructions in *Getting Started with Your VXIpc-486 Model 500 Series*.

Note: ***If you are using a Model 200 Series VXIpc-486, you should use*** *Getting Started with Your Model 200 Series VXIpc-486 and the NI-VXI Software for Windows* ***as your hardware reference whenever this manual refers to*** *Getting Started with Your VXIpc-486 Model 500 Series*.

The VXIpc-486, an embedded computer based on the Industry Standard Architecture (ISA), is a high-performance, easy-to-use platform for VXIbus systems, featuring complete VXI functionality through interactive utilities and C function calls. In addition, the VXIpc-486 has an IEEE 488 interface that is compatible with the NI-488.2 architecture.

The NI-VXI software for Windows works with all models of the Model 200 Series and Model 500 Series of the VXIpc-486. The major difference in the models is the type and speed of the microprocessor. Because the software works on all Model 200 and Model 500 Series VXIpc-486 machines, you may use this manual as a software reference for any of these models. However, *Getting Started with Your Model 200 Series VXIpc-486 and the NI-VXI Software for Windows* contains both hardware and software information.

How to Use This Manual



What You Need to Get Started

- ❑ Windows version 3.1 or higher, and MS-DOS version 5.0 or higher installed on your computer
- ❑ VXIpc-486 Model 200 Series or Model 500 Series Module
- ❑ Keyboard adapter cable
- ❑ NI-VXI distribution disks for the VXIpc-486 200/500 Series and Microsoft Windows

Optional Software

In addition to the NI-VXI bus software for Windows, you can order the LabVIEW or LabWindows®/CVI software from National Instruments for your VXIpc-486. These programs match the modular virtual instrument capability of VXI and can reduce your VXIbus software development time. 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.

Both LabVIEW and LabWindows/CVI include all the tools needed for instrument control, data acquisition, analysis, and presentation. When you order LabVIEW or LabWindows/CVI, you also get more than 300 complete instrument drivers, which are modular, source-code programs that handle the communication with your instrument so that you do not have to learn the programming details.

The LabVIEW for Windows VXI Development System contains the following components:

- LabVIEW for Windows Full Development System
- LabVIEW for Windows VXI Library
- LabVIEW for Windows/Sun VXI Instrument Library

The LabWindows/CVI for Windows, VXI Development System contains the following components:

- LabWindows/CVI for Windows Full Development System
- LabWindows/CVI for Windows VXI Library
- LabWindows/CVI for Windows Instrument Library

You can also order the Super VGA Video Option for your VXIpc-486 Model 500 Series. You can benefit from the improved graphic image from Super VGA if you want to edit or view an especially complex image in graphics or CAD programs. If you ordered your VXIpc-486 with Super VGA already installed, you must modify the system settings as described in Appendix H, *Configuring the Super VGA Option*, of *Getting Started with Your VXIpc-486 Model 500 Series*. That appendix also includes installation instructions in case you order this option at a later date.

NI-VXI Software Description

The VXIpc-486 Series computers are compatible with numerous software packages and tools available for general-market computers. To program VXI directly, you can use the NI-VXI bus interface software included with this kit. The NI-VXI software for the VXIpc-486 and Windows includes a Resource Manager, an interactive VXI resource editor program, a comprehensive library of software routines for VXI/VME programming, and an interactive control program for interacting with the VXIbus. You can use this software to seamlessly program multiple-mainframe configurations and have software compatibility across a variety of VXI/VME

controller platforms. You can use the same software with any model in the Model 200 Series or Model 500 Series of the VXIpc-486.

NI-488.2 Software Description

The NI-488.2 software kit for the VXIpc-486 and MS-DOS/Windows gives you accessibility to the industry-standard NI-488.2 software for controlling external GPIB instruments through the GPIB port on the front panel of your VXIpc-486. The *VXIpc-486 Documentation Notice: AT-GPIB and Windows Compatibility* directs you to the appropriate sections of the NI-488.2 documentation for information about using the capabilities of the AT-GPIB in your system.

Chapter 2
NI-VXI Software Installation

This chapter contains instructions for installing the NI-VXI software for the VXIpc-486 and Windows. Your VXIpc-486 is shipped from the factory with MS-DOS, Windows, NI-VXI, and NI-488.2 software already installed on the hard disk. The information on installing the NI-VXI software is included in this chapter should you ever need to reinstall it. For instructions on installing MS-DOS and Windows, refer to the documentation included in your Microsoft DOS and Windows kit. Because the VXIpc-486 built-in GPIB port is compatible with the industry-standard AT-GPIB plug-in GPIB interface board for PC AT computers, please refer to *Getting Started with Your AT-GPIB/TNT and the NI-488.2 Software for Windows* for information about using the capabilities of the AT-GPIB and the NI-488.2 software in your system.

For a list of all the files and programs, refer to Appendix A, *NI-VXI Software Overview*. If you encounter problems using the NI-VXI software, consult Appendix B, *Common Questions*, and Appendix C, *Troubleshooting*.

Installing the Software

The VXIpc-486 is shipped with Windows and the NI-VXI software installed on the hard disk. Unless you need to reinstall part or all of the NI-VXI files, skip this chapter and continue with Chapter 3, *NI-VXI Software Configuration*.

The NI-VXI distribution disks that came with your VXIpc-486 contain the NI-VXI files as well as an `INSTALL` program. The `INSTALL` program is used to install a software update or to reinstall software in the event that your files were accidentally erased. Follow these steps to reinstall all or part of the NI-VXI software.

Note: ***The NI-VXI software requires approximately 3 MB of free space on your hard disk. Create the necessary free space on the hard disk before starting the installation.***

You can quit the `INSTALL` program at any time by choosing **Exit** from the initial screen, or by pressing the **Cancel** button once installation has begun. If you do not have a mouse, pressing the <*Esc*> key performs the same action. In either case, `INSTALL` will prompt you to be certain you really want to quit at that time. If you choose to run the `INSTALL` program again, it starts over and recopies all the necessary files. If you made any changes from the default configuration, you will need to specify those changes once again.

Running INSTALL

Run `INSTALL.EXE` on distribution disk 1. You can run this program from the Windows Program Manager's **File** menu, **Run** option, or from the DOS prompt (which loads Windows and then runs `INSTALL`). However, it *cannot* be run from the Windows DOS Shell. `INSTALL` is a

graphical, interactive, self-guiding program, which installs all the necessary files (for a complete or partial installation, as user-specified) on the hard disk.

The program prompts you to enter the following information.

- Destination directory: This is the complete pathname of the directory where you want to install the software.

- Environments to install: You can choose whether to install NI-VXI for DOS, Windows, or both. The default is to install both.

- Languages to install: You can choose whether to install the Microsoft C, Borland C, and/or QuickBASIC libraries. The default is to install all of them, but if you are using only one compiler, you can select only that compiler.

- Modifying DOS initialization files: You can choose whether `INSTALL` modifies the `AUTOEXEC.BAT` file to include the location of the NI-VXI library. If you choose to modify `AUTOEXEC.BAT`, you must also enter the drive where it can be found. For more information about this option, refer to the following section, *Modifying the AUTOEXEC.BAT File*.

- Modifying Windows initialization files: You can choose whether `INSTALL` modifies the `PROGMAN.INI`, `SYSTEM.INI`, and `WIN.INI` files to include the location of the NI-VXI library. If you choose to modify these files, you must also enter the directory in which the files can be found. For more information about this option, refer to the following sections, *Modifying the PROGMAN.INI File*, *Modifying the SYSTEM.INI File*, and *Modifying the WIN.INI File*.

If you choose to let `INSTALL` modify any or all of the initialization files, it changes the extension of the old file to `.BAK` and saves it for safety purposes. If you choose not to let `INSTALL` modify the initialization files, it creates new files with the extension `.VXI`, which contain the changes it proposes you make to the files. You can use the `.VXI` files as a guide to make the changes yourself.

Note: ***The*** `NIVXIPATH` ***variables in the*** `SYSTEM.INI`, `WIN.INI`, ***and*** `AUTOEXEC.BAT` ***files should point to the same subdirectory.***

If `INSTALL` detects a previous installation of the NI-VXI software, it prompts you to determine whether to delete or overwrite the previous version, or to provide an alternate path in which to install the new version.

Caution: ***If you have a previous version of the NI-VXI software installed,*** `INSTALL` ***does not automatically create a backup of the software files. If there are files you want to preserve, be sure to exit the installation program at this time and make a backup before continuing.***

Modifying the AUTOEXEC.BAT File

If you choose to let INSTALL 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 INSTALL modifies the file, it saves the old file as AUTOEXEC.BAK and keeps it for safety purposes. The previously specified directories in PATH, LIB, and INCLUDE remain unchanged. INSTALL also adds a new environment variable NIVXIPATH, and appends a command line to execute VXIINIT.EXE automatically.

If you choose not to let INSTALL 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.

Modifying the PROGMAN.INI File

If you choose to let INSTALL modify your PROGMAN.INI file, it specifies the NIVXI.GRP folder file as shown below.

```
[Settings]
Order= x <etc>

[Groups]
Groupx= <NI-VXI directory>\WIN\NIVXI.GRP
```

where <*etc*> represents the numbers already on this line, and where *x* is a new number between 1 and 99 and is the same on both of the preceding lines.

When INSTALL modifies the file, it saves the old file as PROGMAN.BAK for safety purposes. If you do not let INSTALL modify your PROGMAN.INI file, it creates a file called PROGMAN.VXI in the NI-VXI directory, which you can use as a guide to make the changes yourself.

Modifying the SYSTEM.INI File

If you choose to let INSTALL modify your SYSTEM.INI file, it adds a line to the [386Enh] section to load a device driver that NI-VXI needs for shared memory accesses. The file, which must be loaded at Windows startup, is called NIVXIPHM.386 and is normally located in the WIN subdirectory of the NI-VXI directory as shown below. INSTALL also adds a line that keeps Windows from using the VXI space for its own purposes. In addition, a new section [NIVXI] is created, which NIVXIPHM.386 uses during Windows system startup.

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

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

When INSTALL modifies the file, it saves the old file as SYSTEM.BAK for safety purposes. If you choose not to let INSTALL modify your SYSTEM.INI file, it creates a file called SYSTEM.VXI in the NI-VXI directory, which you can use as a guide to make the changes yourself.

Modifying the WIN.INI File

If you choose to let INSTALL modify your WIN.INI file, it 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 INSTALL modifies the file, it saves the old file as WIN.BAK for safety purposes. If you choose not to let INSTALL modify your WIN.INI file, it creates a file called WIN.VXI in the NI-VXI directory, which you can use as a guide to make the changes yourself.

Completing the Software Installation

After you execute INSTALL, you should exit Windows and reboot your machine (or at least run AUTOEXEC.BAT) to make your system aware of the NI-VXI directory.

After the NI-VXI software is installed, run VXIINIT.EXE and then RESMAN.EXE. At this point you are ready to use the NI-VXI software. The default settings of the NI-VXI software are compatible with most applications. Refer to Chapter 3, *NI-VXI Software Configuration*, if you want to modify the default settings to suit the special needs of your application. If you do not need to make any changes, review the material in Chapter 4, *Using NI-VXI with Windows*.

Chapter 3
NI-VXI Software Configuration

This chapter describes how to use the VXI resource editor program to configure the NI-VXI software. For information on configuring the NI-488.2 driver software, refer to the NI-488.2 documentation.

For a list of all the files and programs, refer to Appendix A, *NI-VXI Software Overview*. If you encounter problems using the NI-VXI software, consult Appendix B, *Common Questions*, and Appendix C, *Troubleshooting*.

Configuring the NI-VXI Software

Run VXIINIT from DOS or the Windows DOS shell to initialize and display the VXIpc-486 configuration settings. The default configuration is Slot 0 System Controller, Resource Manager, Message-Based device. To change the current settings, or to view the configuration options, run one of the VXI Resource Editor programs, either VXIEDIT or VXITEDIT, and select the **Configuration Editor** from the **Main** menu.

You can use VXIEDIT or VXITEDIT to edit the description of the VXIpc-486 or any other device that is installed in the system. You can easily modify the configuration tables required for VXIbus and MXIbus operation. You can also configure interrupts, triggers, and the utility bus for single- or multiple-mainframe system configurations. The VXIEDIT program, which you can access from DOS or the Windows DOS shell, uses a series of interactive graphical screens to display information for the various editors. You can use the text-based VXITEDIT program within Windows to accomplish the same tasks by entering commands at the keyboard.

For more details on how to use VXIEDIT, refer to the *NI-VXI DOS Utilities Reference Manual.* For more details on how to use VXITEDIT, refer to the *NI-VXI Text Utilities Reference Manual.*

Running VXIEDIT from DOS or the Windows DOS Shell

VXIEDIT.EXE is the VXI resource editor program that runs in DOS or the Windows DOS shell. You can use this program 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.

To run VXIEDIT from any directory, make sure that both the PATH and NIVXIPATH environment variables have the destination directory of the NI-VXI software added to it. PATH is used to find the application itself, while 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.

Figure 3-1 shows the functions available under VXIEDIT.

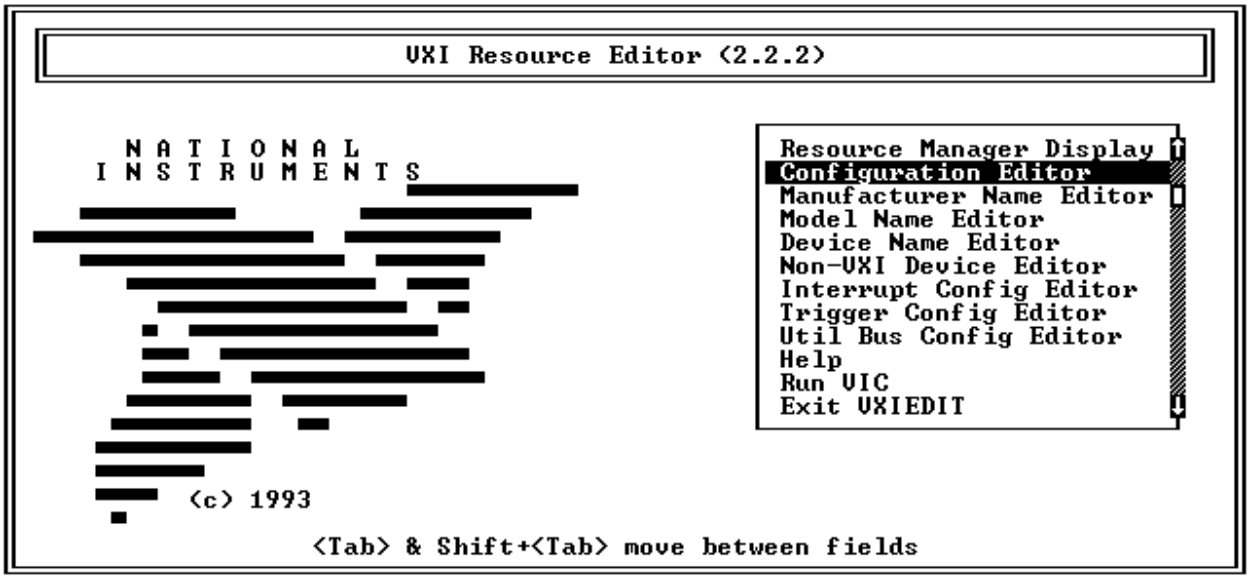


Figure 3-1. VXIEDIT Main Menu (from DOS)

When you select the **Configuration Editor** from the **Main** menu, the Logical Address Configuration Editor menu (shown in Figure 3-2) is displayed. The following sections describe the various configuration options available under the Logical Address Configuration, Bus Configuration, and Device Configuration Editor menus.

Select **Next** and **Previous** to move between the three configuration editors. You can change the default settings to any value within the range shown to the right of each item by using the left and right arrow keys.

You can save your changes by selecting **Save**. Saving the changes updates files VXILA.CFG, VXIBUS.CFG, and VXIDEV.CFG. VXIINIT.EXE uses these files to update the soft-configured settings, including board interrupts, shared RAM size, VXI register contents, and bus interface. You can exit the program at any time without saving any changes you have made by selecting **Abort**.

Logical Address Configuration

The following paragraphs describe options available under the Logical Address Configuration Editor. Figure 3-2 shows the Logical Address Configuration Editor (**VXIla**) menu.

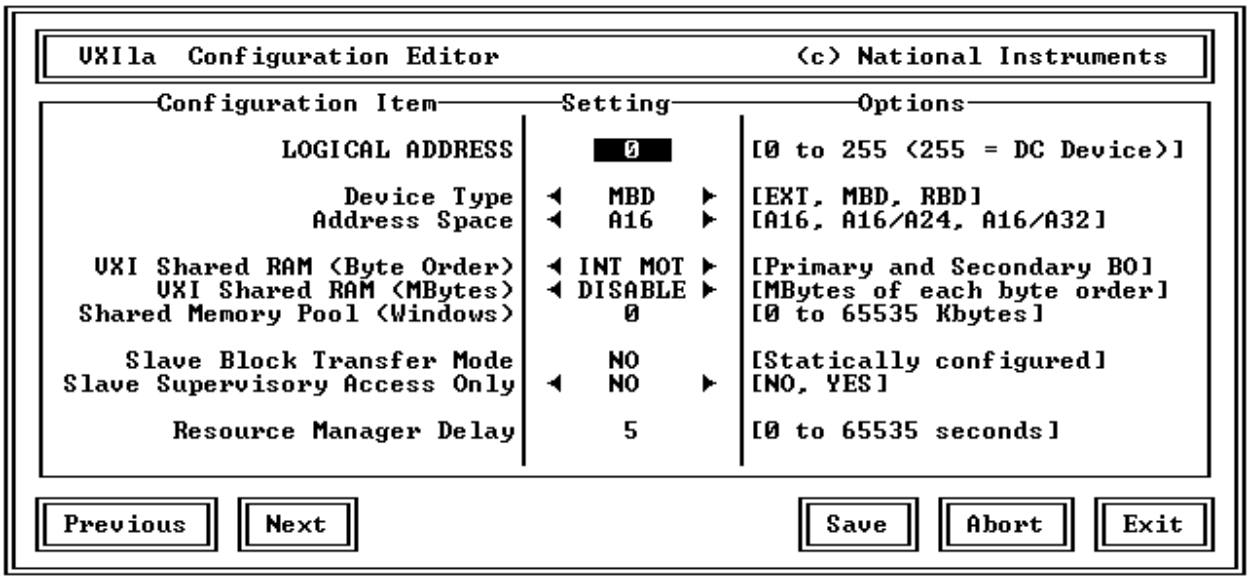


Figure 3-2. VXIpc-486 Logical Address Configuration Editor (from DOS)

Logical Address

The logical address is an 8-bit number that uniquely identifies the VXIpc-486. You can configure it as a static device with a logical address range from 0 through 254. Logical address 255 allows the VXIpc-486 to be a dynamically configured (DC) device, as specified in the VXIbus specification. The default logical address is 0, which is required if you want your VXIpc-486 to be the Resource Manager in your VXI system. If you do not want the VXIpc-486 to be the system Resource Manager, change this value to an unused logical address other than 0.

Device Type

This field indicates the classification of the VXIbus device. The following table defines the possible selections. The default value is shown in ***bold italics***.

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

The VXIpc-486 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-486 can be configured in one of three ways. The following table defines the possible selections. The default value is shown in ***bold italics***.

Addressing Mode	Setting
A16/A24 Device	A16/A24
A16/A32 Device	A16/A32
A16 only Device	***A16***

VXI Shared RAM (Byte Order)

The local RAM of the VXIpc-486 Model 500 Series computer can be slave-accessed in either A24 or A32 space in D08, D16, and D32 data transfer modes (in D08 and D16 data transfer modes only in the case of the Model 200 Series). This field indicates the primary and secondary byte swapping schemes for slave accesses to the local RAM space. The following table defines the possible selections. The default value is shown in ***bold italics***.

Byte-Swapping Scheme	Setting
Intel (Little Endian)	***INT, MOT***
Motorola (Big Endian)	MOT, INT

Note: ***When the* Address Space *field is in the default setting of A16 only, this field is ignored.***

VXI Shared RAM (MBytes)

This field indicates the number of megabytes of local RAM that are shared in either A24 or A32 space. The VXIpc-486 can be set up to have either one or two windows into the local RAM. When set up with one window, all slave accesses through this window use the primary byte order scheme. When there are two windows, accesses through the first (lower) window see the local RAM using the primary byte order scheme, and accesses through the second (upper) window see the same local RAM but use the secondary byte order scheme. In Table 3-1, *primary* and *secondary* byte order refer to the setting of the previous field. When the addressing mode is A16 only, this field should be set to `DISABLED`.

Table 3-1. VXI Shared RAM Options

Address Space	Options	Description
A16	DISABLED	VXI Shared RAM disabled
A16/A24	4, 0	Four megabytes of local RAM are shared in A24 space. Slave accesses to this RAM are in the primary byte order.
	4, 4	In this option, 4 MB of local RAM are shared, but 8 MB are requested in A24 space. Accesses to the lower 4 MB are in the primary byte order, and accesses to the upper 4 MB see the same local RAM in the secondary byte order.
	8, 0	Eight megabytes of local RAM are shared in A24 space, and slave accesses to this RAM are in the primary byte order.

(continues)

Table 3-1. VXI Shared RAM Options (Continued)

Address Space	Options	Description
A16/A32	4, 0	Four megabytes of local RAM are shared in A32 space. Slave accesses to this RAM are in the primary byte order.
	4, 4	In this option, 4 MB of local RAM are shared, but 8 MB are requested in A32 space. Accesses to the lower 4 MB are in the primary byte order, and accesses to the upper 4 MB see the same local RAM in the secondary byte order.
	8, 0	Eight megabytes of local RAM are shared in A32 space, and slave accesses to this RAM are in the primary byte order.
	8, 8	In this option, 8 MB of local RAM are shared, but 16 MB are requested in A32 space. Accesses to the lower 8 MB are in the primary byte order, and accesses to the upper 8 MB see the same local RAM in the secondary byte order.
	ALL, 0	Thirty-two megabytes of local RAM are shared in A32 space (16 MB are shared in the case of the Model 200 Series), and slave accesses to this RAM are in the primary byte order.

Shared Memory Pool (Windows)

This field indicates the size of memory in kilobytes that is allocated on Windows startup to be used exclusively by `VXImemAlloc` function calls from both DOS and Windows applications.

Memory Range	Default Value
0 to 65535 KB	0 KB

Slave Block Transfer Mode

The VXIpc-486 does not currently support block transfers in either slave-mode or master-mode accesses. This field is statically configured to reflect this with a value of `NO`, and cannot be changed.

Slave Supervisory Access Only

This field determines the type of accesses the VXIpc-486 acknowledges. The following table defines the possible selections. The default value is shown in ***bold italics***.

Accesses Acknowledged by the VXIpc-486	Value to Enter
The VXIpc-486 acknowledges both supervisory and nonprivileged accesses.	***NO***
The VXIpc-486 acknowledges only supervisory accesses. Nonprivileged accesses will result in a bus error. This is true for all A16, A24, and A32 slave accesses.	YES

Resource Manager Delay

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 seconds	5 seconds

Bus Configuration

The following paragraphs describe options available under the Bus Configuration Editor. Figure 3-3 shows the VXIbus Configuration Editor (**VXIbus**) menu.

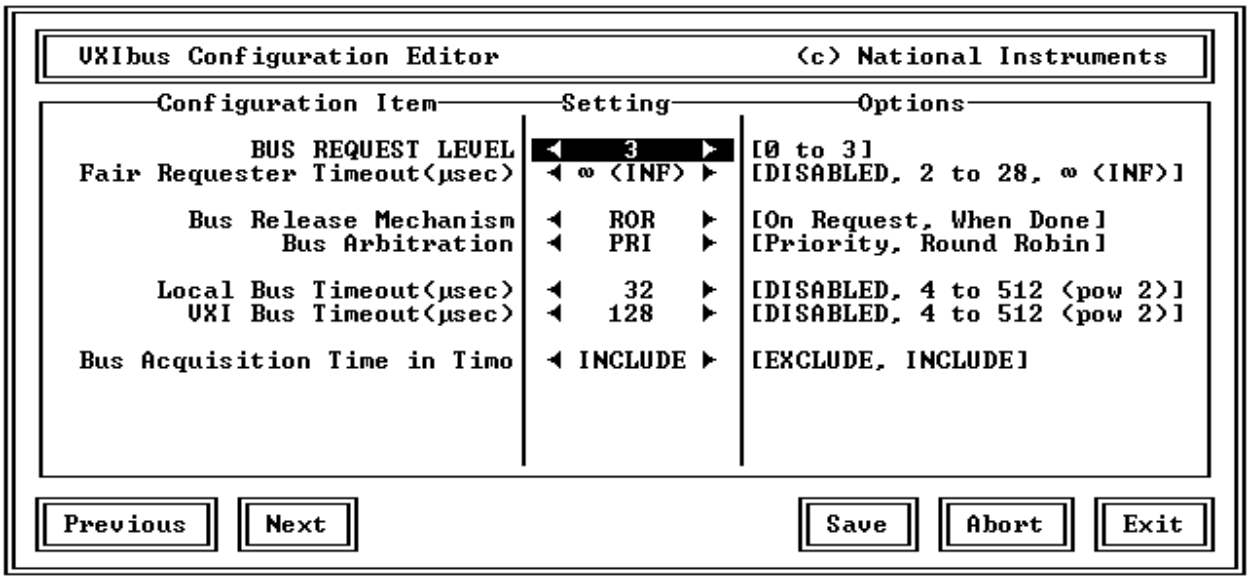


Figure 3-3. VXIpc-486 Bus Configuration Editor (from DOS)

Bus Request Level

This field sets the VXIpc-486's VXIbus request level.

VXIbus Request Levels	Default Value
0 to 3	Level 3

Fair Requester Timeout

The VXIpc-486 can be configured as a fair requester (Request On No request, or RONR), or as an unfair requester.

The following table shows the range of options available for a *fair* requester.

Fair Requester Options	Default Value
2 to 28 µs	
Infinity	INF

If you choose to select a timeout period other than infinity, your options are from 2 to 28 µs in increments of 2.

Note: ***For an unfair requester, select the field value of*** DISABLED**.**

Bus Release Mechanism

This field sets the VXIbus release mechanism. The following table defines the possible selections. The default value is in ***bold italics***.

Release Mechanism	Value to Enter
Release On Request	***ROR***
Release When Done	RWD

Bus Arbitration

The field sets the VXIbus arbitration mode. The bus arbitration mode is relevant only when the VXIpc-486 is the Slot 0 device. The following table defines the possible selections. The default value is in ***bold italics***.

Arbitration Mode	Value to Enter
Priority	***PRI***
Round Robin Select	RRS

Local Bus Timeout

This field sets the timeout period for the VXIpc-486 local bus. The following table shows the range of available options.

Timeout Period	Default Value
4 to 512 µs Disabled	32 µs

To select a new timeout value within the range of 4 to 512 µs you must choose a value with a power of 2 (4, 8, 16, 32, 64, 128, 256, and 512).

Note: ***When the VXIpc-486 is*** **not** ***the System Controller, you should disable this function using the field value of*** `DISABLED`. ***In such a case, you should have configured the VXIpc-486 as a Non-Slot 0 device as described in Chapter 2,*** *VXIpc-486 Configuration and Installation****, of*** *Getting Started with Your VXIpc-486 Model 500 Series****.***

VXIbus Timeout

This field sets the timeout period for the VXIbus bus timeout (BTO) unit. The following table shows the range of options available when the VXIpc-486 is the System Controller.

BTO Period	Default Value
4 to 512 µs Disabled	128 µs

To select a new timeout value within the range of 4 to 512 µs you must choose a value with a power of 2 (4, 8, 16, 32, 64, 128, 256, and 512).

Note: ***When the VXIpc-486 is*** **not** ***the System Controller, the BTO unit is disabled. In such a case, you should have configured the VXIpc-486 as a Non-Slot 0 device as described in Chapter 2,*** *VXIpc-486 Configuration and Installation****, of*** *Getting Started with Your VXIpc-486 Model 500 Series****.***

Bus Acquisition Time in Timeout

This field determines whether the arbitration time to acquire the VXIbus is included in the local bus timeout. The following table defines the possible selections. The default value is shown in ***bold italics***.

Accesses Acknowledged by the VXIpc-486	Value to Enter
The arbitration time is included in the local bus timeout. This is the default setting for embedded-only systems.	***INCLUDE***
The bus acquistion arbitration time is excluded from the overall timeout. This setting is preferable in MXI systems, where the arbitration time may be excessively long.	EXCLUDE

Device Configuration

The following paragraphs describe options available under the Device Configuration Editor. Figure 3-4 shows the Device Configuration Editor (**VXIdev**) menu.

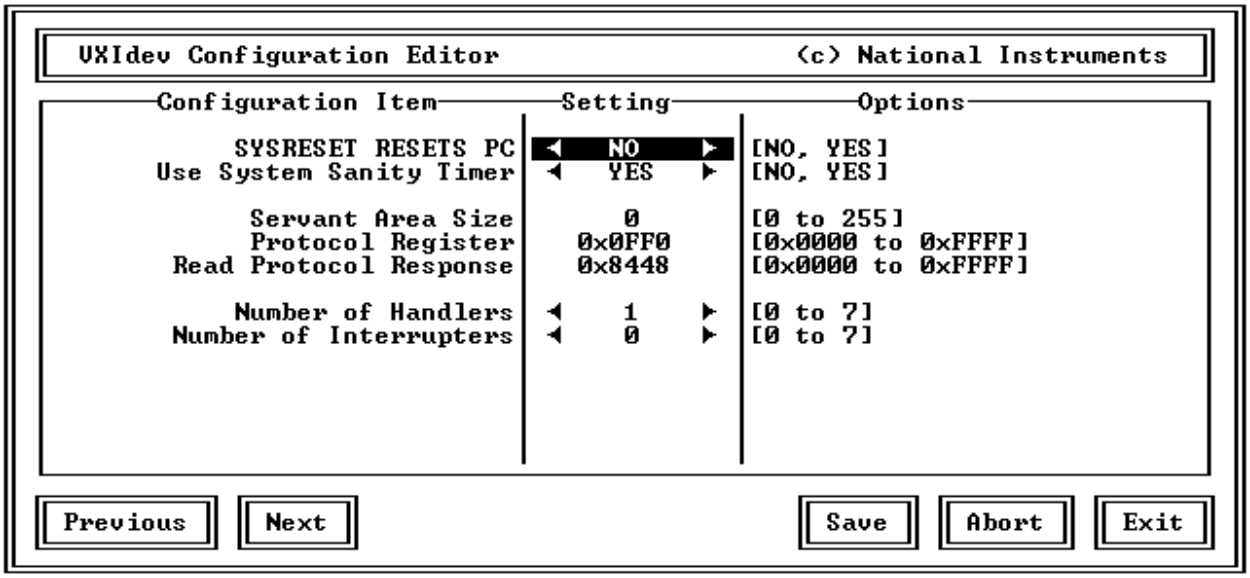


Figure 3-4. VXIpc-486 Device Configuration Editor (from DOS)

SYSRESET Resets PC

This field determines how the VXIpc-486 responds to a SYSRESET on the backplane. The following table defines the possible selections. The default value is shown in ***bold italics***.

SYSRESET Action	Value to Enter
A SYSRESET on the backplane does not reboot the VXIpc-486. However, all the hardware registers on the VXIpc-486 are reset, and the VXIpc-486 asserts SYSFAIL on the backplane. In this situation, you must run VXIINIT to reinitialize the hardware registers and to remove SYSFAIL.	***NO***
A SYSRESET on the backplane causes the VXIpc-486 to reboot.	YES

Use System Sanity Timer

If a misbehaved NI-VXI application causes the system to hang, the setting of this field determines whether the VXIpc-486 asserts the SYSFAIL* signal on the backplane. In general, the sanity timer is useful by alerting you that your application has gone awry. The following table defines the possible selections. The default value is shown in ***bold italics***.

Sanity Timer	Value to Enter
The VXIpc-486 asserts the SYSFAIL* signal on the backplane in response to a misbehaved NI-VXI application that causes the system to hang.	***YES***
The VXIpc-486 does *not* assert the SYSFAIL* signal when the system hangs. You should turn off the sanity timer by selecting this option when debugging an NI-VXI application because some debuggers conflict with the sanity timer mechanism.	NO

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-486 is *not* the Resource Manager in your system). The Servant Area size is an 8-bit value (0 through 255) that indicates the VXIpc-486 Servant area. The Servant area begins at the logical address following the VXIpc-486 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-486 is configured as a Message-Based device.

Servant Area Range	Default Value
0 to 255	0

Note: ***If the VXIpc-486 is the Resource Manager, 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-486 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-486 from the Resource Manager (if the VXIpc-486 is *not* the Resource Manager in your system). This field is meaningful only when the VXIpc-486 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).

Number of Handlers

This field gives the number of interrupt handlers that the VXIpc-486 supports.

Interrupt Handlers	Default Value
0 to 7	1

Number of Interrupters

This field gives the number of interrupters that the VXIpc-486 supports.

Interrupters	Default Value
0 to 7	0

Exiting VXIEDIT and Reinitializing the Hardware

Select **Exit** or press the <Esc> key to quit the VXIEDIT program. If you changed any of the software configuration information, VXIEDIT prompts you to save your changes before exiting the configuration menu. Saving the configuration information updates the configuration files VXILA.CFG, VXIBUS.CFG, and VXIDEV.CFG. Run VXIINIT.EXE to reinitialize the hardware according to the new settings.

If you changed any of the software configuration settings from the default, record the new settings on the *VXIpc-486 Hardware and NI-VXI Software Configuration Form* in Appendix D.

Running VXITEDIT from Windows

As an alternative to VXIEDIT you can use VXITEDIT, a text-based VXI resource editor that runs in the Windows environment. It uses the NIVXIPATH variable set in the WIN.INI file to locate the configuration files, table files, and help files. When NIVXIPATH is not specified in WIN.INI, it assumes the default directory C:\NIVXI.

To run VXITEDIT, select the **File** menu, **Run** command from the Windows Program Manager and type VXITEDIT.EXE. Refer to the *NI-VXI Text Utilities Reference Manual* for further information on how to use VXITEDIT.

Exiting VXITEDIT and Reinitializing the Hardware

To exit VXITEDIT, type exit or the number 11 when the main menu is displayed. If you changed any information, you are prompted to save your changes before exiting the menu. Saving the configuration information updates the configuration files VXILA.CFG, VXIBUS.CFG, and VXIDEV.CFG. Run VXIINIT.EXE to reinitialize the hardware according to the new settings.

If you changed any of the software configuration settings from the default, record the new settings on the *VXIpc-486 Hardware and NI-VXI Software Configuration Form* in Appendix D.

Chapter 4
Using NI-VXI with Windows

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. Remember that 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.

The *NI-VXI Software Reference Manual for C* contains an overview of NI-VXI functionality in Chapter 2, *Introduction to the NI-VXI Functions*. Chapters 3 through 13 contain overviews 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 program (`VIC` from DOS or `VICTEXT` from Windows) to write to and read from your instruments. `VIC` and `VICTEXT` help you learn how to communicate with devices, troubleshoot problems, and develop your application. Both programs display the status of your VXI transactions and inform you of any errors that occur.

The `VIC` or `VICTEXT` command set includes most of the capability of NI-VXI function calls in addition to unique auxiliary commands. You can use this utility to send data and Word Serial commands to devices and to display data received from devices on the screen. After each function executes, the program displays the device's response and the status.

The `VIC` and `VICTEXT` utilities are designed to help you learn how to use the NI-VXI functions to program devices. Once you develop a sequence of steps that works successfully for your system, you can easily incorporate the sequence into an application program using the appropriate language and syntax.

Caution: ***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.***

Refer to the *NI-VXI DOS Utilities Reference Manual* on how to use `VIC` from DOS and to learn about its features, and to the *NI-VXI Text Utilities Reference Manual* on how to use `VICTEXT` from Windows and to learn about its features.

Example Programs

The `EXAMPLES` subdirectories contain various example programs along with makefiles 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. 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 Windows applications that use the NI-VXI bus interface software.

Memory Model

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 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 can not only 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 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 trying to use the NI-VXI library. Because the NI-VXI functions execute to their completion without yielding control of the microprocessor, function calls from different applications do not inadvertently conflict with one another.

However, you do need to be careful in cases when you have multiple applications using the low-level VXIbus 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 Owner Access first before attempting to perform low-level VXIbus access with `VXIpeek()` or `VXIpoke()`. The `MapVXIAddress()` call prevents the hardware interface setting of the VXIpc-486 from being altered by other applications. 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.

Low-Level Access Functions

The function `MapVXIAddress()` returns a pointer for use with low-level access functions. On the VXIpc-486, it is possible to directly dereference this pointer. However, you must use the `VXIpeek()` and `VXIpoke()` functions when working with the AT-MXI. If you want your source code to be portable between these two platforms, it is strongly recommended that you use the NI-VXI low-level access functions to access memory. See Chapter 6, *Low-Level Access Functions*, in the *NI-VXI Software Reference Manual for C* for more information.

Local Resource Access Functions

By using `VXIEDIT` or `VXITEDIT`, you can make local resources such as RAM directly accessible from the VXIbus without intervention from the local CPU. This can be accomplished by setting the appropriate parameters to share the local resources. Regardless of the amount of RAM that you have selected for sharing, the amount of contiguous memory available to your application is limited by the **Shared Memory Pool (Windows)** field.

Therefore, when using local resource access functions, you must start by calling `VXImemAlloc()` to obtain the VXI shared RAM region. Any calls to `VXImemAlloc()` that succeed will receive a block of memory to the nearest 16-byte boundary. See Chapter 8, *Local Resource Access Functions*, in the *NI-VXI Software Reference Manual for C* for more information.

When having multiple applications or multiple instances of the same application all calling `VXImemAlloc()`, it is important to remember that the amount of memory you assign to the **Shared Memory Pool (Windows)** field is a pool for the entire system, not per application. This means that if one or more applications combined allocate the total amount of memory specified in that field, all other applications will receive a failure from `VXImemAlloc()` until the original applications free the memory with `VXImemFree()`.

System Configuration Functions

The System Configuration functions provide the lowest level initialization of your VXI interface. 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. Then you can modify the sample program to try out different things.

The sample C program for the Microsoft C compiler is in the `\nivxi\win\msc\examples` directory and the sample C program for the Borland C compiler is in the `\nivxi\win\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.

The following symbols are usually required. You must define them when using the Microsoft C or Borland C compiler.

`VXIWIN` This means that the application is a Windows application. You can use the same NI-VXI header files to compile DOS programs by defining `VXIDOS` instead.

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

`__PROTOTYPES__` This means that you are using an ANSI C-compliant compiler. If you are using a C++ compiler or if your compiler defines the `__STDC__` symbol, this symbol is automatically defined.

The following symbols are optional.

`BINARY_COMPATIBLE`	This makes the application binary compatible with embedded VXI controllers, such as the VXIpc-486 series of embedded controllers. Using this option may cause a slight performance degradation when using low-level VXIbus access functions.
`__I386__`	If you have an 80386 or higher end machine and you have the compiler set up to generate 80386 instructions, you can define this symbol to achieve higher performance when using the 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 __PROTOTYPES__
#define BINARY_COMPATIBLE
#define __I386__
.
.
.
#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 /D__PROTOTYPES__ /DBINARY_COMPATIBLE /D__I386__
```

For the Borland C compiler:

```
-DVXIWIN;__PROTOTYPES__;BINARY_COMPATIBLE;__I386__
```

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

Appendix A
NI-VXI Software Overview

This appendix lists all the programs and files located on the NI-VXI distribution diskettes.

Main Programs and Files

The main programs and files of the NI-VXI software package are found in the `C:\NIVXI` directory.

- `VXIINIT.EXE` is the VXIpc-486 initialization program. This is a DOS program that removes the **FAILED** LED and initializes the board interrupts, shared RAM, VXI register configurations, and bus configurations. `VXIINIT.EXE` must be executed from DOS (or the Windows DOS shell). It is typically included in the DOS batch file `AUTOEXEC.BAT` so that the VXIpc-486 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 multimainframe Resource Manager. You can execute `RESMAN` from DOS, Windows, or the Windows DOS shell, although the latter is not recommended. `RESMAN.EXE` may be executed only after `VXIINIT.EXE` has been run and must be run every time the mainframe is reset.

- `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 or the Windows DOS shell, although the latter is not recommended. If you do use `VIC` from the Windows DOS shell, ensure that no other Windows application that uses NI-VXI functions is executing. This program is described in detail in the *NI-VXI DOS 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, or the Windows 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 that runs in DOS or the Windows DOS shell. You can use this program to configure the system, edit the manufacturer name and ID association, and edit the information about the model names of VXI and non-VXI devices in the system, as well as the interrupt configuration information for the system. This program also displays the system configuration information devised by the Resource Manager. This program is described in detail in the *NI-VXI DOS 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, or the Windows 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.

Additional Programs and Files

The C:\NIVXI\HLP directory contains various help files used by the VIC.EXE and VXIEDIT.EXE programs.

The C:\NIVXI\TBL directory contains the following files:

- MFNAMEID.TBL contains the database of manufacturer names and their ID numbers.

- MODEL.TBL contains the database of model names, manufacturer names, and the model codes numbers.

- DEVICE.TBL contains the database of device names, manufacturer names, model names, and frame and slot associations for devices in the system.

- NONVXI.TBL contains the database for all non-VXI devices in the system.

- INTCFG.TBL contains the system interrupt configuration information.

- TRIGCFG.TBL contains the system trigger configuration information.

- UTILBUS.TBL contains the utility bus configuration information.

- CREG.TBL contains system information used by the Resource Manager.

- REGS.TBL contains the register names of VXI devices.

- WSCMDS.TBL contains the VXI Word Serial command values.

- VXIBUS.CFG contains the VXIpc-486 VXIbus configuration information.

- VXILA.CFG contains the VXIpc-486 logical address configuration information.

- VXIDEV.CFG contains the VXIpc-486 device-specific configuration information.

- VXIMF.CFG, if present, contains information specific to the board manufacturer.

- DVXI.CFG contains information for internal use by the driver.

The C:\NIVXI\INCLUDE directory contains files for the Microsoft C, Borland C, and QuickBASIC language interfaces. The include files are as follows:

- 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 Word Serial Commander and Servant functions.
- NIVXI.INC is the include file for the Microsoft QuickBASIC Language interface.

The C:\NIVXI\DOS directory contains the following files:

- RESMAND.EXE is used by RESMAN in the DOS environment. RESMAN calls this file directly.
- VICD.EXE is used by VIC in the DOS environment. VIC calls this file directly.
- VICTEXTD.EXE is used by VICTEXT in the DOS environment. VICTEXT calls this file directly.
- VXIEDITD.EXE is used by VXIEDIT in the DOS environment. VXIEDIT calls this file directly.
- VXITEDID.EXE is used by VXITEDIT in the DOS environment. VXITEDIT calls this file directly.

The C:\NIVXI\DOS directory contains up to three subdirectories.

The C:\NIVXI\DOS\BORLANDC subdirectory is installed if you choose to install both the DOS environment and the Borland C libraries. It contains the following file and subdirectory:

- NIVXIDOS.LIB is the NI-VXI library, which must be included along with other DOS libraries when linking your application program from Borland C. The NI-VXI library uses the large memory model.

- The C:\NIVXI\DOS\BORLANDC\EXAMPLES subdirectory contains various example programs that show how to use the NI-VXI software with Borland C.

The C:\NIVXI\DOS\MSC subdirectory is installed if you choose to install both the DOS environment and the Microsoft C libraries. It contains the following file and subdirectory:

- NIVXIDOS.LIB is the NI-VXI library, which must be included along with other DOS libraries when linking your application program from Microsoft C. The NI-VXI library uses the large memory model.

- The C:\NIVXI\DOS\MSC\EXAMPLES subdirectory contains various example programs that show how to use the NI-VXI software with Microsoft C.

The C:\NIVXI\DOS\QB subdirectory contains the following object, library, and batch files to generate the Microsoft library for QuickBASIC:

- QBNIVXI.LIB contains the NI-VXI function library for the QuickBASIC interface.

- QBNIVXI.OBJ contains the NI-VXI symbols used to generate the QuickBASIC library for all versions.

- MKQLB*.BAT and MKQLB*.LNK are batch and response files used to generate a QuickBASIC library for the desired version.

- MKAPPBC.BAT and MKAPPBC.LNK are batch and response files that contain an example of how to build a BASIC application using NI-VXI functions.

Also included are the following files containing system functions used in the generation of QuickBASIC libraries for version 4.5 (default) and version 7.0 (7F and 7N):

- QB.OBJ
- QBMEM.OBJ
- QB1.LIB
- QB1_7F.LIB
- QB1_7N.LIB
- QB2.LIB
- B1_71F.LIB
- B1_71N.LIB
- B2_71.LIB

The C:\NIVXI\WIN directory contains the following files:

- NIVXI.DLL is the NI-VXI dynamic link library executable file, which contains the executable code for all the NI-VXI functions. This file is loaded when any Windows application that uses the NI-VXI library is executed. The directory where it resides should be added to the PATH variable to ensure that it is loaded properly.

- RESMANW.EXE is used by RESMAN in the Windows environment. RESMAN calls this file directly.

- VICTEXTW.EXE is used by VICTEXT in the Windows environment. VICTEXT calls this file directly.

- VXITEDIW.EXE is used by VXITEDIT in the Windows environment. VXITEDIT calls this file directly.

- NIVXIPHM.386 is a virtual memory driver that the NI-VXI library uses for shared memory under Windows only.

- NIVXI.GRP is a Windows group (folder) which contains icons for easy execution of NI-VXI utilities (such as RESMAN and VICTEXT).

- NIVXI.PIF is a Windows Program Information File that lets you run the DOS NI-VXI utilities in a Windows DOS shell.

The C:\NIVXI\WIN directory contains up to two subdirectories.

The C:\NIVXI\WIN\BORLANDC subdirectory is installed if you choose to install both the Windows environment and the Borland C libraries. It contains the following file and subdirectory:

- NIVXIWIN.LIB is the NI-VXI library, which must be included along with other Windows libraries when linking your application program from Borland C. The NI-VXI library uses the large memory model. However, you are not restricted to using only the large memory model for your application. You can also use the medium, compact, or small memory models.

- The C:\NIVXI\WIN\BORLANDC\EXAMPLES subdirectory contains various example programs that show how to use the NI-VXI software with Borland C.

The C:\NIVXI\WIN\MSC subdirectory is installed if you choose to install both the Windows environment and the Microsoft C libraries. It contains the following file and subdirectory:

- NIVXIWIN.LIB is the NI-VXI library, which must be included along with other Windows libraries when linking your application program from Microsoft C. The NI-VXI library uses the large memory model. However, you are not restricted to using only the large memory model for your application. You can also use the medium, compact, or small memory models.

- The C:\NIVXI\WIN\MSC\EXAMPLES subdirectory contains various example programs that show how to use the NI-VXI software with Microsoft C.

Appendix B
Common Questions

This appendix answers common questions you may have when using the NI-VXI bus interface software on the VXIpc-486 platform. This material does not cover problems that you may have with the system. Please see Appendix C, *Troubleshooting*, if you have system-related problems. For hardware-related questions, refer to the similar sections in *Getting Started with Your VXIpc-486 Model 500 Series*.

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

Run the NI-VXI interactive control utility program, either `VIC` or `VICTEXT`. (If you run `VIC`, stay in the Text Window.) At the prompt type `ver`, and the program will display the versions of `VIC`/`VICTEXT` and NI-VXI, and the hardware revision of the VXIpc-486 that the NI-VXI software supports. However, this may not reflect the actual hardware revision of your VXIpc-486.

How can I determine what is the revision of the VXIpc-486 hardware I have in my system?

Running the NI-VXI interactive control utility program as described above will display the versions of `VIC`/`VICTEXT` and NI-VXI, and the hardware revision of the VXIpc-486 that the NI-VXI software supports. Because this may not reflect the actual hardware revision of your VXIpc-486, the only method currently available to determine the actual hardware revision is to check the label affixed to the module.

Which NI-VXI utility program must I use to configure the VXIpc-486?

You must use one of the VXI Resource Editor programs, either `VXIEDIT` or `VXITEDIT`, to configure the VXIpc-486. You will find these programs in the `NIVXI` directory.

Which NI-VXI utility program must I use to initialize the VXIpc-486 hardware?

You must use the hardware initialization program, `VXIINIT`, to initialize the VXIpc-486. You will find this program in the `NIVXI` directory. `VXIINIT` uses the settings in the Configuration Editor of the VXI Resource Editor program, either `VXIEDIT` or `VXITEDIT`.

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

You must use the RESMAN program to perform startup Resource Manager operations. You will find this program in the NIVXI directory. RESMAN uses the settings in the Configuration Editor of the VXI Resource Editor program, either VXIEDIT or VXITEDIT. It initializes your VXI/VMEbus system and stores the information that it collects in 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 VXIbus 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.

Finally, if you are using LabVIEW, LabWindows/CVI, or LabWindows for DOS, 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.

How do I install the VXIpc-486 in a slot other than Slot 0 and prevent the VXIpc-486 from being the Resource Manager?

1. Refer to Chapter 2, *VXIpc-486 Configuration and Installation*, of *Getting Started with Your VXIpc-486 Model 500 Series* for information on configuring the VXIpc-486 hardware for a Non-Slot 0 application. If you are using a Model 200 Series VXIpc-486, refer to the documentation that came with your machine.

2. Run the VXI Resource Editor program, either VXIEDIT or VXITEDIT, and select the Configuration Editor.

3. In the Logical Address Configuration Editor, change the **Logical Address** parameter from 0 to a non-zero value.

4. In the Bus Configuration Editor, change the **Local Bus Timeout** parameter to DISABLE.

5. In the Bus Configuration Editor, change the **VXI Bus Timeout** parameter to DISABLE.

6. Save your changes and rerun VXIINIT and RESMAN.

Notes:

1. If your board is set up as a Message-Based device, it must wait for the Word Serial command *Begin Normal Operation* from the actual Resource Manager device (the device at Logical Address 0).

2. If your board is set up as a Register-Based or Extended device, you don't have to run RESMAN. But running RESMAN will not cause any problems.

3. To set whether your board is Extended, Message-Based, or Register-Based, use the Logical Address Configuration Editor in the VXI Resource Editor program to edit the **Device Type** field. The following table defines the possible selections. The default value is shown in ***bold italics***.

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

Appendix C
Troubleshooting

This appendix addresses system-related problems you may encounter when using the NI-VXI bus interface software on the VXIpc-486 platform. Refer also to Appendix B, *Common Questions*, to see if your problem is addressed by that material. This appendix covers the following issues:

- Running the `VXIINIT` initialization program results in an error message or causes the system to hang.
- Running the `RESMAN` Resource Manager program results in incorrect data or causes the system to hang.
- The mouse degrades system performance.
- The VXIpc-486 and/or its installed components (such as GPIB, COM Port) fail to work once ISA bus plug-in cards are installed.
- The network card installed does not operate properly.
- The hard disk drive reports failures or other errors.

If you encounter any of these problems, your system may have a conflict with one of the VXIpc-486 configuration settings. For proper operation, the VXIpc-486 requires the allocation of the following system resources:

- One interrupt level that cannot be shared with another device
- Two blocks of memory space in the upper memory region of the PC AT address space

In addition, the VXIpc-486 components require Port I/O register space, interrupts and DMA channels. These resources are statically allocated. For more details, refer to Appendix G, *VXIpc-486 Hardware Configuration*, in *Getting Started with Your VXIpc-486 Model 500 Series* (Appendix F in the Model 200 Series manual).

If you are unable to resolve your problem, please fill out the appropriate forms in the *Customer Communication* appendix of this manual and the hardware manual, and contact National Instruments.

What information should I have before I call National Instruments?

Make sure that none of the following questions relate to the problem you are having, or the suggestions contained in the answers have not corrected the problem. As stated previously, you should also refer to the hardware manual to see if your questions are answered there. When you call National Instruments, make sure you have filled out the *VXIpc-486 Hardware and NI-VXI Software Configuration Form* in Appendix D. Also, if you are having problems with `RESMAN`, you can get `RESMAN` to create a file of its output by issuing

```
resman -o
```

from the Program Manager **Run** option, a Windows MS-DOS Shell, or by changing the Command Line to the above statement in the **Program Item Properties** window of the **RESMAN** icon. This creates a file `RESMAN.OUT` in the `TBL` subdirectory of the `NIVXI` directory.

What does `VXIINIT` mean by `File` *`filename`* `not found`?

When `VXIINIT` runs, it attempts to access some of the configuration or table files, such as `VXILA.CFG`, `VXIBUS.CFG`, and `VXIDEV.CFG`. If `VXIINIT` cannot find any of these files, you have two options.

1. The file(s) may have been deleted accidentally. If the file does not exist in your `TBL` directory after you installed your software, you will need to reinstall NI-VXI from your software distribution media.

2. You can modify the `NIVXIPATH` environment variable. Check the `AUTOEXEC.BAT` file in your root directory, and the `SYSTEM.INI` and `WIN.INI` files in the Windows directory. Make sure that the user environment variable `NIVXIPATH` points to the right directory. Notice that this should be the same for all three files. By default this directory is `C:\NIVXI`, but you may have chosen another destination directory during the installation process.

What does `VXIINIT` mean by `Error writing configuration to file`?

This is only a warning message, and does not indicate a problem with the configuration. Either the log file could not be created, or only a partial log was written.

`VXIINIT` creates a file `VXIINIT.OUT` in the `TBL` subdirectory. This file is a text file that contains the same configuration information you see printed to the screen. The likely cause of this message is that the disk drive is full.

Why does RESMAN report warning/errors about devices that don't exist in my system?

If the VXIINIT program runs successfully but you encounter problems with RESMAN, the Resource Manager program, the cause is probably that your system contains a device that already occupies all or some of the upper memory area that is allocated for the master windows. The master windows consist of 32 KB and 64 KB blocks of contiguous memory in the PC AT upper memory space that the VXIpc-486 uses to access the VXI/VME bus. The upper memory space of a PC AT is the region from 640 KB (A000h) to 1024 KB (FFFFh). Several system resources, such as system ROM and video memory, reside in this area. On a VXIpc-486, the master windows occupy the memory regions from a base address of C800h to CFFFh, and E000h to EFFFh.

To solve this problem, first determine what area of upper memory your devices use. Then move your device to the unused area of upper memory from D000h to DFFFh. If you have Ethernet networking cards, be aware that many of them occupy upper memory. Many Ethernet cards have an upper memory base address of D000h or E000h.

In addition to the above problems, your system may contain an expanded-memory manager or an expanded-memory emulator that uses a block of upper memory within the range that is allocated for the master-mode windows.

To solve this problem, verify which expanded-memory manager or emulator you are using, and prevent it from using upper memory within the address ranges for the master-mode windows. The following examples show how to prevent various expanded-memory managers or emulators from using upper memory within the address ranges for the master-mode windows, from C800h to CFFFh and E000h to EFFFh.

If you are using EMM386, which is included with MS-DOS, prevent it from using the address ranges for the master-mode windows by editing the CONFIG.SYS file and appending the flags X=C800-CFFF and X=E000-EFFF to the line DEVICE=EMM386.EXE.

If you are using QEMM386 from Quarterdeck, prevent it from using the address ranges for the master-mode windows by editing the CONFIG.SYS file and appending the flags X=C800-CFFF and X=E000-EFFF to the line DEVICE=QEMM386.SYS.

If you are using 386MAX from Qualitas, prevent it from using the address ranges for the master-mode windows by editing the 386MAX.PRO file located in the 386MAX directory, and inserting the lines RAM=C800-CFFF and RAM=E000-EFFF.

If you are using any other expanded-memory manager, refer to its documentation to learn how to prevent it from using the address ranges for the master-mode windows.

Why does VXIINIT work fine but RESMAN hangs the system?

This is most likely caused by a conflict with the master window. See the previous question, *Why does RESMAN report warning/errors about devices that don't exist in my system?*

Why is it that when I install my mouse driver, the system performance degrades?

If your mouse appears to slow down system performance, add the `/y` option to your mouse command. This action switches off the hardware cursor and gives you full system performance.

Why does the VXIpc-486 and/or its installed components (GPIB, COM Port) fail to work once ISA bus plug-in cards are installed?

The VXIpc-486 and its components use port I/O register space, interrupts and DMA, just like a desktop PC. If installing an ISA bus card causes something to stop working, it is very likely that you have a conflict of these resources between the VXIpc-486 and the plug-in card. See Appendix G, *VXIpc-486 Hardware Configuration*, in *Getting Started with Your VXIpc-486 Model 500 Series* (Appendix F in the Model 200 Series manual) for the resources that the VXIpc-486 uses.

Why does my network not work properly?

There can be a possible conflict with the network card in your system. See the question *Why does the VXIpc-486 and/or its installed components (GPIB, COM Port) fail to work once ISA bus plug-in cards are installed?* above.

In addition, most network cards use some portion of the upper memory area, usually between D000h to EFFFh, for their operation. Because the VXIpc-486 also uses some of this memory area, the computer and the network card may be conflicting. See *Why does RESMAN report warning/errors about devices that don't exist in my system?* earlier in this section for more information.

When my computer boots, it says that it cannot access the hard disk drive (HDD), or says `System Not Found`, or `Fixed Disk Failure`. Is my HDD damaged?

It is not uncommon for the computer to be unable to access the HDD if the system loses power during certain times of operation. Try the following procedure to fix the problem.

1. Reboot the VXIpc-486 and, during the system tests, press the <F2> key.
2. Select **Setup** from the menu.
3. Press the <F9> key to Auto Configure the system.
4. Press <F10> to save the configuration.
5. Press <Esc> until you exit the BIOS `Setup` utility.

6. If the errors do not go away, power down your mainframe, remove the VXIpc-486, and check the hard drive settings on its label.

7. Reinsert the VXIpc-486, power on the mainframe, get back into the BIOS setup, and make sure that the hard drive parameters match those printed on the label.

8. If the errors do not go away, open up the VXIpc-486 and make sure the hard drive cable and power cable are securely attached to the hard drive.

If this did not fix the problem, check to see if you can boot off a floppy DOS disk and then change drive to C. If not, your HDD is probably damaged and may need to be reformatted or repaired.

Caution: ***You should NOT reformat the disk unless directed by National Instruments support personnel. The HDD controller may have a problem, or a fuse may have blown, causing the HDD to fail. Reformatting the drive prematurely may not fix the problem but could result in a loss of data. Contact National Instruments for repair information.***

Appendix D
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.

Corporate Headquarters
(512) 795-8248
Technical support fax: (800) 328-2203
(512) 794-5678

Branch Offices	**Phone Number**	**Fax Number**
Australia	(03) 879 9422	(03) 879 9179
Austria	(0662) 435986	(0662) 437010-19
Belgium	02/757.00.20	02/757.03.11
Denmark	45 76 26 00	45 76 71 11
Finland	(90) 527 2321	(90) 502 2930
France	(1) 48 14 24 00	(1) 48 14 24 14
Germany	089/741 31 30	089/714 60 35
Italy	02/48301892	02/48301915
Japan	(03) 3788-1921	(03) 3788-1923
Mexico	95 800 010 0793	95 800 010 0793
Netherlands	03480-33466	03480-30673
Norway	32-848400	32-848600
Singapore	2265886	2265887
Spain	(91) 640 0085	(91) 640 0533
Sweden	08-730 49 70	08-730 43 70
Switzerland	056/20 51 51	056/20 51 55
Taiwan	02 377 1200	02 737 4644
U.K.	0635 523545	0635 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. Use additional pages if necessary.

Name _____

Company _____

Address _____

_____

Fax (___) _____ Phone (___) _____

National Instruments hardware product model _____ Revision _____

Configuration _____

Operating system _____

Speed _____ MHz RAM _____ MB Display adapter _____

Mouse _____ yes _____ no Other adapters installed _____

Hard disk capacity _____ MB Brand _____

Instruments used _____

National Instruments software product _____ Version _____

Configuration _____

The problem is _____

_____

_____

_____

_____

List any error messages _____

_____

_____

_____

_____

The following steps will reproduce the problem _____

_____

_____

_____

_____

VXIpc-486 Hardware and NI-VXI 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

Hardware Information (refer to your VXIpc-486 hardware manual)

- VXIpc-486 ID Label Information
  - Model Number _____ 200 _____ 240 _____ 500 _____ 566 _____ 599
  - Complete Part Number (including 3-digit suffix) _____
  - Serial Number _____
  - Revision Number _____
  - Fixed Disk Type (See ID label on module cover) _____
- VXIpc-486 Hardware Configuration Information
  - Factory Configuration _____ yes _____ no

    If no, give the following information.

    VXIbus Slot/External CLK10 Configuration

    _____ A* _____ B _____ C _____ D _____ E _____ F
  - External Trigger Input Configuration

    _____ Unterminated* _____ Terminated
  - External Audio/Trigger Output Configuration

    _____ Trigger Out* _____ Audio Out
- VXIpc-EFD External Floppy Drive

  _____ Not Installed _____ Installed. If installed, give the following information.

  Serial Number _____

  Revision _____

  * Factory Configuration

- 80387SX Math Coprocessor (Model 200 Series only)

  _____ Not Installed _____ Installed.

Software Information

- NI-488.2 Software for DOS/Windows
  Version _____
- NI-VXI for VXIpc-486 and MS-DOS/Windows
  Version _____
- Logical Address (VXIla) Configuration Editor

  Logical Address _____

  Device Type _____

  Address Space _____

  VXI Shared RAM (Byte Order) _____

  VXI Shared RAM (Mbytes) _____

  Shared Memory Pool (Windows) _____

  Slave Block Transfer Mode _____

  Slave Supervisory Access Only _____

  Resource Manager Delay (seconds) _____
- Bus (VXIbus) Configuration Editor

  Bus Request Level _____

  Fair Requester Timeout (µs) _____

  Bus Release Mechanism _____

  Bus Arbitration _____

  Local Bus Timeout (µs) _____

  VXIbus Timeout (µs) _____

  Bus Acquisition Time in Timeout _____
- Device (VXIdev) Configuration Editor

  SYSRESET Resets PC _____

  Use System Sanity Timer _____

  Servant Area Size _____

  Protocol Register _____

Read Protocol Response _____

Number of Handlers _____

Number of Interrupters _____

- Super VGA Option (Model 500 Series only; described in your hardware manual)

  _____ Not Installed _____ Factory-Installed _____ User-Installed

  If installed, give the following information.

  System Settings Modified for Super VGA? _____

  Super VGA Option Selected _____

Other Products

- Windows Version _____
- Windows Mode (Enhanced, Standard, or Real) _____
- Any Other Software In System _____
- I/O Expansion Cards–VXIpc-486/2 only (Manufacturer, Model) _____

  _____
- Monitor (Manufacturer, Model) _____
- Keyboard (Manufacturer, Model) _____
- Mouse (Manufacturer, Model) _____
- VXIbus Mainframe Manufacturer and Model _____
- Other VXIbus Devices

Manufacturer	Model	Function	Slot	Logical Address

Continue on a separate sheet, if necessary.

- Interrupt Level(s) of Other VXI Devices (Handler/Interrupter) _____

_____

_____

- IEEE 488 Devices (Manufacturer, Model, GPIB Address) _____

_____

_____

_____

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: **Getting Started with Your NI-VXI™ Software for the VXIpc™-486 Model 200/500 Series and Windows**

Edition Date: **March 1995**

Part Number: **320930A-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
National Instruments Corporation
6504 Bridge Point Parkway, MS 53-02
Austin, TX 78730-5039

Fax to: Technical Publications
National Instruments Corporation
MS 53-02
(512) 794-5678

Glossary

Prefix	Meaning	Value
μ-	micro-	10^{-6}
m-	milli-	10^{-3}
k-	kilo-	10^{3}
M-	mega-	10^{6}
g-	giga-	10^{9}

Symbols

°	degrees
%	percent
Ω	ohms

A

A	amperes
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.
ANSI	American National Standards Institute
ASIC	Application Specific Integrated Circuit (a semi-custom chip)

B

B	bytes
backplane	An assembly, typically a printed circuit board, with 96-pin connectors and signal paths that bus the connector pins.
BIOS	Basic Input/Output System
block-mode transfer	An uninterrupted transfer of data elements in which the master sources only the first address at the beginning of the cycle. The slave is then responsible for incrementing the address on subsequent transfers so that the next element is transferred to or from the proper storage location.
BTO unit	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.
bus master	A device that is capable of requesting the Data Transfer Bus (DTB) for the purpose of accessing a slave device.
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

C	Celsius
CLK10	A 10 MHz, ± 100 ppm, individually buffered (to each module slot), differential ECL system clock that is sourced from Slot 0 and distributed to Slots 1 through 12 on P2. It is distributed to each slot as a single-source, single-destination signal with a matched delay of under 8 ns.
Commander	A Message-Based device which is also a bus master and can control one or more Servants.

D

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

dynamic configuration	A method of automatically assigning logical addresses to VXIbus devices at system startup or other configuration times.
dynamically configured device	A device that has its logical address assigned by the Resource Manager. A VXI device initially responds at Logical Address 255 when its MODID line is asserted. A MXIbus device responds at Logical Address 255 during a priority select cycle. The Resource Manager subsequently assigns it a new logical address, which the device responds to until powered down.

E

ECL	Emitter-Coupled Logic
EFD	External Floppy Disk
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.
EMI	electromagnetic interference
extended controller	A mainframe extender with additional VXIbus controller capabilities.
Extended Longword Serial	A form of Word Serial communication in which Commanders and Servants communicate with 48-bit data transfers.
external controller	In this configuration, a plug-in interface board in a computer is connected to the VXI mainframe via one or more VXIbus extended controllers. The computer then exerts overall control over VXIbus system operations.

F

fair requester	A MXIbus master that will not arbitrate for the MXIbus after releasing it until it detects the bus request signal inactive. This ensures that all requesting devices will be granted use of the bus.
FCC	Federal Communications Commission

G

GPIB	General Purpose Interface Bus; the industry-standard IEEE 488 bus.
GPIO	General Purpose Input/Output, a module within the National Instruments TIC chip. GPIOs are used for connecting external signals to the TIC chip for routing/conditioning to the VXIbus trigger lines.

H

Hz	hertz; events per second.

I

IEEE	Institute of Electrical and Electronics Engineers
in.	inches
I/O	input/output; the techniques, media, and devices used to achieve communication between entities.
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.
ISA	Industry Standard Architecture

K

KB	1,024 or 2^{10}; kilobytes of memory

L

LED	light emitting diode
logical address	An 8-bit number that uniquely identifies the location of a VXIbus device's configuration registers in a system. The A16 base address of a device is C000h + Logical Address * 40h.
Longword Serial	A form of Word Serial communication in which Commanders and Servants communicate with 32-bit data transfers instead of 16-bit data transfers as in the normal Word Serial Protocol.

M

MB	1,048,576 or 2^{20}; megabytes of memory
m	meters

master	A functional part of a MXI/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.
MIGA	Message-Based Interface Gate Array; a proprietary National Instruments chip.
MODID	A set of 13 signal lines on the VXI backplane that VXI systems use to identify which modules are located in which slots in the mainframe.

N

NI-VXI	The National Instruments bus interface software for VME/VXIbus systems.
nonprivileged access	One of the defined types of VMEbus data transfers; indicated by certain address modifier codes. Each of the defined VMEbus address spaces has a defined nonprivileged access mode.

P

PC AT	Personal Computer Advanced Technology
PRI	Priority
privileged access	See *supervisory access*.

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.

RONR	Request On No Request
ROR	Release On Request; a type of VMEbus arbitration where the current VMEbus master relinquishes control of the bus only when another bus master requests the VMEbus.
RRS	Round Robin Select
RWD	Release When Done

S

s	seconds
Servant	A device controlled by a Commander.
statically configured device	A device whose logical address cannot be set through software; that is, it is not dynamically configurable.
supervisory access	One of the defined types of VMEbus data transfers; indicated by certain address modifier codes.
SYSCLK	System Clock Driver; a VMEbus functional module that provides a 16 MHz timing signal on the utility bus.
SYSFAIL	System Failure Driver; asserts a signal to indicate an internal failure.
SYSRESET	System Reset Driver; asserts a signal to indicate a system reset or power-up condition.
System Controller	A functional module that has arbiter, daisy-chain driver, and MXIbus cycle timeout responsibility.

T

TIC	Trigger Interface Chip; a proprietary National Instruments chip used for direct access to the VXI trigger lines. The TIC contains a 16-bit counter, a dual 5-bit tick timer, and a full crosspoint switch.
trigger	Either TTL or ECL lines used for intermodule communication.
tristated	Defines logic that can have one of three states: low, high, and high-impedance.
TTL	Transistor-Transistor Logic

V

V	volts
VDC	volts direct current
VIC	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.
VICTEXT	Text-based VXI Interactive Control program, functionally equivalent to VIC.
VME	Versa Module Eurocard or IEEE 1014
VXIbus	VMEbus Extensions for Instrumentation
VXIINIT	A program in the NI-VXI bus interface software package that initializes the board interrupts, shared RAM, VXI register configurations, and bus configurations. This program must be run at least once after rebooting your machine to initialize the system.
VXIEDIT	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.
VXITEDIT	Text-based VXI Resource Editor program, functionally equivalent to VXIEDIT.

W

W	watts
Word Serial Protocol	The simplest required communication protocol supported by Message-Based devices in the VXIbus system. It utilizes the A16 communication registers to transfer data using a simple polling handshake method.

Index

A

address space, setting, 3-4
AT-GPIB compatibility, 1-4, 2-1
AUTOEXEC.BAT file, modifying, 2-3

B

bus configuration, 3-8 to 3-11
    bus acquisition time in timeout, 3-11
    bus arbitration, 3-9
    bus release mechanism, 3-9
    bus request level, 3-8
    fair requester timeout, 3-9
    local bus timeout, 3-10
    VXIbus Configuration Editor (VXIbus) menu (illustration), 3-8
    VXIbus timeout, 3-10

C

common questions, B-1 to B-3
compiling your C program, 4-4
    symbols, 4-4 to 4-5
configuration. *See* NI-VXI software.
customer communication, *xi*, D-1

D

defining symbols, 4-4 to 4-5
device configuration, 3-11 to 3-14
    Device Configuration Editor (VXIdev) menu (illustration), 3-11
    number of handlers, 3-13
    number of interrupters, 3-14
    Protocol register, 3-13
    read protocol response, 3-13
    Servant Area size, 3-13
    SYSRESET, 3-12
    system sanity timer, 3-12
Device Configuration Editor (VXIdev) menu (illustration), 3-11
device type, setting, 3-4
documentation
    conventions used in manual, *xi*
    how to use documentation set, *ix*
    organization of manual, *x*
    related documentation, *xi*

E

example programs, 4-2, 4-4

F

FAILED LED, A-1
fair requester timeout, 3-9
files included with NI-VXI software, A-1 to A-5

G

GPIB connector
    illustration, D-7
    signals (table), D-7
GPIO connector. *See* COM2/GPIO connector.

H

handlers, specifying number of, 3-13
hardware configuration, 1-1

I

IEEE 488 interface, 1-1, 1-4
include files, A-3
INCLUDE variable, 2-3
installation, NI-VXI software, 2-1 to 2-4
    AUTOEXEC.BAT file, modifying, 2-2 to 2-4

procedure for, 2-1 to 2-2, 2-4
PROGMAN.INI file, modifying, 2-3
space requirements, 2-1
SYSTEM.INI file, modifying, 2-4
WIN.INI file, modifying, 2-4
INSTALL.EXE, 2-1 to 2-4
interrupters, specifying number of, 3-14

L

LabVIEW for Windows, 1-3
LabWindows/CVI for Windows, 1-3
LIB variable, 2-3
local bus timeout, 3-10
local resource access functions, 4-3
logical address configuration, 3-3 to 3-7
address space, 3-4
device type, 3-4
Logical Address Configuration Editor (VXIla) menu (illustration), 3-3
logical address, 3-3
Resource Manager delay, 3-7
shared memory pool (Windows), 3-6
slave block transfer mode, 3-6
slave supervisory access only, 3-7
VXI shared RAM (Byte Order), 3-4
VXI shared RAM (MBytes), 3-5 to 3-6
Logical Address Configuration Editor (VXIla) menu (illustration), 3-3
low-level access functions, 4-3

M

manual. *See also* documentation.
flowchart, 1-2
memory model, 4-2
multiple applications using NI-VXI library, 5-2

N

networks
connecting VXIpc-486, I-4
problems, J-3 to J-4
NI-VXI software
bus configuration, 3-8 to 3-11
bus acquisition time in timeout, 3-11
bus arbitration, 3-9
bus release mechanism, 3-9
bus request level, 3-8
fair requester timeout, 3-9
local bus timeout, 3-10
VXIbus Configuration Editor (VXIbus) menu (illustration), 3-8
VXIbus timeout, 3-10
defining symbols, 4-4 to 4-5
device configuration, 3-11 to 3-14
Device Configuration Editor (VXIdev) menu (illustration), 3-11
number of handlers, 3-13
number of interrupters, 3-14
Protocol register, 3-13
read protocol response, 3-13
Servant Area size, 3-13
SYSRESET, 3-12
system sanity timer, 3-12
installation, 2-1 to 2-4
AUTOEXEC.BAT file, modifying, 2-2 to 2-4
procedure for, 2-1 to 2-2
PROGMAN.INI file, modifying, 2-3
space requirements, 2-1
SYSTEM.INI file, modifying, 2-4
WIN.INI file, modifying, 2-4
logical address configuration, 3-3 to 3-7
address space, 3-4
device type, 3-4
Logical Address Configuration Editor (VXIla) menu (illustration), 3-3
logical address, 3-3
Resource Manager delay, 3-7
shared memory pool (Windows), 3-6
slave block transfer mode, 3-6
slave supervisory access only, 3-7
VXI shared RAM (Byte Order), 3-4
VXI shared RAM (MBytes), 3-5 to 3-6
programming considerations, 4-2 to 4-3
example programs, 4-2
interactive control of NI-VXI, 4-1
local resource access functions, 4-3
low-level access functions, 4-3
memory model, 4-2

multiple applications using NI-VXI library, 4-2
using VIC or VICTEXT utilities, 4-1
programs and files
additional programs and files, A-2 to A-5
main programs and files, A-1 to A-2
questions and answers, B-1 to B-3
reinitializing hardware after configuration, 3-14
NIVXIPATH variable, 2-2 to 2-4
Number of Handlers, 3-13
Number of Interrupters, 3-14

P

PATH variable, 2-3
PROGMAN.INI file, modifying, 2-3
programming considerations
example programs, 4-2
interactive control of NI-VXI, 4-1
local resource access functions, 4-3
low-level access functions, 4-3
memory model, 4-2
multiple applications using NI-VXI library, 4-2
Protocol register, 3-13
Protocol response, setting, 3-13

R

Read Protocol response, 3-13
README.DOC file, 4-1, A-2
reinitializing hardware after configuration, 3-14
RESMAN program
hanging the system, C-3
performing startup Resource Manager operations, B-2
purpose and use, 4-1, A-1
setting delay period for Resource Manager, 3-7
testing the system, B-2
warning/error messages for nonexistent devices, C-3

S

sanity timer, setting, 3-12
Servant Area size, setting, 3-13
shared memory pool (Windows), 3-6
slave block transfer mode, 3-6
slave supervisory access only, setting, 3-7
software. *See also* NI-VXI software.
compatible software for VXIpc-486, 1-3
NI-488.2, 1-4
optional, 1-3
Super VGA option, configuring, 1-3
SYSFAIL* signal, 3-12
SYSRESET, setting, 3-12
system sanity timer, setting, 3-12
SYSTEM.INI file, modifying, 2-4

T

technical support
fax and phone numbers, D-1
information to gather before calling National Instruments, C-1 to C-2
troubleshooting, C-1 to C-5
cannot access hard disk (HDD), C-4
information to gather before calling National Instruments, C-1 to C-2
ISA bus plug-in cards, C-4
mouse problems, C-4
network problems, C-4
questions and answers, B-1 to B-3
RESMAN problems, C-3
VXIINIT error messages, C-2

V

VIC program
communicating with VXI devices, 4-1
help files, A-2
purpose and use, A-1
VICTEXT program
communicating with VXI devices, 4-1
help files, A-2
purpose and use, A-1
VXI shared RAM (Byte Order), 3-4
VXI shared RAM (MBytes)
options (table), 3-5 to 3-6

purpose and use, 3-5
VXIbus Configuration Editor (VXIbus)
menu (illustration), 3-8
VXIBUS.CFG file, 3-2, 3-14
VXIDEV.CFG file, 3-2, 3-14
VXIEDIT program
exiting, 3-14
help files, A-2
Main Menu (illustration), 3-2
purpose and use, A-1
running, 3-1 to 3-2
VXIINIT program
purpose and use, 4-1, A-1
troubleshooting error messages, C-2
VXILA.CFG file, 3-2, 3-14
VXIpc-486
definition, 1-1
Model 200 Series, 1-1
Model 500 Series, 1-1
optional software, 1-3
questions and answers, B-1 to B-3
reinitializing after NI-VXI software configuration, 3-14
software description, 1-3 to 1-4
troubleshooting, C-1 to C-5
VXITEDIT program
exiting, 3-14
purpose and use, A-1
running, 3-14

W

WIN.INI file, modifying, 2-4

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP