

Using Your NI-488.2™ Software with Microsoft Windows

March 1994 Edition

Part Number 320319-01

**© Copyright 1990, 1994 National Instruments Corporation.
All Rights Reserved.**

National Instruments Corporate Headquarters

6504 Bridge Point Parkway

Austin, TX 78730-5039

(512) 794-0100

Technical support fax: (800) 328-2203

(512) 794-5678

Branch Offices:

Australia (03) 879 9422, Austria (0662) 435986, Belgium 02/757.00.20,

Canada (Ontario) (519) 622-9310, Canada (Québec) (514) 694-8521,

Denmark 45 76 26 00, Finland (90) 527 2321, France (1) 48 14 24 24,

Germany 089/741 31 30, Italy 02/48301892, Japan (03) 3788-1921,

Netherlands 03480-33466, Norway 32-848400, Spain (91) 640 0085,

Sweden 08-730 49 70, Switzerland 056/20 51 51, U.K. 0635 523545

Limited Warranty

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

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

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

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THEREFORE 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 Instrument must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments

installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

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

Trademarks

NI-DAQ[®], NI-488[®], and NI-488.2[™] are trademarks of National Instruments Corporation.

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

Warning Regarding Medical and Clinical Use of National Instruments Products

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

Contents

About This Manual	vii
Organization of This Manual	vii
Conventions Used in This Manual	viii
Related Documentation.....	ix
Customer Communication	ix
Chapter 1	
Introduction	1-1
Documentation Useful for Developing Applications.....	1-1
NI-488.2 Software Package for Windows	1-1
NI-488.2 Files for the Windows Operating Environment.....	1-2
NI-488.2 Files for the Development of Windows Applications	1-2
GPIB Sample Windows Application Program.....	1-3
Chapter 2	
Installing and Configuring Your NI-488.2 Software	2-1
Step 1. Run INSTALL	2-1
Quick Installation	2-1
Interactive Installation.....	2-2
Step 2. Set Up the Windows Applications	2-4
Step 3. Configure the Software (Optional)	2-4
Step 4. Verify the Software Installation.....	2-5
Chapter 3	
Using Your NI-488.2 Software	3-1
Using WIBIC	3-1
Writing Windows Programs That Use the GPIB	3-1
The WINSAMP Sample.....	3-2
Microsoft C++ and Borland C++	3-2
General Rules for Using GPIB.LIB with Windows	3-2

Appendix A
DLL Direct Entry NI-488 Functions and
NI-488.2 RoutinesA-1

Appendix B
Customer Communication.....B-1

GlossaryG-1

Tables

Table A-1. Direct Entry NI-488.2 Style RoutinesA-3
Table A-2. Direct Entry NI-488 Style Functions.....A-7
Table A-3. Direct Entry NI-488.2 Style RoutinesA-12
Table A-4. Direct Entry NI-488 Style Functions.....A-15

About This Manual

The NI-488.2 software for Microsoft Windows 3.x (hereafter referred to as *Windows*) is a dynamic link library (DLL). This manual describes how to install the files needed to execute Windows application programs that communicate over the IEEE 488 bus, and how to develop Windows application programs for GPIB instrument control.

Organization of This Manual

This manual is organized as follows:

- Chapter 1, *Introduction*, contains guidelines for executing existing and developing new Windows applications. It also contains a list of files that are copied to your destination and Windows directories when you install the NI-488.2 software.
- Chapter 2, *Installing and Configuring Your NI-488.2 Software*, contains instructions for installing and configuring the NI-488.2 software.
- Chapter 3, *Using Your NI-488.2 Software*, describes two methods that you can use to communicate with GPIB devices from Windows: the Windows Interface Bus Interactive Control (WIBIC) program and a Windows application program that you develop.
- Appendix A, *DLL Direct Entry NI-488 Functions and NI-488.2 Routines*, explains and gives examples of how to use the DLL Direct Entry NI-488 functions and NI-488.2 routines to access the GPIB.DLL file. Following the examples are tables that list all NI-488.2 routines and NI-488 functions, including their calling syntax and ordinal entry values.
- Appendix B, *Customer Communication*, contains forms you can use to request help from National Instruments or to comment on our products and manuals.
- The *Glossary* contains an alphabetical list and description of terms used in this manual including abbreviations, acronyms, metric prefixes, mnemonics, and symbols.

Conventions Used in This Manual

The following conventions are used to distinguish elements of text throughout this manual.

bold	Bold text denotes menus, menu items, window names, or dialog box buttons or options.
<i>italic</i>	Italic text denotes emphasis, a cross reference, or an introduction to a key concept.
<i>bold italic</i>	Bold italic text denotes a note, caution, or warning.
monospace	Lowercase text in this font denotes text or characters that are to be literally input from the keyboard, sections of code, and programming examples. This font is also used for the proper names of disk drives, directories, programs, and filenames, and for statements and comments taken from program code.
<>	Angle brackets enclose the name of a key on the keyboard—for example, <Delete>.
<Return>	Key names begin with a capital letter.
NI-488.2	NI-488.2 is used throughout this manual to refer to the NI-488.2 software for Microsoft Windows 3.x unless otherwise noted.
Windows	Windows is used throughout this manual to refer to the Microsoft Windows software package, version 3.0 or later.

Related Documentation

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

- *NI-488.2 Software Reference Manual for MS-DOS*, National Instruments (part number 320282-01)
- Getting Started manual that came with your GPIB interface board
- *Microsoft Windows 3 Users Manual*
- ANSI/IEEE Standard 488.1-1987, *IEEE Standard Digital Interface for Programmable Instrumentation*
- ANSI/IEEE Standard 488.2-1987, *IEEE Standard Codes, Formats, Protocols, and Common Commands*

Customer Communication

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

Chapter 1

Introduction

This chapter contains guidelines for executing existing and developing new Windows applications. It also contains a list of files that are copied to your destination and Windows directories when you install the NI-488.2 software.

Documentation Useful for Developing Applications

Note: *For information on installing and configuring your hardware, refer to the Getting Started manual that came with your GPIB interface board.*

If you are using this manual to execute existing Windows applications, you need the following items:

- Getting Started manual that came with your GPIB interface board
- *NI-488.2 Software Reference Manual for MS-DOS*
- *Microsoft Windows 3 Users Manual*
- *Microsoft Windows 3 Software Development Kit*

If you are using this manual to develop Windows applications, you need the software development kit for your development environment and the items listed above.

NI-488.2 Software Package for Windows

Before installing the software, you should understand the files that will be copied from the distribution disk(s) and the purpose of each file. The following section describes the files contained on the distribution disk(s).

NI-488.2 Files for the Windows Operating Environment

The `INSTALL` program copies the following files to the specified Windows directory.

- `GPIB.DLL` is a dynamic link library (DLL) that is accessed by a NI-488.2 application for Windows as the application executes. The DLL contains all of the NI-488 functions and NI-488.2 routines.
- `GPIB.INI` is the private profile file which is used by `GPIB.DLL` to determine the software configuration parameters for each GPIB board and device in the system. You can modify `GPIB.INI` by using either the `WIBCONF.EXE` file or a text editor.

NI-488.2 Files for the Development of Windows Applications

The `INSTALL` program copies the following files to the specified destination directory.

- `GPIB.LIB` is the import library for the DLL. It must be linked into your NI-488.2 application for Windows just like any other library.
- `WINDECL.H` is an include file that contains prototypes of the NI-488 functions and NI-488.2 routines, and useful constants that you may want to use in your NI-488.2 application for Windows. It must be included at the beginning of any file that makes NI-488 function calls.
- `WIBCONF.EXE`, a DOS application, is a software configuration program that you can use to change the software parameters and other data used by the DLL. It has the same basic functionality as the NI-488.2 driver for MS-DOS `IBCONF` program, which is described in Chapter 2 of the *NI-488.2 Software Reference Manual for MS-DOS*.
- `WIBCONF.PIF` contains configuration information about the program `WIBCONF.EXE` that is used by Windows.
- `IBDIAG.EXE`, a DOS application, is a program that tests the hardware settings on your GPIB board. It ensures that the board is properly installed and that the hardware is accessible.

- `IBDIAG.PIF` contains configuration information about the program `IBDIAG.EXE` that is used by Windows.
- `WIBTEST.EXE`, a Windows application, is a program that tests the software installation. It verifies that the software configuration is consistent with the GPIB hardware.
- `WIBIC.EXE`, a Windows application, is the Windows Interface Bus Interactive Control program that executes NI-488 functions and NI-488.2 routines that you enter from the keyboard. It can help you learn how to use the NI-488 functions and NI-488.2 routines, program instruments or other GPIB devices, and develop your particular Windows application program.
- `NIVDMAD.386`, the replacement virtual DMA device, is an improved version of the virtual DMA device (`vdmad`) used in Windows 386-enhanced mode. `nivdmad.386` must be loaded for the `GPIB.DLL` to use DMA when running Windows version 3.0. With Windows versions later than 3.0, it is not necessary to use the `nivdmad.386`. The `INSTALL` program offers to modify the `SYSTEM.INI` file to use `nivdmad.386`; if you have the Windows 3.0 version on your computer, you should let `INSTALL` make the modification. If you are also installing NI-DAQ for Windows, you must use `NIVDMAD.386`, regardless of your Windows version. NI-DAQ for Windows software is used with National Instruments plug-in data acquisition boards.

GPIB Sample Windows Application Program

The `INSTALL` program also copies the following sample files into a new subdirectory, named `C`, in the specified destination directory.

- `WINSAMP.EXE` is a compiled Windows application program for Windows that communicates over the GPIB. It is based primarily on the `GENERIC` Windows application example provided with the Windows Software Development Kit.
- `WINSAMP` is the makefile used to compile and link the sample Windows application.

- `WINSAMP.C` is the C language source file containing the Windows functions `WinMain`, `MainWndProc`, `About`, `InitApplication`, and `InitInstance`.
- `GPIBSAMP.C` is the C language source file containing NI-488 function calls to the DLL.
- `WINSAMP.H` is the header file containing definitions and declarations required by `WINSAMP.C`.
- `WINSAMP.RC` is the resource script file that defines the menus and the dialog-box template for the `About` dialog box.
- `WINSAMP.DEF` is the module definition file that contains module definitions.

Chapter 2

Installing and Configuring Your NI-488.2 Software

This chapter contains instructions for installing and configuring the NI-488.2 software. For information on installing and configuring your hardware, refer to the Getting Started manual that you received with your GPIB interface board.

Step 1. Run INSTALL

You can install the NI-488.2 software for Windows using one of two methods: quick installation or interactive installation. Please review the section titled *NI-488.2 Software Package for Windows* in Chapter 1, *Introduction*, before you install.

Quick Installation

This quick version of the INSTALL program assumes that Windows is installed in the default directory (C:\WINDOWS). INSTALL copies files to C:\WINDOWS and the GPIB destination directory, named C:*boardname*W (for example, C:\AT-GPIBW for the AT-GPIB or C:\MC-GPIBW for the MC-GPIB). If Windows is not in its default directory or you do not want the default destination directory to be created, you must install the NI-488.2 software for Windows interactively (refer to the *Interactive Installation* section later in this chapter).

After starting your computer, run the INSTALL program on the distribution diskette by entering the following command:

```
X:install /qw
```

where X is the name of the drive containing the distribution diskette (this is usually A).

The quick version of INSTALL copies the NI-488.2 files and then automatically runs the hardware diagnostics (IBDIAG.EXE). If no error

message appears, the NI-488.2 driver is successfully installed and you can proceed to [Step 2. Set Up the Windows Applications](#). If an error occurs during the quick installation, you may need to run the INSTALL program interactively (refer to the next section, *Interactive Installation*). For more information on error codes, refer to Chapter 3, *Understanding the NI-488.2 Software*, in the *NI-488.2 Software Reference Manual for MS-DOS*.

Interactive Installation

If you choose to interactively install the NI-488.2 software, complete the following instructions.

After starting your computer, run the INSTALL program on the distribution diskette by entering the following command:

```
X:install
```

where *X* is the name of the drive containing the distribution diskette (this is usually A).

This is the interactive version of the INSTALL program. When the program prompts you for the type of software to install (**DOS** or **Windows**), select **Windows**. After you select the Windows option, INSTALL displays a main menu with four options: **Partial GPIB Installation**, **Full GPIB Installation**, **Diagnostics**, and **Return to DOS**.

Select the type of installation that you want to use (**Partial** or **Full**). If you select **Partial GPIB Installation**, you are prompted to choose which parts of the NI-488.2 software to install. If you select **Full GPIB Installation**, all of the NI-488.2 software is installed.

Next, you are prompted to give the location of your Windows directory and you are prompted for the name of the directory where Windows is stored and the name of a new directory into which the files can be copied. The INSTALL program creates the specified destination directory and copies the files listed in Chapter 1, [Introduction](#), to their appropriate directories.

After the files have been copied, the INSTALL program offers to modify SYSTEM.INI so that the correct virtual DMA device is used. If you do not let INSTALL make the changes and you are using Windows version 3.0, you should make them yourself. If you have a version of Windows later than 3.0 and are not planning to install NI-DAQ software for Windows with

a National Instruments plug-in data acquisition boards, this modification is not necessary.

Now select the **Diagnostics** option from within the main menu of the INSTALL program to run the hardware diagnostics.

INSTALL leads you through a hardware diagnostics check of your GPIB interface board using the program IBDIAG. If you are using an interface board with the default hardware settings, press <Enter> in response to each of the prompts. If you are not using the default settings, type in the correct response to each prompt and then press <Enter>.

If IBDIAG returns an error message, check the following:

- Check the GPIB interface board to ensure that it is not connected to a GPIB device. IBDIAG requires that the GPIB interface board *not* be connected to a GPIB device.
- Check the hardware configuration to ensure that the switch and jumper settings are correct:
 - The hardware configuration settings should be the same as the values you entered when prompted by the IBDIAG program.
 - The hardware configuration settings should not be configured to the same setting as another board or device in your computer. Try reconfiguring the hardware and running IBDIAG again.

Note: *You can run IBDIAG either by selecting the **Diagnostics option of the INSTALL menu or directly from DOS. IBDIAG . EXE is copied to the destination directory by the INSTALL program.***

Refer to the Getting Started manual that came with your interface board for more information on changing hardware and software settings. If IBDIAG still returns an error message, refer to Appendix B, [Customer Communication](#).

After you have completed IBDIAG with no errors, proceed to [Step 2. Set Up the Windows Applications](#).

Step 2. Set Up the Windows Applications

To set up the NI-488.2 applications for Windows, complete the following steps:

1. Run **Windows Setup** in the **Main** window.
2. Select **Set Up Applications** from the **Options** pull down menu.
3. Add WIBIC and WIBTEST to the **Windows Applications** window.
4. WIBCONF and IBDIAG are DOS applications. If you want to set them up as non-windows applications, refer to Chapter 3 in the *Microsoft Windows 3 Users Manual*.

Refer to the *Microsoft Windows 3 Users Manual* for a more detailed description of the Windows Setup procedure.

Step 3. Configure the Software (Optional)

If you changed the hardware settings for your interface board, you must run WIBCONF to configure the software to match these new hardware settings. WIBCONF is *not* a Windows application. You can run it to configure the software either from DOS by executing WIBCONF . EXE, or directly from Windows.

If you run WIBCONF from Windows, a prompt appears asking for a parameter, which is the path and name of the GPIB . INI file to be configured.

If your Windows files are located in the C : \WINDOWS directory, press <Enter>.

If you did not install Windows in the C : \WINDOWS directory, enter *x* : \ *path* \ GPIB . INI, where *x* and *path* are the drive and directory path of your Windows files.

To avoid entering the path each time you run WIBCONF, the file WIBCONF . PIF can be edited with the Windows PIF editor to reflect your Windows directory.

WIBCONF has the same basic functionality as the NI-488.2 driver for MS-DOS IBCONF program, which is fully described in your *NI-488.2 Software Reference Manual for MS-DOS*. However, the following differences exist between the IBCONF for DOS and the IBCONF for Windows:

- Instead of configuring the handler file, GPIB.COM, WIBCONF actually modifies the configuration file, GPIB.INI.
- WIBCONF always searches for the GPIB.INI file in the WINDOWS directory first and then searches in the current directory.
- Auto-configuration for NI-488.2 GPIB DLLs is not supported.
- The option of configuring the installed driver does not apply.

Step 4. Verify the Software Installation

The program WIBTEST can be used to verify that the software is properly installed and configured for your interface board. WIBTEST is a Windows application and can be run by selecting the WIBTEST icon in the **Windows Applications** window.

WIBTEST requires no interaction with the user and takes about 10 seconds to complete. Any GPIB cables should be disconnected from the interface board while the test is in progress. If any step of WIBTEST fails, follow the instructions on the screen. If you need to change a setting, consult the Getting Started manual that came with your interface board.

Chapter 3

Using Your NI-488.2 Software

This chapter describes two methods that you can use to communicate with GPIB devices from Windows: the Windows Interface Bus Interactive Control (WIBIC) program and a Windows application program that you develop.

Using WIBIC

The WIBIC program is the Windows version of the NI-488.2 driver for MS-DOS IBIC program. It has the same general appearance and exactly the same functionality as the DOS IBIC program. Refer to Chapter 6, *IBIC*, of your *NI-488.2 Software Reference Manual for MS-DOS* for a complete description of how to use IBIC. The WIBIC program is designed to help you learn the GPIB routines, program GPIB instruments and other devices, and develop your application program.

To run WIBIC, change to the *Windows Applications* window and select the WIBIC icon.

Writing Windows Programs That Use the GPIB

There are two methods for writing a Windows application that uses the GPIB. The first method is to write an application that uses the standard NI-488 functions and NI-488.2 routines and is linked to one of the Windows 3 language interfaces. The NI-488.2 software includes the Microsoft C language interface. Contact National Instruments for information on other language interfaces that you can use with Windows 3.

The second method of writing a NI-488.2 for Windows application is to use the DLL direct entry NI-488 functions and NI-488.2 routines. Using direct entry, you do not need to have a special language interface to link with your application. Refer to Appendix A, [*DLL Direct Entry NI-488 Functions and NI-488.2 Routines*](#), for more information.

The remainder of this chapter describes the WINSAMP sample program that illustrates how GPIB calls can be made from a simple Microsoft C Windows application using the National Instruments Microsoft C language interface. It also details how the Microsoft C language interface is used to create Microsoft Visual C++ and Borland C++ Windows applications. The chapter concludes with a general set of rules to follow when using the DLL in your own Windows application.

The WINSAMP Sample

There are two primary parts to the WINSAMP sample: WINSAMP . C and GPIBSAMP . C. WINSAMP . C handles most of the details for interfacing with Windows and GPIBSAMP . C makes GPIB calls and then displays the results on the screen.

To execute WINSAMP, set it up as a Windows Application (refer to Chapter 2, *Installing and Configuring Your NI-488.2 Software, Step 2. Set Up the Windows Applications*), change to the *Windows Applications* window, and select the WINSAMP icon.

To make changes to WINSAMP and rebuild it, add the desired changes and enter the following command.

```
make WINSAMP
```

Microsoft C++ and Borland C++

You need the Microsoft C file WINDECL . H and the import library GPIB . LIB to create Windows applications that use GPIB . DLL. Include the file WINDECL . H in your Windows application and add the library GPIB . LIB to your project file.

General Rules for Using GPIB.LIB with Windows

By following these general rules, any application can use the GPIB . DLL.

- Make the same GPIB calls that you do under DOS (refer to the *NI-488.2 Software Reference Manual for MS-DOS* for a list of these calls).

- Add GPIB.LIB to the library list in the link command line.

Note: *All NI-488.2 GPIB.DLL files for Windows share the same .LIB file; therefore, you do not have to relink applications to switch between GPIB boards.*

- Ensure that the correct GPIB.DLL is in the directory in which Windows is installed or in the DOS search path when the application is run. Unlike the GPIB.LIB file, GPIB.DLL files are unique for each National Instruments GPIB board.
- Ensure that GPIB.INI is in the directory in which Windows is installed when the application is run so that it can be used to properly initialize the GPIB.DLL file. The GPIB.INI file is also unique for each GPIB board.

Appendix A

DLL Direct Entry NI-488 Functions and NI-488.2 Routines

This appendix explains and gives examples of how to use the DLL Direct Entry NI-488 functions and NI-488.2 routines to access the GPIB.DLL file. Following the examples are tables that list all NI-488.2 routines and NI-488 functions, including their calling syntax and ordinal entry values.

The DLL Direct Entry NI-488 functions and NI-488.2 routines can be used to access the GPIB.DLL file from any language or programming environment that runs under Windows 3 and supports access to standard Windows DLL functions. As with all functions exported by a DLL, these functions conform to the PASCAL calling conventions. A few examples of using these entry points follow. Tables A-1 and A-2 contain a complete list of all of the entry points in C. Tables A-3 and A-4 contain the declarations for the NI-488.2 routines and NI-488 functions in Visual BASIC.

For specific information on the variables `ibsta`, `iberr`, and `ibcntl`, refer to Chapter 3 in the *NI-488.2 Software Reference Manual for MS-DOS*. For specific information on a routine or function, refer to Chapters 4 and 5 in the *NI-488.2 Software Reference Manual for MS-DOS*. For information about accessing dynamic link library (DLL) functions from a given language or environment or using ordinal entry values which some environments do not support, see the documentation provided with that package.

Example 1, accessing the GPIB.DLL file from Turbo Pascal for Windows:

```
(* First, import the DLL functions you plan to use. *)

function DLLibfind(udname: PChar;
                  var ibsta: integer;
                  var iberr: integer;
                  var ibcntl: longint) : integer; far;
external 'GPIB' index 22;

function DLLibsic(ud: integer;
                 var: ibsta: integer;
                 var: iberr: integer;
                 var: ibcntl: longint) : integer; far;
```

```

external 'GPIB' index 42;
procedure DLLSendIFC(board: integer;
                    var ibsta: integer;
                    var iberr: integer;
                    var ibcntl: longint) ; far;
external 'GPIB' index 119;

(* Your application can now use the functions. *)

var BoardHandle: integer;
var ibsta: integer;
var iberr: integer;
var ibcntl: longint;
var temp: integer;

BoardHandle:= DLLlibfind('GPIB0', ibsta, iberr, ibcntl);
temp:= DLLibsic(BoardHandle, ibsta, iberr, ibcntl);

                (* or *)

DLLSendIFC(0, ibsta, iberr, ibcntl);

```

Example 2, accessing the GPIB.DLL file from Microsoft Visual Basic:

```

'First declare the DLL functions you plan to use.

Declare function DLLlibfind Lib "gpib.dll"
    (ByVal udname$, ibsta%, iberr%, ibcntl&) As Integer
Declare function DLLibsic Lib "gpib.dll"
    (ByVal ud%, ibsta%, iberr%, ibcntl&) As Integer
Declare sub DLLSendIFC Lib "gpib.dll"
    (ByVal board%, ibsta%, iberr%, ibcntl&)

'Your application can now use the functions.

Global BoardHandle As Integer
Global ibsta As Integer
Global iberr As Integer
Global ibcntl As Long

BoardHandle% =
    DLLlibfind("GPIB0", ibsta%, iberr%, ibcntl&)

temp% = DLLibsic(BoardHandle%, ibsta%, iberr%, ibcntl&)

'or

call DLLSendIFC(0, ibsta%, iberr%, ibcntl&)

```

Note: *All of the routines listed in Table A-1 are of type void _far _pascal.*

Table A-1. Direct Entry NI-488.2 Style Routines in C

Routine (ordinal entry value)	Syntax
AllSpoll (100)	DLLAllSpoll (short board, short_far* addresslist, short_far* resultlist, short_far*ibsta, short_far*iberr, unsigned long_far*ibcntl)
DevClear (101)	DLLDevClear (short board, short address, short_far*ibsta, short_far *iberr, unsigned long_far*ibcntl)
DevClearList (102)	DLLDevClearList (short board, short_far*addresslist, short_far*ibsta, short_far*iberr, unsigned long_far*ibcntl)
EnableLocal (103)	DLLEnableLocal (short board, short_far*addresslist, short_far*ibsta, short_far*iberr, unsigned long_far*ibcntl)
EnableRemote (104)	DLLEnableRemote (short board, short_far*addresslist, short_far*ibsta, short_far*iberr, unsigned long_far*ibcntl)
FindLstn (105)	DLLFindLstn (short board, short_far*addresslist, short_far*resultlist, short limit, short_far*ibsta, short_far*iberr, unsigned long_far*ibcntl)
FindRQS (106)	DLLFindRQS (short board, short_far*addresslist, short_far*result, short_far*ibsta, short_far*iberr, unsigned long_far*ibcntl)

(continues)

Table A-1. Direct Entry NI-488.2 Style Routines in C (Continued)

Routine (ordinal entry value)	Syntax
GenerateREQF (53)	DLLGenerateREQF (short board, short addr, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
GenerateREQT (52)	DLLGenerateREQT (short board, short addr, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
GotoMultAddr (129)	DLLGotoMultAddr (short board, unsigned short type, unsigned short (_far_loads *addrfunc()), unsigned short (_far_loads *spollfunc()), short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
PassControl (107)	DLLPassControl (short board, short address, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
PPoll (108)	DLLPPoll (short board, short_far *result, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
PPollConfig (109)	DLLPPollConfig (short board, short address, short dataline, short sense, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
PPollUnconfig (110)	DLLPPollUnconfig (short board, short_far *addresslist, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)

(continues)

Table A-1. Direct Entry NI-488.2 Style Routines in C (Continued)

Routine (ordinal entry value)	Syntax
RcvRespMsg (111)	DLLRcvRespMsg (short board, char_far *data, long count, short termination, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ReadStatusByte (112)	DLLReadStatusByte (short board, short address, short_far *result, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
Receive (113)	DLLReceive (short board, short address, char_far *data, unsigned long count, short termination, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ReceiveSetup (114)	DLLReceiveSetup (short board, short address, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ResetSys (115)	DLLResetSys (short board, short_far *addresslist, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
Send (116)	DLLSend (short board, short address, char_far *data, long count, short eotmode, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
SendCmds (117)	DLLSendCmds (short board, char_far *commands, unsigned long count, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)

(continues)

Table A-1. Direct Entry NI-488.2 Style Routines in C (Continued)

Routine (ordinal entry value)	Syntax
SendDataBytes (118)	DLLSendDataBytes (short board, char_far *data, long count, short eotmode, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
SendIFC (119)	DLLSendIFC (short board, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
SendList (120)	DLLSendList (short board, short_far *addresslist, char_far *data, long count, short eotmode, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
SendLLO (121)	DLLSendLLO (short board, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
SendSetup (122)	DLLSendSetup (short board, short_far *addresslist, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
SetRWLS (123)	DLLSetRWLS (short board, short_far *addresslist, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
TestSRQ (124)	DLLTestSRQ (short board, short_far *result, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
TestSys (125)	DLLTestSys (short board, short_far *addresslist, short_far *resultlist, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)

(continues)

Table A-1. Direct Entry NI-488.2 Style Routines in C (Continued)

Routine (ordinal entry value)	Syntax
Trigger (126)	DLLTrigger (short board, short address, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
TriggerList (127)	DLLTriggerList (short board, short_far *addresslist, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
WaitSRQ (128)	DLLWaitSRQ (short board, short_far *result, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)

Note: *All of the functions listed in Table A-2 are of type short_far_pascal.*

Table A-2. Direct Entry NI-488 Style Functions in C

Functions (ordinal entry value)	Syntax
ibbna (10)	DLLibbna (short ud, char_far *bname, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibcac (11)	DLLibcac (short ud, short v, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibclr (12)	DLLibclr (short ud, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)

(continues)

Table A-2. Direct Entry NI-488 Style Functions in C (Continued)

Routine (ordinal entry value)	Syntax
ibcmd (13)	DLLibcmd (short ud, char_far *cmd, long cnt, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibcmda (14)	DLLibcmda (short ud, char_far *cmd, long cnt, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibconfig (15)	DLLibconfig (short ud, unsigned short option, unsigned short value, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibdev (16)	DLLibdev (short boardindex, short pad, short sad, short tmo, short eot, short eos, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibdma (18)	DLLibdma (short ud, short v, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibeos (19)	DLLibeos (short ud, short v, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibeot (20)	DLLibeot (short ud, short v, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibevent (21)	DLLibevent (short ud, short event, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibfind (22)	DLLibfind (char_far *udname, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)

(continues)

Table A-2. Direct Entry NI-488 Style Functions in C (Continued)

Functions (ordinal entry value)	Syntax
ibgts (23)	DLLibgts (short ud, short v, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibist (24)	DLLibist (short ud, short v, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
iblines (25)	DLLiblines (short ud, short_far *clines, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibln (26)	DLLiblin (short ud, short pad, short sad, short_far *listen, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibloc (27)	DLLibloc (short ud, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibonl (28)	DLLibonl (short ud, short v, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibpad (29)	DLLibpad (short ud, short v, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibpct (30)	DLLibpct (short ud, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibppc (32)	DLLibppc (short ud, short v, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibrd (33)	DLLibrd (short ud, short_far *rd, unsigned long cnt, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)

(continues)

Table A-2. Direct Entry NI-488 Style Functions in C (Continued)

Functions (ordinal entry value)	Syntax
ibrda (34)	DLLibrda (short ud, char_far *rd, unsigned long cnt, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibrdf (35)	DLLibrdf (short ud, char_far *flname, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibrdkey† (36)	DLLibrdkey (short ud, char_far *rd, unsigned short cnt, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibrpp (37)	DLLibrpp (short ud, char_far *ppr, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibrsc (38)	DLLibrsc (short ud, short v, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibrsp (39)	DLLibrsp (short ud, char_far *spr, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibrsv (40)	DLLibrsv (short ud, short v, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibsad (41)	DLLibsad (short ud, short v, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ibsic (42)	DLLibsic (short ud, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)

(continues)

Table A-2. Direct Entry NI-488 Style Functions in C (Continued)

Functions (ordinal entry value)	Syntax
ib sre (43)	DLLib sre (short ud, short v, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ib stop (44)	DLLib stop (short ud, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ib tmo (45)	DLLib tmo (short ud, short v, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ib trg (46)	DLLib trg (short ud, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ib wait (47)	DLLib wait (short ud, short mask, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ib wrt (48)	DLLib wrt (short ud, char_far *wrt, unsigned long cnt, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ib wrta (49)	DLLib wrta (short ud, char_far *wrt, unsigned long cnt, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
ib wrtf (50)	DLLib wrtf (short ud, char_far *flname, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
(51)	DLLib wrtkey (short ud, char_far *wrt, unsigned short cnt, short_far *ibsta, short_far *iberr, unsigned long_far *ibcntl)
† ibrdkey and ibwrtkey are OEM functions. Refer to the <i>NI-488 Hardware Key Functions Reference Guide</i> , for a detailed description of these functions.	

Table A-3 lists the declarations of the NI-488.2 routines for Visual BASIC.

Note: *When you call a subroutine that has an array as one or more of its parameters, pass the first element of the array. For example, you would include the addresslist array as follows:*

Call DLLFindRQS (board%, addresslist%(0), result%,
ibsta%, iberr%, ibcntl&)

Table A-3. Declarations for NI-488.2 Routines in Visual BASIC

Functions (ordinal entry value)	Syntax
AllSpoll (100)	Declare sub DLLAllSpoll Lib "gpib.dll" (ByVal board%, addresslist As Integer, resultlist As Integer, ibsta%, iberr%, ibcntl&)
DevClear (101)	Declare sub DLLDevClear Lib "gpib.dll" (ByVal board%, ByVal address%, ibsta%, iberr%, ibcntl&)
DevClearList (102)	Declare sub DLLDevClearList Lib "gpib.dll" (ByVal board%, addresslist As Integer, ibsta%, iberr%, ibcntl&)
EnableLocal (103)	Declare sub DLEnableLocal Lib "gpib.dll" (ByVal board%, addresslist As Integer, ibsta%, iberr%, ibcntl&)
EnableRemote (104)	Declare sub DLEnableRemote Lib "gpib.dll" ByVal board%, addresslist As Integer, ibsta%, iberr%, ibcntl&)
FindLstn (105)	Declare sub DLLFindLstn Lib "gpib.dll" (ByVal board%, addresslist As Integer, resultlist As Integer, ByVal limit%, ibsta%, iberr%, ibcntl&)
FindRQS (106)	Declare sub DLLFindRQS Lib "gpib.dll" (ByVal board%, addresslist As Integer, result%, ibsta%, iberr%, ibcntl&)

(continues)

Table A-3. Declarations for NI-488.2 Routines in Visual BASIC (Continued)

Functions (ordinal entry value)	Syntax
PassControl (107)	Declare sub DLLPassControl Lib "gpib.dll" (ByVal board%, ByVal address%, ibsta%, iberr%, ibcntl&)
PPoll (108)	Declare sub DLLPPoll Lib "gpib.dll" (ByVal board%, result%, ibsta%, iberr%, ibcntl&)
PPollConfig (109)	Declare sub DLLPPollConfig Lib "gpib.dll" (ByVal board%, ByVal address%, ByVal dataline%, ByVal sense%, ibsta%, iberr%, ibcntl&)
PPollUnconfig (110)	Declare sub DLLPPollUnconfig Lib "gpib.dll" (ByVal board%, addresslist As Integer, ibsta%, iberr%, ibcntl&)
RcvRespMsg (111)	Declare sub DLLRcvRespMsg Lib "gpib.dll" (ByVal board%, ByVal data\$, ByVal count&, ByVal termination%, ibsta%, iberr%, ibcntl&)
ReadStatusByte (112)	Declare sub DLLReadStatusByte Lib "gpib.dll" (ByVal board%, ByVal address%, result%, ibsta%, iberr%, ibcntl&)
Receive (113)	Declare sub DLLReceive Lib "gpib.dll" (ByVal board%, ByVal address%, ByVal data\$, ByVal count&, ByVal termination%, ibsta%, iberr%, ibcntl&)
ReceiveSetup (114)	Declare sub DLLReceiveSetup Lib "gpib.dll" (ByVal board%, ByVal address%, ibsta%, iberr%, ibcntl&)

(continues)

Table A-3. Declarations for NI-488.2 Routines in Visual BASIC (Continued)

Functions (ordinal entry value)	Syntax
ResetSys (115)	Declare sub DLLResetSys Lib "gpib.dll" (ByVal board%, addresslist As Integer, ibsta%, iberr%, ibcntl&)
Send (116)	Declare sub DLLSend Lib "gpib.dll" (ByVal board%, ByVal address%, ByVal data\$, ByVal count&, ByVal eotmode% ibsta%, iberr%, ibcntl&)
SendCmds (117)	Declare sub DLLSendCmds Lib "gpib.dll" (ByVal board%, ByVal commands\$, ByVal count&, ibsta%, iberr%, ibcntl&)
SendDataBytes (118)	Declare sub DLLSendDataBytes Lib "gpib.dll" (ByVal board%, ByVal data\$, ByVal count&, ByVal eotmode%, ibsta%, iberr%, ibcntl&)
SendIFC (119)	Declare sub DLLSendIFC Lib "gpib.dll" (ByVal board%, ibsta%, iberr%, ibcntl&)
SendList (120)	Declare sub DLLSendList Lib "gpib.dll" (ByVal board%, addresslist As Integer, ByVal data\$, ByVal count&, ByVal eotmode%, ibsta%, iberr%, ibcntl&)
SendLLO (121)	Declare sub DLLSendLLO Lib "gpib.dll" (ByVal board%, ibsta%, iberr%, ibcntl&)
SendSetup (122)	Declare sub DLLSendSetup Lib "gpib.dll" (ByVal board%, addresslist As Integer, ibsta%, iberr%, ibcntl&)
SetRWLS (123)	Declare sub DLLSetRWLS Lib "gpib.dll" (ByVal board%, addresslist As Integer, ibsta%, iberr%, ibcntl&)

(continues)

Table A-3. Declarations for NI-488.2 Routines in Visual BASIC (Continued)

Functions (ordinal entry value)	Syntax
TestSRQ (124)	Declare sub DLLTestSRQ Lib "gpib.dll" (ByVal board%, result%, ibsta%, iberr%, ibcntl&)
TestSys (125)	Declare sub DLLTestSys Lib "gpib.dll" (ByVal board%, addresslist As Integer, resultlist As Integer, ibsta%, iberr%, ibcntl&)
Trigger (126)	Declare sub DLLTrigger Lib "gpib.dll" (ByVal board%, ByVal address%, ibsta%, iberr%, ibcntl&)
TriggerList (127)	Declare sub DLLTriggerList Lib "gpib.dll" (ByVal board%, addresslist As Integer, ibsta%, iberr%, ibcntl&)
WaitSRQ (128)	Declare sub DLLWaitSRQ Lib "gpib.dll" (ByVal board%, result%, ibsta%, iberr%, ibcntl&)
GenerateREQF (53)	Not available in Visual BASIC
GenerateREQT (52)	Not available in Visual BASIC
GotoMultAddr (129)	Not available in Visual BASIC

Table A-4 lists the declarations of the NI-488 functions for Visual BASIC.

Table A-4. Declarations for NI-488 Functions in Visual BASIC

Functions (ordinal entry value)	Syntax
ibbna (10)	Declare function DLLibbna Lib "gpib.dll" (ByVal ud%, ByVal bname\$, ibsta%, iberr%, ibcntl&) As Integer
ibcac (11)	Declare function DLLibcac Lib "gpib.dll" (ByVal ud%, ByVal v%, ibsta%, iberr%, ibcntl&) As Integer
ibclr (12)	Declare function DLLibclr Lib "gpib.dll" (ByVal ud%, ibsta%, iberr%, ibcntl&) As Integer
ibcmd (13)	Declare function DLLibcmd Lib "gpib.dll" (ByVal ud%, ByVal cmd\$, ByVal cnt&, ibsta%, iberr%, ibcntl&) As Integer
ibcmda (14)	Declare function DLLibcmda Lib "gpib.dll" (ByVal ud%, ByVal cmd\$, ByVal cnt&, ibsta%, iberr%, ibcntl&) As Integer
ibconfig (15)	Declare function DLLibconfig Lib "gpib.dll" (ByVal ud%, ByVal option%, ByVal value%, ibsta%, iberr%, ibcntl&) As Integer
ibdev (16)	Declare function DLLibdev Lib "gpib.dll" (ByVal index%, ByVal pad%, ByVal sad%, ByVal tmo%, ByVal eot%, ByVal eos%, ibsta%, iberr%, ibcntl&) As Integer
ibdma (18)	Declare function DLLibdma Lib "gpib.dll" (ByVal ud%, ByVal v%, ibsta%, iberr%, ibcntl&) As Integer

(continues)

Table A-4. Declarations for NI-488 Functions in Visual BASIC
(Continued)

Functions (ordinal entry value)	Syntax
ibeos (19)	Declare function DLLibeos Lib "gpib.dll" (ByVal ud%, ByVal v%, ibsta%, iberr%, ibcntl&) As Integer
ibeot (20)	Declare function DLLibeot Lib "gpib.dll" (ByVal ud%, ByVal v%, ibsta%, iberr%, ibcntl&) As Integer
ibevent (21)	Declare function DLLibevent Lib "gpib.dll" (ByVal ud%, ByVal event%, ibsta%, iberr%, ibcntl&) As Integer
ibfind (22)	Declare function DLLibfind Lib "gpib.dll" (ByVal udname\$, ibsta%, iberr%, ibcntl&) As Integer
ibgts (23)	Declare function DLLibgts Lib "gpib.dll" (ByVal ud%, ByVal v%, ibsta%, iberr%, ibcntl&) As Integer
ibist (24)	Declare function DLLibist Lib "gpib.dll" (ByVal ud%, ByVal v%, ibsta%, iberr%, ibcntl&) As Integer
iblines (25)	Declare function DLLiblines Lib "gpib.dll" (ByVal ud%, clines%, ibsta%, iberr%, ibcntl&) As Integer
ibln (26)	Declare function DLLiblin Lib "gpib.dll" (ByVal ud%, ByVal pad%, ByVal sad%, listen%, ibsta%, iberr%, ibcntl&) As Integer
ibloc (27)	Declare function DLLibloc Lib "gpib.dll" (ByVal ud%, ibsta%, iberr%, ibcntl&) As Integer

(continues)

Table A-4. Declarations for NI-488 Functions in Visual BASIC
(Continued)

Functions (ordinal entry value)	Syntax
ibonl (28)	Declare function DLLibonl Lib "gpib.dll" (ByVal ud%, ByVal v%, ibsta%, iberr%, ibcntl&) As Integer
ibpad (29)	Declare function DLLibpad Lib "gpib.dll" (ByVal ud%, ByVal v%, ibsta%, iberr%, ibcntl&) As Integer
ibpct (30)	Declare function DLLibpct Lib "gpib.dll" (ByVal ud%, ibsta%, iberr%, ibcntl&) As Integer
ibppc (32)	Declare function DLLibppc Lib "gpib.dll" (ByVal ud%, ByVal v%, ibsta%, iberr%, ibcntl&) As Integer
ibrđ (33)	Declare function DLLibrđ Lib "gpib.dll" (ByVal ud%, ByVal rd\$, ByVal cnt&, ibsta%, iberr%, ibcntl&) As Integer
ibrda (34)	Declare function DLLibrda Lib "gpib.dll" (ByVal ud%, ByVal rd\$, ByVal cnt&, ibsta%, iberr%, ibcntl&) As Integer
ibrdf (35)	Declare function DLLibrdf Lib "gpib.dll" (ByVal ud%, ByVal flname\$, ibsta%, iberr%, ibcntl&) As Integer
ibrđkey† (36)	Declare function DLLibrđkey Lib "gpib.dll" (ByVal ud%, ByVal rd\$, ByVal cnt%, ibsta%, iberr%, ibcntl&) As Integer
ibrpp (37)	Declare function DLLibrpp Lib "gpib.dll" (ByVal ud%, ppr%, ibsta%, iberr%, ibcntl&) As Integer

(continues)

Table A-4. Declarations for NI-488 Functions in Visual BASIC
(Continued)

Functions (ordinal entry value)	Syntax
ibrsc (38)	Declare function DLLibrsc Lib "gpib.dll" (ByVal ud%, ByVal v%, ibsta%, iberr%, ibcntl&) As Integer
ibrsp (39)	Declare function DLLibrsp Lib "gpib.dll" (ByVal ud%, spr%, ibsta%, iberr%, ibcntl&) As Integer
ibrsv (40)	Declare function DLLibrsv Lib "gpib.dll" (ByVal ud%, ByVal v%, ibsta%, iberr%, ibcntl&) As Integer
ibsad (41)	Declare function DLLibsad Lib "gpib.dll" (ByVal ud%, ByVal v%, ibsta%, iberr%, ibcntl&) As Integer
ibsic (42)	Declare function DLLibsic Lib "gpib.dll" (ByVal ud%, ibsta%, iberr%, ibcntl&) As Integer
ibsre (43)	Declare function DLLibsre Lib "gpib.dll" (ByVal ud%, ByVal v%, ibsta%, iberr%, ibcntl&) As Integer
ibstop (44)	Declare function DLLibstop Lib "gpib.dll" (ByVal ud%, ibsta%, iberr%, ibcntl&) As Integer
ibtmo (45)	Declare function DLLibtmo Lib "gpib.dll" (ByVal ud%, ByVal v%, ibsta%, iberr%, ibcntl&) As Integer
ibtrg (46)	Declare function DLLibtrg Lib "gpib.dll" (ByVal ud%, ibsta%, iberr%, ibcntl&) As Integer

(continues)

Table A-4. Declarations for NI-488 Functions in Visual BASIC
(Continued)

Functions (ordinal entry value)	Syntax
ibwait (47)	Declare function DLLibwait Lib "gpib.dll" (ByVal ud%, ByVal mask%, ibsta%, iberr%, ibcntl&) As Integer
ibwrt (48)	Declare function DLLibwrt Lib "gpib.dll" (ByVal ud%, ByVal wrt\$, ByVal cnt&, ibsta%, iberr%, ibcntl&) As Integer
ibwrta (49)	Declare function DLLibwrta Lib "gpib.dll" (ByVal ud%, ByVal wrt\$, ByVal cnt&, ibsta%, iberr%, ibcntl&) As Integer
ibwrtf (50)	Declare function DLLibwrtf Lib "gpib.dll" (ByVal ud%, ByVal fname\$, ibsta%, iberr%, ibcntl&) As Integer
ibwrtkey† (51)	Declare function DLLibwrtkey Lib "gpib.dll" (ByVal ud%, ByVal wrt\$, ByVal cnt%, ibsta%, iberr%, ibcntl&) As Integer
† ibrdkey and ibwrtkey are OEM functions. Refer to the <i>NI-488 Hardware Key Functions Reference Guide</i> , for a detailed description of these functions.	

Appendix B

Customer Communication

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

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

Corporate Headquarters

(512) 795-8248

Technical support fax: (800) 328-2203
(512) 794-5678

Branch Offices	Phone Number	Fax Number
Australia	(03) 879 9422	(03) 879 9179
Austria	(0662) 435986	(0662) 437010-19
Belgium	02/757.00.20	02/757.03.11
Denmark	45 76 26 00	45 76 71 11
Finland	(90) 527 2321	(90) 502 2930
France	(1) 48 14 24 00	(1) 48 14 24 14
Germany	089/741 31 30	089/714 60 35
Italy	02/48301892	02/48301915
Japan	(03) 3788-1921	(03) 3788-1923
Netherlands	03480-33466	03480-30673
Norway	32-848400	32-848600
Spain	(91) 640 0085	(91) 640 0533
Sweden	08-730 49 70	08-730 43 70
Switzerland	056/20 51 51	056/20 51 55
U.K.	0635 523545	0635 523154

Technical Support Form

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

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

Name _____

Company _____

Address _____

Fax (____) _____ Phone (____) _____

Computer brand _____

Model _____ Processor _____

Operating system _____

Speed _____MHz RAM _____MB

Display adapter _____

Mouse _____yes _____no

Other adapters installed _____

Hard disk capacity _____MB Brand _____

Instruments used _____

National Instruments hardware product model _____

Revision _____

Configuration _____

(continues)

National Instruments software product _____

Version _____

Configuration _____

The problem is _____

List any error messages _____

The following steps will reproduce the problem _____

GPIB for Windows Hardware and Software Configuration Form

Record the settings and revisions of your hardware and software on the line to the right of or underneath each item. Update this form each time you revise your software or hardware configuration, and use this form as a reference for your current configuration.

National Instruments Products

- NI-488.2 Software Revision/Version Number on Distribution Medium:

- Type of National Instruments GPIB boards installed and their respective hardware settings:

Board Type	Interrupt Level	DMA Channel	Base I/O Address
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

Other Products

- Microsoft Windows Version: _____
- Type of other boards installed and their respective hardware settings:

Board Type	Interrupt Level	DMA Channel	Base I/O Address
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

Glossary

ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
DLL	dynamic link library
DMA	direct memory access
EMI	electromagnetic interference
GPIB	General Purpose Interface Bus
hex	hexadecimal
Hz	hertz
IEEE	Institute of Electrical and Electronic Engineers
MB	megabytes of memory
M-	mega- (10^6)
OEM	Original Equipment Manufacturer
RAM	random-access memory