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 PCI-MXI-2* at our website: *Click HERE*

MXI™

Getting Started with Your PCI-Based MXI-2 Interface for Windows

August 2002 Edition
Part Number 321712C-01

Worldwide Technical Support and Product Information

ni.com

National Instruments Corporate Headquarters

11500 North Mopac Expressway Austin, Texas 78759-3504 USA Tel: 512 683 0100

Worldwide Offices

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 55 11 3262 3599,
Canada (Calgary) 403 274 9391, Canada (Montreal) 514 288 5722, Canada (Ottawa) 613 233 5949,
Canada (Québec) 514 694 8521, Canada (Toronto) 905 785 0085, China 86 21 6555 7838,
Czech Republic 02 2423 5774, Denmark 45 76 26 00, Finland 09 725 725 11, France 01 48 14 24 24,
Germany 089 741 31 30, Greece 01 42 96 427, Hong Kong 2645 3186, India 91 80 4190000,
Israel 03 6393737, Italy 02 413091, Japan 03 5472 2970, Korea 02 3451 3400, Malaysia 603 9596711,
Mexico 001 800 010 0793, Netherlands 0348 433466, New Zealand 09 914 0488, Norway 32 27 73 00,
Poland 22 3390 150, Portugal 210 311 210, Russia 095 238 7139, Singapore 65 6 226 5886,
Slovenia 3 425 4200, South Africa 11 805 8197, Spain 91 640 0085, Sweden 08 587 895 00,
Switzerland 056 200 51 51, Taiwan 02 2528 7227, United Kingdom 01635 523545

For further support information, see the *Technical Support and Professional Services* appendix. To comment on the documentation, send email to techpubs@ni.com.

© 1997–2002 National Instruments Corporation. All rights reserved.

Important Information

Warranty

The MXIbus boards and accessories are warranted against defects in materials and workmanship for a period of one year from the date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

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 document 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

CVI™, LabVIEW™, Measurement Studio™, MITE™, MXI™, National Instruments™, NI™, NI-488™, NI-488.2™, ni.com™, NI-DAQ™, NI-VISA™, NI-VXI™, and VXIpc™ are trademarks of National Instruments Corporation.

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

Patents

For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your CD, or `ni.com/patents`.

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

Compliance

FCC/Canada Radio Frequency Interference Compliance

Determining FCC Class

The Federal Communications Commission (FCC) has rules to protect wireless communications from interference. The FCC places digital electronics into two classes. These classes are known as Class A (for use in industrial-commercial locations only) or Class B (for use in residential or commercial locations). Depending on where it is operated, this product could be subject to restrictions in the FCC rules. (In Canada, the Department of Communications (DOC), of Industry Canada, regulates wireless interference in much the same way.)

Digital electronics emit weak signals during normal operation that can affect radio, television, or other wireless products. By examining the product you purchased, you can determine the FCC Class and therefore which of the two FCC/DOC Warnings apply in the following sections. (Some products may not be labeled at all for FCC; if so, the reader should then assume these are Class A devices.)

FCC Class A products only display a simple warning statement of one paragraph in length regarding interference and undesired operation. Most of our products are FCC Class A. The FCC rules have restrictions regarding the locations where FCC Class A products can be operated.

FCC Class B products display either a FCC ID code, starting with the letters **EXN**, or the FCC Class B compliance mark that appears as shown here on the right.



Consult the FCC Web site at `http://www.fcc.gov` for more information.

FCC/DOC Warnings

This equipment generates and uses radio frequency energy and, if not installed and used in strict accordance with the instructions in this manual and the CE Mark Declaration of Conformity*, may cause interference to radio and television reception. Classification requirements are the same for the Federal Communications Commission (FCC) and the Canadian Department of Communications (DOC).

Changes or modifications not expressly approved by National Instruments could void the user's authority to operate the equipment under the FCC Rules.

Class A

Federal Communications Commission

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Canadian Department of Communications

This Class A digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulations.

Cet appareil numérique de la classe A respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

Class B

Federal Communications Commission

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Canadian Department of Communications

This Class B digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulations.

Cet appareil numérique de la classe B respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

Compliance to EU Directives

Readers in the European Union (EU) must refer to the Manufacturer's Declaration of Conformity (DoC) for information* pertaining to the CE Mark compliance scheme. The Manufacturer includes a DoC for most every hardware product except for those bought for OEMs, if also available from an original manufacturer that also markets in the EU, or where compliance is not required as for electrically benign apparatus or cables.

To obtain the DoC for this product, click **Declaration of Conformity** at `ni.com/hardref.nsf/`. This Web site lists the DoCs by product family. Select the appropriate product family, followed by your product, and a link to the DoC appears in Adobe Acrobat format. Click the Acrobat icon to download or read the DoC.

* The CE Mark Declaration of Conformity will contain important supplementary information and instructions for the user or installer.

Contents

About This Manual

How to Use This Documentation Set x
Conventions xi
Related Documentation xii

Chapter 1
Introduction

How to Use This Manual 1-1
What You Need to Get Started 1-2
PCI-Based MXI-2 Interface Kit Overview 1-2
Hardware Description 1-3
VME Users 1-4
Software Description 1-4
National Instruments Application Software 1-5

Chapter 2
Setup

Configuring the Hardware 2-1
Installing the Hardware 2-1
Installing Your PCI-Based MXI-2 Interface 2-1
Installing Your Mainframe Extender 2-2
Connecting the MXI-2 Cable Properly 2-3
Software Installation *2-3*
Installing the Software 2-4
Completing the Software Installation 2-5
Verifying Your System Configuration 2-5

Chapter 3
Developing Your Application

NI-VXI, NI-VISA, and Related Terms 3-1
Configuration 3-2
Device Interaction 3-3
Programming for VXI 3-6
Optimizing Large VXIbus Transfers 3-7
Shared Memory 3-8

NI-VXI API Notes 3-8
Compiler Symbols 3-8
Compatibility Layer Options 3-9
Debugging 3-10

Appendix A
Default Settings

Appendix B
Common Questions

Appendix C
Technical Support and Professional Services

Glossary

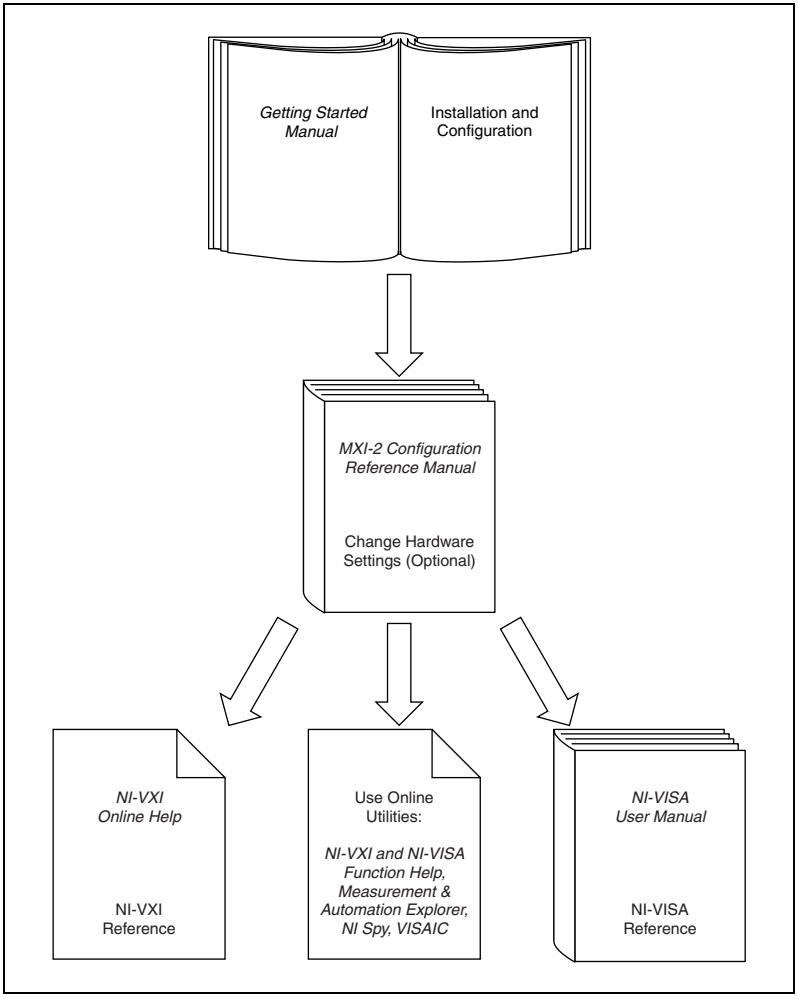
Index

About This Manual

This manual contains instructions for installing and configuring the National Instruments PCI-based MXI-2 interface kit for Windows. This manual is meant to be used with the *MXI-2 Configuration Reference Manual*.

Your kit contains either a VXI-MXI-2 or VME-MXI-2, which plugs into your VXI or VME mainframe and links your computer to the VXIbus or VMEbus, respectively. The kit also contains a PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320 interface board, which links your PCI-based or PXI/CompactPCI computer to the MXIbus. Your software consists of the NI-VXI bus interface software, which is fully VXI*plug&play* compliant, and the NI-VISA API, which is the National Instruments implementation of the VISA I/O software standard on which all VXI*plug&play* software components are based.

How to Use This Documentation Set



This getting started manual contains an overview of the MXI-2 hardware and the NI-VXI/NI-VISA software, guides you through setting up your kit, and helps you get started with application development. You also can use this manual as a reference for the hardware and software default settings and to find the answers for commonly asked questions.

The *MXI-2 Configuration Reference Manual* contains information on configuring, installing, and cabling your MXI-2 hardware. Use this manual in conjunction with the getting started manual.

When you have successfully set up your system, you can begin to develop applications in NI-VISA and/or NI-VXI. Refer to the *NI-VISA User Manual* to learn about VISA and how to use it in your system. The NI-VISA help describes the attributes, events, and operations you can use in NI-VISA. The user manual is available in the `c:\VXIpnp\os\NIvisa\manuals` directory (where *os* corresponds to the version of Windows you are using) under the name `NIVISAUM.pdf`. Use Acrobat Reader 3.0 or later to open this file.

The NI-VXI help presents the concepts of VXI and detailed explanations of the NI-VXI functions. Study the descriptions of each function in the help to fully understand the purpose and syntax of each function. You also can access the NI-VXI help in the `NIVXI` folder.

Conventions

The following conventions appear in this manual:

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



This icon denotes a note, which alerts you to important information.



This icon denotes a caution, which advises you of precautions to take to avoid injury, data loss, or a system crash.

bold Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter and LED names.

italic Italic text denotes variables, emphasis, a cross reference, an introduction to a key concept, or text as it appears on a hardware label. This font also denotes text that is a placeholder for a word or value that you must supply.

`monospace` Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames and extensions, and code excerpts.

monospace bold Bold text in this font denotes the messages and responses that the computer automatically prints to the screen. This font also emphasizes lines of code that are different from the other examples.

monospace italic Italic text in this font denotes text that is a placeholder for a word or value that you must supply.

Related Documentation

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

- ANSI/IEEE Standard 1014-1987, *IEEE Standard for a Versatile Backplane Bus: VMEbus*
- ANSI/IEEE Standard 1155-1993, *IEEE VMEbus Extensions for Instrumentation: VXIbus*
- ANSI/VITA 1-1994, *VME64*
- *CompactPCI Specification*, Revision 2.0, PCI Industrial Computers Manufacturers Group
- *Multisystem Extension Interface Bus Specification*, Version 2.0, National Instruments Corporation
- *PCI Local Bus Specification*, Revision 2.1, PCI Special Interest Group
- *PXI Specification*, Revision 2.0, National Instruments Corporation
- *VME-MXI-2 User Manual*, National Instruments Corporation
- *VXI-MXI-2 User Manual*, National Instruments Corporation
- VXI-6, *VXIbus Mainframe Extender Specification*, Rev. 1.0, VXIbus Consortium

1

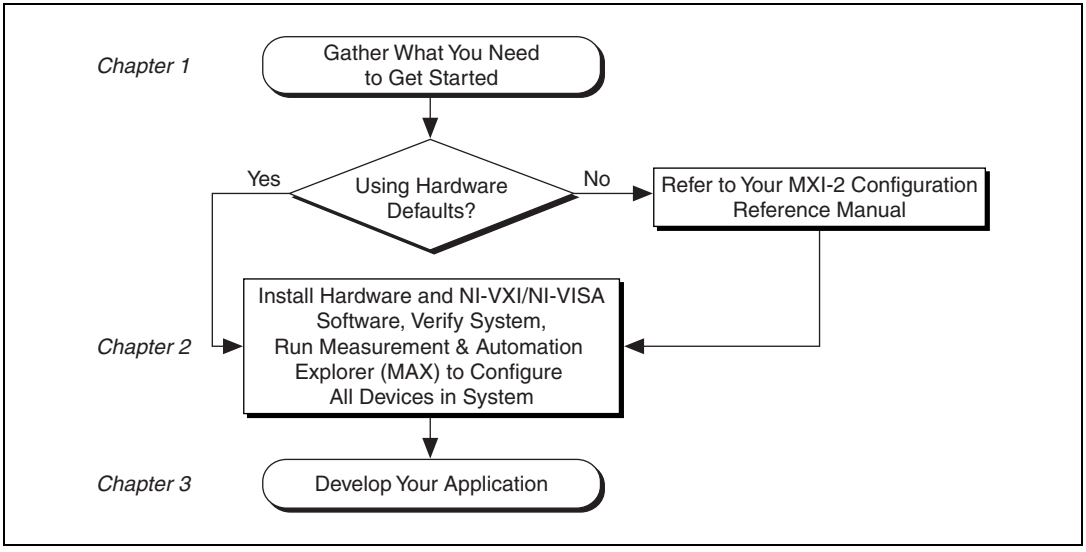
Introduction

This chapter describes your MXI-2 interface kit, lists what you need to get started, and includes a brief description of the hardware and software.

This manual uses the term *VXI/VME-MXI-2* when information applies to either the VXI-MXI-2 or the VME-MXI-2. This manual also uses the term *NI-VXI/NI-VISA* when software information applies to both NI-VISA and NI-VXI.

How to Use This Manual

The following flowchart shows where to turn for more details on configuring and using the hardware and software.



Notice that for hardware configuration you should refer to the *MXI-2 Configuration Reference Manual* for a full description of how to configure and install the MXI-2 boards and an overview of MXI-2 itself. Then return to this manual for more information.

What You Need to Get Started

You need the following items to set up your VXI or VME system:

- ❑ A PCI-based computer or PXI/CompactPCI chassis running Windows
- ❑ VXIbus or VMEbus mainframe
- ❑ PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320 interface board
- ❑ VXI-MXI-2, VXI-MXI-2/B, or VME-MXI-2 interface module
- ❑ MXI-2 cable
- ❑ National Instruments software media

PCI-Based MXI-2 Interface Kit Overview

The interface kits described in this manual link a PCI-based computer or a PXI or CompactPCI chassis directly to the VXIbus or VMEbus using the high-speed Multisystem eXtension Interface bus (MXI-2). The MXI-2 kits, which include the NI-VXI/NI-VISA software for Windows, are as follows:

- VXI-PCI8000, containing a PCI-MXI-2 or PCI-MXI-2 Universal and a C- or B-size VXI-MXI-2
- VXI-PXI8000, containing a PXI-8320 and a C- or B-size VXI-MXI-2
- VME-PCI8000, containing a PCI-MXI-2 or PCI-MXI-2 Universal and a VME-MXI-2
- VME-PXI8000, containing a PXI-8320 and a VME-MXI-2

A PCI-based computer equipped with a VXI-PCI8000 interface, or a PXI/CompactPCI chassis equipped with a VXI-PXI8000 interface, can function as a VXI Commander and Resource Manager. A PCI-based computer equipped with a VME-PCI8000 interface, or a PXI/CompactPCI chassis equipped with a VME-PXI8000, can function as a VMEbus master and/or slave device. The MXI-2 interface kit makes your computer or chassis behave as if it were plugged directly into the VXI/VME backplane as an embedded CPU VXI/VME module.

The CD included with the kits contains software for Windows-based computers.

Hardware Description

The PCI-MXI-2 and PCI-MXI-2 Universal are half-size, PCI-compatible plug-in circuit boards that plug into one of the expansion slots in your PCI-based computer. The PCI-MXI-2 is for 5 V PCI systems, whereas the PCI-MXI-2 Universal is for 5 V or the newer 3.3 V PCI systems. The PXI-8320 is a 3U-size, PXI/CompactPCI-compatible circuit board that plugs into one of the peripheral slots in your PXI/CompactPCI chassis. All three boards link your computer directly to the MXIbus and vice versa.

Because the PCI-MXI-2, PCI-MXI-2 Universal, and PXI-8320 use the same communication register set that other VXIbus message-based devices use, other MXIbus devices view the board as a VXIbus device. The PCI-MXI-2, PCI-MXI-2 Universal, and PXI-8320 can also function as the MXIbus System Controller and can terminate the MXIbus signals directly. In addition, you can have up to 16 MB of onboard DRAM on your board that can be shared with the MXIbus and VXI/VMEbus and used as a dedicated data buffer.

The VXI-MXI-2 module is an extended-class, register-based VXIbus device with optional VXIbus Slot 0 capability so that it can reside in any slot in a C- or D-size chassis. Optionally, you can use the VXI-MXI-2/B, which can reside in any B-size VXIbus slot.

Note D-size VXI mainframes have connections for a P3 connector. The VXI-MXI-2, however, does not have this connector and, if configured as a Slot 0 controller, cannot provide the necessary control for VXI devices that need P3 support.

The VXI-MXI-2 uses address mapping to convert MXIbus cycles into VXIbus cycles and vice versa. By connecting to the PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320 board, the VXI-MXI-2 links the PCI bus or the PXI/CompactPCI bus to the VXIbus. The VXI-MXI-2 can automatically determine whether it is in VXI Slot 0 and/or if it is the MXIbus System Controller.

The VME-MXI-2 module is a single-slot, double-height VMEbus device with optional VMEbus System Controller functions. It uses address mapping to convert MXIbus cycles into VMEbus cycles and vice versa, just like the VXI-MXI-2. By connecting to the PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320 board, it links the PCI bus or the PXI/CompactPCI bus to the VMEbus. The VME-MXI-2 can automatically determine whether it is located in the first slot of a VMEbus chassis and if it is the MXIbus System Controller.

Also, the VXI/VME-MXI-2 automatically terminates the MXIbus if installed as the first or last device in the MXIbus. If installed in the middle of the MXIbus, the VXI/VME-MXI-2 automatically disables MXIbus termination. In addition, you can have up to 64 MB of onboard DRAM on the VXI/VME-MXI-2 module that either can be shared with the VXI/VMEbus and MXIbus or used as a dedicated data buffer.

VME Users

When used with a VXI-MXI-2, VXI Resource Manager (Resman) identifies and configures the VXI devices, including the VXI-MXI-2. When used with a VME-MXI-2, Resman configures the VME-MXI-2 to allow the PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320 to access devices in the VME chassis. Resman does not configure VME devices. The VME specification does not include the initialization and configuration procedures that the VXI specification requires.

If you want to include VME devices in your system, you can enter information about your VME devices into the Measurement & Automation Explorer (MAX) utility using the Create New wizard. Resman then can properly configure the various device-specific VME address spaces and VME interrupt lines. Notice that the VME-MXI-2 itself does conform to the VXIbus register set and is therefore configured automatically by Resman.

For each address space in which your device has memory, you must create a separate pseudo-device entry with a logical address between 256 and 511. For example, a VME device with memory in both A24 and A32 spaces would require two entries. You also can specify which interrupt level(s) the device uses. Interrupt levels cannot be shared by VXI and VME devices. You then can access the device from NI-VXI/NI-VISA just as you would a VXI device, by specifying the address space and the offset from the base at which you have configured it. NI-VISA support for VME devices includes the register access operations (both high-level and low-level) and the block move operations, as well as the ability to receive interrupts.

Software Description

The NI-VXI/NI-VISA bus interface software includes a Resource Manager, an interactive configuration and troubleshooting program, a comprehensive library of software routines for VXI/VME programming, a logging utility you can use for debugging, and graphical interactive control programs for interacting with VISA. You can use this software to

seamlessly program multiple-mainframe configurations and have software compatibility across a variety of controller platforms.

NI-VISA has a comprehensive library of software routines not only for VXI/VME programming but also for GPIB, GPIB-VXI, and serial. Use this software to program instruments connected via different types of interfaces.

Measurement & Automation Explorer (MAX) helps merge NI-VXI and NI-VISA closer together. Use MAX to view your entire T&M system and configure various components, whether they are VXI, GPIB, or Serial devices. This utility not only takes the place of the NI-VXI T&M Explorer utility, but it also adopts the functionality of the NI-DAQ Configuration utility so you can configure National Instruments VXI-DAQ cards. You can also add VME devices to your system easily with MAX and view them on a screen display along with the rest of your system.

MAX also features various options for running Resman. You can still execute Resman independently to configure your instruments after a power cycle. But you also can perform resource manager operations directly from MAX or configure it to run Resman automatically at startup.

The NI Spy utility tracks the calls your application makes to National Instruments T&M drivers, including NI-VXI, NI-VISA, and NI-488.2. NI Spy helps you debug your application by clearly highlighting the functions that return errors. You can let NI Spy keep a log of your program's calls to these drivers so that you can check them for errors at your convenience.

National Instruments Application Software

In addition to the NI-VXI/NI-VISA software, you can use the National Instruments LabVIEW, LabWindows™/CVI™, and Measurement Studio application programs and instrument drivers to ease your programming task. These standardized programs match the modular virtual instrument capability of VXI and can reduce your VXI/VME software development time. These programs are fully VXI*plug&play* compliant and feature extensive libraries of VXI instrument drivers written to take full advantage of direct VXI control. LabVIEW and Measurement Studio include all the tools needed for instrument control, data acquisition, analysis, and presentation.

LabVIEW is an easy-to-use, graphical programming environment you can use to acquire data from thousands of different instruments, including IEEE 488.2 devices, VXI devices, serial devices, PLCs, and plug-in data acquisition boards. After you have acquired raw data, you can convert it into meaningful results using the powerful data analysis routines in LabVIEW. LabVIEW also comes with hundreds of instrument drivers, which dramatically reduce software development time, because you do not have to spend time programming the low-level control of each instrument.

LabWindows/CVI is an interactive ANSI C programming environment designed for building virtual instrument applications. LabWindows/CVI delivers a drag-and-drop editor for building user interfaces, a complete ANSI C environment for building your test program logic, and a collection of automated code generation tools, as well as utilities for building automated test systems, monitoring applications, or laboratory experiments.

Measurement Studio includes support for Microsoft Visual C++, Visual Basic, and the .NET platform.

If you want to use these application programs, install them during the NI-VXI/NI-VISA software installation. LabVIEW, LabWindows/CVI, and Measurement Studio integrate the VXI and VISA libraries that are required to support PCI-based MXI-2 products. You also get hundreds of complete instrument drivers, which are modular source-code programs that handle the communication with your instrument to speed application development.

2

Setup

This chapter contains the instructions to set up the VXI or VME system using the MXI-2 hardware and NI-VXI/NI-VISA software.

Configuring the Hardware

This section contains basic information about configuring the MXI-2 hardware.

Note Install the NI-VXI software for Windows first, followed by the hardware.

The default settings for the MXI-2 hardware are acceptable for most typical applications. Refer to Appendix A, *Default Settings*, for a complete listing of the hardware and software default settings.

The *MXI-2 Configuration Reference Manual* fully describes the configuration and installation of each MXI-2 board discussed in this getting started manual. Refer to the *MXI-2 Configuration Reference Manual* if you want to try a different hardware configuration or want more information on a particular setting.

Use the MAX utility in NI-VXI/NI-VISA to change PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320 configuration settings. For information on the software, including optional settings, use MAX and MAX help. Access MAX from the MAX icon on the desktop. To access the MAX help, open the Help menu and select **Help Topics**.

Installing the Hardware

This section describes how to install the MXI-2 hardware.

Installing Your PCI-Based MXI-2 Interface

You received either a PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320 in your kit.

Caution To guard against electrostatic discharge, touch the antistatic plastic package to a metal part of your computer or chassis before removing the board from the package. Your computer or chassis should be plugged in but powered off.

Install the PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320 in an available peripheral slot in your PCI-based computer or PXI/CompactPCI chassis. For more information, refer to the PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320 chapter in the *MXI-2 Configuration Reference Manual*.

Installing Your Mainframe Extender

You also received either a VXI-MXI-2, VXI-MXI-2/B, or VME-MXI-2 in your kit.

Caution To guard against electrostatic discharge, touch the antistatic plastic package to a metal part of your chassis before removing the module from the package. The VXI or VME chassis should be plugged in but powered off.

Install the VXI-MXI-2 or VXI-MXI-2/B in the first slot of a VXI chassis, or install the VME-MXI-2 in the first slot of a VME chassis.

The VXI/VME-MXI-2 default configuration automatically detects whether it should be the VXI/VMEbus system controller. The VXI/VMEbus system controllers operate certain VXI/VMEbus lines as required for VXI/VME systems. Verify that any other VXI/VME devices with system controller capability that are in the same chassis are not configured as system controller.

Caution Having more than one device configured as system controller can damage the VXI/VME system.

For VXI systems that include VME devices, ensure that the VME devices are not configured in the upper 16 KB (starting from 0xC000) of the A16 address space. This region is reserved for VXI device configuration registers, which are used for initializing, configuring, and interacting with VXI devices. The PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320 and VME-MXI-2 also use this region for this purpose.

Also, ensure that no VXI devices in your system are configured for either logical addresses 0 or 1. These are the default configurations for the PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320 and the VXI-MXI-2, respectively.

For more information, refer to the VXI-MXI-2, VXI-MXI-2/B, or VME-MXI-2 chapter in the *MXI-2 Configuration Reference Manual*.

Connecting the MXI-2 Cable Properly

By default, the PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320 automatically detects whether it should be the system controller on the MXIbus. Verify that the correct cable end labeled *Connect This End To Device Closest To MXIbus Controller In This Daisy Chain* is attached securely to the PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320. The cable must be connected in this manner so that the MXI board can correctly detect whether it should be the system controller on the MXIbus. Attach the other end of the cable to the VXI/VME-MXI-2.

Software Installation

Use the Setup program that came with the NI-VXI/NI-VISA software to install the entire software package or a software update, or to reinstall software in the event that your files were accidentally erased.

Some of the utilities rely on the LabWindows/CVI Run-Time Engine. This software is installed, if necessary, during the NI-VXI/NI-VISA installation.

Depending on the type of installation you choose, you may need up to 40 MB of free space available to accommodate the NI-VXI/NI-VISA software. If you choose the Custom installation method, Setup displays the amount of memory required for the options you select.

To be compliant with VXI*plug&play* specifications, a VXI controller must provide the VISA I/O driver library standardized by VXI*plug&play.* VISA ensures that your controller can run all VXI*plug&play*-compatible software now and in the future.

The NI-VISA software in this kit is compatible with the WIN95/GWIN95 and WINNT/GWINNT frameworks. With NI-VISA installed on your computer, you can run any VXI*plug&play* software compatible with these frameworks. This includes instrument drivers and executable soft front panel software included with VXI*plug&play*-compatible instruments from a variety of vendors.

Installing the Software

This section describes how to install the 32-bit NI-VXI/NI-VISA software. Please carefully read these directions along with any messages on the screen before making your selections.

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

Setup is an interactive, self-guiding program that installs the NI-VXI/NI-VISA software and configures your system to use the software with the PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320. Follow these steps to complete the installation:

1. Insert the CD-ROM labeled *NI-VXI/NI-VISA for Windows Ver. 3.3*.

   The setup program should run automatically.

2. Click the **Next** button at the Welcome screen to start the installation and accept the license agreement.



Note If Setup detects an earlier version of the NI-VXI software, it prompts you to remove it. Setup can quit so you can uninstall the old software or install the new version over the previous version.



Caution If you want to keep the manufacturer/model name tables or the VME device configuration from a previous installation, be sure to back them up before starting Setup.

3. Select the destination folders for the files being installed.
4. Select the type of installation from the Select Installation Type screen.
   - **Typical** setup is the fastest and simplest installation option. This option installs all the NI-VXI/NI-VISA software without prompting you to make any further choices.
   - **Custom** setup gives you control over which files and utilities you want installed on your system. This option is recommended for advanced users.
5. Click the **Next** button to begin the installation.

Setup now copies the necessary files to your hard drive and creates program icons.

Completing the Software Installation

1. Review the information in any `README` files that Setup prompts you to read.
2. When the installation process completes, you must reboot your computer for the changes to take effect. The NI-VXI driver is loaded at this time.
3. If you backed up the manufacturer and model name files, restore them to the `TBL` subdirectory of the NI-VXI directory before running MAX.
4. After you install the NI-VXI/NI-VISA software, run MAX. It will prompt you to run Resman. You must run Resman every time the chassis power is cycled so that the application can access devices in the VXI/VME chassis. You also can configure MAX to run Resman automatically at every computer startup.
5. After you run Resman, you are ready to use MAX to interactively configure the National Instruments hardware in your system. Use the right-click help for information about the various configuration options.

Verifying Your System Configuration

After you finish configuring the system through MAX, verify the system configuration through the interactive control utility, VISAIC.

For more details about the utilities in NI-VXI/NI-VISA, refer to Chapter 3, *Developing Your Application*.

3

Developing Your Application

This chapter discusses the software utilities you can use to start developing applications that use NI-VXI.

After installing the NI-VXI software, you can begin developing a VXI/VME application. Be sure to check the release notes for the latest application development notes and changes.

NI-VXI, NI-VISA, and Related Terms

Before you develop an application, it is important to understand the difference between *NI-VXI*, *NI-VISA*, and similar terms:

- *NI-VXI* is the software package that ships with National Instruments VXI and VME controllers. NI-VXI includes Measurement & Automation Explorer (MAX), NI-VISA, NI Spy, Resource Manager (Resman), VXI device drivers, and other utilities for configuring and controlling the VXI or VME system.
- *NI-VISA* is the native API for communicating with VXI/VME devices. NI-VISA is the National Instruments implementation of the VISA I/O standard, which is a common interface to many types of instruments (such as VXI, GPIB, PXI, Serial, TCP/IP, etc.). NI-VXI is optimized for use through NI-VISA, and NI recommends using NI-VISA to develop all new VXI/VME applications.
- The *NI-VXI API* is an optional development environment that is not part of the default NI-VXI installation. The NI-VXI API was developed before NI-VISA; while NI-VXI still supports the NI-VXI API, NI recommends using NI-VISA for all new VXI/VME applications. If you must develop an application using the older NI-VXI API, run the NI-VXI installer and select the appropriate option in the custom installation screen. Be sure to review the *NI-VXI API Notes* section later in this chapter.
- The *NI-VXI compatibility layer* allows older programs that use the NI-VXI API to communicate with VXI devices through VISA. Using this compatibility layer, older programs can run in NI-VXI 3.0 or later without being rewritten to use the VISA interface. This layer installs with NI-VXI by default. It should be completely transparent and

provide a high level of performance; however, there may be some slight changes in behavior for certain applications.

The software features several system development utilities including MAX, Resman, NI Spy, VISA Interactive Control (VISAIC), and VXI Interactive Control (VIC, optional). You also can access help and a variety of examples to learn how to use NI-VXI for certain tasks.

Each component assists you with one of four development steps: configuration, device interaction, programming, and debugging.

You can access the utilities, help files, and release notes through the Windows Start menu by opening the **National Instruments»VXI** or **National Instruments»VISA** program groups.

Configuration

The configuration utilities in the software kit are Resman and MAX.

Resman performs VXI Resource Manager functions as described in the VXIbus specification. Resman configures all devices on the VXI backplane for operation and allocates memory for devices that request it. Resman does not require you to specify any settings; it automatically performs the VXI resource management whenever you run it.

Note Power cycling resets all devices, so you must run Resman to reconfigure your system every time you cycle the power on the chassis.

MAX presents a graphical display of the entire test and measurement system to help you configure various components. When you launch MAX, you see all of the devices (including VXI) on the screen. You can view the properties (such as logical address, address space, and so on) of each device by clicking the device in the configuration tree. To see additional configuration options for a given device, right-click the device in the configuration tree. When you access the properties of most National Instruments devices by right-clicking, you can configure the hardware settings by selecting **Hardware Configuration**.

MAX and Resman are designed to work together. You can run Resman through MAX by either clicking the **Run VXI Resource Manager** button in the toolbar or right-clicking a specific VXI system on which to run Resman (refer to Figure 3-1). You also can select **Tools»NI-VXI»VXI Resource Manager** to run Resman on all VXI systems. From the VXI Options dialog box in the **Tools»NI-VXI** menu, you also can use MAX to

configure Resman to run on all VXI systems automatically when the computer boots. Resman reports to MAX all errors it finds in the system; when you view the VXI system in MAX, you can easily spot any errors that Resman found while configuring the system.

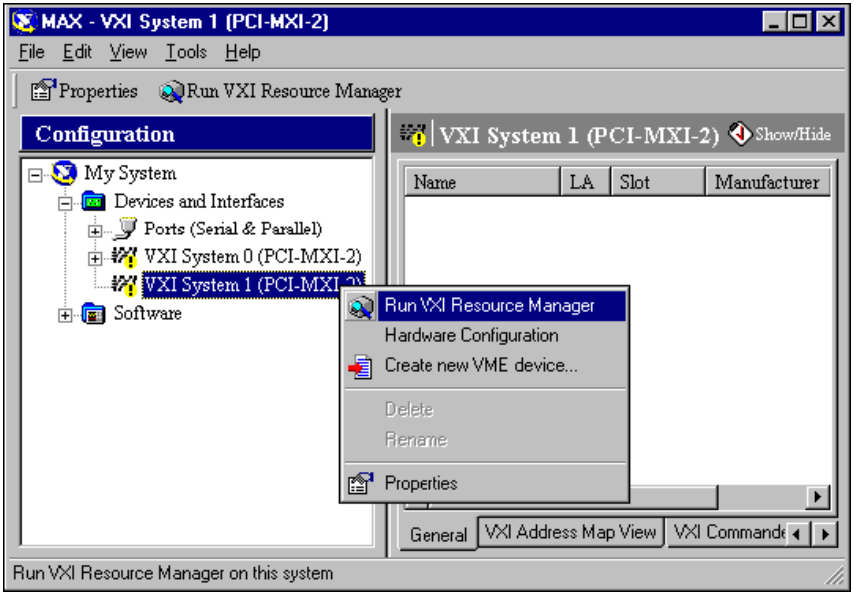


Figure 3-1. Right-Click on a VXI System in MAX to Run Resman on that System

After Resman detects and configures all of the VXI/VME devices, you can use MAX to view specific information about each device in the system. The default MAX view of a VXI system shows the General tab window, which contains a summary of key information about each device, including its device name, logical address, model name, and other data.

For more information about MAX, refer to MAX help by selecting the **Help»Help Topics** menu.

Device Interaction

You can interact with the VXI/VME devices using the VISA Interactive Control (VISAIC) utility. VISAIC allows you to control the VXI/VME devices without using a conventional programming language: LabVIEW, LabWindows/CVI, or Measurement Studio. You also can control the devices in MAX by right-clicking a device name and selecting **Open VISA Session**.

Note You also can use VXI Interactive Control Program (VIC) to control the VXI/VME devices and develop and debug VXI application programs. VIC is not included in the default NI-VXI installation; to install it, select **NI-VXI API Development** from the custom installation screen in the installer.

You can launch VISAIC (or VIC) from the Tools menu in MAX or from the VISA or VXI subgroups in **Start»Programs»National Instruments**.

Try the following in VISAIC: In the tree view, navigate using your mouse to the VISA resource for your controller—probably **VXI0::0::INSTR**, representing the VXI system 0, logical address 0 instrument resource (refer to Figure 3-2).

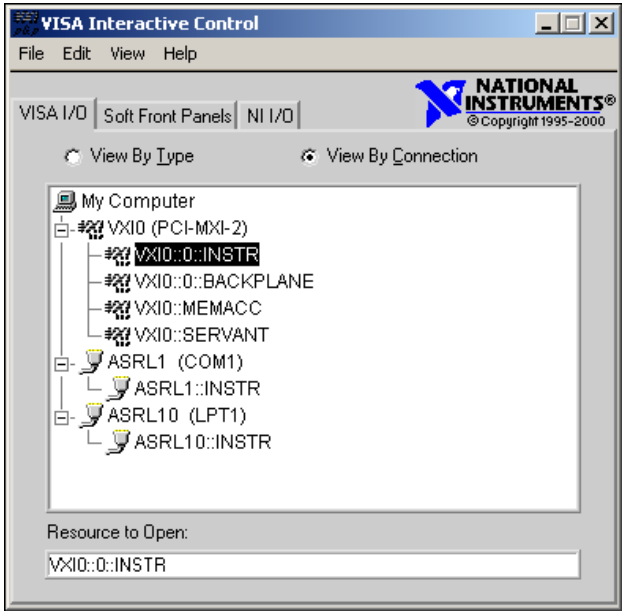


Figure 3-2. Select Your Controller in VISAIC

Open the selected resource and navigate to the Register I/O tab. In this tab, you can read registers on the device, such as the VXI device configuration registers. Execute the **viIn** operation (called **In** in LabVIEW compatibility mode) with the default parameters. The Data Value field shows the I/O operation result, such as `0x9ff6`. The Return Value field shows the function status, such as `0` for `VI_SUCCESS`. This dialog box may look slightly different for LabVIEW users. Refer to Figure 3-3.

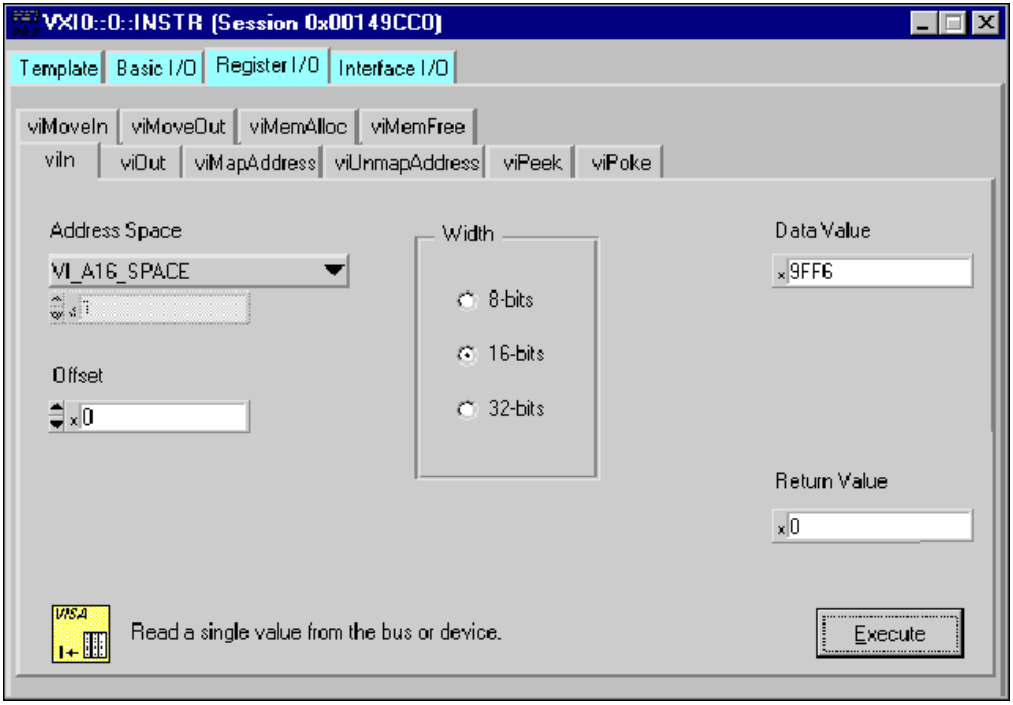


Figure 3-3. Successful viIn Access in the VISAIC Register I/O Tab

If the data value ends in `ff6`, you have successfully read the National Instruments manufacturer ID from your VXI/VME controller's ID register.

You may now want to read the configuration registers from other VXI devices in the system by opening the devices in VISAIC. Try reading a register from each device listed in the MAX view of the VXI system. This way, you can verify that the VXI controller can access each device in the VXI system successfully.

You also can access VXI and VME devices configured in A16, A24, or A32 space by opening the VXI MEMACC resource, which is VISA's representation of VXI memory. For more information about VISAIC operations and commands, refer to the help in the Help menu and the context-sensitive help (such as **What's This?**), accessible by right-clicking in any panel.

Programming for VXI

NI-VISA and the NI-VXI API are the two National Instruments programming interfaces for accessing VXI/VME instruments. With NI-VXI 3.0 or later, NI-VISA is the native API for communicating with a VXI or VME system, and NI recommends using it for all new applications. Older programs that use the NI-VXI API now use the NI-VXI-to-NI-VISA compatibility layer to communicate with the VXI devices. Using this layer, older programs can run in NI-VXI 3.0 or later without being rewritten to use the VISA interface.



Note The NI-VXI API development environment is not installed by default as part of the NI-VXI installation. If you must develop an application using the older NI-VXI API, run the NI-VXI installer and select the appropriate option in the custom installation screen. Be sure to review the *NI-VXI API Notes* section later in this chapter.

NI-VISA is the National Instruments implementation of the VISA API as the VXI*plug&play* standard defines. It provides a common interface to many types of instruments (such as VXI, GPIB, PXI, Serial, TCP/IP, etc.) and therefore is especially useful in situations in which you are using multiple types of instruments.

Both NI-VISA and the NI-VXI API include functions for register-level access to VXI instruments and messaging capability to message-based devices. You also can use either interface to service asynchronous events such as triggers, signals, and interrupts, and also assert them. Compatibility with the NI-VXI API is included for legacy applications only—NI recommends that you write all new VXI/VME applications in VISA.

The best way to learn NI-VISA programming is by reviewing the example programs the software includes. The examples directory contains working VISA programs that illustrate many different types of applications. You can find these examples in the `VXIpnp\WinNT\NIvisa\Examples` directory.

If you are just getting started, you should learn how to access registers with high-level calls and send messages with word-serial functions. The NI-VISA examples for these tasks are `HighReg.c` and `RdWrt.c`. Refer to the other examples as you try more advanced techniques. Refer to the *NI-VISA User Manual* or help for additional information on these topics.

Table 3-1 summarizes the topics the example programs address. All files are in the `VXIpnp\WinNT\NIvisa\Examples` directory, in the subdirectories listed below.

Table 3-1. NI-VISA/NI-VXI Examples

Coverage	NI-VISAExample	NI-VXI Example (Optional)
Message-Based Access	`General\RdWrt.c`	`VXIws.c`
High-Level Register Access	`VXI-VME\HighReg.c`	`VXIhigh.c`
Low-Level Register Access	`VXI-VME\LowReg.c`	`VXIlow.c`
Sharing Memory	`VXI-VME\ShareSys.c`	`VXImem.c`
Interrupt Handling	`VXI-VME\AsyncIntr.c` and `WaitIntr.c`	`VXIint.c`
Trigger Handling	`VXI-VME\WaitTrig.c`	`VXItrig.c`

Note MAX includes configuration options that affect low-level functions and shared memory, as well as trigger mappings and other attributes of the VXI system. Refer to the MAX help for information regarding these options.

Optimizing Large VXIbus Transfers

For best performance, keep the following in mind when using `viMove()` or `VXImove()`:

- Make sure your buffers are 32-bit aligned.
- Transfer 32-bit data whenever possible.
- Use VXI block access privileges to significantly improve performance to devices that are capable of accepting block transfers.
- To optimize move performance on virtual memory systems, lock the user buffer in memory yourself so the move operation does not need to lock the buffer.
- To optimize move performance on paged memory systems, use a contiguous buffer so the move operation does not need to build a scatter-gather list for the user buffer.

Note `viMemAlloc()` or `VXImemAlloc()` returns 32-bit aligned, page-locked, continuous buffers which work efficiently with the move operations.

Shared Memory

In the **Hardware Configuration** settings of the PCI-MXI-2 and PCI-MXI-2 Universal in MAX, you can share memory on your computer or from DRAM added to the PCI-MXI-2 and PCI-MXI-2 Universal. Right-click any of the settings or refer to the help in MAX for more information. You can access shared memory on your computer using `VXImemAlloc()` in NI-VXI and `viMemAlloc()` in VISA.

NI-VXI API Notes

The following sections apply only if you are using the NI-VXI API. NI recommends that all new VXI/VME applications use the NI-VISA API, but you can still develop with the older NI-VXI API for compatibility with legacy code.

Compiler Symbols

You may need to define certain compiler symbols so that the NI-VXI library can work properly with your program. The required symbol indicates your operating system platform; for example, `VXINT` designates the application as a Windows 2000/NT/Me/98 application.

Note LabWindows/CVI automatically defines the correct symbol. You do not need to define `VXINT` when using LabWindows/CVI.

The additional symbol `BINARY_COMPATIBLE` is optional. It ensures that the resulting application is binary compatible with other National Instruments VXI controllers using the same operating system. This symbol may cause a slight performance degradation when you use low-level VXIbus access functions.

You can define these symbols using `#define` statements in your source code or using the appropriate option in your compiler (typically either `-D` or `/D`). If you use `#define` statements, they must appear in the code before the line that includes the NI-VXI API header `nivxi.h`.

Compatibility Layer Options

Although NI-VXI supports multiple VXI controllers through NI-VISA, the NI-VXI API supports only a single controller. To specify which controller the emulation layer should use, run MAX. Select **Tools»NI-VXI»VXI Options**. Select the VXI system that will support the emulation layer.

In NI-VXI 3.0 or later, when you enable for triggers or interrupts, only the local controller is enabled. In the NI-VXI API functions for enabling triggers and interrupts, the controller parameter is ignored. If you need to enable a remote controller for triggers, use the MAX frame resource to map the trigger back to the local controller.

The interrupt and trigger routing in the NI-VXI 3.0 or later low-level drivers is somewhat different from the default routing in previous versions of NI-VXI. Therefore, the compatibility layer may behave differently than the original NI-VXI API with regard to these settings. In particular, if you are receiving triggers on an external controller, you may need to modify the trigger configuration on the extender module using MAX. In general, interrupts are routed automatically based on the interrupt configuration the resource manager detects. Whether the changed routing behavior affects the program is application dependent.

Because VISA is an instrument-centric API, certain functions from the more controller-centric NI-VXI API do not match perfectly with a VISA counterpart. When an application enables an event with the NI-VXI API compatibility layer, each logical address is enabled for that event separately. For example, if the application enables an interrupt level, VISA will enable the interrupt on each logical address, one at a time, until all the devices are enabled. This means that some interrupts could be lost from devices with higher numbered logical addresses. MAX provides an option for users to pick which logical address is enabled first. Select **Tools»NI-VXI»VXI Options**. Set **Prioritized Signal LA** to the logical address of the device that generates the events. This prevents possible loss of events from that device.

Debugging

NI Spy and VISAIC are useful utilities for identifying the causes of problems in an application.

NI Spy tracks the calls an application makes to National Instruments programming interfaces, including NI-VISA, NI-VXI, and NI-488. NI Spy highlights functions that return errors, so during development you can quickly spot which functions failed during a program's execution. NI Spy can log the calls the program makes to these drivers so you can check them for errors at your convenience, or use the NI Spy log as a reference when discussing the problem with National Instruments technical support. Figure 3-4 shows an example of a normal error returned from a call to `viMemAlloc` when no memory has been shared.

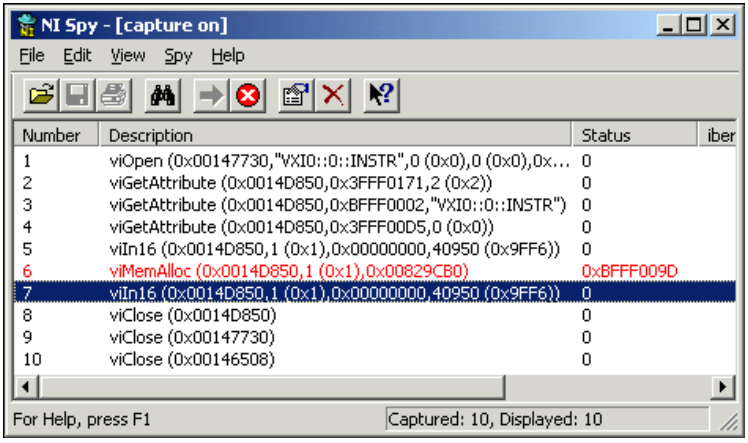


Figure 3-4. NI-Spy

VISAIC, discussed in the *Device Interaction* section, is an excellent platform for quickly testing instruments and learning how to communicate with them.

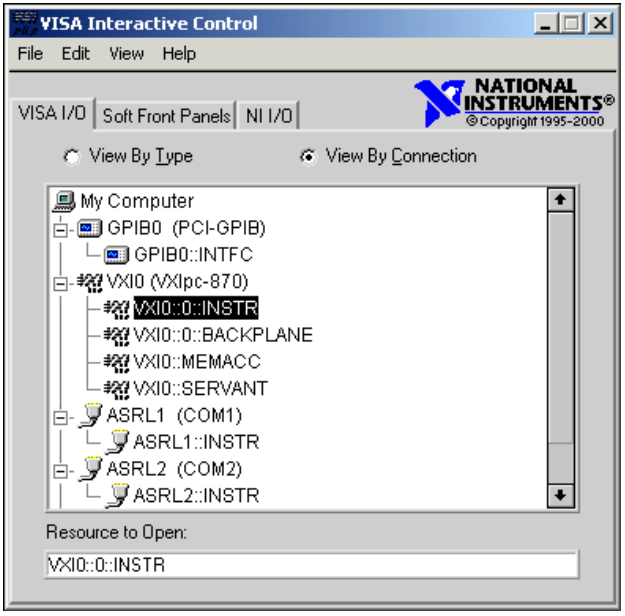


Figure 3-5. VISAIC

A

Default Settings

This section summarizes the hardware and software default settings for the hardware and software in your kit. For more information about a particular setting, or to try a different configuration, refer to the appropriate hardware or software chapters in this documentation set. Refer to the *MXI-2 Configuration Reference Manual* for your hardware reference and the MAX help for your software reference.

PCI-MXI-2/PCI-MXI-2 Universal/PXI-8320

This section summarizes the hardware and software default settings for the PCI-MXI-2, PCI-MXI-2 Universal, and PXI-8320.

Hardware Settings

Table A-1. PCI-MXI-2 Hardware Default Settings

Hardware Component	Default Setting
U17 Switch 1 (FOV)	OFF: PCI-MXI-2 boots off the user-configured half of the EEPROM
U17 Switch 2 (TST)	OFF: Factory configuration of the EEPROM is protected
U17 Switch 3 (POS)	OFF: *Do not alter this setting*
U17 Switch 4 (CT)	ON: *Do not alter this setting*
DRAM SIMM installed	Per customer order

Table A-2. PCI-MXI-2 Universal Hardware Default Settings

Hardware Component	Default Setting
SW1 Switch 1 (FOV)	OFF: PCI-MXI-2 Universal boots off the user-configured half of the EEPROM
SW1 Switch 2 (TST)	OFF: Factory configuration of the EEPROM is protected
SW1 Switch 3 (POS)	OFF: *Do not alter this setting*
SW1 Switch 4 (CT)	ON: *Do not alter this setting*
DRAM SODIMM installed	Per customer order

Table A-3. PXI-8320 Hardware Default Settings

Hardware Component	Default Setting
U6 Switch 1 (FOV)	OFF: PXI-8320 boots off the user-configured half of the EEPROM
U6 Switch 2 (TST)	OFF: Factory configuration of the EEPROM is protected
U6 Switch 3 (POS)	OFF: *Do not alter this setting*
U6 Switch 4 (CT)	ON: *Do not alter this setting*
DRAM SODIMM installed	Per customer order

MAX Settings

Table A-4. Device Tab Default Settings

Editor Field	Default Setting
Logical address	0
Device class	Message-based

Table A-4. Device Tab Default Settings (Continued)

Editor Field	Default Setting
Size of Servant area	0
Number of handlers	1
Number of interrupters	0

Table A-5. Shared Memory Tab Default Settings

Editor Field	Default Setting
Memory sharing	Don't share memory
Shared RAM size	0 KB
Reserved physical memory	0 KB
Lower half window byte swapping	Disabled
Lower half window memory selection	System memory
Upper half window byte swapping	Disabled
Upper half window memory selection	System memory
Map upper and lower halves at same PCI address	Disabled

Table A-6. MXI-2 Bus Tab Default Settings

Editor Field	Default Setting
Bus timeout	1 ms
System controller	Auto-detect
MXI-2 auto retry	Enabled

Table A-6. MXI-2 Bus Tab Default Settings (Continued)

Editor Field	Default Setting
A24/A32 write posting	Disabled
VXImove uses Synchronous MXI	Enabled
MXI transfer limit	Unlimited
MXI CLK10 signal	Receive

Table A-7. PCI Tab Default Settings

Editor Field	Default Setting
Low-level register access API support	Enabled
User window size	64 KB
Expansion ROM	Enabled

VXI/VME-MXI-2

This section summarizes the hardware and software default settings for the VXI-MXI-2, VXI-MXI-2/B, and VME-MXI-2.

Hardware Settings

Table A-8. VXI-MXI-2 Hardware Default Settings

Hardware Component	Default Setting
Logical address (U43)	1
VXIbus Slot 0/Non-Slot 0 (W2)	Automatic detection
VXIbus local bus (S8, S9)	Both OFF: single VXI-MXI-2
VXIbus CLK10 routing (W3)	From onboard oscillator

Table A-8. VXI-MXI-2 Hardware Default Settings (Continued)

Hardware Component	Default Setting
External trigger termination (S2)	OFF: unterminated
SMB CLK10 direction (S3)	OUT: drive CLK10 signal
SMB CLK10 termination (S4)	Ignored; effective only when S3 is set to IN
Polarity of external SMB CLK10 (S5)	Inverted
MXIbus CLK10 signal (S7)	Receive CLK10 from MXIbus
MXIbus termination (U35 Switches 1 and 2)	Automatic MXIbus termination: Switch 2 set to NO; Switch 1 ignored
Configuration EEPROM (U35 Switches 3 and 4)	User-modifiable; factory settings protected: both switches set to NO
DRAM SIMMs installed	Per customer order
SIMM size configuration (S6)	OFF if SIMMS are 4 M × 32 or larger; ON if smaller than 4 M × 32

Table A-9. VXI-MXI-2/B Hardware Default Settings

Hardware Component	Default Setting
Logical address (U20)	1
VXIbus Slot 0/Non-Slot 0 (W3)	Automatic detection
VXIbus local bus (W2)	Single VXI-MXI-2/B in frame
VXIbus CLK10 routing (W1)	From onboard oscillator
External trigger termination (S5)	OFF: unterminated
SMB CLK10 direction (S7)	OUT: drive CLK10 signal

Table A-9. VXI-MXI-2/B Hardware Default Settings (Continued)

Hardware Component	Default Setting
SMB CLK10 termination (S6)	Ignored; effective only when S7 is set to IN
Polarity of external SMB CLK10 (S3)	Inverted
MXIbus CLK10 signal (S1)	Receive CLK10 from MXIbus
MXIbus termination (U21 Switches 3 and 4)	Automatic MXIbus termination: Switch 3 set to OFF; Switch 4 ignored
Configuration EEPROM (U21 Switches 1 and 2)	User-modifiable; factory settings protected: both switches set to OFF

Table A-10. VME-MXI-2 Hardware Default Settings

Hardware Component	Default Setting
A16 base address (U20)	Hex C040
VME-MXI-2 intermodule signaling (W2)	No user-defined pin selected
MXIbus termination (U21 Switches 3 and 4)	Automatic MXIbus termination: Switch 3 OFF; Switch 4 ignored
Configuration EEPROM (U21 Switches 1 and 2)	User-modifiable; factory settings protected: both switches OFF
DRAM SIMMs installed	Per customer order
SIMM size configuration (S2)	OFF if SIMMS are 4 M × 32 or larger; ON if smaller than 4 M × 32

MAX Settings

Table A-11. Device Tab Default Settings

Editor Field	Default Setting
Logical address	Use DIP switch
Address space	A24*
Requested memory	16 KB*
A24/A32 write posting	Disabled
A16 write posting	Disabled
Interlocked mode	Disabled
* Assumes no DRAM is installed. If DRAM is installed, the Address space should be A32, and Requested memory should match the amount of DRAM. If you install the DRAM yourself, you must manually specify these changes.	

Table A-12. VXI/VME Bus Tab Default Settings

Editor Field	Default Setting
Bus timeout value	125 µs
Slot 0 configuration	Auto-detect
Auto retry	Disabled
Transfer limit	256
Arbiter type	Priority
Fair requester	Enabled
Arbiter timeout	Enabled
Request level	3

Table A-13. MXI-2 Bus Tab Default Settings

Editor Field	Default Setting
System controller	Auto-detect
Bus timeout value	1 ms
MXI-2 auto retry	Disabled
MXI transfer limit	Unlimited
MXI fair requester	Disabled
Perform parity checking	Enabled
MXI-2 CLK10 signal direction	Switch determines signal direction (VXI-MXI-2 only)

B

Common Questions

This appendix addresses common questions you may have about using the NI-VXI/NI-VISA software on the PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320 platform.

What are some of the differences between the old utilities and the new ones?

The old utility components are as follows:

- VXIinit—This utility initialized the National Instruments controller hardware with settings determined in VXIedit.
- Resman—This utility initializes and configures all the other devices in your VXI system.
- VXIedit—This utility configured the National Instruments hardware.
- VXItedit—This was a console-based version of VXIedit.
- VIC—This utility interactively communicates with VXI devices over the VXIbus using the NI-VXI API.
- VICtext—This was a console-based version of VIC.
- VISAconf—This utility configured settings used by NI-VISA
- VISAIC—This utility interactively communicates with VISA devices (GPIB, VXI, serial) using NI-VISA.
- T&M Explorer—This utility configured, viewed, and initialized the system.

VXI system integration with the old utilities typically proceeded as follows:

1. Install components and boot the system.
2. Configure the hardware with VXIedit.
3. Reboot and run VXIinit to initialize the National Instruments hardware.
4. Run Resman to initialize the VXIbus.
5. Optionally run VXIedit to configure any extender devices on the VXIbus.
6. Run VIC to verify device operation.

7. Run VISAIC to verify that you can communicate with your system using VISA.

The new utility components are as follows:

- MAX—This utility configures, views, and initializes your system.
- Resman—You can still use this as before. However, you can perform resource manager operations directly from MAX or configure it to run Resman automatically at startup. Refer to *What about running Resman?* later in this section.
- VISAIC—You can still use this utility as before.
- NI Spy—This utility debugs an NI-VXI or NI-VISA application.

Your setup might now include the following steps:

1. Install components and boot the system.
2. Launch MAX. Execute VXI Resource Manager responsibilities. Run Resman by clicking the **Run VXI Resource Manager** button in the MAX toolbar.

What happened to VXIinit?

You no longer need to run VXIinit to initialize settings on the hardware. NI takes care of loading hardware settings in the driver, completely eliminating the need for VXIinit.

Where do I find the information that VXIinit used to print?

You can view information about your controller from the properties pages and the hardware configuration pages. For example, you can view logical address and user window size in the configuration view in MAX.

What happened to T&M Explorer?

The functionality of this utility has been integrated into a powerful new utility called MAX. This utility starts with a graphical view of the VISA devices (GPIB, VXI, and Serial) that it finds in your system. Right-click an individual device in the configuration tree to see its properties. You can further configure National Instruments devices by selecting the **Hardware Configuration** option. This includes National Instruments VXI-DAQ cards, which means you can configure these devices from MAX without running the DAQ Configuration Utility.

What about running Resman?

Resman is the name of the utility that performs the duties of a VXI Resource Manager as discussed in the VXIbus specification. When you set a National Instruments controller to Logical Address 0, at some point you will need to run Resman to configure the VXI instruments. If your controller uses a different (non-zero) logical address and is a message-based device, you need to start Resman before running it on the Logical Address 0 computer.

When do you need to run Resman?

Run Resman whenever you need to configure your VXI instruments. For example, if you power-cycle your VXI chassis, your instruments will be reset, and you will need to run Resman to configure them. You can get into trouble if you run Resman when your devices are not in a reset state. Therefore, if you need to run Resman after running it once, you should reset all of your VXI instruments.

In the new architecture, you can perform resource manager operations from within MAX. Additionally, you can tell MAX to run Resman when the computer first boots. In this case, you may never need to run Resman explicitly again. This is common when you use an embedded PC such as the VXIpc-860. You can configure the computer to run Resman at startup, so when you power on the chassis, Resman runs. If you power-cycle the chassis, the PC reboots, forcing Resman to run again.

With the PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320, you may need to run the Resman utility if you boot your computer before powering on your VXI chassis or if you power-cycle your VXI chassis while the external PC remains on. In these cases, the instruments would have been reset without the computer rebooting. You will need to run the Resman utility or configure your system in MAX to initialize your VXI system.

What if I have a system that requires the old utilities?

The new installer for NI-VXI/NI-VISA gives you the option of installing some of the old NI-VXI API utilities and documents. Thus, if you have a documented procedure for configuring your system that relies on the old configuration utilities, you may be able to install them on your system. Use the **Custom** installer option to explicitly add the old utilities. If the custom installation does not install the utility you need, contact NI technical support for a solution.

How do I handle VME devices?

Although there is no way to automatically detect VME devices in a system, you can add them easily through the Create New wizard in MAX. Through this procedure, you can reserve resources for each of your VME devices and configure MAX to show VME devices on the screen with all your other devices.

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

There are several ways to find this information.

- Under Windows, you can find version information by right-clicking any component and selecting the **Properties** option. This displays a property sheet with a version tab. This tab has version information about the product (NI-VXI) and the component (`NIVXINT.DLL`, for example).
- In MAX, select **Software** in the Configuration panel and select **NI-VISA**. The version information appears in the Attributes panel.
- You can find version information about the NI-VXI driver by running the VIC utility program. Type `ver` at the prompt, and the utility displays the versions of VIC and NI-VXI, and the latest PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320 board revision that this NI-VXI driver supports.
- You can find version information about the VISA driver through VISAIC by selecting **About** from the Help menu.

How can I determine the serial number and hardware revision of the MXI-2 boards?

Run MAX and right-click the name of the MXI-2 board in the configuration tree. Select **Hardware Configuration**, and the dialog box for the MXI-2 board is displayed. The title bar includes the serial number and hardware revision of the board.

Which NI-VXI utility program must I use to configure the PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320?

Use MAX to configure the PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320. Access MAX from the desktop icon.

Which NI-VXI utility program must I use to initialize the PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320?

Windows automatically initializes the board at system startup.

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

Use the Resman program to perform startup Resource Manager operations. You can run Resman by clicking the **Run VXI Resource Manager** button in the MAX toolbar. Resman uses the settings configured in MAX. It initializes the VXI/VMEbus system and stores the information that it collects in the `RESMAN.TBL` file in the `TBL` subdirectory of the `NIVXI` directory.

Through MAX, you also can configure Resman to run automatically at computer startup.

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

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

To test individual devices, you can use the VISAIC program to interactively issue NI-VISA operations. You can use the `ViIn` and `ViOut` 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 `viRead` and `viWrite` functions.

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

What should I do if I get a `Configuration EEPROM is Invalid` message?

There are several reasons why you might get the **`Configuration EEPROM is Invalid`** message. If you turned off the computer while the configuration update process was still in progress, the board functions normally except when running MAX. To correct these problems, reboot the computer with the Factory Override (FOV) switch set as described in Appendix B, *EEPROM Configuration*, of the *MXI-2 Configuration Reference Manual* and update the configuration, or load the configuration from file.

Two other reasons you might receive this error message are that the board might have an incorrect base address assigned for the driver window, or there may be a conflict with another adapter or memory management software.

What should I do if Resman hangs?

1. Ensure that the MXI-2 cable is plugged in and that the end labeled *Connect this end to the device closest to the MXIbus System Controller* is connected to the MXIbus system controller (by default, the PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320). Because the MXI-2 cables are polarized, it matters which end is connected to which device.
2. Check for bent or broken pins on the MXI-2 connectors.
3. If you are using a VME-MXI-2 in the first slot of a VMEbus chassis, the chassis may be causing problems with the First Slot Detection circuit on the VME-MXI-2. Use MAX to change the **Slot 0 Configuration** setting of the VME-MXI-2 to **Slot 0** rather than **Auto-detect** before running Resman again. You must then enter the logical address of the VME-MXI-2 to configure.
4. If problems persist, run Resman with the VXI/VME-MXI-2 located in the leftmost slot of the chassis (VXI Slot 0) and no other instruments installed. If this works, try adding instruments until the problem occurs again. Contact National Instruments for further assistance.

Where can I find hardware specifications?

Refer to Appendix A, *Specifications*, of the *MXI-2 Configuration Reference Manual*.

What do the LEDs on the front of the VXI-MXI-2 or VME-MXI-2 mean?

The **SYSFAIL** LED shows the state of the VXIbus/VMEbus SYSFAIL line. This line is asserted whenever any device in the chassis has not yet passed the self-test, if it has failed the self-test, or if it has detected a failure after originally passing the self-test. The **MXI** LED indicates that the VXI-MXI-2 or VME-MXI-2 is acting as a slave to another device on the MXIbus, such as when the PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320 communicates with either the VXI-MXI-2 or VME-MXI-2 or with another device in the chassis. The **VXI (VME)** LED, when lit, indicates that the VXI-MXI-2 or VME-MXI-2 is acting as a slave to another device in the VXI (VME) chassis, such as when a bus master inside the chassis wants to talk to either the VXI-MXI-2 or VME-MXI-2 or another device outside the chassis.

Are the VXI-MXI-2 and PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320 two devices or one with respect to the VXIbus?

The PCI-MXI-2, PCI-MXI-2 Universal, PXI-8320, and VXI-MXI-2 are unique VXIbus devices with their own logical addresses. However, the MXIbus allows the PCI or PXI/CompactPCI computer to behave as if it is inside the chassis with the VXI-MXI-2 by transparently converting PCI bus cycles to MXIbus cycles to VXIbus cycles, and vice versa.

I have a system that requires rugged chassis and bulkhead cables. Can I still use MXIbus?

Yes, National Instruments sells MXIbus bulkhead cables. Contact National Instruments for further information.

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

CLK10 is a differential ECL signal on the VXIbus backplane. However, the oscillator for the VXI-MXI-2 and the EXTCLK input from the front panel use TTL. Therefore, supply a TTL-level signal for EXTCLK; the voltage converters will convert the signal to differential ECL.

CLK10 is not applicable to VME.

What is the accuracy of the CLK10 signal?

The CLK10 generated by the VXI-MXI-2 is 100 ppm (0.01%) as per the VXIbus specification. If you need a more accurate CLK10 signal, you can use the EXTCLK input at the front of the VXI-MXI-2.

CLK10 is not applicable to VME.

What are the user and driver windows?

The PCI-MXI-2, PCI-MXI-2 Universal, and PXI-8320 driver requires the use of two PCI windows: a user window and a driver window. NI-VXI uses the driver window to perform high-level functions such as `ViIn` and `ViOut`, and to access registers on the MXI-2 boards in the system. The user window is reserved for low-level function calls, such as `ViPeek`, `ViPoke`, and `ViMapAddress`. The driver window is system defined and not configurable, but you can increase the size of your user window through MAX if you expect to initiate transfers to a wide variety of addresses in both A24 and A32 address spaces.

What is shared memory and dual-ported memory?

These terms refer to a block of memory that is accessible to both a client and a server. The memory block operates as a message buffer for communications. Shared memory is applicable only if you are using either A24 or A32 address space.

How should I assign logical addresses in a multiple-mainframe system?

A simple algorithm for a system containing only one level of hierarchy—a single chain of MXI cables—is to use the upper nibble (most significant four bits) as a *frame* number and the lower nibble (least significant four bits) as a *device* number. For example, the *fifth* device in the *third* mainframe would be logical address 35 (hex).

For more detailed information on this topic, refer to the *VXI-MXI-2 User Manual*, the *VME-MXI-2 User Manual*, or the *VXI-6, VXIbus Mainframe Extender Specification*.

C

Technical Support and Professional Services

Visit the following sections of the National Instruments Web site at ni.com for technical support and professional services:

- **Support**—Online technical support resources include the following:
  - **Self-Help Resources**—For immediate answers and solutions, visit our extensive library of technical support resources available in English, Japanese, and Spanish at ni.com/support. These resources are available for most products at no cost to registered users and include software drivers and updates, a KnowledgeBase, product manuals, step-by-step troubleshooting wizards, hardware schematics and conformity documentation, example code, tutorials and application notes, instrument drivers, discussion forums, a measurement glossary, and so on.
  - **Assisted Support Options**—Contact NI engineers and other measurement and automation professionals by visiting ni.com/ask. Our online system helps you define your question and connects you to the experts by phone, discussion forum, or email.
- **Training**—Visit ni.com/custed for self-paced tutorials, videos, and interactive CDs. You also can register for instructor-led, hands-on courses at locations around the world.
- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, NI Alliance Program members can help. To learn more, call your local NI office or visit ni.com/alliance.

If you searched ni.com and could not find the answers you need, contact your local office or NI corporate headquarters. Phone numbers for our worldwide offices are listed at the front of this manual. You also can visit the Worldwide Offices section of ni.com/niglobal to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

Glossary

Prefix	Meaning	Value
p-	pico-	10^{-12}
n-	nano-	10^{-9}
µ-	micro-	10^{-6}
m-	milli-	10^{-3}
k-	kilo-	10^{3}
M-	mega-	10^{6}
G-	giga-	10^{9}
t-	tera-	10^{12}

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

ACFAIL — A VMEbus backplane signal that is asserted when a power failure has occurred (either AC line source or power supply malfunction), or if it is necessary to disable the power supply (such as for a high-temperature condition)

address — Character code that identifies a specific location (or series of locations) in memory. In VISA, it identifies a resource.

address modifier — One of six signals in the VMEbus specification used by VMEbus masters to indicate the address space in which a data transfer is to take place

address space	A set of 2^n memory locations differentiated from other such sets in VXI/VMEbus systems by six addressing lines known as address modifiers. *n* is the number of address lines required to uniquely specify a byte location in a given space. Valid numbers for *n* are 16, 24, and 32. In VME/VXI, because there are six address modifiers, there are 64 possible address spaces.
address window	A portion of address space that can be accessed from the application program
ANSI	American National Standards Institute
API	Application Programming Interface; the direct interface that an end user sees when creating an application
arbitration	A process in which a potential bus master gains control over a particular bus
asynchronous	Not synchronized; not controlled by time signals

B

B	bytes
backplane	An assembly, typically a printed circuit board, with 96-pin connectors and signal paths that bus the connector pins. A C-size VXIbus system will have two sets of bused connectors called J1 and J2. A D-size VXIbus system will have three sets of bused connectors called J1, J2, and J3.
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. In VME, the data transfer may have no more than 256 elements; MXI does not have this restriction.
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 error	An error that signals failed access to an address. Bus errors occur with low-level accesses to memory and usually involve hardware with bus mapping capabilities. For example, nonexistent memory, a nonexistent register, or an incorrect device access can cause a bus error.
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

CLK10	A 10 MHz, ±100 ppm, individually buffered (to each module slot), differential ECL system clock that is sourced from Slot 0 of a VXIbus mainframe 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 that is also a bus master and can control one or more Servants
CompactPCI	An adaptation of the PCI specification for industrial and/or embedded applications that require a more robust mechanical form factor than desktop PCI. CompactPCI provides a standard form factor for those applications requiring the high performance of PCI as well as the small size and ruggedness of a rack-mount system.
configuration registers	A set of registers through which the system can identify a module device type, model, manufacturer, address space, and memory requirements. To support automatic system and memory configuration, the VXIbus specification requires that all VXIbus devices have a set of such registers.

D

daisy-chain	A method of propagating signals along a bus, in which the devices are prioritized on the basis of their position on the bus
Data Transfer Bus	DTB; one of four buses on the VMEbus backplane. The DTB is used by a bus master to transfer binary data between itself and a slave device.

DMA	Direct Memory Access; a method by which data is transferred between devices and internal memory without intervention of the central processing unit
DRAM	Dynamic RAM (Random Access Memory); storage that the computer must refresh at frequent intervals
driver window	A region of PCI address space that is decoded by the PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320 for use by the NI-VXI software
DTB	*See* Data Transfer Bus.
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
EEPROM	Electronically Erasable Programmable Read Only Memory
embedded controller	An intelligent CPU (controller) interface plugged directly into the VXI backplane, giving it direct access to the VXIbus. It must have all of its required VXI interface capabilities built in.
expansion ROM	An onboard EEPROM that may contain device-specific initialization and system boot functionality
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 (IEEE 488)

H

hex	hexadecimal; the numbering system with base 16, using the digits 0 to 9 and letters A to F
Hz	hertz; cycles per second

I

IEEE	Institute of Electrical and Electronics Engineers
I/O	input/output; the techniques, media, and devices used to achieve communication between machines and users
instrument driver	A set of routines designed to control a specific instrument or family of instruments, and any necessary related files for LabWindows/CVI or LabVIEW
interrupt	A means for a device to request service from another device
interrupt handler	A VMEbus functional module that detects interrupt requests generated by Interrupters and responds to those requests by requesting status and identify information
interrupt level	The relative priority at which a device can interrupt

K

KB	Kilobytes of memory

L

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

M

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.
master-mode operation	A device is in master mode if it is performing a bus cycle which it initiated
MAX	Measurement & Automation Explorer
MB	Megabytes of memory
MBLT	Eight-byte block transfers in which both the Address bus and the Data bus are used to transfer data
message-based device	An intelligent device that implements the defined VXIbus registers and communication protocols. These devices are able to use Word Serial Protocol to communicate with one another through communication registers.
MITE	A National Instruments custom ASIC, a sophisticated dual-channel DMA controller that incorporates the Synchronous MXI and VME64 protocols to achieve high-performance block transfer rates
MXI-2	The second generation of the National Instruments MXIbus product line. MXI-2 expands the number of signals on a standard MXIbus cable by including VXI triggers, all VXI interrupts, CLK10, SYSFAIL*, SYSRESET*, and ACFAIL*. MXI-2 also defines new higher-performance data transfer protocols.
MXIbus	Multisystem eXtension Interface Bus; a high-performance communication link that interconnects devices using round, flexible cables
MXIbus System Controller	A functional module that has arbiter, daisy-chain driver, and MXIbus cycle timeout responsibility. Always the first device in the MXIbus daisy-chain

N

NI-488 or NI-488.2	The National Instruments software for GPIB systems
NI-DAQ	The National Instruments software for data acquisition instruments

NI-VISA	The National Instruments implementation of the VISA standard; an interface-independent software that provides a unified programming interface for VXI, GPIB, and serial instruments
NI-VXI	The software package that ships with National Instruments VXI and VME controllers. NI-VXI includes Measurement & Automation Explorer (MAX), NI-VISA, NI Spy, Resource Manager (Resman), VXI device drivers, and other utilities for configuring and controlling your VXI or VME system.
Non-Slot 0 device	A device configured for installation in any slot in a VXIbus mainframe other than Slot 0. Installing such a device into Slot 0 can damage the device, the VXIbus backplane, or both.

O

Onboard RAM	The optional RAM installed into the SIMM slots of the MXI-2 board

P

PCI	Peripheral Component Interconnect. The PCI bus is a high-performance 32- or 64-bit bus with multiplexed address and data lines.
PXI	PCI eXtensions for Instrumentation; an open implementation of CompactPCI that adds electrical features that meet the high-performance requirements of instrumentation applications by providing triggering, local buses, and system clock capabilities. PXI also offers two-way interoperability with CompactPCI products.

R

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

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

S

s	seconds
Servant	A device controlled by a Commander; there are message-based and register-based Servants
Shared Memory Protocol	A communication protocol that uses a block of memory accessible to both a client and a server. The memory block operates as a message buffer for communications. This is unique to register-based interfaces such as VXI.
SIMM	Single In-line Memory Module
slave	A functional part of a MXI/VME/VXIbus device that detects data transfer cycles initiated by a VMEbus master and responds to the transfers when the address specifies one of the device's registers
slave-mode operation	A device is in slave mode it if is responding to a bus cycle.
Slot 0 device	A device configured for installation in Slot 0 of a VXIbus mainframe. This device is unique in the VXIbus system in that it performs the VMEbus System Controller functions, including clock sourcing and arbitration for data transfers across the backplane. Installing such a device into any other slot can damage the device, the VXIbus backplane, or both.
SODIMM	Small Outline Dual In-line Memory Module
statically configured device	A device whose logical address cannot be set through software; that is, it is not dynamically configurable
SYSFAIL	A VMEbus signal that is used by a device to indicate an internal failure. A failed device asserts this line. In VXI, a device that fails also clears its PASSed bit in its Status register.

SYSRESET	A VMEbus signal that is used by a device to indicate a system reset or power-up condition
System RAM	RAM installed on your personal computer and used by the operating system, as contrasted with onboard RAM, which is installed on the MXI-2 board

T

trigger	Either TTL or ECL lines used for intermodule timing
TTL	Transistor-Transistor Logic

U

user window	A region of PCI address space reserved by the PCI-MXI-2, PCI-MXI-2 Universal, or PXI-8320 for use via the NI-VXI low-level function calls. `MapVXIAddress()` uses this address space to allocate regions for use by the `VXIpeek()` and `VXIpoke()` macros.

V

VIC	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.
VISA	Virtual Instrument Software Architecture. This is the general name given to VISA and its associated architecture. The architecture consists of two main VISA components: the VISA Resource Manager and the VISA Instrument Control Resources.
VISAIC	VISA Interactive Control Program, a part of the NI-VISA software. Used to program devices, and develop and debug application programs.
VME	Versa Module Eurocard or IEEE 1014

VMEbus System Controller	A device configured for installation in Slot 0 of a VXIbus mainframe or Slot 1 of a VMEbus chassis. This device is unique in the VMEbus system in that it performs the VMEbus System Controller functions, including clock sourcing and arbitration for data transfers across the backplane. Installing such a device into any other slot can damage the device, the VMEbus/VXIbus backplane, or both.
VXIbus	VMEbus eXtensions for Instrumentation

W

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

Index

A

application development
    programming with VXI
        example programs (table), 3-7

B

bulkhead cables, B-7

C

cables
    bulkhead cables, B-7
    cable connection for MXI-2, 2-3
CLK10 signal, B-7
common questions, B-1
configuration EEPROM is invalid message, B-6
contacting National Instruments, C-1
conventions used in the manual, *xi*
Create New wizard, 1-4
customer
    education, C-1
    professional services, C-1
    technical support, C-1

D

default settings
    PCI-MXI-2/PCI-MXI-2
      Universal/PXI-8320
        hardware settings, A-1
        MAX settings, A-2
    VXI/VME-MXI-2
        hardware settings, A-4
        MAX settings, A-7
diagnostic resources, C-1
documentation
    flowchart for using manual, 1-1
    how to use documentation set, *x*
    online library, C-1
    related documentation, *xii*
driver window, B-8
drivers
    instrument, C-1
    software, C-1
D-size mainframes with P3 connector (note), 1-3
dual-ported memory, B-8

E

example code, C-1

F

frequently asked questions, C-1

H

hardware configuration, 2-1
hardware description, 1-3
    P3 support, 1-3
    PCI-MXI-2 interface board, 1-3
    PCI-MXI-2 Universal interface board, 1-3
    PXI-8320 circuit board, 1-3
    using VME devices, 1-4
    VXI/VME-MXI-2 module, 1-4
hardware installation, 2-1
hardware settings
    PCI-MXI-2/PCI-MXI-2
      Universal/PXI-8320, A-1
    VXI-VME-MXI-2, A-4

help
professional services, C-1
technical support, C-1
how to use documentation set, *x*

I

in, B-4
installation and configuration, 2-1
hardware configuration, 2-1
hardware installation, 2-1
configuring more than one device as system controller (caution), 2-2
connecting MXI-2 cable, 2-3
electrostatic discharge (caution), 2-2
mainframe extender, 2-2
PCI-based MXI-2 interface, 2-1
software installation, 2-3
completing installation, 2-5
procedure for installing software, 2-4
verifying system configuration, 2-5
instrument drivers, C-1

K

KnowledgeBase, C-1

L

LabWindows/CVI, 1-6
LEDs on front of VXI-MXI-2 or VME-MXI-2, B-7
logical address in multiple-mainframe system, B-8

M

mainframe extender, installing, 2-2
MAX
changing configuration settings, 2-1
default settings
PCI-MXI-2/PCI-MXI-2 Universal/PXI-8320, A-2
VXI/VME-MXI-2, A-7
overview, 1-5
Measurement and Automation Explorer. *See* MAX
Measurement Studio, 1-6
memory, shared and dual-ported, B-8
MXI LED, B-7
MXI-2 interface kit. *See* PCI-based MXI-2 interface
MXI-2. *See* PCI-MXI-2/PCI-MXI-2 Universal/PXI-8320
MXIbus system controller, 1-3

N

National Instruments
customer education, C-1
professional services, C-1
system integration services, C-1
technical support, C-1
worldwide offices, C-1
National Instruments application software, 1-5
NI Spy utility, 1-5
NI-VISA/NI-VXI software
example programs (table), 3-7
NI-VXI/NI-VISA software
determining which version is installed, B-4
overview, 1-4

O

old utilities, installing, B-3
online technical support, C-1

P

P3 connector on D-size VXI mainframes (note), 1-3
PCI-based MXI-2 interface
  common questions, B-1
  hardware description, 1-3
  installation, 2-1
  overview, 1-2
  requirements for getting started, 1-2
  software description, 1-4
    National Instruments application software, 1-5
  VME users, 1-4
PCI-MXI-2/PCI-MXI-2 Universal/PXI-8320
  connecting cable properly, 2-3
  determining serial number and hardware revision, B-4
  hardware description, 1-3
phone technical support, C-1
professional services, C-1
programming examples, C-1
PXI-8320. *See* PCI-MXI-2/PCI-MXI-2 Universal/PXI-8320

R

related documentation, *xii*
requirements for getting started, 1-2
Resman
  performing startup Resource Manager operations, B-5
  running, B-3
  testing your system, B-5
  troubleshooting hangups, B-6
  when to use, B-3
Resource Manager. *See* Resman
rugged chassis, B-7

S

settings. *See* default settings
setup. *See* installation and configuration
shared memory, B-8
software description
  National Instruments application software, 1-5
  NI-VXI/NI-VISA bus interface software, 1-4
software drivers, C-1
software installation, 2-3
  completing installation, 2-5
  procedure for installing software, 2-4
startup Resource Manager operations, B-5
support
  technical, C-1
SYSFAIL LED, B-7
system controller
  configuring more than one device as system controller (caution), 2-2
  optional MXIbus system controller, 1-3
system integration services, C-1
system requirements, 1-2

T

T&M Explorer, B-1
  replacement by MAX, B-2
technical support, C-1
telephone technical support, C-1
training
  customer, C-1
troubleshooting resources, C-1

U

user window, B-8
utilities
  questions about, B-1

V

VIC utility, B-1
VICtext, B-1
VISAconf utility, B-1
VISAIC utility, B-1
VME devices
  adding with Add Device Wizard, B-4
  configuration considerations, 2-2
  using with PCI-based MXI-2 interface, 1-4
VME-MXI-2. *See* VXI/VME-MXI-2
VXI (VME) LED, B-7
VXI/VME-MXI-2
  hardware description, 1-3
  hardware settings, A-4
  LEDs on front panel, B-7
  MAX settings, A-7
VXIedit utility, B-1
VXIinit, B-1, B-2
VXI*plug&play* specifications, 2-3
VXItedit, B-1

W

Web
  professional services, C-1
  technical support, C-1
WIN95/GWIN95 framework, 2-3
WINNT/GWINNT framework, 2-3
worldwide technical support, C-1

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