

**Getting Started with Your NI-VXI™
Software for the VXIpc™ -486
Model 500 Series and LynxOS**



September 1995 Edition

Part Number 321006A-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 9 879 9422, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Canada (Ontario) 519 622 9310,
Canada (Québec) 514 694 8521, Denmark 45 76 26 00, Finland 90 527 2321, France 1 48 14 24 24,
Germany 089 741 31 30, Hong Kong 2645 3186, Italy 02 48301892, Japan 03 5472 2970, Korea 02 596 7456,
Mexico 95 800 010 0793, Netherlands 0348 433466, Norway 32 84 84 00, Singapore 2265886,
Spain 91 640 0085, Sweden 08 730 49 70, Switzerland 056 200 51 51, Taiwan 02 377 1200,
U.K. 01635 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

NI-488.2™, NI-488.2M™, 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	vii
How to Use This Documentation Set.....	vii
Organization of This Manual.....	viii
Conventions Used in This Manual.....	ix
Related Documentation.....	ix
Customer Communication.....	ix
Chapter 1	
Introduction	1-1
How to Use This Manual.....	1-2
What You Need to Get Started.....	1-2
NI-VXI Software Description.....	1-3
NI-488.2M Software Description.....	1-3
Chapter 2	
NI-VXI Software Installation	2-1
Installing the Software.....	2-1
Prepare for Installation.....	2-1
Extract the NI-VXI Software.....	2-2
Install the NI-VXI Driver into the LynxOS Kernel.....	2-2
Configure the System.....	2-3
Chapter 3	
NI-VXI Software Configuration	3-1
Configuring the NI-VXI Software.....	3-1
Running vxitedit.....	3-1
Logical Address Configuration.....	3-2
Logical Address.....	3-2
Device Type.....	3-2
Address Space.....	3-2
VXI Shared RAM (MBytes).....	3-2
Slave Block Transfer Mode.....	3-4
Slave Supervisory Access Only.....	3-4
Resource Manager Delay.....	3-4
VXI Shared RAM (Primary Byte Order).....	3-4
Bus Configuration.....	3-4
Bus Request Level.....	3-4
Fair Requester Timeout.....	3-4
Bus Release Mechanism.....	3-5
Bus Arbitration.....	3-5
Local Bus Timeout.....	3-5
VXIbus Timeout.....	3-5
Bus Acquisition Time in Timeout.....	3-5

Device Configuration.....	3-5
SYSRESET Resets PC.....	3-5
Servant Area Size.....	3-6
Protocol Register.....	3-6
Read Protocol Response.....	3-6
Number of Handlers.....	3-6
Number of Interrupters.....	3-6
Use System Sanity Timer.....	3-6
Word Serial Poll Value	3-7
Exiting vxitedit and Reinitializing the Hardware	3-7
Chapter 4	
Using NI-VXI with LynxOS	4-1
Interactive Control of NI-VXI	4-1
Example Programs	4-2
Compiling and Linking	4-2
Multiple Applications Using the NI-VXI Library	4-2
Low-Level Access Functions.....	4-3
Local Resource Access Functions.....	4-3
Using Callbacks from NI-VXI.....	4-3
Errors Closing the VXI Library	4-3
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

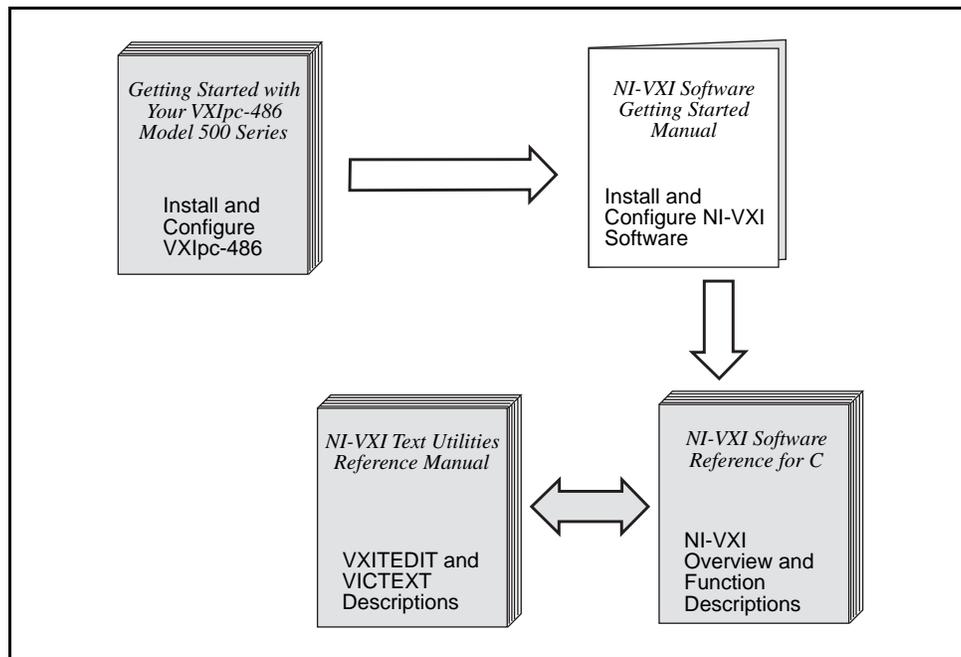
Table

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

About This Manual

This manual contains instructions for installing and configuring the NI-VXI bus interface software for the VXIpc-486 and LynxOS. The NI-VXI software works with all models of 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. 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 LynxOS on the VXIpc-486. 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 Text Utilities Reference Manual* to learn more about the capabilities of the NI-VXI 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-488.2M Software for LynxOS* with your VXIpc-486 kit. Refer to the *VXIpc-486 Documentation Notice: AT-GPIB and LynxOS Compatibility* for more information about using this documentation.

Organization of This Manual

This manual is organized as follows:

- Chapter 1, *Introduction*, describes the NI-VXI and NI-488.2M software kits for the VXIpc-486 and LynxOS and lists what you need to get started.
- Chapter 2, *NI-VXI Software Installation*, contains instructions for installing the NI-VXI software for the VXIpc-486 and LynxOS.
- Chapter 3, *NI-VXI Software Configuration*, describes how to use the VXI resource editor program to configure the NI-VXI software for the VXIpc-486 computer.
- Chapter 4, *Using NI-VXI with LynxOS*, 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 that make up the NI-VXI software.
- 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.
<i>italic</i>	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.
<i>bold italic</i>	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:

- ANSI/IEEE Standard 1014-1987, *IEEE Standard for a Versatile Backplane Bus: VMEbus*
- ANSI/IEEE Standard 1155-1993, *IEEE VMEbus Extensions for Instrumentation: VXIbus*
- VXI-6, *VXIbus Mainframe Extender Specification*, Rev. 1.0, VXIbus Consortium

Customer Communication

National Instruments wants to receive your comments on our products and manuals. We are interested in the applications you develop 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.

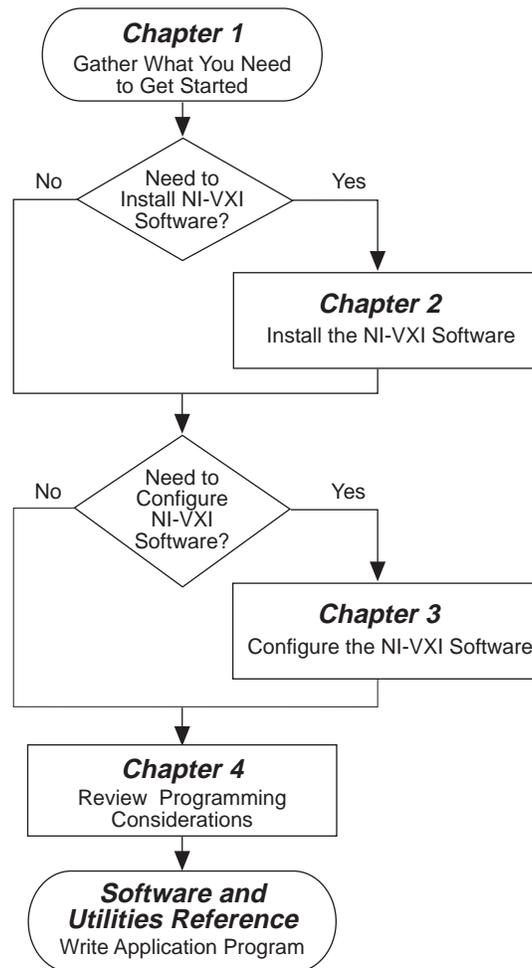
Chapter 1

Introduction

This chapter describes the NI-VXI and NI-488.2M software kits for the VXIpc-486 and LynxOS and lists what you need to get started. 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*.

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.

How to Use This Manual



What You Need to Get Started

- LynxOS version 2.3 or higher installed on your computer
- VXIpc-486 Model 500 Series module
- Keyboard adapter cable
- NI-VXI software for the VXIpc-486 Model 500 Series and LynxOS

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 LynxOS 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.

NI-488.2M Software Description

The NI-488.2M software kit for the VXIpc-486 and LynxOS 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 LynxOS Compatibility* directs you to the appropriate sections of the *Getting Started with Your AT-GPIB/TNT and the NI-488.2M Software for LynxOS* 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 LynxOS. For instructions on installing LynxOS, refer to the documentation included with your LynxOS 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-488.2M Software for LynxOS* for information about using the capabilities of the AT-GPIB and the ESP-488 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

Note: *You must have super-user privileges to install the NI-VXI software. If you do not have these privileges, please contact your system administrator.*

Because you must have super-user privileges to install the software, you must first either log in as `root` and set the `NIVXIPATH` environment variable, or you can set the environment variable under your login and then use the `su` command to obtain super-user privileges.

Prepare for Installation

Before you install the NI-VXI software, you should first prepare the LynxOS system for the driver.

1. Determine the directory where you wish to install the software. The standard directory is `/usr/nivxi`, but you can install the driver in any directory.
2. Set up the environment variable, `NIVXIPATH`. This variable should contain the absolute directory path to the NI-VXI driver. The technique to set the variable depends on the shell that you are using.

For example, if you plan to install the NI-VXI software in `/usr/nivxi` and you are using `sh` or `bash`, execute the following commands:

```
NIVXIPATH=/usr/nivxi
export NIVXIPATH
```

If you are using either `csh` or `tcsh`, execute the following command:

```
setenv NIVXIPATH /usr/nivxi
```

If you are using another shell, refer to the documentation for that shell to determine how to set the environment variable.

Extract the NI-VXI Software

Now that the system is ready for the software, you are ready to extract the software from the disk.

1. Insert the disk containing the NI-VXI software into the floppy drive.
2. Create the NI-VXI directory.
3. Change (`cd`) to that directory.
4. Extract the NI-VXI driver with the `tar` command.

For example, on most LynxOS systems, the device `fd0` points to the floppy drive. In this case, you can use the following commands:

```
mkdir /usr/nivxi
cd /usr/nivxi
tar xf /dev/fd0
```

Install the NI-VXI Driver into the LynxOS Kernel

NI-VXI is a static driver, which means that you need to compile it into the LynxOS kernel. As a result, the driver is loaded when the system starts up and is present throughout the time that LynxOS is running.

1. From the NI-VXI directory, execute the `buildvxi` script, which installs the NI-VXI driver into the system and rebuilds the kernel.

Note: *The message that is displayed is the same step procedure listed below.*

2. Back up the `/sys/lynx.os/CONFIG.TBL` file.
3. Add the following line to the end of the `CONFIG.TBL` file:

```
I:vxi.cfg
```

This adds the NI-VXI driver to the list of drivers to be included in the LynxOS kernel. To remove the driver at a later time, comment out the above line using a pound (`#`) sign, as shown below:

```
#I:vxi.cfg
```

4. From the `/sys/lynx.os`, type the following command:

```
make install
```

5. Reboot the system using the following command:

```
reboot -aN
```

Configure the System

At this point, the NI-VXI driver should be present in the system. For verification, you can use the `devices` and `drivers` utilities and look for an entry called `vxi`. In addition, execute the command `vxiinit` to initialize the hardware. If you have any problems, please see Appendix B, *Common Questions*, and Appendix C, *Troubleshooting*.

Note: *You must run the `vxiinit` program every time the LynxOS system is booted or if you make any changes to the NI-VXI configuration.*

To use the NI-VXI software successfully, you must include the `NIVXIPATH` environment variable with every login, as described above. In addition, ensure that the access permissions on the NI-VXI directory (`/usr/nivxi`) are set to allow read and write access to all users who will use NI-VXI. The files installed in your NI-VXI directory will be owned by `root`. Make no assumptions about the group setting. Remember to change the owner of the files (`chown` command) and the group (`chgrp` command) to set the appropriate permissions. Finally, be sure to add the NI-VXI directory to your path.

Note: *The VXIpc-486 uses computer resources (interrupt line, I/O space, and memory windows) that are at fixed locations. If you add a plug-in card to the VXIpc-486 that also uses these resources (such as an Ethernet card), you need to ensure against conflicts with the VXIpc-486. For information on the resources that the VXIpc-486 uses, refer to Appendix G, VXIpc-486 Hardware Configuration, in Getting Started with Your VXIpc-486 Model 500 Series. Refer to your Lynx documentation for more information on how to configure your plug-in cards.*

Chapter 3

NI-VXI Software Configuration

This chapter describes how to use the VXI resource editor program to configure the NI-VXI software for the VXIpc-486 computer. 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 `vxiiinit` from your LynxOS 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 the `vxitedit` VXI Resource Editor program.

You can use `vxitedit` to edit the description of the VXIpc-486 or any other device that is installed in the VXI 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.

Running `vxitedit`

`vxitedit` is the VXI resource editor program that runs in your LynxOS shell. You can use this program 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. This program also displays the system configuration information generated by the Resource Manager.

To run `vxitedit` 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.

To start the program, type `vxitedit`. When you select the **Configuration Editor** from the **Main** menu, three options are available for you to configure: the Logical Address, Bus, and Device Configuration options.

Type `help` to obtain online assistance within any editor, including determining which commands are applicable to the editor you are using. Type `save` to apply any changes you have made, and then type `exit`. Saving the changes updates files `vxila.cfg`, `vxibus.cfg`, and `vxidev.cfg`. `vxiiinit.exe` uses these files to update the soft-configured settings, including 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 typing `abort`. When you return to the main menu, you can enter another editor by selecting the number for that editor.

For further instructions on the use of the `vxitedit` program, refer to the *NI-VXI Text Utilities Reference Manual*. The following sections describe the configuration options available under the three editors.

Logical Address Configuration

The following paragraphs describe options available under the Logical Address Configuration Editor.

Logical Address

The logical address is an 8-bit number that uniquely identifies the VXIpc-486. It can be configured as a static device with a logical address range from 0 through 255. Logical address 255 configures the VXIpc-486 as a dynamic device, as specified in the *VXIbus System Specification*. The default logical address is 0. 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 options are extended device (EXT), message-based device (MBD), and register-based device (RBD). The default is MBD.

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 as an A16 only, A16/A24, or A16/A32 device. The default is A16 only.

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 `DISABLE`.

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.
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, and slave accesses to this RAM are in the primary byte order.

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 0, and cannot be changed.

Slave Supervisory Access Only

When YES is selected, the VXIpc-486 will acknowledge only supervisory accesses. Nonprivileged accesses will result in a bus error. This is true for all A16, A24, and A32 slave accesses. When NO is selected, the VXIpc-486 will acknowledge both supervisory and nonprivileged accesses. The default option is NO.

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. The allowable range is from 0 to 65535 seconds. The default value is 5 s.

VXI Shared RAM (Primary Byte Order)

The local RAM of the Model 500 Series VXIpc-486 can be slave-accessed in either A24 or A32 space in D08, D16, and D32 data transfer modes. This field, which indicates the primary and secondary byte swapping schemes for slave accesses to the local RAM space, can be set to either Intel (Little Endian) or Motorola (Big Endian). The default primary byte ordering is Intel for backward compatibility reasons. When the addressing mode is A16 only, this field is ignored.

Bus Configuration

The following paragraphs describe options available under the Bus Configuration Editor.

Bus Request Level

This field sets the VXIbus request level (BR0-3) for the VXIpc-486. The default value is level 3.

Fair Requester Timeout

You can configure the VXIpc-486 as a fair requester (Request On No Request) with a timeout period of either infinity or within the range of 2 to 28 μ s (increments of 2 μ s), or you can configure it as an unfair requester by specifying 0. The default value specifies an infinite timeout period.

Bus Release Mechanism

This field sets the VXIbus release mechanism, which can be Release On Request (ROR) or Release When Done (RWD). The default value is ROR.

Bus Arbitration

The field sets the VXIbus arbitration mode. The two options are Priority (PRI) or Round Robin Select (RRS). The default value is PRI. The bus arbitration mode is relevant only when the VXIpc-486 is the Slot 0 device.

Local Bus Timeout

This field sets the timeout period for the VXIpc-486 local bus. You can set the timeout within the range from 4 to 512 μ s, in powers of 2 (4, 8, 16, 32, 64, 128, 256, and 512), or you can disable this function. When the VXIpc-486 is not the System Controller, you should select 0 to disable the timeout. In such a case, you should have configured the VXIpc-486 as a Non-Slot 0 device as described in *Getting Started with Your VXIpc-486 Model 500 Series*. The default value is 32 μ s.

VXIbus Timeout

This field sets the timeout period for the VXIbus bus timeout (BTO) unit. When the VXIpc-486 is the System Controller, you can set the BTO period within the range from 4 to 512 μ s in powers of 2 (4, 8, 16, 32, 64, 128, 256, and 512). When the VXIpc-486 is not the System Controller, you should select 0 to disable the timeout. In such a case, you should have configured the VXIpc-486 as a Non-Slot 0 device as described in *Getting Started with Your VXIpc-486 Model 500 Series*. The default value is 128 μ s.

Bus Acquisition Time in Timeout

This field determines whether the arbitration time to acquire the VXIbus is included in the local bus timeout. In embedded-only systems, the default is to include bus acquisition time. In MXI systems, however, the arbitration time may be excessively long. In such a case, it may be preferable to exclude the bus acquisition time from the overall timeout.

Device Configuration

The following paragraphs describe options available under the Device Configuration Editor.

SYSRESET Resets PC

When YES is selected, a SYSRESET on the backplane causes the VXIpc-486 to reboot. When NO is selected, a SYSRESET on the backplane does not reboot the VXIpc-486. The default value for this field is 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. You can set the Servant area size within the range 0 to 255. The default value is 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).

Note: *If the VXIpc-486 is the Resource Manager, this setting is irrelevant.*

Number of Handlers

This field gives the number of interrupt handlers that the VXIpc-486 supports, which can range from 0 to 7. The default value is 1.

Number of Interrupters

This field gives the number of interrupters that the VXIpc-486 supports, which can range from 0 to 7. The default value is 0.

Use System Sanity Timer

When YES is selected, if a misbehaved NI-VXI application causes the system to hang, the VXIpc-486 asserts the SYSFAIL* signal on the backplane. When NO is selected, if the system hangs, the VXIpc-486 does not assert SYSFAIL*. In general, the sanity timer is useful by alerting you that your application has gone awry. However, you should turn off the sanity timer when debugging an NI-VXI application because some debuggers conflict with the sanity timer mechanism. The default value for this field is YES.

Word Serial Poll Value

This field indicates the number of times the driver will loop checking for *Write Ready* and *Read Ready* for writing a command to the Data Low register. The driver sleeps (yields the CPU) at regular intervals during polling, which further degrades the Word Serial transfer rate. You may want to make the value high in a uni-process system so that the driver never sleeps during polling, giving the maximum Word Serial transfer rate. We recommend a low value in a multi-process system to allow other processes CPU time. The default value is 120.

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, the program prompts you to save your changes before exiting the menu. Saving the configuration information updates the configuration files `vxila.cfg`, `vxibus.cfg`, and `vxidev.cfg`. Run `vxiiinit.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 LynxOS

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

After you have verified that the driver software has been installed successfully, you can begin to develop your VXI application software. Be sure to check the `readme.first` file for the latest application development notes. Remember that you must run the `vxiiinit` 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

Use the VXI Text Interactive Control (`victext`) program to communicate with VXI devices through commands you enter at the keyboard. This feature helps you learn how to communicate with devices, troubleshoot problems, and develop your application.

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

The `victext` utility is 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.

For details on how to use the `victext` program, refer to the *NI-VXI Text Utilities Reference Manual*.

Example Programs

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

Compiling and Linking

To compile and link your application `appl.c` so that it can use the NI-VXI driver, enter the following commands.

```
gcc -c appl.c
gcc -o appl appl.o cvxi.o
```

The `cvxi.c` file is the source code for the object file `cvxi.o` and can be used to recreate `cvxi.o`, if necessary. Notice that you should use the GNU `cc` compiler (`gcc`) to create NI-VXI applications.

Multiple Applications Using the NI-VXI Library

Multi-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.

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()`. This guarantees your application exclusive use of the window. 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 and/or macros 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 and/or macros to access memory. See Chapter 6, *Low-Level VXIbus Access Functions*, in the *NI-VXI Software Reference Manual for C* for more information.

Local Resource Access Functions

By using `vxitedit`, it is possible to have 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 in `vxitedit` to share the local resources. However, when LynxOS is asked for a block of contiguous memory, which is needed for `VXImemAlloc()` to work, the memory returned is often above 4 MB. It is therefore recommended that you always share the maximum available memory when sharing memory to ensure that `VXImemAlloc()` will succeed.

See Chapter 8, *Local Resource Access Functions*, in the *NI-VXI Software Reference Manual for C* for more information.

Using Callbacks from NI-VXI

When an interrupt occurs (VXI interrupt, VXI signal, bus error, and so on) the kernel driver performs basic interrupt handling and then reports the event back to the user application (if the application is enabled to detect them) as a UNIX signal. It is important to remember that the user handler is *not* invoked at interrupt time. Since the user handler is actually a signal handler, it is invoked whenever the LynxOS scheduler decides. Therefore, if you require very low interrupt latency, it is vital that you adjust the priority of your process accordingly.

Errors Closing the VXI Library

A `CloseVXIlibrary()` call that returns a failure may have partially freed some resources used by the program. For this reason, it is important in such a case that you do not continue using resources, such as the master window or shared memory windows, since we cannot guarantee their status. A failure of a `CloseVXIlibrary()` is uncommon and means a more serious system failure. Please contact National Instruments for assistance.

Appendix A

NI-VXI Software Overview

This appendix lists all the programs and files that make up the NI-VXI software.

Main Programs and Files

The main programs and files of the NI-VXI software package are found in the NI-VXI subdirectory, `$NIVXIPATH/bin`.

Note: *The \$ in `$NIVXIPATH` indicates that it is an environment variable.*

- `vxinit` is the VXIpc-486 initialization program. This program removes the **FAILED** LED and initializes the board interrupts, shared RAM, VXI register configurations, and bus configurations. It is typically included in the shell configuration so that the VXIpc-486 is automatically initialized at startup. You can modify the configuration settings using the `vxitedit` program.
- `resman` is the National Instruments multimainframe Resource Manager. Execute `resman` only after running `vxinit`. You must also run `resman` every time the mainframe is reset.
- `victext` is the text-based interactive control program that executes functions you enter from the keyboard. It helps you learn the functions, program your VXI device, and develop and debug your application program. This program is described in detail in the *NI-VXI Text Utilities Reference Manual*.
- `vxitedit` is the text-based VXI Resource Editor program you use to edit system and device information. 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 Text Utilities Reference Manual*.

The following files are in the root level of the NI-VXI directory:

- `readme.first` contains the latest updates and corrections to the manual when appropriate.
- `buildvxi` is a script used to install the NI-VXI driver during installation.

Additional Programs and Files

The `$NIVXIPATH/hlp` directory contains various help files used by the `vxitedit` program.

The `$NIVXIPATH/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.

The `$NIVXIPATH/INCLUDE` directory contains files for creating NI-VXI applications.

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.

- `vxint.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.

The NI-VXI subdirectory `$NIVXIPATH/src` contains the NI-VXI object file, `cvxi.o`, that you need when creating a NI-VXI application. This directory also includes the `cvxi.c` source file for `cvxi.o`, which you will need if you ever have to recreate the object file. A makefile is provided to recompile the file.

The NI-VXI subdirectory `$NIVXIPATH/example` contains a set of sample source files that use a variety of NI-VXI functions. This example application is not intended to actually execute a program, but rather to present code segments that can illustrate how to use the various NI-VXI functions.

The NI-VXI subdirectory `$NIVXIPATH/sys` contains a variety of files that are necessary to create the NI-VXI driver in the LynxOS kernel. It is not expected that any user should need to access these files directly, but only through the `buildvxi` script file.

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 `victext`. At the prompt type `ver`, and the program will display the versions of `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 number revision of your VXIpc-486, only the letter revision.

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

See above.

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

You must use the VXI Resource Editor program, `vxitedit`, to configure the VXIpc-486. You will find this program in the NI-VXI subdirectory `$NIVXIPATH/bin`.

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

You must use the hardware initialization program, `vxiiinit`, to initialize the VXIpc-486. You will find this program in the NI-VXI subdirectory `$NIVXIPATH/bin`. `vxiiinit` uses the settings in the Configuration Editor of the VXI Resource Editor program, `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 NI-VXI subdirectory `$NIVXIPATH/bin` directory. `resman` uses the settings in the Configuration Editor of the VXI Resource Editor program, `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 NI-VXI 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 `vixtext` 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.

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.
2. Run the VXI Resource Editor program, `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 `vxiiinit` 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 do not 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
<i>Message-Based Device</i>	<i>MBD</i>
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.

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
```

This creates a file `resman.out` in the TBL subdirectory of the NI-VXI directory.

`VXImemAlloc()` returns errors whenever I try to execute it.

`VXImemAlloc()` requires that the VXIpc-486 has been configured to share its local memory with the VXIbus. This is accomplished by using `vxitedit` to set the share options (See Chapter 3, *NI-VXI Software Configuration*) and `vxiiinit` to initialize the hardware. In addition, you should also run `resman` to make sure that the memory has been allocated a VXIbus address.

If you have done all of this and you still receive errors, you may not have shared enough local memory. Sharing local memory through `VXImemAlloc()` requires that the memory be contiguous under the virtual memory system running in LynxOS. When LynxOS returns a block of contiguous memory, it is usually above the 4 MB boundary of your local RAM. Therefore, if you have decided to share only 4 MB of memory and errors occur, you should try increasing the amount of memory you share.

Remember that sharing memory by using `vxitedit` and `vxiiinit` as described above does not use up LynxOS system resources. Only after you use `VXImemAlloc()` to request that memory space are LynxOS system resources used.

Finally, if you are sharing enough memory but you still receive errors, try using `VXImemAlloc()` to request a single byte of memory. If this succeeds, the amount you requested previously was more than LynxOS was able to provide. Try reducing the amount of your memory request. If you cannot allocate even a single byte of shared memory, please contact National Instruments for assistance.

What does `vxiiinit` mean by File `filename` not found?

When `vxiiinit` runs, it attempts to access some of the configuration or table files, such as `vxila.cfg`, `vxibus.cfg`, and `vxidev.cfg`. If `vxiiinit` 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. Make sure that the user environment variable `NIVXIPATH` points to the right directory.

What does `vxiiinit` 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.

`vxiiinit` creates a file `vxiiinit.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 write permissions of the NI-VXI directory and/or its subdirectories do not give you write access.

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* for the resources that the VXIpc-486 uses.

When my computer boots, it reports that it can not 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.*

My Ethernet/SCSI/Plug-in card doesn't work or stops working after I run `vxiinit`.

The VXIpc-486 uses computer resources (interrupt line, I/O space, and memory windows) that are at fixed locations. If you add a plug-in card to the VXIpc-486 that also uses these resources (such as an Ethernet card), the conflict will cause the VXIpc-486, the plug-in card, or both to stop working or cause unpredictable results. For details of what resources are used by the VXIpc-486, refer to Appendix G, *VXIpc-486 Hardware Configuration*, in *Getting Started with your VXIpc-486 Model 500 Series*. Refer to your Lynx documentation for more information on how to configure your plug-in cards.

Appendix D

Customer Communication

For your convenience, this appendix contains forms you can use to request help from National Instruments. 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 9 879 9422	03 9 879 9179
Austria	0662 45 79 90 0	0662 45 79 90 19
Belgium	02 757 00 20	02 757 03 11
Canada (Ontario)	519 622 9310	519 622 9311
Canada (Quebec)	514 694 8521	514 694 4399
Denmark	45 76 26 00	45 76 71 11
Finland	90 527 2321	90 502 2930
France	1 48 14 24 24	1 48 14 24 14
Germany	089 741 31 30	089 714 60 35
Hong Kong	2645 3186	2686 8505
Italy	02 48301892	02 48301915
Japan	03 5472 2970	03 5472 2977
Korea	02 596 7456	02 596 7455
Mexico	95 800 010 0793	5 520 3282
Netherlands	0348 433466	0348 430673
Norway	32 84 84 00	32 84 86 00
Singapore	2265886	2265887
Spain	91 640 0085	91 640 0533
Sweden	08 730 49 70	08 730 43 70
Switzerland	056 200 51 51	056 200 51 55
Taiwan	02 377 1200	02 737 4644
U.K.	01635 523545	01635 523154

Technical Support Form

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

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

Software Information

- NI-488.2M Software for LynxOS
Version _____
- NI-VXI for VXIpc-486 and LynxOS
Version _____
- Logical Address (VXIla) Configuration Editor
 - Logical Address _____
 - Device Type _____
 - Address Space _____
 - VXI Shared RAM (Mbytes) _____
 - Slave Block Transfer Mode _____
 - Slave Supervisory Access Only _____
 - Resource Manager Delay _____
 - VXI Shared RAM (Primary Byte Order) _____
- Bus Configuration
 - Bus Request Level _____
 - Fair Requester Timeout (μ s) _____
 - Bus Release Mechanism _____
 - Bus Arbitration _____
 - Local Bus Timeout (μ s) _____
 - VXIbus Timeout (μ s) _____
 - Bus Acquisition Time in Timeout _____
- Device Configuration
 - SYSRESET Resets PC _____
 - Servant Area Size _____
 - Protocol Register _____
 - Read Protocol Response _____
 - Number of Handlers _____
 - Number of Interrupters _____
 - Use System Sanity Timer _____
 - Word Serial Poll Value _____

Other Products

- LynxOS Version _____
- 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 500 Series and LynxOS**

Edition Date: **September 1995**

Part Number: **321006A-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

A

- A16 space** VXIbus address space equivalent to the VME 64 KB *short* address space. In VXI, the upper 16 KB of A16 space is allocated for use by VXI devices configuration registers. This 16 KB region is referred to as VXI configuration space.
- A24 space** VXIbus address space equivalent to the VME 16 MB *standard* address space.
- A32 space** VXIbus address space equivalent to the VME 4 GB *extended* address space.
- access permissions** The file security system used by UNIX operating systems.
- ANSI** American National Standards Institute

B

- B** bytes
- 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.
- 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

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

D

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

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.

environment variable A variable used by the shell.

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.

G

GPIB General Purpose Interface Bus; the industry-standard IEEE 488 bus.

I

IEEE Institute of Electrical and Electronics Engineers

I/O input/output; the techniques, media, and devices used to achieve communication between entities.

Intel (Little Endion) Byte Swapping. See *byte order*.

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

kernel The main component of the operating system.

L

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 $C000h + \text{Logical Address} * 40h$.

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.
Motorola (Big Endian)	Byte Swapping. See <i>byte order</i> .

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.

O

object file	A binary file which in the output of a compiler, but which has not been through a linker yet.
-------------	---

P

PC AT	Personal Computer Advanced Technology
PRI	Priority
privileged access	See <i>supervisory access</i> .

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 <i>Resource Manager</i> .
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
script	A text file containing commands to be run by the shell.
Servant	A device controlled by a Commander.
shell	A process executed by the operating system to allow interacting control of the operating system.
statically configured device	A device whose logical address cannot be set through software; that is, it is not dynamically configurable.
super-user	The user with full access to all files on the system.
supervisory access	One of the defined types of VMEbus data transfers; indicated by certain address modifier codes.
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

trigger Either TTL or ECL lines used for intermodule communication.

V

victext VXI Interactive Control program, a part of the NI-VXI bus interface software package. Used to program VXI devices, and develop and debug VXI application programs.

VME Versa Module Eurocard or IEEE 1014

VXIbus VMEbus Extensions for Instrumentation

vxiiinit 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.

vxitedit VXI Resource Editor program, a part of the NI-VXI bus interface software package. Used to configure the system, edit the manufacturer name and ID numbers, edit the model names of VXI and non-VXI devices in the system, as well as the system interrupt configuration information, and display the system configuration information generated by the Resource Manager.

W

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

access permissions, 2-3
address space field, logical address
configuration, 3-2

B

bus configuration, 3-4 to 3-5
bus acquisition time in timeout, 3-5
bus arbitration, 3-5
bus release mechanism, 3-5
bus request level, 3-4
fair requester timeout, 3-4
local bus timeout, 3-5
VXIbus timeout, 3-5

C

callbacks from NI-VXI, 4-3
CloseVXIlibrary function, 4-3
common questions about NI-VXI software,
B-1 to B-3. *See also* troubleshooting.
compiling and linking programs, 4-2
configuration
bus configuration, 3-4 to 3-5
bus acquisition time in timeout, 3-5
bus arbitration, 3-5
bus release mechanism, 3-5
bus request level, 3-4
fair requester timeout, 3-4
local bus timeout, 3-5
VXIbus timeout, 3-5
device configuration, 3-5 to 3-7
number of handlers, 3-6
number of interrupters, 3-6
Protocol register, 3-6
read protocol response, 3-6
servant area size, 3-6
SYSRESET resets PC, 3-5
use system sanity timer, 3-6
word serial poll value, 3-7

exiting vxitedit and reinitializing
hardware, 3-7

logical address configuration, 3-2 to 3-4
address space, 3-2
device type, 3-2
logical address, 3-2
Resource Manager delay, 3-4
slave block transfer mode, 3-4
slave supervisory access only, 3-4
VXI shared RAM (MBytes), 3-2
VXI shared RAM (primary byte
order), 3-4
VXI shared RAM options
(table), 3-3
running vxitedit, 3-1 to 3-2
saving changes, 3-1

configuration files, error messages related
to, C-2
conflicts with plug-in cards, 2-3, C-3
customer communication, ix, D-1

D

device configuration, 3-5 to 3-7
number of handlers, 3-6
number of interrupters, 3-6
Protocol register, 3-6
read protocol response, 3-6
servant area size, 3-6
SYSRESET resets PC, 3-5
use system sanity timer, 3-6
word serial poll value, 3-7
device type field, logical address
configuration, 3-2
documentation
conventions used in manual, ix
how to use documentation set, vii
how to use this manual, 1-2
organization of manual, viii
related documentation, ix

E

environment variable for NI-VXI software
including with every login, 2-3
setting up, 2-1

error messages. *See* troubleshooting.
errors closing VXI library, 4-3
example programs, 4-2

F

fair requester timeout field, bus
configuration, 3-4
files in NI-VXI software, A-1 to A-3
fixed disk error messages, C-2 to C-3

G

getting started with NI-VXI software, 1-2

H

hard disk drive error messages, C-2 to C-3
hardware, reinitializing after
configuration, 3-7

I

installation of NI-VXI software, 2-1 to 2-3
extracting NI-VXI software, 2-2
installing NI-VXI driver into LynxOS
kernel, 2-2 to 2-3
preparing for, 2-1
super-user privileges required (note), 2-1
system configuration, 2-3
interrupt handlers, specifying number of, 3-6
interrupt handling under LynxOS, 4-3
interrupters, specifying number of, 3-6

L

linking programs, 4-2

local bus timeout field, bus
configuration, 3-5
local resource access functions, 4-3
logical address configuration, 3-2 to 3-4
address space, 3-2
device type, 3-2
logical address, 3-2
Resource Manager delay, 3-4
slave block transfer mode, 3-4
slave supervisory access only, 3-4
VXI shared RAM (MBytes), 3-2
VXI shared RAM (primary byte
order), 3-4
VXI shared RAM options (table), 3-3
low-level access functions, 4-3
LynxOS kernel, installing NI-VXI driver
into, 2-2 to 2-3

M

manual. *See* documentation.
MapVXIAddress function, 4-2, 4-3
memory, sharing, 4-3. *See also* VXI shared
RAM (MBytes) field.
multiple applications using NI-VXI
library, 4-2

N

NI-488.2M software, 1-3
NI-VXI library
errors closing VXI library, 4-3
multi-application support, 4-2
NI-VXI software
common questions about NI-VXI
software, B-1 to B-3
configuring. *See* configuration.
getting started, 1-2
installation, 2-1 to 2-3
extracting NI-VXI software, 2-2
installing NI-VXI driver into
LynxOS kernel, 2-2 to 2-3
preparing for, 2-1
super-user privileges required
(note), 2-1
system configuration, 2-3
overview, 1-3

- programming. *See* programming.
- programs and files
 - additional, A-2 to A-3
 - primary, A-1
 - version, determining, B-1
- NIVXIPATH environment variable
 - including with every login, 2-1
 - setting up, 2-1
- number of handlers field, device
 - configuration, 3-6
- number of interrupters field, device
 - configuration, 3-6

P

- permissions, 2-3
- plug-in cards, conflicts with, 2-3, C-3
- programming
 - compiling and linking, 4-2
 - errors closing VXI library, 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
 - multiple applications using NI-VXI library, 4-2
 - using callbacks from NI-VXI, 4-3
- programs and files in NI-VXI software
 - additional programs and files, A-2 to A-3
 - common questions about NI-VXI software, B-1
 - main programs and files, A-1 to A-2
- Protocol register field, device
 - configuration, 3-6

Q

- questions about NI-VXI software, B-1 to B-3. *See also* troubleshooting.

R

- RAM. *See* VXI shared RAM (MBytes) field.
- read protocol response field, device
 - configuration, 3-6
- reinitializing hardware after

- configuration, 3-7
- resman program
 - location in \$NIVXIPATH/bin directory, B-1
 - overview, A-1
 - testing your system, B-2
- Resource Manager, preventing VXIpc-486 from becoming, B-2 to B-3
- Resource Manager delay field, logical address configuration, 3-4
- resources used by VXIpc-486, 2-3, C-3

S

- servant area size field, device
 - configuration, 3-6
- shared RAM. *See* VXI shared RAM (MBytes) field.
- slave block transfer mode field, logical address configuration, 3-4
- slave supervisory access only field, logical address configuration, 3-4
- super-user privileges for NI-VXI software installation (note), 2-1
- SYSRESET resets PC field, device
 - configuration, 3-5
- system, testing, B-2
- system sanity timer, selecting, 3-6

T

- technical support
 - address, phone, and fax numbers, D-1
 - information required before calling technical support, C-1
- troubleshooting
 - common questions about NI-VXI software, B-1 to B-3
 - Error writing configuration to file error message, C-2
 - failure of VXIpc-486 with ISA bus plug-in cards installed, C-2
 - File *filename* not found error message, C-2

- fixed disk error messages, C-2 to C-3
- information required before calling technical support, C-1
- VXImemAlloc returns errors on attempted execution, C-1

U

- UnMapVXIAddress function, 4-2
- use system sanity timer field, device configuration, 3-6

V

- version of NI-VXI software, determining, B-1
- version of VXIpc-486 hardware, determining, B-1
- victext utility
 - command set, 4-1
 - overview, A-1
 - purpose, 4-1
 - testing devices, B-2
- VXI library, errors closing, 4-3
- VXI shared RAM (MBytes) field
 - description, 3-2
 - options (table), 3-3
- VXI shared RAM (primary byte order) field, logical address configuration, 3-4
- VXI Text Interactive Control (victext) program, 4-1, A-1
- VXIbus timeout field, bus configuration, 3-5
- vxiiinit program
 - location in \$NIVXIPATH/bin directory, B-1
 - overview, A-1
 - running every time LynxOS is booted (note), 2-3
- VXImemAlloc function
 - errors on attempted execution, C-1
 - sharing memory for, 4-3
- VXIpc-486 hardware
 - determining version, B-1
 - failure to work with ISA bus plug-in cards installed, C-2
 - installing in slot other than Slot 0, B-2
 - overview, 1-1

- preventing from becoming Resource Manager, B-2 to B-3
- VXIpeek function, 4-2, 4-3
- VXIpoke function, 4-2, 4-3
- vxitedit program. *See also* configuration.
 - exiting, 3-7
 - location in \$NIVXIPATH/bin directory, B-1
 - overview, A-1
 - running, 3-1 to 3-2
 - saving changes, 3-1

W

- word serial poll value field, device configuration, 3-7