VXI/VME𝗣𝗖™ 600 S𝗘𝗥𝗜𝗘𝗦 𝗙𝗢𝗥 W𝗜𝗡𝗗𝗢𝗪𝗦 95/NT

This document contains information to help you understand the components of your kit, start setting up your kit, and learn about the NI-VXI/VISA features.

Contents

What Do You Have?

Your kit contains hardware, software, and documentation. You may also have ordered some optional equipment or software.

Hardware

Your hardware includes either the VXIpc 600 Series or the VMEpc 600 Series controller, which you install in your VXI or VME chassis. You also receive the following accessories:

- Electromagnetic emissions kit
- AT-PS/2 cable
- Serial adapter cable
- Parallel adapter cable

Software

Your VXI/VMEpc 600 Series controller comes with National Instruments software already installed on its hard drive.

Windows 95 Kit

The Windows 95 kit contains the following software:

- Windows 95 CD-ROM
- *NI-VXI/VISA for VXI/VMEpc 600 Series and Windows 95*
- *NI-488.2M Software for Windows 95*

Windows NT Kit

The Windows NT kit contains the following software:

- Windows NT CD-ROM with boot disk
- *NI-VXI/VISA for VXI/VMEpc 600 Series and Windows NT*
- *NI-488.2M Software for Windows NT*

Printed Documentation

Aside from this document, your kit contains the following printed manuals:

- *Getting Started with Your VXI/VMEpc 600 Series for Windows 95/NT* contains an overview of the VXI/VMEpc hardware and the NI-VXI/VISA software, guides you through setting up your kit, and helps you get started with application development. You can also use this manual as a reference for the hardware and software default settings, to find the answers for commonly asked questions, and for information on reinstalling the software, if necessary.

- The *VXI/VMEpc 600 Series User Manual* contains information on configuring and installing your VXI/VMEpc 600 Series controller. You may not require this manual if you are performing a simple hardware installation that uses the standard default settings. However, you should keep this manual handy in case you decide to try a different jumper setting at a later time. The user manual also contains information on BIOS, the front panel and connectors, LEDs, and system resources.

- *Setup Information for Windows NT Users with VXI Embedded Controllers* (for Windows NT users only)

- NI-488.2M documentation includes the following manuals:

  – *NI-488.2M User Manual for Windows 95 and Windows NT*

  – *NI-488.2M Function Reference Manual for Win32*

Acrobat (Online) Documentation

Your kit includes manuals that are not in printed form. These other manuals are available as Adobe Acrobat version 3.0 portable document format (PDF) files. You can view these manuals online, navigate through them, and print them from your computer.

The Acrobat manuals are as follows:

- The *NI-VXI User Manual* describes the features of the NI-VXI software.

- The *NI-VXI Programmer Reference Manual* describes in detail the VXI/VME function calls in the C/C++ and BASIC languages.

- The *NI-VISA User Manual* describes how to program using VISA.

- The *NI-VISA Programmer Reference Manual* describes in detail the attributes, events, and operations you use in NI-VISA.

If you do not already have Adobe Acrobat Reader 3.0 installed on your machine, you can obtain the latest Adobe Acrobat Reader directly from Adobe. Adobe maintains a Bulletin Board Service (BBS), which you can use to upload files to Adobe Technical Support and download certain Adobe files. The number for the Adobe BBS is 206-623-6984. It is located in Seattle, Washington, which is in the Pacific time zone. If you live outside the United States or Canada, you are welcome to connect to the BBS, but may first want to try Adobe's Web site (`http://www.adobe.com/`) or FTP site (`ftp://ftp.adobe.com/`), which may be accessible at a significantly lower connection cost.

Here are some tips on using PDF files efficiently:

- To view and print the online documentation, launch the Acrobat Reader and open the PDF documents. You can assemble your own

printed documentation by printing out either the entire document or only the sections relevant to your application.

- To use hypertext links, click on any entry in the table of contents to jump directly to that topic's page. You can jump to major sections within each manual quickly and easily by clicking on the bookmarks. When you open your PDF documents, Acrobat displays the bookmarks. A bookmark with a triangle in front of it has bookmarks within it. To reveal the next level, click on the triangle. To collapse to a higher level, again click on the triangle.

- You can change the magnification of the documents either by selecting the magnifying glass tool in the toolbar and clicking anywhere on the manual page or by using the various options in the **View** menu.

- You can quickly find and go to every occurrence of any word or phrase in the entire document. Select the **Tools**»**Find...** option and type in the key word or phrase you want to find.

- The PDF file assigns consecutive numbers to each page throughout the document and displays this page number in the lower-left corner of the Acrobat window. When you choose the **View**»**Go To Page...** option or are printing pages, you must use the page numbers in the lower-left corner of the window, not the number displayed at the bottom center of each manual page or listed in the contents and index. The number shown in the bottom center of the page is for printed documents only.

Available Options

You may have ordered the following optional software:

- LabVIEW Development software (CD-ROM). You also receive the LabVIEW Configuration Disk so you can activate LabVIEW.

- LabWindows/CVI Development software (CD-ROM). You also receive the LabWindows/CVI Configuration Disk so you can activate LabWindows/CVI.

All of the software listed previously is already installed on your VXI/VMEpc controller's hard drive, including LabVIEW and LabWindows/CVI. The National Instruments NI-DAQ software is also pre-installed on all VXI/VMEpc controllers to support the National Instruments data acquisition boards, including VXI-DAQ. A copy of all software media, excluding LabVIEW and LabWindows/CVI, is available in the c:\images directory for immediate access.

Both LabVIEW and LabWindows/CVI integrate the VXI and VISA library interfaces that are required to support your VXI/VMEpc embedded controller. You also get hundreds of complete instrument drivers, which are modular, source-code programs that handle the communication with your instrument to speed your application development.

Where Do You Start?

1. Compare your kit contents with the description in the preceding *What Do You Have?* section. Contact National Instruments regarding any discrepancies.

2. If you ordered the LabVIEW or LabWindows/CVI software option, your kit includes either a LabVIEW or LabWindows/CVI configuration disk. Because both LabVIEW and LabWindows/CVI are pre-installed on your hard drive in a nonfunctional mode, you need to insert the configuration disk when prompted to activate the installed software.

☞ **Note**   *If you do not insert the configuration disks when prompted, Setup deletes the pre-installed LabVIEW and LabWindows/CVI software from your hard drive. However, you can reinstall them from disk, as described in the next section.*

If you ordered a kit that does not contain the configuration disks, you can install LabVIEW or LabWindows/CVI from disk if you want to use either programming environment. Then reinstall the NI-VXI/VISA software from disk or from the copy located in the c:\images directory. The NI-VXI/VISA installer updates necessary files in the LabVIEW and LabWindows/CVI directories to ensure that the latest development files are being used.

For more detailed setup information, refer to Chapter 2, *Setting up Your VXI/VME System*, in your getting started manual. This chapter describes the steps for installing both the hardware and software. If you ordered either LabVIEW or LabWindows/CVI, pay particular attention to the *Booting Your System for the First Time* section.

3. After you finish setting up your system, refer to Chapter 3, *Developing Your Application*, in your getting started manual to learn how you can use your VXI/VME system and to ensure it is operating properly.

4. Please refer to the README.TXT files for important information that may affect your application program, including known issues and software corrections with this release, and additional information relevant for NI-VXI and NI-VISA API development:

You can also reference the National Instruments sites http://www.natinst.com or ftp.natinst.com for driver updates, examples, and product news.

What Is NI-VXI?

The NI-VXI system-level software is the driver that controls your VXI/VMEpc and VXI/VME system. NI-VXI includes a VXI Resource Manager, an interactive configuration and troubleshooting program,

libraries of software routines for test and measurement programming, interactive control programs for both NI-VXI and NI-VISA, and a logging utility you can use for debugging your applications. You can use this software to seamlessly program multiple-mainframe configurations and have software compatibility across a variety of controller platforms.

♦ **Note to VME users**—VXI is an extension of the VMEbus architecture. VXI uses a unique method to automatically detect and configure VXI instruments. NI-VXI, the National Instruments software for VXI and VME controllers, has a utility known as *Resman* that performs this procedure. Because VME does not use this same device detection and configuration scheme, you need to perform a few extra configuration steps so that the NI-VXI software will recognize VME devices in the system. NI-VXI has a utility known as *T&M Explorer* that not only displays the system devices but also includes an editor for configuring your VME devices in the system. For more information regarding these utilities and others, please refer to Chapter 3, *Developing Your Application*, in your getting started manual.

What Is VISA?

VISA is a standard I/O Application Programming Interface (API) for instrumentation programming. VISA by itself does not provide instrumentation programming functionality for interfaces other than Serial. VISA is a high-level API that calls into system-level drivers. For example, the NI-VISA implementation of VISA uses the NI-VXI system-level driver for National Instruments VXI/VME controllers, and the NI-488.2 system-level driver for National Instruments GPIB and GPIB-VXI devices.

VISA can control VXI/VME, PXI, GPIB, or Serial instruments, making the appropriate driver calls depending on the type of instrument being used. VISA uses the same operations to communicate with instruments regardless of the interface type. For example, the VISA command to write an ASCII string to a message-based instrument is the same whether the instrument is Serial, GPIB, or VXI. As a result, VISA gives you interface independence. This makes it easier to switch bus interfaces and means that users who must program instruments for multiple interfaces need learn only one API.

Another advantage of VISA is that it is an object-oriented API that will easily adapt to new instrumentation interfaces as they evolve, making application migration to the new interfaces easy.

Because VISA is the industry standard for developing instrument drivers, most instrument drivers currently written by National Instruments use VISA and therefore support Windows 95, Windows NT, Windows 3.*x*,

Solaris 1, Solaris 2, HP-UX, and Macintosh, if the system-level drivers are available for that platform.

NI-VXI/VISA Release Notes

This section describes the new utilities and features in this release of NI-VXI/VISA for Windows 95/NT.

New Utilities

This release of NI-VXI/VISA for Windows 95/NT includes two new utilities to help you configure, develop, and debug your system quickly: T&M Explorer and NI Spy.

You can use T&M Explorer to view your entire T&M system and configure various components. When you launch T&M Explorer, a list of your VXI, GPIB, and Serial devices appears on your screen. To view the properties of each device (such as logical address, address space used, primary address), right-click on the device in the list. When you view the properties of a National Instruments device, you can configure the hardware settings directly from the properties list.

♦ **VME users**—Remember that since the Resource Manager cannot auto-detect VME devices, they do not automatically appear when you launch T&M Explorer. Go to **Edit**»**Add Device...**»**Add Device Wizard** to integrate each VME device into your system.

T&M Explorer replaces many earlier utilities, such as VXIedit and VISAconf, and integrates with the NI-DAQ Configuration Utility (for VXI-DAQ instruments). T&M Explorer also has new features, such as an option to run Resource Manager at startup, and troubleshooting to guide you through configuration conflicts and errors.

NI Spy tracks the calls your application makes to National Instruments T&M drivers, including NI-VXI, NI-VISA, and NI-488.2. It highlights functions that return errors, so you can quickly determine which functions failed during your development. NI Spy can also log your program's calls to these drivers, so you can check them for errors at your convenience.

Supported Application Development Environments

This release of NI-VXI/VISA for Windows 95/NT supports the following Application Development Environments (ADEs):

• LabVIEW version 4.*x*, 5.*x*

• LabWindows/CVI version 4.*x*, 5.*x*

• Borland C/C++ version 4.5.*x*

- Microsoft Visual C/C++ version 4.*x*, 5.*x*
- Microsoft Visual Basic version 4.*x*, 5.*x*

☞ **Note** *Although NI-VXI and NI-VISA have been tested and found to work with these ADEs, other ADEs or higher versions of the ADEs listed above may also work.*

New Features and Terminology

NI-VXI/VISA exploits the MITE ASIC to provide a rich implementation of the VXI interface, especially in regard to the following:

- Window mapping
- MITE DMA
- Shared memory
- Remote controllers

Window Mapping

The MITE architecture allows much more flexibility in low-level mapping of VXI/VME address spaces. In particular, the CPU interface of the MITE has windows that can be dynamically resized and relocated from CPU space to VXI/VME space. The low-level functions have new extensions that reflect this feature. Refer to the NI-VXI online help or the *NI-VXI Programmer Reference Manual* for information about these functions in NI-VXI. The NI-VISA online help and the *NI-VISA Programmer Reference Manual* cover this information for NI-VISA applications. Use the Acrobat Reader to view and navigate through these manuals.

As in earlier versions of the drivers, the functions MapVXIAddress() and viMapAddress() check whether a window that can be shared already maps to the desired address space and location. If so, they return a pointer to that window. If the desired space is not already mapped, they set up a new MITE window to the VXI/VME address and return a pointer to the new window.

The MapVXIAddressSize() function is the standard mechanism for specifying how large a window the driver should map on a call to MapVXIAddress(). The default size of a mapped window when using NI-VXI is 64 KB. In VISA, you specify the window size directly in viMapAddress().

The success of this allocation depends on the availability of three factors:

- Address space in the user window
- Number of MITE windows
- Memory for allocating data structures for the map

Address Space

The VXI/VMEpc 600 controller can decode any 32-bit address on the PCI bus as a VXI/VME cycle, giving 4 GB of addressability that can be used for windows on your controller. The operating system or computer architecture may limit which addresses can be assigned to your VXI/VMEpc 600 Series.

To change the address space, go to T&M Explorer»VXI/VMEpc **Hardware Configuration**»**PCI**»**user window size**. This setting limits the total amount of memory you can map with `MapVXIAddress()` or `viMapAddress()`. If the user window is disabled, the `MapVXIAddress()` function returns `NO_HARDWARE_SUPPORT (-1)`.

The *NI-VXI Programmer Reference Manual* implies that the error code `MAP_TIMEOUT (-8)` is returned when the window is in use. Because the MITE-based products have multiple hardware windows of variable size, the meaning of this error has been modified. `MapVXIAddress()` now returns the error code `MAP_TIMEOUT (-8)` whenever there are not enough resources to map the window.

For example, if you use `MapVXIAddressSize()` and `MapVXIAddress()` to request a 1 MB window to A32 space and you request a user window in T&M Explorer of only 64 KB, `MapVXIAddress()` returns the error code `MAP_TIMEOUT` because there are not enough resources to complete the request.

Number of MITE Windows

The MITE has eight CPU windows. NI-VXI uses four of these windows, leaving four for user applications.

Memory for Allocating Data Structures

You need to have sufficient memory available to set up the necessary page tables. If you request a very large window—hundreds of megabytes, for example—you may run out of memory.

MITE DMA

The MITE has two DMA channels to improve the throughput of block transfers to and from the VXI/VME system. The DMA channels can use various high-speed bus protocols, such as the following:

* VME block

* Burst mode (on the PCI bus)

* VME64 (on the VXI/VME bus)

The DMA channels can transfer data between a VXI/VME device and local memory, or between VXI/VME devices. The DMA channel can handle contiguous or noncontiguous local memory. If it is handling noncontiguous memory, it can perform scatter-gather operations on the noncontiguous memory.

The `VXImove()` and `viMoveXX()` functions automatically use appropriate bus protocols and transfer types to efficiently perform the data transfer specified in the function. You can also use configuration options in T&M Explorer to instruct the NI-VXI/VISA software to use DMA channels for particular types of operations and to designate what protocols the channel should use. In addition, you can programmatically control which protocols to use in NI-VXI. See the NI-VXI online help or the *NI-VXI Programmer Reference Manual* for complete descriptions of `VXImove()` and other high-level functions. Notice that previously written NI-VXI and NI-VISA code uses the DMA capabilities of the MITE without modification.

To take full advantage of the throughput of the DMA channels, you should perform 32-bit transfers where both the source and the destination are longword aligned. If you need to transfer character data between devices of different byte orders—for example, between a big-endian device and an Intel 80*x*86-based Windows NT PC—transfer the data as longwords but adjust the byte-ordering parameters in `VXImove()` to get the correct data in the most efficient manner.

NI-VXI Examples:

```
/* Transferring 32-bit data to a big-endian A32 device */
VXImove(0x0, userBuffer, 0x3, deviceOffset, numDataPoints, 4);

/* Transferring 8-bit data to a big-endian A32 device */
VXImove(0x80, userBuffer, 0x3, deviceOffset, numDataPoints / 4, 4);
```

Shared Memory

In the **Hardware Configuration** settings of the VXI/VMEpc-650 in T&M Explorer, you can share memory on your computer. Right-click on any of the settings or consult the online help in T&M Explorer for more information. You can access shared memory on your computer using `VXImemAlloc()` in NI-VXI and `viMemAlloc()` in VISA.

Remote Controllers

This section applies to multiframe systems where you use interrupts, triggers, or utility bus signals.

Remote controllers, when configured to detect asynchronous events such as a VXI/VME interrupt or VXI trigger, need to inform the local controller that such an asynchronous event has occurred. The remote controllers

report these events back to the local controller via a VXI/VME IRQ line. This IRQ line is called the *system IRQ line*. You can use T&M Explorer to select which VXI/VME interrupt line the remote controller uses to report remote events to the local controller. You need to map the system IRQ line back to the local controller to receive remote controller interrupts. This mapping is performed automatically by the Resource Manager in the parent-side VXI/VME-MXI-2 controllers, but not in other mainframe extenders. You can map interrupts through T&M Explorer, or with the MapVXIint() function, which is described in the NI-VXI online help or the *NI-VXI Programmer Reference Manual*.

The system IRQ line is treated differently than other IRQ lines used by NI-VXI:

• The Resource Manager (Logical Address 0) always acknowledges the system IRQ line.

• You cannot disable the system IRQ line on the Resource Manager. Calling DisableVXIint() on the system IRQ line does *not* disable it.

• Devices other than remote controllers can also interrupt on the system IRQ line, provided that the device at Logical Address 0 is the handler for the interrupt.

• National Instruments recommends that you do *not* route the system IRQ line to the signal queue. Because the system IRQ line cannot be disabled, this routing could lose interrupts.

Passing the value –1 as the logical address of a controller in NI-VXI causes NI-VXI to select the *local* controller. Notice that on external controllers such as the PCI-MXI-2, –1 refers to the first *remote* controller in your system. This is to maintain compatibility with older systems where the external controller needed an extender to assert and receive interrupts.

Enhancements to the NI-VXI Software

The following sections describe the additional options beyond what is documented in the *NI-VXI User Manual* and the *NI-VXI Programmer Reference Manual*.

Compatibility

NI-VXI applications that follow the guidelines documented in the *NI-VXI User Manual* will work with NI-VXI for the VXI/VMEpc 600 Series.

System Configuration Functions

The InitVXIlibrary() function has a new return value of INIT_RET_OK_RMERROR (2). If this value is returned, it means "The NI-VXI library successfully initialized, but the Resource Manager has not

been run successfully." Always run the Resource Manager before using the NI-VXI library, even if you are using only VME.

Low-Level VXIbus Access Functions

Do not make any assumptions about the size and features of a window returned from `MapVXIAddress()`. You should use `GetWindowRange()` to determine the size of a window.

The 32-bit value returned from `GetContext()` and passed to `SetContext()` has a new format. Applications that set the context bits directly for use in `SetContext()` may not be compatible with the new format for context. Because the MITE allows more flexible window mapping, extra bits have been added to this field to reflect these new features. Do not manipulate the context bits directly.

High-Level VXIbus Access Functions

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

- Make sure your buffers are 32-bit aligned.

- Transfer 32-bit data whenever possible.

- Using VXI/VME block access privileges significantly improves performance to devices that are capable of accepting block transfers.

- `VXImove()` must lock the user buffer in memory on virtual memory systems, so locking the buffer yourself optimizes `VXImove()`.

- Because `VXImove()` must build a scatter-gather list for the user buffer on paged memory systems, using a contiguous buffer optimizes `VXImove()`.

`VXImemAlloc()` returns 32-bit aligned, page-locked, contiguous buffers, which work efficiently with `VXImove()`, but only if the function returns `MEM_OK (0)`. A status of `MEM_OK_USE_MEMCOPY (1)` means this buffer cannot be used directly with `VXImove()`.

`VXImove()` can also move blocks of data to and from a single VXI/VME address. This is commonly referred to as *FIFO mode*. For more information refer to the *NI-VXI Programmer Reference Manual*.