



Serial Hardware and Software Help

July 2007, 370447J-01

This help file contains instructions to help you install and configure the NI serial hardware and software for Windows 2000/XP/Server 2003/Vista, Windows XP x64/Server 2003 x64/Vista 64, and LabVIEW Real-Time.

For more information about this help file, refer to the following topics:

[Using Help](#)

[Related Documentation](#)

[Glossary](#)

[Important Information](#)

[Technical Support and Professional Services](#)

To comment on National Instruments documentation, refer to the [National Instruments Web site](#).

© 2002–2007 National Instruments Corporation. All rights reserved.

About this Help File

This help file contains instructions to help you install and configure the NI serial hardware and software. This help file includes information about the following serial hardware:

- PCI-8430 (two, four, eight, and 16 port), PCI-8431 (two, four, and eight port), PCI-8432 (two and four port), and PCI-8433 (two and four port)
- PCI-232 (two, four, eight, and 16 port), PCI-485 (two, four, and eight port), PCI-232 isolated (two and four port), and PCI-485 isolated (two and four port)
- PXI-8430 (two, four, eight, and 16 port), PXI-8431 (two, four, and eight port), PXI-8432 (two and four port) and PXI-8433 (two and four port)
- PXI-8420 (two, four, eight, and 16 port), PXI-8421 (two, four, and eight port), PXI-8422 (two and four port), and PXI-8423 (two and four port)
- USB-232 (one, two, and four port) and USB-485 (one, two, and four port)
- ENET-232 (two and four port) and ENET-485 (two and four port)
- ExpressCard-8420 (two port) and ExpressCard-8421 (two port)
- PCMCIA-232 (one, two, and four port) and PCMCIA-485 (one and two port)

This help file assumes that you are already familiar with Windows and LabVIEW Real-Time.

Related Documentation

The following documents contain information that you may find helpful as you use this help file:

- ANSI/EIA-232-D Standard, *Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange*
- EIA/RS-422-A Standard, *Electrical Characteristics of Balanced Voltage Digital Interface Circuits*
- EIA-485 Standard, *Standard for Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems*
- *Microsoft Platform Software Developer Kit*, Microsoft Corporation
- *NI Serial Hardware Guide*
- *NI-VISA Help*

Using Help

[Conventions](#)

[Navigating Help](#)

[Searching Help](#)

[Printing Help File Topics](#)

Conventions

This help file uses the following formatting and typographical conventions:

- < > Angle brackets that contain numbers separated by an ellipsis represent a range of values associated with a bit or signal name—for example, AO <0..3>.
- » 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 names.
- dark red** Text in this color denotes a caution.
- ExpressCard ExpressCard serial boards refers to all versions of the serial boards ExpressCard serial boards.
- green Underlined text in this color denotes a link to a help topic, help file, or Web address.
- italic* Italic text denotes variables, emphasis, cross references, or an introduction to a key concept. Italic text 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.

**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.

PCI serial
boards

PCI serial boards refers to all versions of the PCI serial boards.

PCMCIA
serial boards

PCMCIA serial boards refers to all versions of the PCMCIA serial boards.

PXI serial
boards

PXI serial boards refers to all versions of the PXI serial boards.

Serial ENET

Serial ENET refers to all versions of the ENET serial boards.

USB serial
boards

USB serial boards refers to all versions of the USB serial boards.

Navigating Help (Windows Only)

To navigate this help file, use the **Contents**, **Index**, and **Search** tabs to the left of this window or use the following toolbar buttons located above the tabs:

- **Hide**—Hides the navigation pane from view.
- **Locate**—Locates the currently displayed topic in the **Contents** tab, allowing you to view related topics.
- **Back**—Displays the previously viewed topic.
- **Forward**—Displays the topic you viewed before clicking the **Back** button.
- **Options**—Displays a list of commands and viewing options for the help file.

Searching Help (Windows Only)

Use the **Search** tab to the left of this window to locate content in this help file. If you want to search for words in a certain order, such as "related documentation," add quotation marks around the search words as shown in the example. Searching for terms on the **Search** tab allows you to quickly locate specific information and information in topics that are not included on the **Contents** tab.

Wildcards

You also can search using asterisk (*) or question mark (?) wildcards. Use the asterisk wildcard to return topics that contain a certain string. For example, a search for "prog*" lists topics that contain the words "program," "programmatically," "progress," and so on.

Use the question mark wildcard as a substitute for a single character in a search term. For example, "?ext" lists topics that contain the words "next," "text," and so on.

Nested Expressions

Use nested expressions to combine searches to further refine a search. You can use Boolean expressions and wildcards in a nested expression. For example, "example AND (program OR VI)" lists topics that contain "example program" or "example VI." You cannot nest expressions more than five levels.

Boolean Expressions

Click the  button to add Boolean expressions to a search. The following Boolean operators are available:

- **AND** (default)—Returns topics that contain both search terms. You do not need to specify this operator unless you are using nested expressions.
- **OR**—Returns topics that contain either the first or second term.
- **NOT**—Returns topics that contain the first term without the second term.
- **NEAR**—Returns topics that contain both terms within eight words of each other.

Search Options

Use the following checkboxes on the **Search** tab to customize a search:

- **Search previous results**—Narrows the results from a search that returned too many topics. You must remove the checkmark from this checkbox to search all topics.
- **Match similar words**—Broadens a search to return topics that contain words similar to the search terms. For example, a search for "program" lists topics that include the words "programs," "programming," and so on.
- **Search titles only**—Searches only in the titles of topics.

Printing Help File Topics (Windows Only)

Complete the following steps to print an entire book from the **Contents** tab:

1. Right-click the book.
2. Select **Print** from the shortcut menu to display the **Print Topics** dialog box.
3. Select the **Print the selected heading and all subtopics** option.
 **Note** Select **Print the selected topic** if you want to print the single topic you have selected in the **Contents** tab.
4. Click the **OK** button.

Printing PDF Documents

This help file may contain links to PDF documents. To print PDF documents, click the print button located on the Adobe Acrobat Viewer toolbar.

Introduction

The following topics explain how to use this help file, list what you need to get started and optional equipment you can order, and briefly describe the NI serial hardware and software.

[How to Use this Help File](#)

[What You Need to Get Started](#)

[Optional Equipment](#)

[Serial Hardware Overview](#)

[PCI Kits](#)

[PXI Kits](#)

[USB Kits](#)

[ENET Kits](#)

[ExpressCard Kits](#)

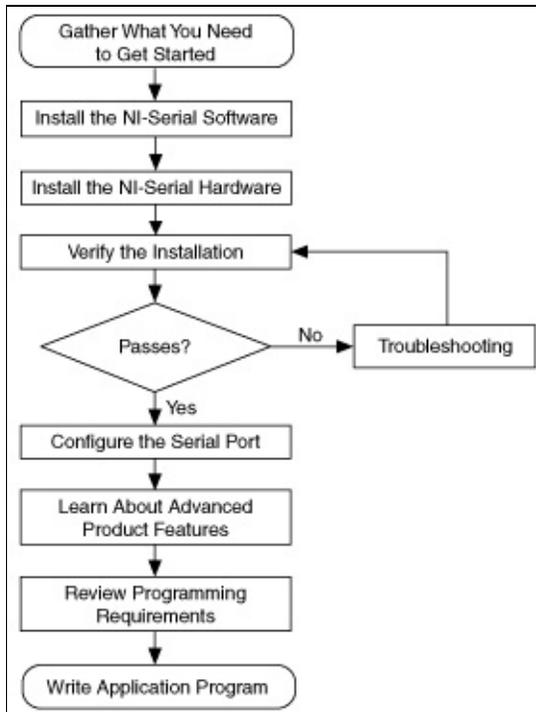
[PCMCIA Kits](#)

[NI-Serial Software Overview](#)

[Time-Saving Development Tools](#)

How to Use this Help File

To use this help file, refer to the following flow chart and the appropriate topics.



What You Need to Get Started

Before you install your serial hardware and the NI-Serial software, make sure you have all of the following items:

- One of the following hardware kits:

PCI Kits

- PCI-8430 (two, four, eight, or 16 port)
- PCI-8431 (two, four, or eight port)
- PCI-8432 (two or four port)
- PCI-8433 (two or four port)
- PCI-232 (two, four, eight, or 16 port)
- PCI-485 (two, four, or eight port)
- PCI-232 isolated (two or four port)
- PCI-485 isolated (two or four port)

PXI Kits

- PXI-8430 (two, four, eight, or 16 port)
- PXI-8431 (two, four, or eight port)
- PXI-8432 (two or four port)
- PXI-8433 (two or four port)
- PXI-8420 (two, four, eight, or 16 port)
- PXI-8421 (two, four, or eight port)
- PXI-8422 (two or four port)
- PXI-8423 (two or four port)

USB Kits

- USB-232 (one, two, or four port)
- USB-485 (one, two, or four port)

ENET Kits

- ENET-232 (two or four port)
- ENET-485 (two or four port)

ExpressCard Kits

- ExpressCard-8420 (two port)
- ExpressCard-8421 (two port)

PCMCIA Kits

- PCMCIA-232 (one, two, or four port)
- PCMCIA-485 (one or two port)

- The following CD:
 - NI-Serial Software, which is included in your kit

Optional Equipment

For more information about ordering the following optional equipment, contact National Instruments:

- DB-9 RS-485 termination connector (PCI-485, PCI-8431, PCI-8433, ExpressCard-8421, PCMCIA-485, PXI-8421, PXI-8423, PXI-8431, PXI-8433, USB-485, and ENET-485 only)
- 10-position modular jack to DB-9 cable (PCI and PXI four-port boards only)
- 10-position modular jack to DB-25 cable (PCI and PXI four-port boards only)
- 68-pin to DB-9 adapter cable (PCI and PXI eight-port and PCI/PXI-8430/16 boards only)
- RS-232 9-pin to 9-pin null modem cable
- RS-232 9-pin to 9-pin straight-through cable
- RS-232 9-pin to 25-pin null modem cable
- RS-485 9-pin to 9-pin null modem cable
- Breakout box (PCI-232/16 and PXI-8420/16 boards only)
- External power supply (ENET and USB-485/4 only)
- USB cable (USB two and four port only)
- One of the following Category 5 twisted-pair network cards (ENET only)
 - Straight-through cable for connecting the serial ENET to a network hub
 - Crossover cable for connecting the serial ENET directly to a computer's network interface card (NIC)

Serial Hardware Overview

[PCI Kits](#)

[PXI Kits](#)

[USB Kits](#)

[ENET Kits](#)

[ExpressCard Kits](#)

[PCMCIA Kits](#)

PCI Kits

The PCI serial hardware gives you a variety of solutions for serial communication. The RS-232 boards (PCI-8430, PCI-8432, and PCI-232) work with the RS-232 protocol. The RS-485 boards (PCI-8431, PCI-8433, and PCI-485) work with the RS-422 and RS-485 protocols. You can use the RS-232 ports for serial communication up to distances of 50 ft. You can connect the RS-485 ports to up to 31 devices using serial cable lengths up to 4,000 ft.

Additionally, the PCI serial boards are available in two-port, four-port, eight-port, and 16-port versions. The two-port versions use DB-9 connectors. The four-port versions use 10-position modular jacks to provide all four connections on a single back panel. Optional cable accessories convert the 10-position modular jacks to either DB-9 or DB-25 connectors with standard pinouts. The eight-port versions use adapter cables to convert the 68-pin connector on the board to eight DB-9 connectors. The PCI-8430/16 uses two adapter cables to convert the two 68-pin connectors on the board to 16 DB-9 connectors. The PCI-232/16 uses a breakout box to convert the 100-pin connector on the board to 16 DB-9 connectors. Throughout this help file, *PCI serial boards* refers to all versions of the PCI serial boards.

The isolated PCI boards are designed for applications in harsh environments. Isolated ports provide reliable communication in situations involving ground loops from different ground levels or high common mode voltage induced on the lines in noisy environments. Nonisolated ports may not provide reliable communication in those situations. The isolation between each communication port and the host PC ensures the safe operation of the PC and the devices connected to other ports on the same board, in case of accidental high voltages or line surges on communication lines.

The RS-232 ports are DTE. In the RS-232 specification, Data Terminal Equipment (DTE) and Data Communications Equipment (DCE) refer to the types of equipment on either end of a serial connection. In general, DTE and DCE refer to computer equipment and modems, respectively. For more information about cabling a RS-232 port to other devices, refer to [RS-232 DTE vs. DCE](#).

The RS-485 ports support four hardware transceiver control modes for

reliable communication with two and four-wire devices. For more information about transceiver control modes, refer to [RS-485 Transceiver Control](#).

All PCI serial hardware uses standard 16550-compatible UARTs (Universal Asynchronous Receiver/Transmitters) for complete compatibility with standard PC COM ports. The serial hardware contains FIFO (First-In-First-Out) buffers to reduce susceptibility to interrupt latency for faster transmission rates. Full Plug and Play compatibility allows switchless configuration and installation. For more information about the serial hardware specifications and operating conditions, refer to the *NI Serial Hardware Guide*.

The following table lists the PCI serial board numbers and corresponding board descriptions.

PCI Board Names and Descriptions

PCI Board Name	Description
PCI-8430	RS-232 two port RS-232 four port RS-232 eight port RS-232 16 port
PCI-8431	RS-485 two port RS-485 four port RS-485 eight port
PCI-8432	RS-232 two port isolated RS-232 four port isolated
PCI-8433	RS-485 two port isolated RS-485 four port isolated
PCI-232	RS-232 two port RS-232 four port RS-232 eight port RS-232 16 port
PCI-485	RS-485 two port RS-485 four port RS-485 eight port
PCI-232I	RS-232 two port isolated RS-232 four port isolated

PCI-485I	RS-485 two port isolated
	RS-485 four port isolated

PXI Kits

The PXI serial hardware gives you a variety of solutions for serial communication. The RS-232 boards (PXI-8430, PXI-8432, PXI-8420, and PXI-8422) work with the RS-232 protocol. The RS-485 boards (PXI-8431, PXI-8433, PXI-8421, and PXI-8423) work with the RS-422 and RS-485 protocols. You can use the RS-232 ports for serial communication up to distances of 50 ft. You can connect the RS-485 ports to up to 31 devices using serial cable lengths up to 4,000 ft.

Additionally, the PXI serial boards are available in two-port, four-port, eight-port, and 16-port versions. The two-port versions use DB-9 connectors. The four-port versions use 10-position modular jacks to provide all four connections on a single front panel. Optional cable accessories convert the 10-position modular jacks to either DB-9 or DB-25 connectors with standard pinouts. The eight-port versions use adapter cables to convert the 68-pin connector on the board to eight DB-9 connectors. The PXI-8430/16 uses two adapter cables to convert the two 68-pin connectors on the board to 16 DB-9 connectors. The PXI-8420/16 uses a breakout box to convert the 100-pin connector on the board to 16 DB-9 connectors. Throughout this help file, *PXI serial boards* refers to all versions of the PXI serial boards.

The isolated PXI boards are designed for applications in harsh environments. Isolated ports provide reliable communication in situations involving ground loops from different ground levels or high common mode voltage induced on the lines in noisy environments. Nonisolated ports may not provide reliable communication in those situations. The isolation between each communication port and the host PC ensures the safe operation of the PC and the devices connected to other ports on the same board, in case of accidental high voltages or line surges on communication lines.

The RS-232 ports are DTE. In the RS-232 specification, Data Terminal Equipment (DTE) and Data Communications Equipment (DCE) refer to the types of equipment on either end of a serial connection. In general, DTE and DCE refer to computer equipment and modems, respectively. For more information about cabling a RS-232 port to other devices, refer to [RS-232 DTE vs. DCE](#).

The RS-485 ports support four hardware transceiver control modes for

reliable communication with two and four-wire devices. For more information about transceiver control modes, refer to [RS-485 Transceiver Control](#).

All PXI serial hardware uses standard 16550-compatible UARTs (Universal Asynchronous Receiver/Transmitters) for complete compatibility with standard PC COM ports. The serial hardware contains FIFO (First-In-First-Out) buffers to reduce susceptibility to interrupt latency for faster transmission rates. Full Plug and Play compatibility allows switchless configuration and installation. For more information about the serial hardware specifications and operating conditions, refer to the *NI Serial Hardware Guide*.

The following table lists the PXI serial board numbers and corresponding board descriptions.

PXI Board Names and Descriptions

PXI Board Name	Description
PXI-8430	RS-232 two port RS-232 four port RS-232 eight port RS-232 16 port
PXI-8431	RS-485 two port RS-485 four port RS-485 eight port
PXI-8432	RS-232 two port isolated RS-232 four port isolated
PXI-8433	RS-485 two port isolated RS-485 four port isolated
PXI-8420	RS-232 two port RS-232 four port RS-232 eight port RS-232 16 port
PXI-8421	RS-485 two port RS-485 four port RS-485 eight port
PXI-8422	RS-232 two port isolated RS-232 four port isolated

PXI-8423

RS-485 two port isolated
RS-485 four port isolated

USB Kits

The USB serial hardware gives you a variety of solutions for serial communication. The USB-232 hardware works with the RS-232 interface standard, and the USB-485 hardware works with the RS-422 and RS-485 interface standards. You can use the RS-232 ports for serial communication up to distances of 50 ft. You can connect the RS-485 ports to up to 31 devices using serial cable lengths up to 4,000 ft.

Additionally, the USB serial hardware is available in one-port, two-port, and four-port versions. All versions use DB-9 connectors. Throughout this help file, *USB serial hardware* refers to all versions of the USB serial hardware.

The USB-232 two and four-port hardware supports three hardware transceiver modes supporting both DTE and DCE configurations. For more information about transceiver control modes, refer to [USB-232 Transceiver Modes](#).

The RS-485 hardware supports four hardware transceiver control modes for reliable communication with two and four-wire devices. For more information about transceiver control modes, refer to [RS-485 Transceiver Control](#). In addition, the USB-485 hardware supports programmatic control of the bias resistors. For more information, refer to [USB-485 Programmatically Controlled Bias Resistors](#).

All USB serial hardware uses standard 16550-compatible UARTs (Universal Asynchronous Receiver/Transmitters) for complete compatibility with standard PC COM ports. The serial hardware contains FIFO (First-In-First-Out) and USB buffers to reduce susceptibility to system latency for faster transmission rates. Full Plug and Play compatibility allows switchless configuration and installation. For more information about the serial hardware specifications and operating conditions, refer to the *NI Serial Hardware Guide*.

ENET Kits

The ENET serial hardware gives you a variety of solutions for serial communication. The RS-232 interfaces work with the RS-232 protocols. The RS-485 interfaces work with the RS-422 and RS-485 protocols. You can use the RS-232 ports for serial communication up to distances of 50 ft. You can connect the RS-485 ports to up to 31 devices using serial cable lengths up to 4,000 ft.

The RS-232 ports are DTE. In the RS-232 specification, Data Terminal Equipment (DTE) and Data Communications Equipment (DCE) refer to the types of equipment on either end of a serial connection. In general, DTE and DCE refer to computer equipment and modems, respectively. For more information about cabling a RS-232 port to other devices, refer to [RS-232 DTE vs. DCE](#).

The RS-485 boards support four hardware transceiver control modes for reliable communication with two and four-wire devices. For more information about transceiver control modes, refer to [RS-485 Transceiver Control](#).

All serial ENET interfaces contain FIFO (First-In-First-Out) buffers to reduce susceptibility to interrupt latency for faster transmission rates. Also, serial ENET interfaces contain additional onboard buffers to reduce susceptibility to Ethernet network traffic.

With a serial ENET interface, you gain all the features inherent in a networked device: fewer cabling distance restrictions, device sharing, and communication with devices throughout the Internet. In a serial application, you usually are restricted to the distance limitations of the RS-232, RS-422, or RS-485 specifications. Because the serial ENET interface uses Ethernet, you can add an unlimited distance to your application by exploiting the distances available using a networked device. Device sharing with a serial ENET interface happens on a per-port basis. That is, although the serial ENET interface supports network connections from multiple hosts, each serial port is associated with only one host at a time.

Dynamic Host Configuration Protocol (DHCP) is available on many networks to configure network parameters automatically. DHCP simplifies the installation and configuration process for the serial ENET. For more information about DHCP, refer to [Using DHCP](#).

If your network does not support DHCP, use the NI Ethernet Device Configuration utility as described in [Serial ENET Configuration](#) to configure network parameters for the serial ENET interface.

ExpressCard Kits

The ExpressCard serial hardware gives you a variety of solutions for serial communication. The ExpressCard-8420 interface works with the RS-232 protocol, and the ExpressCard-8421 interface works with the RS-422 and RS-485 protocols. You can use the RS-232 ports for serial communication up to distances of 50 ft. You can connect the RS-485 ports to up to 31 devices using serial cable lengths up to 4,000 ft.

The ExpressCard-8420 and ExpressCard-8421 are available with two ports. All ExpressCard cards include cables for each port that terminates in a standard DB-9 connector. Throughout this help file, *ExpressCard serial interfaces* refers to all versions of the interfaces.

The RS-232 ports are DTE. In the RS-232 specification, Data Terminal Equipment (DTE) and Data Communications Equipment (DCE) refer to the types of equipment on either end of a serial connection. In general, DTE and DCE refer to computer equipment and modems, respectively. For more information about cabling a RS-232 port to other devices, refer to [RS-232 DTE vs. DCE](#).

The RS-485 ports support four hardware transceiver control modes for reliable communication with two and four-wire devices. For more information about transceiver control modes, refer to [RS-485 Transceiver Control](#).

All ExpressCard serial hardware uses standard 16550-compatible UARTs (Universal Asynchronous Receiver/Transmitters) for complete compatibility with standard PC COM ports. The serial hardware contains FIFO (First-In-First-Out) buffers to reduce susceptibility to interrupt latency for faster transmission rates. Full Plug and Play compatibility allows switchless configuration and installation. For more information about the serial hardware specifications and operating conditions, refer to the *NI Serial Hardware Guide*.

ExpressCard Board Names and Descriptions

ExpressCard Board Name	Description
ExpressCard-8420	RS-232 two port
ExpressCard-8421	RS-485 two port

PCMCIA Kits

The PCMCIA serial hardware gives you a variety of solutions for serial communication. The PCMCIA-232 interfaces work with the RS-232 protocol, and the PCMCIA-485 interfaces work with the RS-422 and RS-485 protocols. You can use the RS-232 ports for serial communication up to distances of 50 ft. You can connect the RS-485 ports to up to 31 devices using serial cable lengths up to 4,000 ft.

The PCMCIA-232 is available in one-port, two-port, and four-port versions. The PCMCIA-485 is available in one-port or two-port versions. All PCMCIA cards come with cables for each port that terminates in a standard DB-9 connector. Throughout this help file, *PCMCIA serial interfaces* refers to all versions of the interfaces.

The RS-232 ports are DTE. In the RS-232 specification, Data Terminal Equipment (DTE) and Data Communications Equipment (DCE) refer to the types of equipment on either end of a serial connection. In general, DTE and DCE refer to computer equipment and modems, respectively. For more information about cabling a RS-232 port to other devices, refer to [RS-232 DTE vs. DCE](#).

The RS-485 ports support four hardware transceiver control modes for reliable communication with two and four-wire devices. For more information about transceiver control modes, refer to [RS-485 Transceiver Control](#).

All PCMCIA serial hardware uses standard 16550-compatible UARTs (Universal Asynchronous Receiver/Transmitters) for complete compatibility with standard PC COM ports. The serial hardware contains FIFO (First-In-First-Out) buffers to reduce susceptibility to interrupt latency for faster transmission rates. Full Plug and Play compatibility allows switchless configuration and installation. For more information about the serial hardware specifications and operating conditions, refer to the *NI Serial Hardware Guide*.

NI-Serial Software Overview

The supported interfaces and operating systems are shown in the following table.

Supported Interfaces and Operating Systems

Hardware Interface	Windows Vista 32	Windows Vista 64	Windows 2000/XP/Server 2003	Windows XP x64/Server 2003 x64	LabVIEW Real-Time
PCI	X	X	X	X	X
PXI	X	X	X	X ¹	X
USB	X		X		
ENET	X	X ²	X	X ²	X ²
ExpressCard	X		X		
PCMCIA	X		X	X	

Windows Vista, Windows 2000/XP/Server 2003, Windows XP x64/Server 2003 x64, and LabVIEW Real-Time allow a maximum of 256 COM ports. If you require additional ports, contact National Instruments.

The NI-Serial software includes the following components:

- Device driver
- Measurement & Automation Explorer (MAX)
- Troubleshooting Wizard (Windows only)
- Configuration utilities
- Help files

¹Currently, National Instruments PXI controllers do not offer Windows XP x64 support.

²NI-VISA support only.

Time-Saving Development Tools

Your kit includes the NI-Serial software. In addition, you can order LabVIEW or LabWindows™/CVI™ software from National Instruments to speed your application development time and make it easier to communicate with your instruments.

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.

After you install your serial hardware and the NI-Serial software, you can use NI-VISA in LabVIEW and LabWindows/CVI with your serial interface. If you already have one or more of these applications and want to use them with your serial interface, refer to your product documentation for information about serial I/O functions. For ordering information, contact National Instruments.

NI-VISA is a standard I/O application programming interface (API) for instrumentation programming.

In its full implementation, NI-VISA can control USB, FireWire, VXI/VME, PXI, GPIB, TCP/IP, or serial instruments, making the appropriate driver calls depending on the type of instrument being used. NI-VISA uses the same operations to communicate with instruments regardless of the interface type. For example, the NI-VISA command to write an ASCII string to a message-based instrument is the same whether the instrument is serial, USB, GPIB, or VXI. As a result, NI-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 to learn only one API.

Another advantage of NI-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.

VISA is the industry standard for developing instrument drivers. Most current drivers written by National Instruments use NI-VISA and support Windows, Pocket PC, Linux, Mac OS X, and LabVIEW RT, as long as the appropriate system-level drivers are available for that platform.

NI-Serial Software Installation

Windows

To install the NI-Serial software on your Windows system, follow these steps:

1. Log on as Administrator or as a user with administrator privileges.
2. Insert the *NI-Serial Software* CD.
3. Click **Install Software** and follow the onscreen instructions.
4. [Install your Windows serial hardware.](#)

LabVIEW RT

To install the NI-Serial software on your LabVIEW RT target, first follow steps 1–3 in the preceding section to install the software onto the Windows host machine. Then complete the following steps:

1. [Launch MAX](#).
2. Expand **Remote Systems** by clicking the + next to it.
3. Find your remote system in the list and expand it.
4. Right-click **Software** and select **Add/Remove Software**.
5. In the **Add/Remove Software** window, click **NI-Serial** and select **Install**. To use VISA Interactive Control and VISA test panels, you must also install NI-VISA Server.
6. Click **Next** to start the software download. Click **Finish** when the download is complete.
7. Shut down your RT controller and [install your RT serial hardware](#).

Install the Hardware

[PCI Hardware Installation](#)

[PXI Hardware Installation](#)

[USB Hardware Installation](#)

[ENET Hardware Installation](#)

[ExpressCard Installation](#)

[PCMCIA Hardware Installation](#)

PCI Hardware Installation



Note If you are installing a PCI RS-485 interface, you might need to adjust the value of the bias resistors, depending on your application. Bias resistors are not available on eight-port PCI hardware. For more information, refer to [Bias Resistors](#).

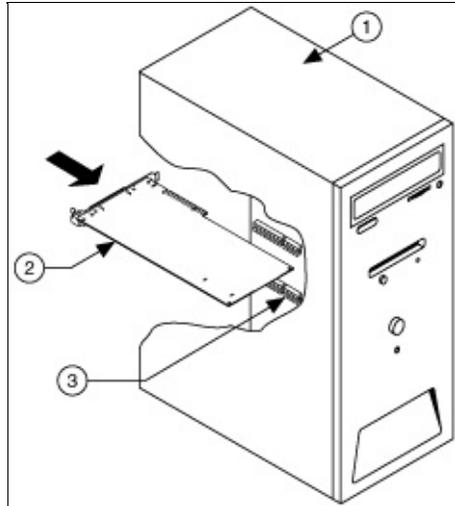


Caution Before you remove your board from the package, touch the antistatic plastic package to a metal part of your system chassis to discharge electrostatic energy, which can damage several components on your serial board.

Before installing your hardware, follow the [NI-Serial Software Installation](#) instructions.

To install your PCI serial board, complete the following steps:

1. Turn off your computer. Keep the computer plugged in so that it remains grounded while you install the PCI serial board.
2. Remove the top or side cover of the computer.
3. Find an unused PCI expansion slot in your computer.
4. Remove the corresponding expansion slot cover on the back panel of the computer.
5. Touch a metal part on your chassis to discharge any static electricity.
6. Insert the PCI serial board into the slot with the serial connectors toward the opening on the back panel. Make sure that you insert the board all the way into the slot. The board might seem to click firmly into place, even though it is only part of the way in. The following figure shows how to install the PCI serial board into an expansion slot.



- 1 PC
- 2 PCI Serial Board
- 3 PCI Slot

PCI Serial Board Installation

7. Screw the PCI serial board mounting bracket to the back panel mounting rail of the computer.
8. Replace the cover.
9. Turn on your computer.
10. The operating system will automatically detect your hardware.

The serial hardware installation is complete. Continue to [Verify the Installation](#).

PXI Hardware Installation



Note If you are installing a PXI RS-485 interface, you might need to adjust the value of the bias resistors, depending on your application. Bias resistors are not available on eight-port PXI hardware. For more information, refer to [Bias Resistors](#).

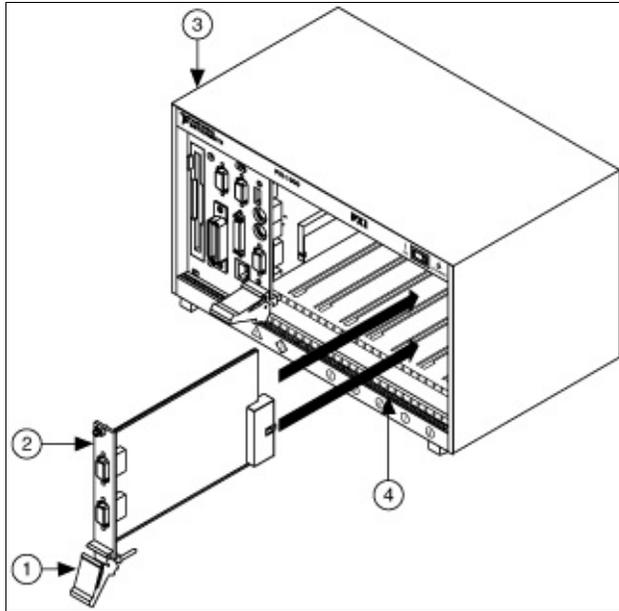


Caution Before you remove your board from the package, touch the antistatic plastic package to a metal part of your system chassis to discharge electrostatic energy, which can damage several components on your serial board.

Before installing your hardware, follow the [NI-Serial Software Installation](#) instructions.

To install your PXI serial board, complete the following steps:

1. Turn off your PXI or CompactPCI chassis. Keep the chassis plugged in so that it remains grounded while you install the PXI serial board.
2. Find an unused PXI or CompactPCI peripheral slot.
3. Remove the corresponding filler panel.
4. Touch a metal part on your chassis to discharge any static electricity.
5. Insert the PXI serial board into the slot. Use the injector/ejector handle to fully inject the device into place. The following figure shows how to install the PXI serial board into a PXI or CompactPCI chassis.



- 1 Injector/Ejector Handle (In Down Position)
- 2 PXI Serial Board
- 3 PXI Chassis
- 4 Injector/Ejector Rail

Installing the PXI Serial Board

6. Screw the PXI serial board front panel to the front panel mounting rail of the PXI or CompactPCI chassis.
7. Turn on your PXI or CompactPCI chassis.
8. The operating system will automatically detect your hardware.

The serial hardware installation is complete. Continue to [Verify the Installation](#).

USB Hardware Installation



Note If you are installing a USB-485/2 or USB-485/4, you might need to adjust the value of the bias resistors, depending on your application. For more information, refer to [Bias Resistors](#).

Before installing your hardware, follow the [NI-Serial Software Installation](#) instructions.

To install your USB serial hardware, complete the following steps:

1. If you are installing a USB-485/4, connect the external power supply.
2. Connect the USB cable from the USB hardware to an available USB port on your computer or USB hub.
3. If your computer is not already on, turn it on.
4. Windows will automatically detect your hardware.

The serial hardware installation is complete. Continue to [Verify the Installation](#).

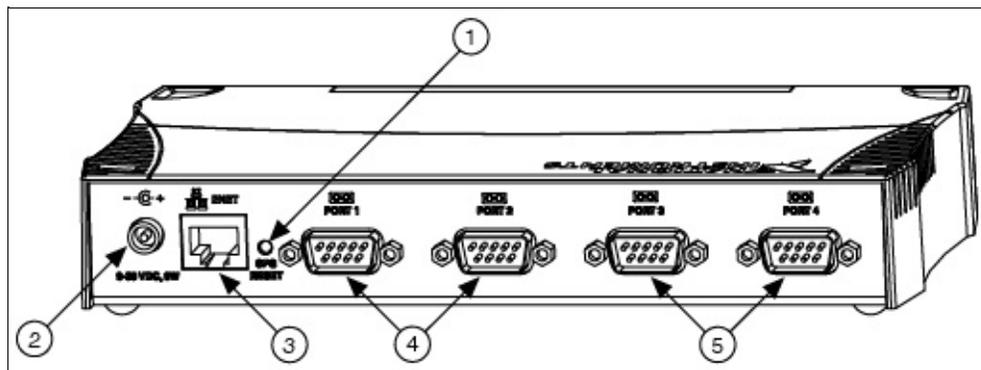
ENET Hardware Installation

Before installing your hardware, follow the [NI-Serial Software Installation](#) instructions.

To install your serial ENET hardware, complete the following steps:

1. Connect one end of your Ethernet cable to your serial ENET. (The rear panel of the serial ENET is shown below.) Connect the other end of the Ethernet cable to your Ethernet network.
2. Connect one end of the power cord to the power supply. Screw the power connector on the other end of the power supply onto the power jack of the serial ENET. Plug the other end of the power cord into an AC outlet.

! **Caution** If you do not use the power supply included in your kit, ensure that you do not operate your serial ENET interface at any voltage other than the one specified on the rear panel. Doing so could damage the unit.



- 1 Configuration Reset Switch
- 2 Power Connection
- 3 Ethernet Connection
- 4 Serial Ports 1 and 2
- 5 Serial Ports 3 and 4 (on 4-Port Models Only)

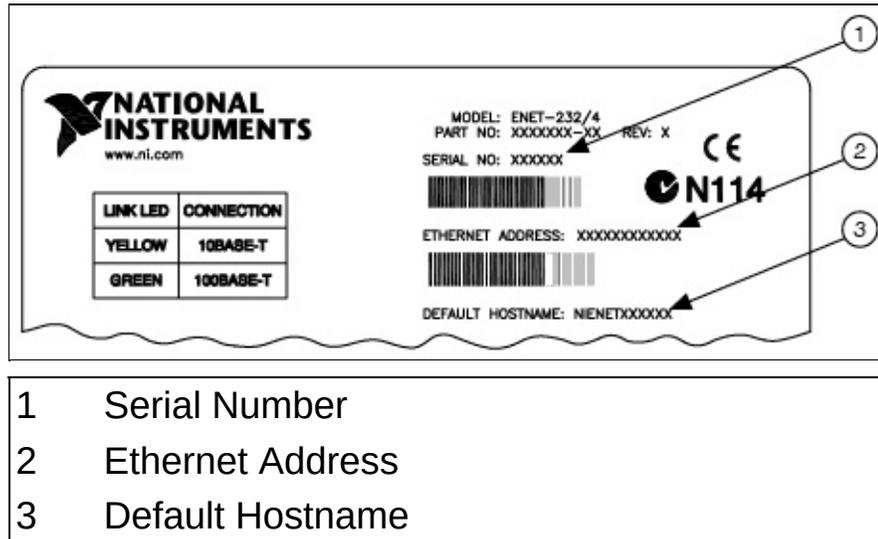
Rear Panel of a Four-Port Serial ENET

3. Refer to the Baseplate Identification Label, shown below, and make a note of the serial number, Ethernet address, and default hostname. Recording this information is not necessary for proper installation. However, you will find it convenient when you

configure the serial ENET.



Note The Ethernet address is not the IP address. All devices on an Ethernet network are assigned a unique physical address—the Ethernet address—so they can communicate with each other.



Serial ENET Baseplate Identification Label

4. Before you power-on your serial ENET interface, contact your network administrator to determine whether you need to configure your network settings manually using the NI Ethernet Device Configuration utility or use the Dynamic Host Configuration Protocol (DHCP) to perform the configuration automatically.

Turn on the front-panel power switch. The **PWR/RDY** LED alternates rapidly between red and yellow while the unit completes its power-on self-tests and attempts to acquire its network parameters. By default, the serial ENET interface attempts its network configuration through DHCP.

The time required for assigning the IP address depends on your network and the configuration of your serial ENET interface. Allow up to 90 seconds and observe the state of the **PWR/RDY** LED to determine the outcome of the self tests. One of the following should occur:

- A steady yellow **PWR/RDY** LED indicates the serial ENET interface passed its self tests and acquired its IP address. The unit is now ready to operate. When using DHCP, the

serial ENET Interface typically is ready to operate about 15 seconds after you power it on. Follow the instructions in [Adding a Serial ENET Interface](#) to add the serial ports to your computer.

- If the **PWR/RDY** LED continues to alternate rapidly between red and yellow, the unit could not use DHCP to configure its network parameters. Follow the instructions in [Adding a Serial ENET Interface](#) to configure the unit properly and add the serial ports to your computer.
- If the **PWR/RDY** LED is steady red, the serial ENET has an unrecoverable error. Contact National Instruments Technical Support.
- If the **PWR/RDY** LED blinks a slow red/yellow pattern, the serial ENET did not pass its self tests. Refer to [ENET PWR/RDY LED Signaling](#) to interpret the flash pattern before calling National Instruments Technical Support.



Note If at any time you want to return the serial ENET interface to its default configuration state as given on the baseplate identification label, press and hold the CFG RESET switch for three seconds at power-on. This switch is on the rear panel, as shown in the figure above. Refer to [Using the CFG RESET Switch](#) for more information.

ExpressCard Hardware Installation

Before installing your hardware, follow the [NI-Serial Software Installation](#) instructions.

To install the ExpressCard serial board in your computer, insert the card into a free ExpressCard socket. The card has no jumpers or switches to set, and you do not need to shut down the system before you insert it.

Windows will automatically detect your hardware.

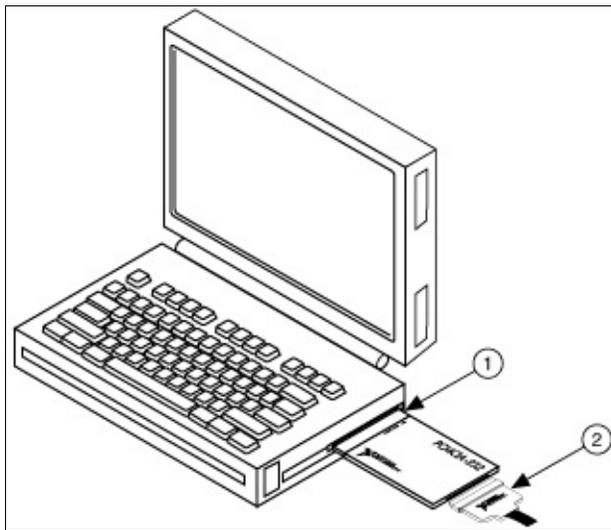
The serial hardware installation is complete. Continue to [Verify the Installation](#).

PCMCIA Hardware Installation

Before installing your hardware, follow the [NI-Serial Software Installation](#) instructions.

To install the PCMCIA serial board in your computer, insert the card into a free PC Card (PCMCIA) socket. The card has no jumpers or switches to set, and you do not need to shut down the system before you insert it.

The following figure shows how to insert a PCMCIA serial card and how to connect the cable.



- | |
|-----------------------|
| 1 PCMCIA Socket |
| 2 PCMCIA Serial Cable |

Inserting a PCMCIA Serial Card

Windows will automatically detect your hardware.

The serial hardware installation is complete. Continue to [Verify the Installation](#).

Measurement & Automation Explorer (MAX)

Overview

[Launching MAX](#)

[Adding a Serial ENET Interface](#)

[Verify Installation and Troubleshoot NI-Serial Problems](#)

[Delete a Serial Interface](#)

[View NI-Serial Software Version](#)

[Monitor, Record, and Display NI-VISA API Calls](#)

[System Report Generation](#)

[View or Change Serial Port Settings](#)

[Port Settings](#)

[Advanced Settings](#)

[Save Pending Changes](#)

[Discard Pending Changes](#)

[Restore Defaults](#)

[Import/Export a Configuration File](#)

[NI-Serial Communicator](#)

[Recover Unused COM Numbers](#)

[Additional Help and Resources](#)

[NI-Serial Online Help](#)

[National Instruments Serial Web Site](#)

MAX Overview

You can perform the following serial-related tasks in MAX:

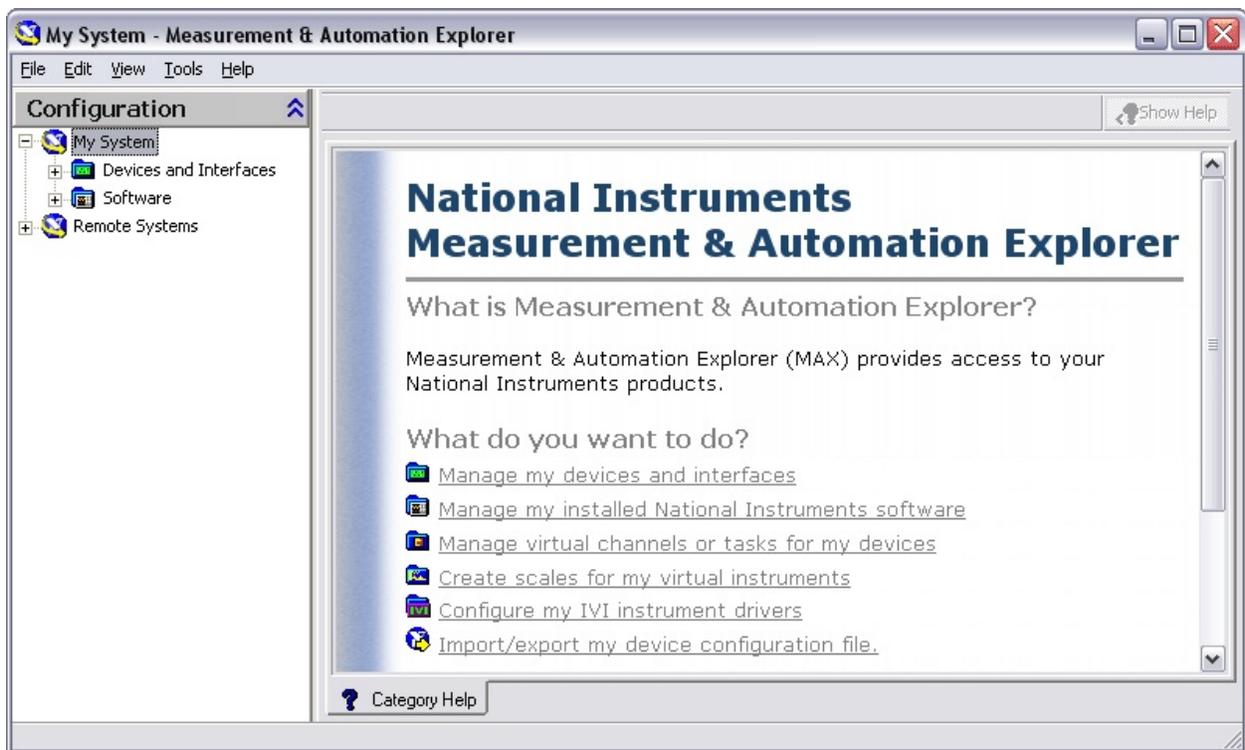
- Establish basic communication with your serial devices.
- Launch the NI-Serial Troubleshooting Wizard to verify installation and troubleshoot serial problems. (Windows only.)
- Launch NI Spy to monitor NI-VISA API calls to serial interfaces. (Windows only.)
- View information about your serial hardware and NI-Serial software.
- Reconfigure the serial interface settings.
- Locate additional help resources for NI-Serial.

Launching MAX

To start MAX, double-click the MAX icon on the desk or select **Start»Programs»National Instruments»Measurement & Automation**.

On Windows Vista, to make changes to serial port settings, you must launch MAX with administrative privileges. To launch MAX with administrative privileges, right-click the MAX icon on the desktop or the **Start** menu and select **Run as administrator**. You must enter an administrative password if you are not an administrator.

When you launch MAX, the following window appears:



Measurement & Automation Explorer

Adding a Serial ENET Interface

LabVIEW RT users do not need to add serial ENET interfaces to their RT controllers. To use a serial ENET interface on your LabVIEW RT system, you simply configure it. Refer to [Viewing Network Settings](#) for instructions on viewing and configuring network settings for serial ENET interfaces.

To add a serial ENET interface on your Windows system, complete the following steps:

1. [Launch MAX](#).
2. Right-click **Devices and Interfaces** and select **Create New...**
3. In the **Create New...** dialog window, select **Serial ENET Interface** and click **Finish**.
4. Follow the prompts in the Add Serial ENET Hardware Wizard to add your serial ENET interface.

If the **PWR/RDY** LED continues to alternate rapidly between red and yellow, the unit could not use DHCP to configure its network parameters. You need to use the NI Ethernet Device Configuration utility to configure your serial ENET interface. Refer to [Configuring Network Settings](#) for more information.

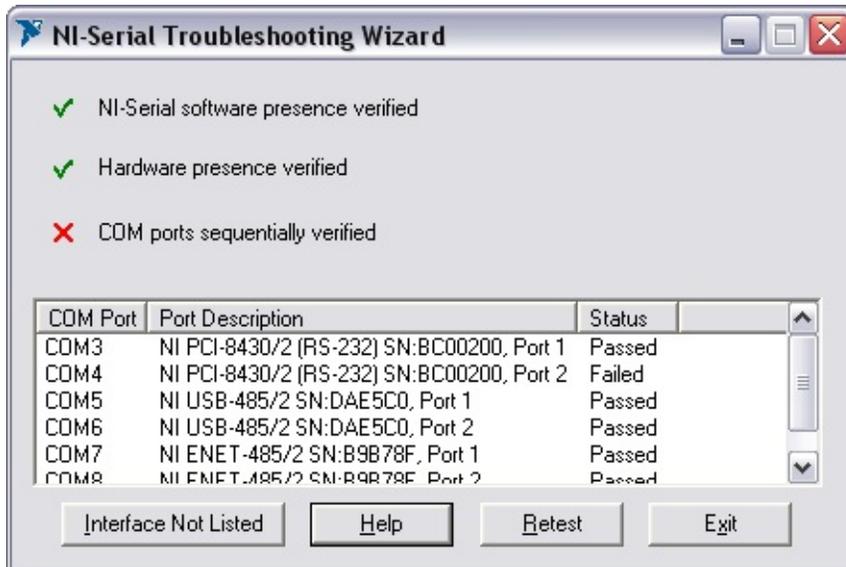
5. When finished, select **Restart Now** to finish the serial ENET installation.
6. After the computer has finished restarting, continue to [Verify the Installation](#).

Verify Installation and Troubleshoot NI-Serial Problems

The NI-Serial Troubleshooting Wizard supports ports only on your local Windows system. It does not test your LabVIEW Real-Time serial ports.

To verify the hardware and software installation and troubleshoot NI-Serial problems, follow these steps to run the Troubleshooting Wizard:

1. [Launch MAX](#).
2. Select **Help»Troubleshooting»NI-Serial Troubleshooting Wizard**. The Troubleshooting Wizard tests your serial interface and displays the results as shown below.



NI-Serial Troubleshooting Wizard

The Troubleshooting Wizard verifies that your serial driver is installed properly, that the configuration of your hardware does not conflict with anything else in your system, and that the serial driver can communicate with your hardware correctly.

To view online help for the Troubleshooting Wizard, click the **Help** button.

Delete a Serial Interface

Before you physically remove a serial interface from your Windows system, follow these steps to remove the hardware information:

1. [Launch MAX](#).
2. Expand the **Devices and Interfaces** directory by clicking the + next to the folder.
3. Right-click on your serial interface and select **Delete** from the drop-down menu that appears.
4. When prompted, click the **Yes** button to confirm the removal of your interface.
5. MAX automatically updates the list of installed serial interfaces. You can also select **View»Refresh** to update the list.

On LabVIEW RT, it is not necessary to programmatically delete hardware from the system. Simply power down your RT controller and remove the hardware. When you power on your RT controller again, the ports are automatically removed from the system. You may need to refresh MAX to see the changes.

View NI-Serial Software Version

To view the NI-Serial software version for Windows, follow these steps:

1. [Launch MAX](#).
2. Expand the **Software** directory by clicking the + next to the folder.
3. Click **NI-Serial**.

MAX displays the NI-Serial software version number in the right pane.

To view the NI-Serial software version installed on a LabVIEW RT controller, follow these steps:

1. [Launch MAX](#).
2. Expand **Remote Systems** by clicking the + next to it.
3. Find your RT controller in the list and expand it.
4. Expand the **Software** directory.
5. Click **NI-Serial**.

MAX displays the NI-Serial software version number in the right pane.

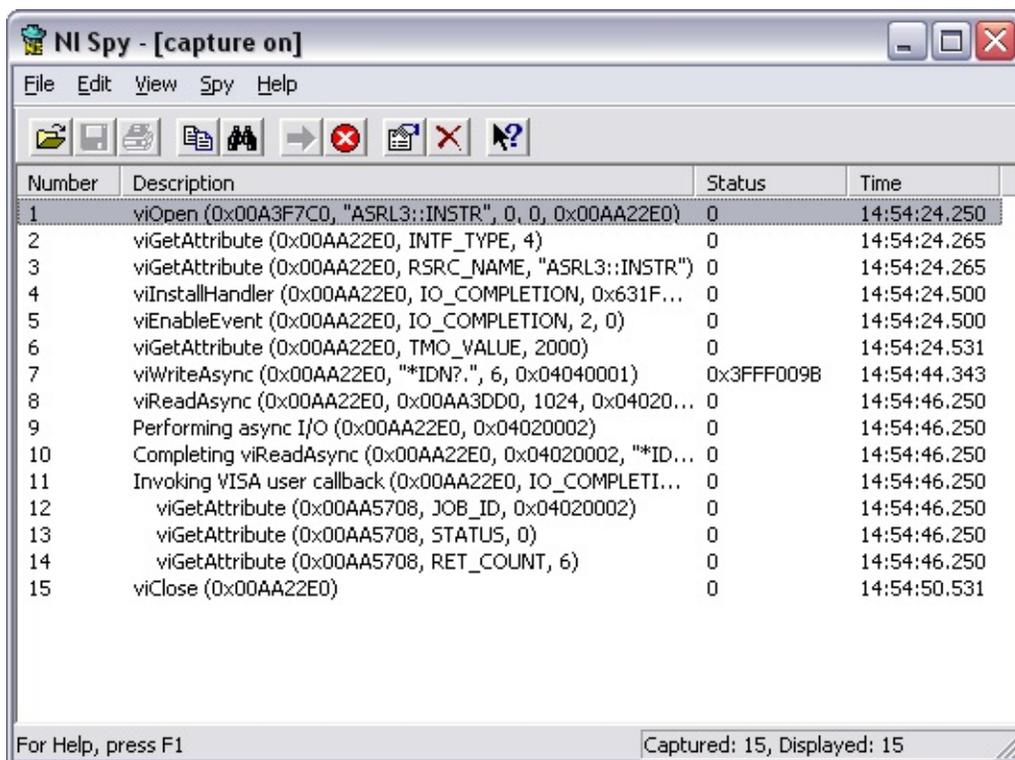
Monitor, Record, and Display NI-VISA API Calls

NI-Spy is available only on Windows.

To monitor NI-VISA calls, use NI Spy as follows:

1. [Launch MAX](#).
2. Expand the **Devices and Interfaces** directory by clicking the + next to the folder.
3. Right-click on your serial interface and select **NI Spy** from the drop-down menu that appears.
4. On the NI Spy toolbar, click the blue arrow button to start a capture.
5. Start the NI-VISA application you want to monitor.

NI Spy records and displays all NI-VISA calls, as shown below.



The screenshot shows the NI Spy application window with a menu bar (File, Edit, View, Spy, Help) and a toolbar. The main area displays a table of captured API calls. The table has four columns: Number, Description, Status, and Time. The data is as follows:

Number	Description	Status	Time
1	viOpen (0x00A3F7C0, "ASRL3::INSTR", 0, 0, 0x00AA22E0)	0	14:54:24.250
2	viGetAttribute (0x00AA22E0, INTF_TYPE, 4)	0	14:54:24.265
3	viGetAttribute (0x00AA22E0, RSRC_NAME, "ASRL3::INSTR")	0	14:54:24.265
4	viInstallHandler (0x00AA22E0, IO_COMPLETION, 0x631F...)	0	14:54:24.500
5	viEnableEvent (0x00AA22E0, IO_COMPLETION, 2, 0)	0	14:54:24.500
6	viGetAttribute (0x00AA22E0, TMO_VALUE, 2000)	0	14:54:24.531
7	viWriteAsync (0x00AA22E0, "*IDN?", 6, 0x04040001)	0x3FFF009B	14:54:44.343
8	viReadAsync (0x00AA22E0, 0x00AA3DD0, 1024, 0x04020...)	0	14:54:46.250
9	Performing async I/O (0x00AA22E0, 0x04020002)	0	14:54:46.250
10	Completing viReadAsync (0x00AA22E0, 0x04020002, "*ID...)	0	14:54:46.250
11	Invoking VISA user callback (0x00AA22E0, IO_COMPLETI...)	0	14:54:46.250
12	viGetAttribute (0x00AA5708, JOB_ID, 0x04020002)	0	14:54:46.250
13	viGetAttribute (0x00AA5708, STATUS, 0)	0	14:54:46.250
14	viGetAttribute (0x00AA5708, RET_COUNT, 6)	0	14:54:46.250
15	viClose (0x00AA22E0)	0	14:54:50.531

At the bottom of the window, it says "For Help, press F1" and "Captured: 15, Displayed: 15".

For more information about using NI Spy, select **Help»Help Topics** in NI Spy.

System Report Generation

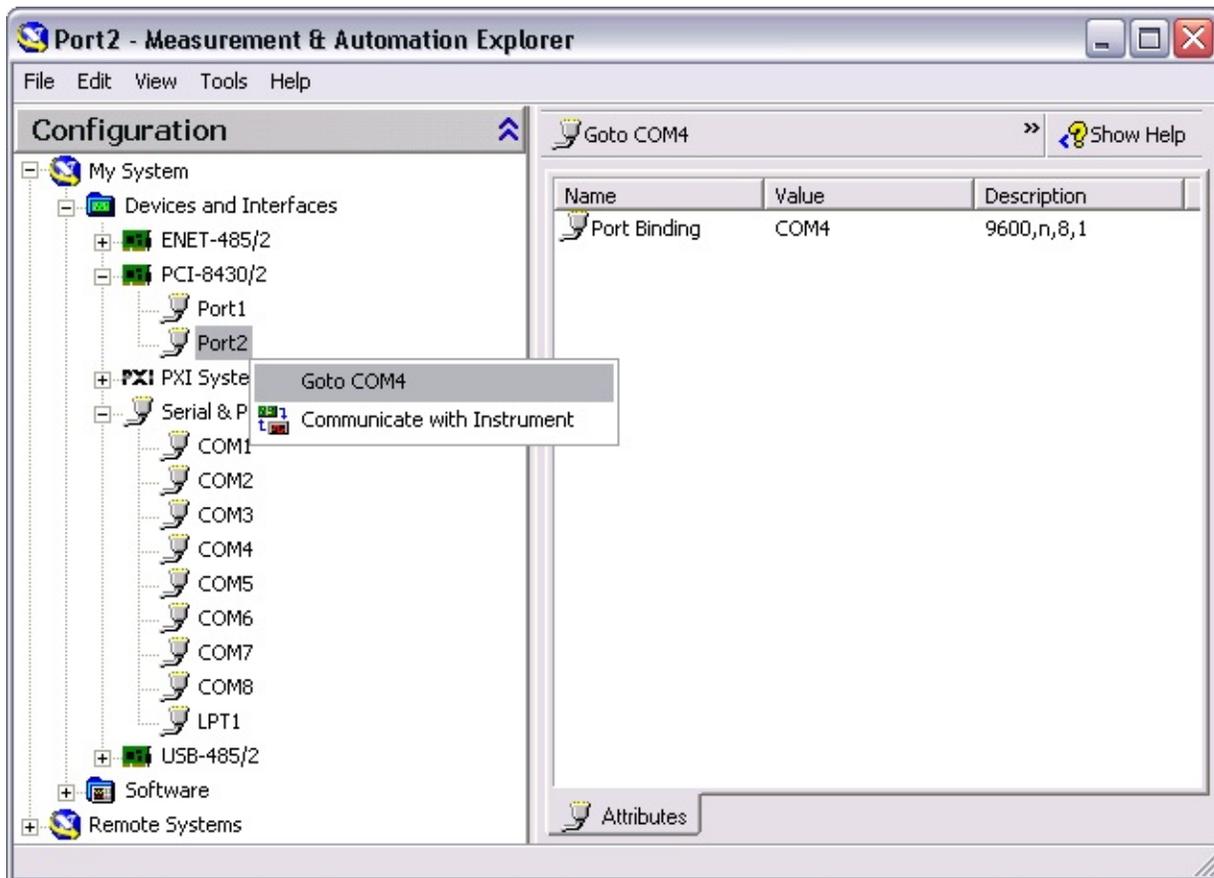
You can use the Report Generation feature in MAX to create a snapshot of the National Instruments hardware and software installed on your system. To create a report, follow these steps:

1. [Launch MAX](#).
2. Click **File»Create Report...**
3. Follow the dialog prompts to select the report type and the system and components you want to include in the report.
4. Select a report filename and location. Click the **View report after the wizard exits** box to launch the report automatically.

View or Change Serial Port Settings

To view or change your Windows serial port settings, follow these steps:

1. [Launch MAX](#).
2. Expand the **Devices and Interfaces** directory by clicking the + next to the folder.
3. Select your serial interface to display hardware information in the right pane.
4. Right-click on the serial port you want to modify and select **Goto COM#**, where # is the serial COM number assigned to that physical port. This directs you to the appropriate serial port under the **Serial & Parallel** section of MAX.

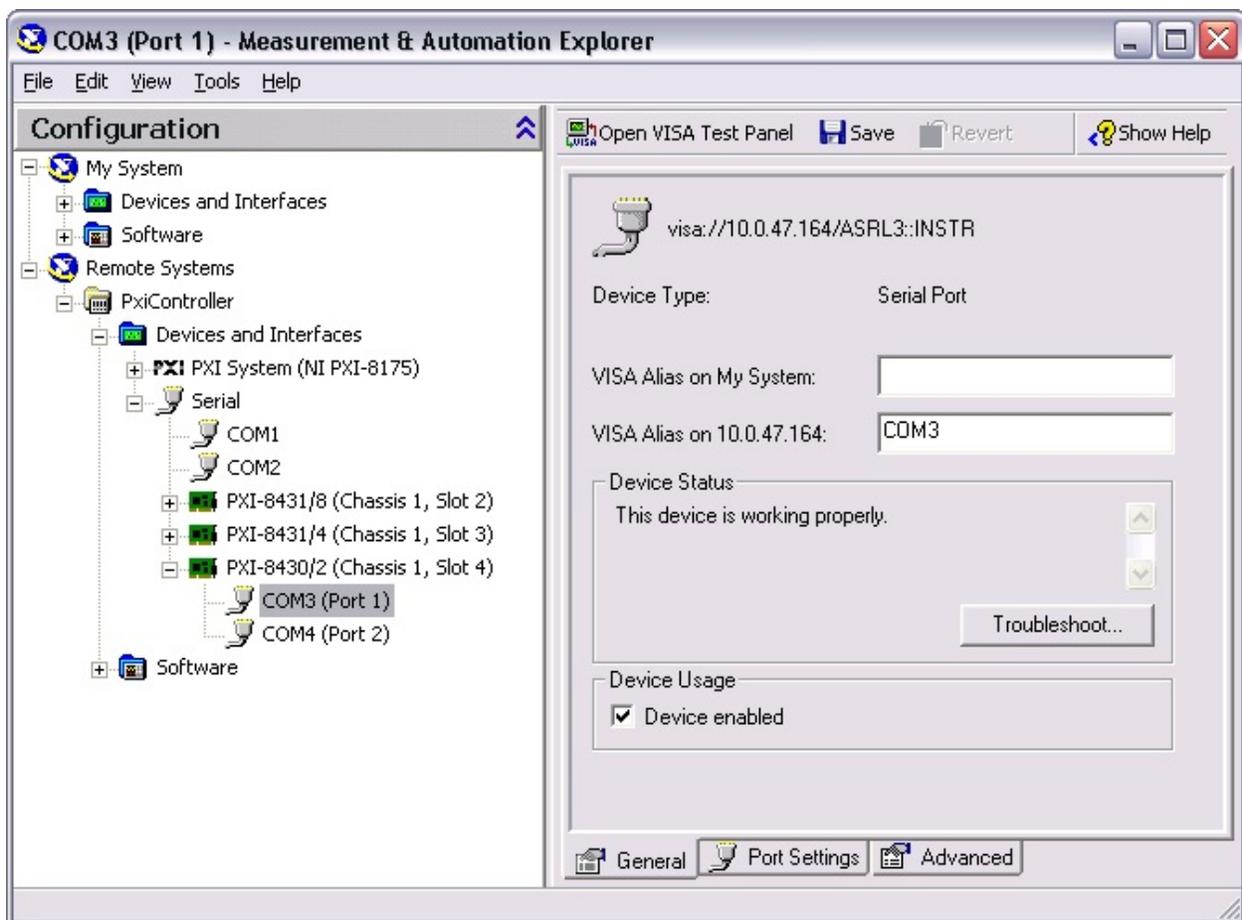


To view or change your LabVIEW RT port settings, follow these steps:

1. [Launch MAX](#).

2. Expand **Remote Systems** by clicking the + next to the folder.
3. Find your RT controller in the list and expand it.
4. Expand the **Devices and Interfaces** directory.
5. Expand the **Serial** entry.
6. Built-in ports and plug-in boards are displayed. To change the settings of a port on a plug-in serial board, select it from the list and expand it to see the ports.
7. Select a port to change its settings.

To see chassis and slot information for plug-in serial boards, identify your PXI system.



To view or change the port settings, click the **Port Settings** tab. For more information about these settings, refer to [Port Settings](#).

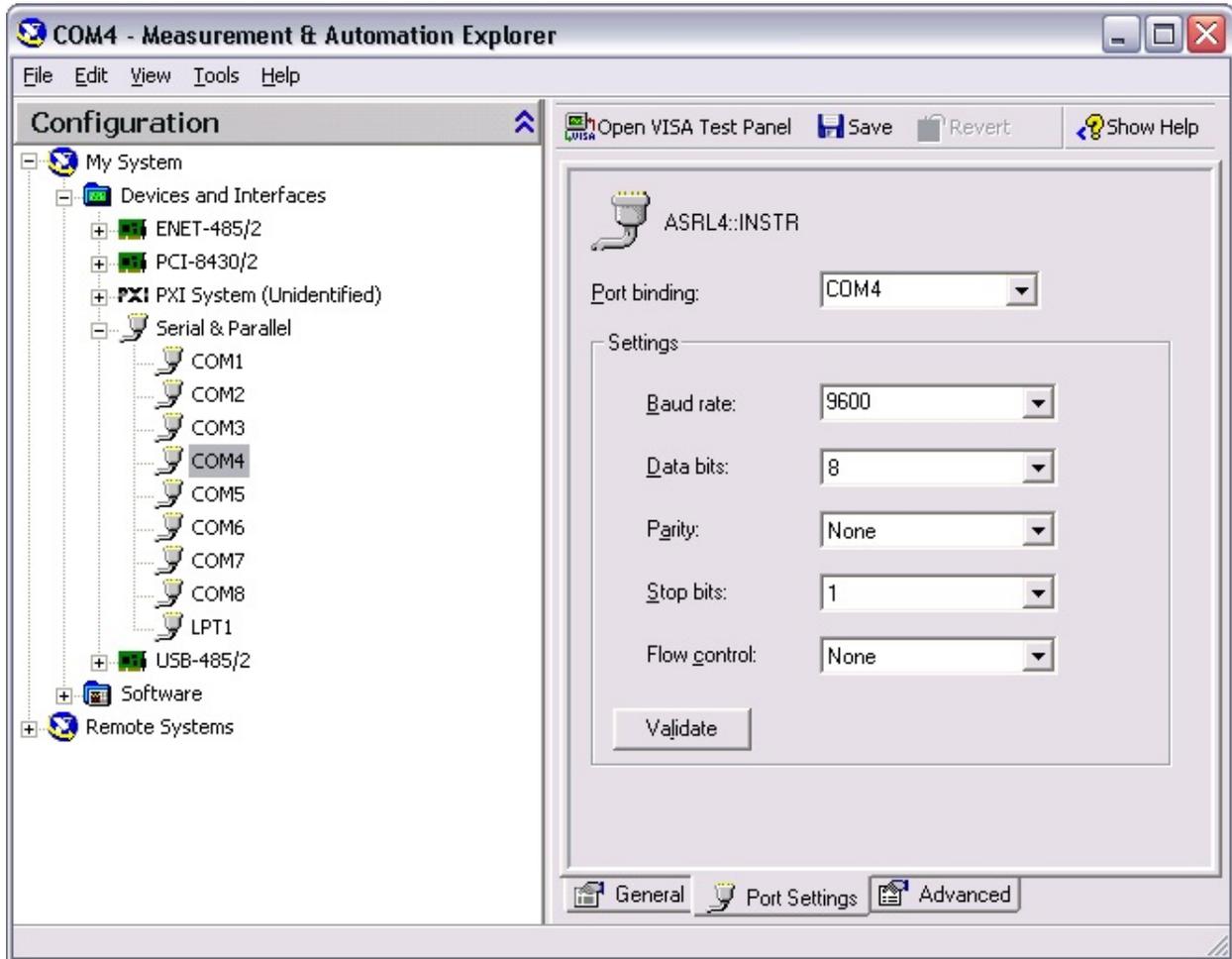
To change the transceiver mode, enable or disable the FIFOs, or enable or disable the bias resistors on the USB-485, click the **Advanced** tab. For more information about these settings, refer to [Advanced Settings](#).



Note Transceiver modes apply to RS-485 and USB-232 (two and four port) interfaces only. For more information about transceiver modes, refer to [Using Your Serial Hardware](#).

Port Settings

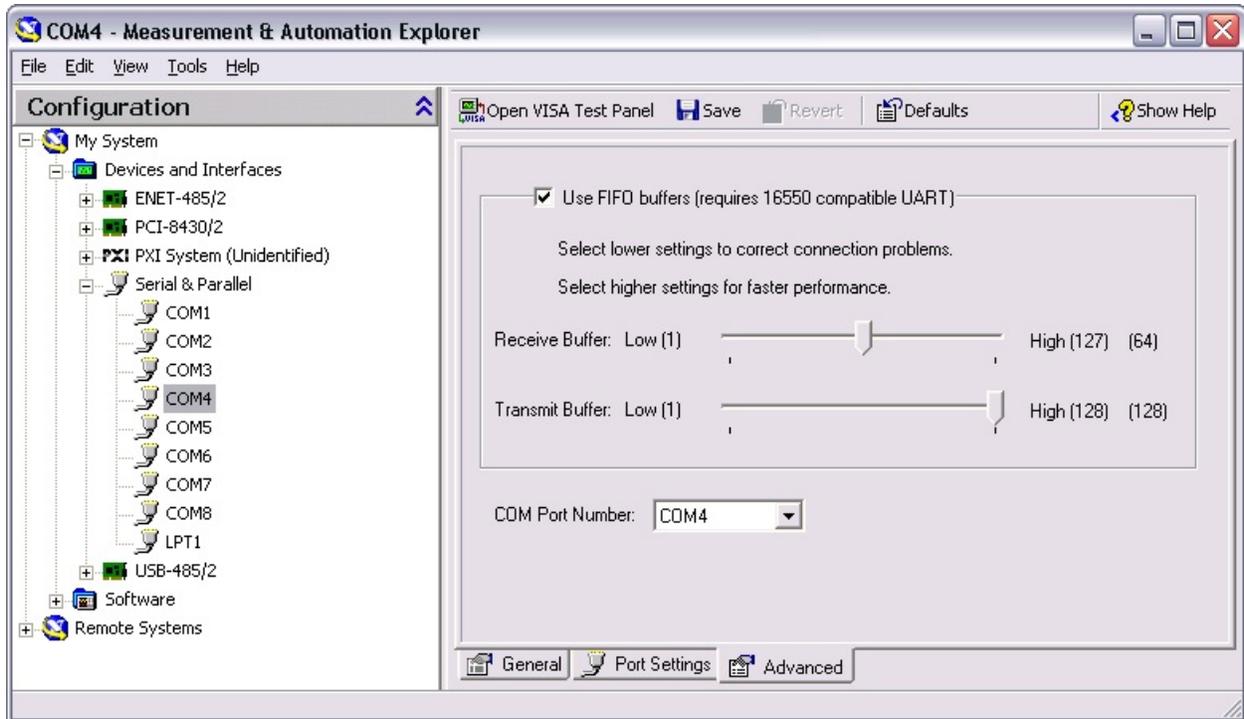
In the **Port Settings** tab, you can change any setting by clicking the arrow button to the right of a field. When you click the arrow button, a list of valid values for that field appears, and you can select the desired setting from the list. The following figure shows the **Port Settings** tab.



Advanced Settings

To view or change the advanced settings, click the **Advanced** tab, shown in the following figure.

The available settings may change depending on what serial interface you are configuring. Click **Show Help** in the upper right corner of the MAX window for information about specific advanced settings.



Save Pending Changes

To save your changes to this item, click **Save**.

Discard Pending Changes

To discard your changes to this item, click **Revert**. **Revert** is available only if you have made at least one change to this item. It undoes any changes you have made and restores the item to the most recently saved state.

Restore Defaults

To reset the fields to their default values at any time, click the **Defaults** button.

Import/Export a Configuration File

Import/export is available only for your LabVIEW Real-Time serial ports. You can export your port settings to a file and use that file later to import those settings back into a LabVIEW Real-Time system. You can use this method to back up and restore your port settings or deploy them to other systems.

To export your port settings to a file, select **File»Export**. Be sure to pick the correct LabVIEW Real-Time system from which to export.

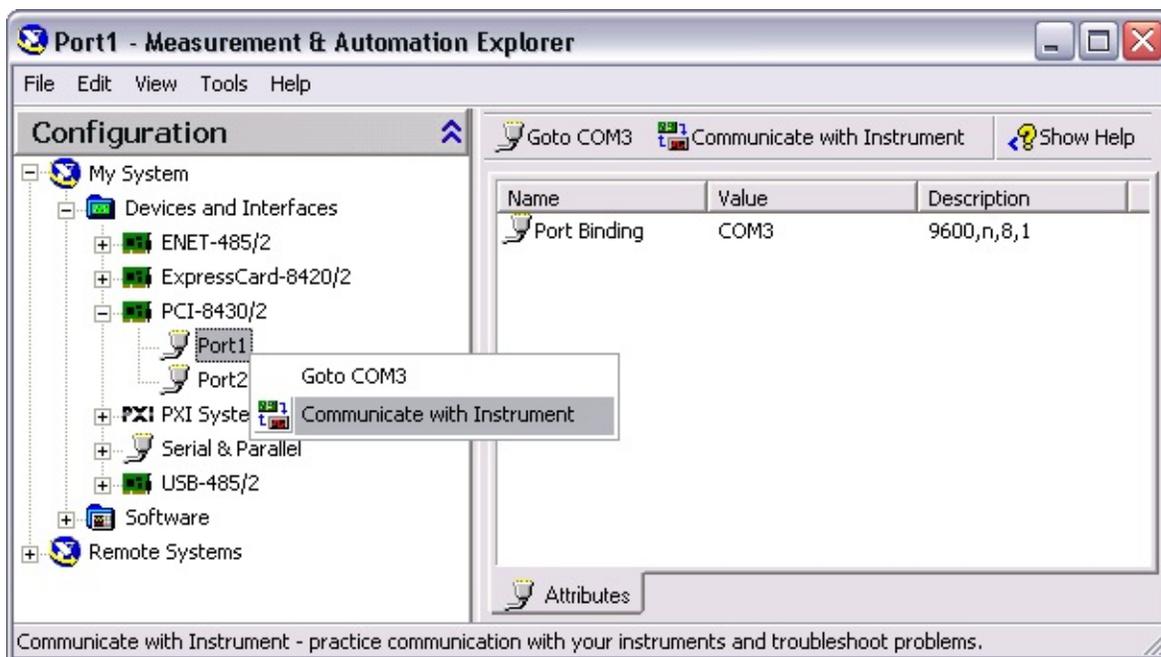
To import settings from a file to your ports, select **File»Import**. Be sure to pick the correct LabVIEW Real-Time system to which to import.

For more information, refer to *Measurement & Automation Explorer Help*. To access this help, select **Help»MAX Help**.

NI-Serial Communicator

The NI-Serial Communicator is a utility that allows you to establish basic communication with your instrument or another serial port. The NI-Serial Communicator is available only for Windows serial ports. Follow these steps to launch the NI-Serial Communicator:

1. [Launch MAX](#).
2. Expand the **Devices and Interfaces** directory by clicking the + next to the folder.
3. Select your serial interface and expand it to see its ports by clicking the + next to its name.
4. Right-click on the serial port you want to use and select **Communicate with Instrument**. This launches the NI-Serial Communicator.



You can use the **General**, **Flow Control**, and **Advanced** tabs to change the port settings. Use the **Test** tab to write and read strings through the COM port.

Recover Unused COM Numbers

When serial hardware is physically removed from a Windows system without first being deleted from the operating system, Windows continues to mark the COM numbers associated with that hardware as "in use." This allows you to reinsert the hardware and have those ports keep their original COM numbers.

To free up the COM number assignments that are no longer associated with hardware physically present in the system, follow these steps:

1. [Launch MAX](#).
2. Select **Tools»NI-Serial»Recover Unused COM Numbers**.

Additional Help and Resources

To access additional help and resources for the NI-Serial software and your serial hardware, refer to the following topics.

[NI-Serial Online Help](#)

[National Instruments Serial Web Site](#)

NI-Serial Online Help

The NI-Serial Help addresses questions you might have about NI-Serial and includes troubleshooting information. Follow these steps to access the NI-Serial online help:

1. [Launch MAX](#).
2. Select **Help»Help Topics»NI-Serial**.

National Instruments Serial Web Site

Follow these steps to access the National Instruments Web site for serial:

1. [Launch MAX](#).
2. Select **Help»National Instruments on the Web»NI-Serial**.

Serial ENET Configuration

[Viewing Network Settings](#)

[Configuring Network Settings](#)

[Using the CFG RESET Switch](#)

[Verifying the Hostname](#)

[Using DHCP](#)

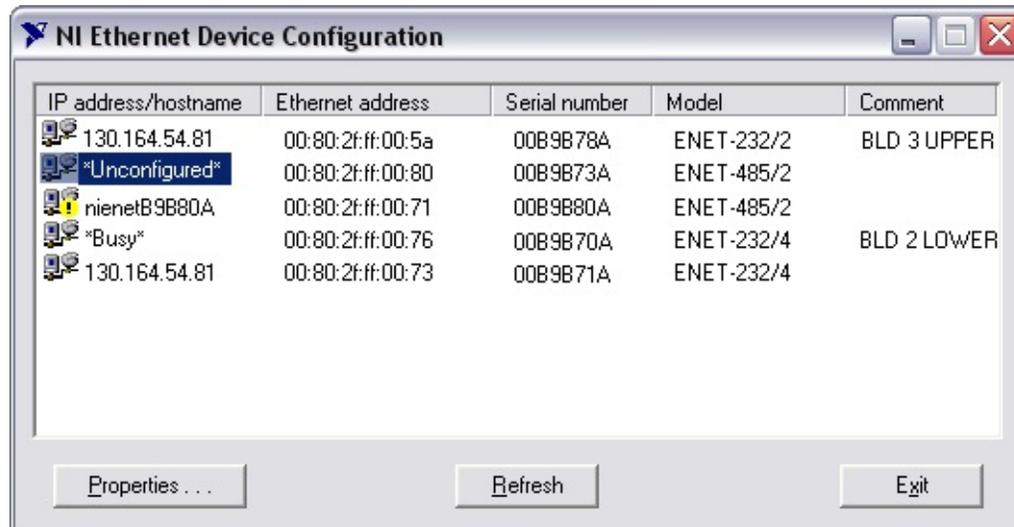
[Using Static IP Parameters](#)

[Updating Firmware](#)

Viewing Network Settings

To view network settings for a serial ENET interface, complete the following steps:

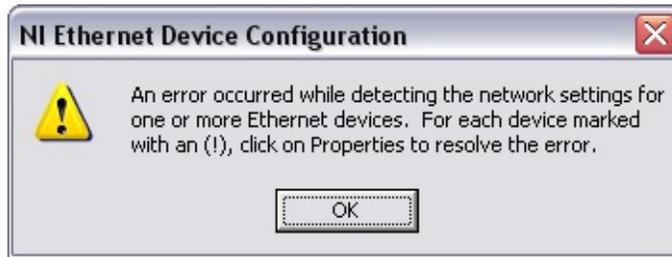
1. [Launch MAX](#).
2. Select **Tools»NI-Serial»Ethernet Device Configuration**.



NI Ethernet Device Configuration Utility

3. The NI Ethernet Device Configuration utility displays a list of devices found on your subnet, sorted by model. You can identify your device by the Ethernet address or the serial number found on the baseplate label. The listed devices can be in one of four possible states, as indicated in the **IP address/hostname** column:
 - A hostname indicates DHCP has successfully configured the device.
 - A numerical IP address indicates the device has successfully been configured with a static IP address.
 - ***Unconfigured*** indicates the device is configured to use DHCP, but DHCP failed to attain network parameters.
 - ***Busy*** indicates the device is configured to use DHCP and currently is attempting to acquire network parameters.
4. The NI Ethernet Device Configuration utility automatically verifies that the hostname for each DHCP-enabled device matches the DNS entry for the assigned IP address. This verification process automatically occurs when you either run the utility or click

Refresh. The utility alerts you as shown below if it detects a problem with the network settings.



Hostname Verification Error Detected

5. To view or change the settings for a particular serial ENET interface, select it in the list and click **Properties**. Refer to [Configuring Network Settings](#) for information about configuring your serial ENET network settings.

Configuring Network Settings

Your serial ENET interface must be in network configuration mode if you want to change its network settings. Any changes you make while in normal operating mode will not take effect. If the serial ENET interface is configured to use DHCP, and DHCP is unavailable, the unit automatically enters network configuration mode after a 60 second timeout. The unit must be in this mode before you can make changes to the network parameters.

You also can enter this mode during normal operation by pressing and holding the CFG RESET switch for three seconds. Refer to [Using the CFG RESET Switch](#) for more information about using this switch.

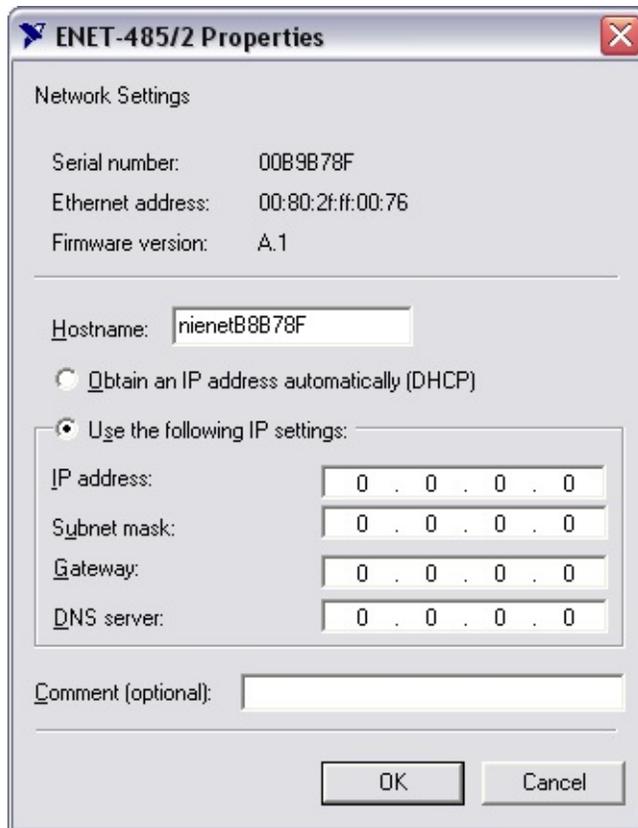
While in network configuration mode, the **PWR/RDY** LED alternates rapidly between red and yellow, and normal operation is halted.



Note If you only want to view the network configuration settings without making changes, you do not need to place the unit into network configuration mode. While in normal operation, you can follow the instructions below to examine the properties for any National Instruments Ethernet device on your subnet. However, any changes you attempt to make to the network parameters while in normal operating mode will not take effect.

To configure network settings, complete the following steps:

1. When you select **Properties...** from the NI Ethernet Device Configuration utility, the utility displays the properties for your serial ENET interface, similar to the following example.



Properties Window for an Unconfigured Serial ENET Interface

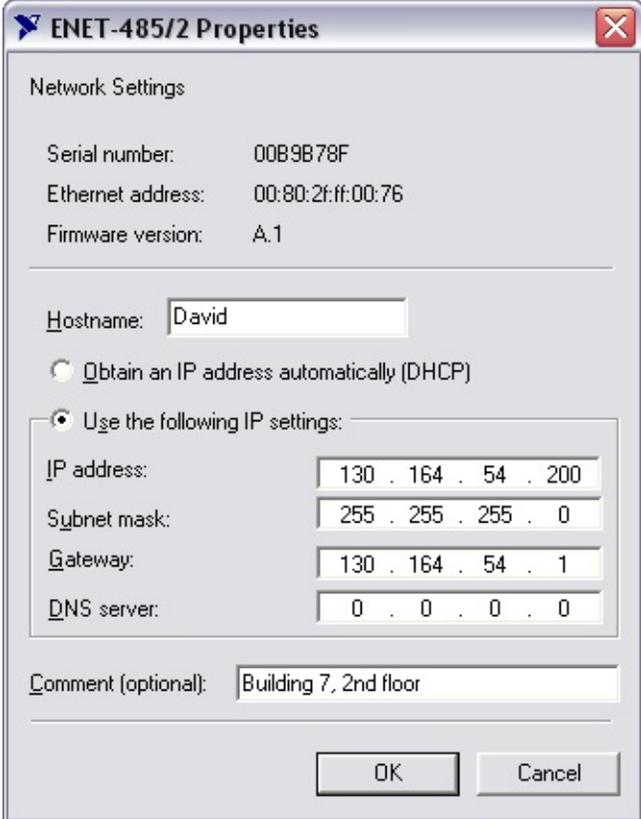
The current hostname is displayed. The hostname associates a name with a numerical IP address. **Hostname** is a required field.

The serial ENET interface attempts to use the hostname when registering with DHCP. Many DHCP servers can register the hostname and assigned IP address. You then can reliably use the hostname to communicate with your serial ENET interface even if the numerical IP address changes.

However, some DHCP servers do not implement hostname registration. The serial ENET interface requires domain name server (DNS) registration when using DHCP. If your DHCP server does not support DNS registration, you must use static network parameters. Consult your network administrator for more details. For more information about DHCP, refer to [Using DHCP](#).

2. In the **Properties** window, select either **Obtain an IP address automatically (DHCP)** or **Use the following IP settings**.
3. If you select **Obtain an IP address automatically (DHCP)**, you do not need to enter any network parameters unless you want to

change the hostname of the Ethernet device. If you select **Use the following IP settings**, enter the network parameters you have chosen for the host IP address, subnet mask, gateway IP, and DNS server IP, as in the example shown below. Refer to [Using Static IP Parameters](#) for more information.



The screenshot shows a dialog box titled "ENET-485/2 Properties". It contains the following fields and options:

- Network Settings:**
 - Serial number: 00B9B78F
 - Ethernet address: 00:80:2f:ff:00:76
 - Firmware version: A.1
- Hostname:** David
- IP Configuration:**
 - Obtain an IP address automatically (DHCP)
 - Use the following IP settings:
 - IP address: 130 . 164 . 54 . 200
 - Subnet mask: 255 . 255 . 255 . 0
 - Gateway: 130 . 164 . 54 . 1
 - DNS server: 0 . 0 . 0 . 0
- Comment (optional):** Building 7, 2nd floor

Buttons: OK, Cancel

Specifying IP Settings



Note The IP settings shown above are shown only as a format example.

4. You can enter an optional comment to help you identify the device.
5. Click **OK** to configure the device, or **Cancel** to exit without saving the configuration changes.
6. Click **Exit** (or **Cancel**) to close the NI Ethernet Device Configuration utility.

The device automatically reboots with the new configuration in effect.

Using the CFG RESET Switch

The Configuration Reset (CFG RESET) switch is a recessed switch beside the Ethernet (ENET) connector on the rear panel. Refer to [Rear Panel of a Four-Port Serial ENET Interface](#) for an illustration of its location.

This switch performs two functions, depending on whether you press it at power-on or during operation.

Using the CFG RESET Switch during Operation

While the serial ENET interface is operational, as indicated by a steady yellow **PWR/RDY** LED, you can use the CFG RESET switch to place the box into network configuration mode.

This special mode ensures that network parameters are not changed while in normal operation mode. Because you cannot change the network parameters unless you deliberately place the serial ENET interface into network configuration mode, the parameters are protected while in normal operation. Hosts cannot connect to the serial ENET interface while it is in network configuration mode.



Note Pressing the CFG RESET switch has no effect if hosts are currently connected.

Close all connections, then press and hold the CFG RESET switch and wait approximately three seconds until the **PWR/RDY** LED becomes solid red. If you release the switch prior to three seconds, the serial ENET interface continues to operate normally.

The **PWR/RDY** LED goes through the following changes during these three seconds:

1. The LED begins slowly alternating between red and yellow.
2. The alternating pattern increases in tempo.
3. At three seconds, the **PWR/RDY** LED becomes steady red. This indicates the serial ENET interface is ready to enter network configuration mode.
4. Now release the CFG RESET switch. The **PWR/RDY** LED alternates rapidly between red and yellow to indicate the serial ENET interface is now in network configuration mode.

This mode remains in effect until you switch off the serial ENET interface or you use the NI Ethernet Device Configuration utility to change its network characteristics.

Using the CFG RESET Switch at Power-On

In the event you forget the network configuration that a particular serial ENET interface is using, you can reset the unit to its default network characteristics. By pressing and holding the CFG RESET switch while you power on the serial ENET interface, the network parameters revert to the default settings as defined on the baseplate label.

You must press and hold the switch for approximately three seconds until the **PWR/RDY** LED becomes solid red. If you release the switch prior to three seconds, no change occurs to the network configuration, and the serial ENET interface continues to boot normally.

The **PWR/RDY** LED goes through the following changes during these three seconds:

1. The LED begins slowly alternating between red and yellow.
2. The alternating pattern increases in tempo.
3. At three seconds, the **PWR/RDY** LED becomes steady red. This indicates the network configuration will be set to the factory default settings.
4. When you release the CFG RESET switch, the box continues to boot as normal, and the **PWR/RDY** LED indicates the boot process as described in [ENET LED Descriptions](#).

Verifying the Hostname

To correct a problem with the hostname, complete the following steps:

1. Locate the device that has a problem. This is indicated by an (!) on the device icon, as shown in the [NI Ethernet Device Configuration utility](#).
2. Select **Properties....** A dialog box similar to the following appears.



Resolving a Hostname Verification Error

3. The utility gives you four options for resolving the verification error. Select the one that best fits your situation and click **OK**.
 - **Change the device's hostname to match the DNS entry.** Use this option if you want to accept the hostname the DHCP server assigns, or if you cannot contact the network administrator to change the DNS entry.
 - **Use static network parameters instead of DHCP.** Use this option if you cannot use the hostname the DHCP server assigns. Contact your network administrator to obtain a valid IP address, subnet, and gateway. This option disables DHCP on the device.
 - **Edit the current hostname.** Use this option to change the hostname to a name other than either the configured hostname or the name the DHCP server assigns. Contact your network administrator to obtain a valid name.

- **Keep the existing hostname.** Use this option to keep the previously assigned hostname. If you select this option, contact your network administrator to change the DNS entry.
4. Review the network parameter settings in the **Properties** window.
 5. Confirm that the device is in network configuration mode and click **OK**. The device reboots with the new settings in effect.
 6. After the device reboots, click **Refresh** to verify that the hostname is now valid.

Using DHCP

The Dynamic Host Configuration Protocol (DHCP) is designed for large networks in which networking devices are transient and network parameters cannot be statically assigned and thus tied to specific devices. DHCP eases the addition of networking devices onto a network by having a server assign necessary network parameters, including the IP address, netmask, and router information, to a newly attached network device. Optionally, if the device provides a hostname with the configuration request, DHCP may attempt to configure your network to recognize the device with the requested hostname.

DHCP requires a Domain Name Server (DNS) to associate the numerical IP address assigned with the requested hostname. Within the past few years, an Internet community standard has emerged to provide a standardized way for these services to provide dynamic domain name services. Using this standard, after DHCP assigns the numerical IP address, it can communicate with DNS to register the newly assigned IP address with the requested hostname. However, the complexity of DHCP and dynamic name registration typically requires active management by a corporate MIS department, or equivalent, because of several risks for failure.

One possible failure can occur if the pool of available addresses contains no more unassigned IP addresses. This problem is evident if DHCP fails to work and the **PWR/RDY** LED continues to flicker for longer than 90 seconds at power on. In this situation, you need to obtain a static IP address from your network administrator and configure the device yourself using the NI Ethernet Device Configuration utility. Notice that if DHCP fails to assign you an address, the current subnet might not have an address available for you to use. In this case, you may need to move your product to another subnet where there is an available IP address.

Failure also happens if communication between DHCP and DNS does not occur. In this situation, DHCP assigns an address to the device, but you cannot communicate to it using the hostname you assigned. Your network administrator may need to insert the hostname manually into the DNS table. Alternately, you can use the NI Ethernet Device Configuration utility to determine the hostname associated with your assigned IP address, and change the hostname to match the DNS entry. Refer to [Verifying the Hostname](#) for more information.

Because there are many other possible reasons for failure, contact your network administrator if you are having problems getting DHCP to work. When you can use DHCP and DNS successfully, you can use the hostname form of the IP address to communicate with your product—even if the numerical IP address changes with successive reboots.

If you are setting up a small network for your instrumentation system, National Instruments recommends using static IP addresses, because they are easier to implement and maintain. In this system, you can safely use the numerical form of the IP address to communicate with your product because the address is unlikely to change. Small networks are less likely to have domain name services available that would resolve the hostname into a numerical IP address.

Using Static IP Parameters

If DHCP is not available, you must provide the serial ENET interface with several important network parameters. These parameters are listed below.

- **IP address**—The unique, computer-readable address of a device on your network. An IP address typically is represented as four decimal numbers separated by periods (for example, 130.164.54.20). Refer to *Choosing a Static IP Address*.
- **Subnet mask**—A code that helps the network device determine whether another device is on the same network or a different network.
- **Gateway IP**—The IP address of a device that acts as a gateway, which is a connection between two networks. If your network does not have a gateway, set this parameter to 0.0.0.0.
- **DNS Server**—The IP address of a network device that stores hostnames and translates them into IP addresses. If your network does not have a DNS server, set this parameter to 0.0.0.0.

Choosing a Static IP Address

For a Network Administered by a Network Administrator

If you are adding the serial ENET interface to an existing Ethernet network, you must choose IP addresses carefully. Contact your network administrator to obtain an appropriate static IP address for your serial ENET interface. Also, have the network administrator assign the proper subnet mask, gateway, and DNS server addresses.

For a Network without a Network Administrator

If you are assembling your own small Ethernet network, you can choose your own IP addresses. The subnet mask determines the IP address format. You should use the same subnet mask as the computer you are using to configure your serial ENET interface. If your subnet mask is 255.255.255.0, the first three numbers in every IP address on the network must be the same. If your subnet mask is 255.255.0.0, only the first two numbers in the IP addresses on the network must match.

For either subnet mask, numbers between 1 and 254 are valid choices for the last number of the IP address. Numbers between 0 and 255 are valid for the third number of the IP address, but this number must be the same as other devices on your network if your subnet mask is 255.255.255.0.

If you are setting up your own network, you probably do not have a gateway or DNS server, so you should set these values to 0.0.0.0.

Updating Firmware

The NI-Serial software contains a firmware update utility you can use to access new features that may be added to the serial ENET interface in the future. It may be necessary to update the firmware to take advantage of these new features.

You need to know either the IP address or hostname of your serial ENET interface before you run the NI Ethernet Device Firmware Update utility. If you do not remember this information, you can obtain it by [Viewing Network Settings](#).



Note The NI Ethernet Device Firmware Update utility cannot update the firmware while any network connections are active. Close any open connections before you attempt a firmware update. Hosts cannot connect to the serial ENET interface while it is updating the firmware.

Complete the following steps to run the NI Ethernet Device Firmware Update utility.

1. [Launch MAX](#).
2. Expand the **Devices and Interfaces** directory by clicking on the + next to the folder.
3. Right-click any serial ENET interface and select **Update Firmware**.



NI Ethernet Device Firmware Update Utility

4. Enter the IP address or the hostname of the serial ENET interface in the **IP address or hostname** edit box. An example of an IP address is 137.65.220.40. An example of a hostname is nienetB9B76A.
5. Enter the full path of the binary file where the firmware is located in the **EEPROM binary image filename** edit box. You also can click **Browse...** to locate the file.
6. Click **Update** to update the firmware. The NI Ethernet Device Firmware Update utility communicates with the specified serial ENET interface to verify the box has no open network connections and determine the current version of the firmware in the unit. A dialog box reports that the update utility is searching for the Ethernet device.
7. The utility prompts you to confirm the change you are about to perform. Click **OK** to continue. As the update utility transfers the firmware image to your serial ENET interface, the **Update Progress** status bar fills, and the **Current status** box describes each step in the process. The **Current status** box reports whether the firmware update completed successfully or failed. The serial ENET interface automatically reboots with the new firmware in effect.
 **Caution** Do *not* power-off the serial ENET interface or disconnect the power supply while running the NI Ethernet Device Firmware Update utility. Doing so will damage the unit.
8. Click **Exit** to close the NI Ethernet Device Firmware Update utility.

Using Your Serial Hardware

The following topics describe how to use the advanced features of your NI serial hardware and list some general programming requirements.

[General Programming Requirements](#)

[Baud Rates](#)

[Performance on LabVIEW Real-Time](#)

[Advanced Features](#)

[FIFO Buffers](#)

[Retrieving the Serial Interface Type](#)

[RS-485 Transceiver Control](#)

[Setting the RS-485 Transceiver Control Mode in Windows](#)

[Setting the RS-485 Transceiver Control Mode in LabVIEW RT](#)

[Bias Resistors](#)

[USB-485 Programmatically Controlled Bias Resistors](#)

[Setting the USB-485 Programmatically Controlled Bias Resistor Mode](#)

[Retrieving the RS-232 Transceiver State with DeviceloControl](#)

[USB-232 Transceiver Control](#)

[Setting the RS-232 Transceiver Mode with DeviceloControl](#)

[Setting Ring Indicate and Data Carrier Detect \(DCD\) with DeviceloControl](#)

[Status LEDs](#)

[USB LED Descriptions](#)

[ENET LED Descriptions](#)

[ENET PWR/RDY LED Signaling](#)

General Programming Requirements

The NI-Serial software for Windows is fully integrated into the standard Windows communications software. NI serial ports are used like any other Windows communications (COM) port. Windows has standard communication functions for use within Windows applications.

When you develop your application, remember that you must use the standard Windows serial communication functions or any API that uses the standard Windows serial communication functions such as NI-VISA. For more information about NI-VISA, see ni.com/VISA. For information about Microsoft Windows serial communication functions, refer to the *Platform Software Development Kit*.

If you have LabVIEW, LabWindows/CVI, or VISA and want to use it with your serial hardware, refer to your LabVIEW, LabWindows/CVI, or VISA documentation for information about serial I/O functions.

Baud Rates

The maximum baud rates supported are listed in the table below.

Hardware Platform	Maximum Baud Rate	
	RS-232	RS-485
PCI-843x	1000 kbit/s	3000 kbit/s ¹
PCI-232/485	115.2 kbit/s	460.8 kbit/s
PXI-843x	1000 kbit/s	3000 kbit/s ¹
PXI-842x	115.2 kbit/s	460.8 kbit/s
USB	230.4 kbit/s	460.8 kbit/s
ENET	230.4 kbit/s	460.8 kbit/s
ExpressCard	230.4 kbit/s	460.8 kbit/s
PCMCIA	921.6 kbit/s ²	921.6 kbit/s

All NI serial hardware supports standard baud rates. In addition, the PCI/PXI-843x family of hardware supports any baud rate between 2 bit/s and the maximum supported baud rate for that interface.

National Instruments considers the following baud rates to be standard. NI serial hardware supports these rates up to the maximum rate specified. Your device may also support additional baud rates not listed below:

110 14400
300 19200
600 38400
1200 57600
2400 115200
4800 230400
9600 460800

To set the baud rate, set the VISA Baud attribute or use the Windows SetCommState function and pass the actual value of the baud rate in the **BaudRate** field of the **DCB** structure.

¹The two-wire auto control mode for the RS-485 transceiver has a maximum baud rate of 2000 kbit/s.

²The PCMCIA-232/4 has a maximum baud rate of 115.2 kbit/s.

Performance on LabVIEW Real-Time

NI-Serial users on LabVIEW Real-Time have control over whether secondary interrupt handling is enabled. When secondary interrupt handling is enabled, I/O is processed at a priority lower than time-critical. Using secondary interrupt handling reduces jitter for time-critical loops and makes serial hardware interrupts more deterministic. However, it may also reduce I/O performance and can increase the possibility of overflowing the RX FIFO buffer. Secondary interrupt handling is a setting at the system-wide level and is enabled by default.

To enable or disable this feature, complete the following steps:

1. Copy to your host the `niserial.dbs` file in the `/ni-rt/system` directory on the RT target.
2. In the `niserial.dbs` file, find the entry **SecondaryInterrupts** and modify it accordingly:
 - To enable secondary interrupt handling, set **SecondaryInterrupts** to 1.
 - To disable secondary interrupt handling, set **SecondaryInterrupts** to 0.
3. Download the updated `niserial.dbs` file to the target, overwriting the existing file in the `/ni-rt/system` directory.
4. Reboot your RT target.

Advanced Features

In addition to supporting standard 16550-compatible UART functionality, some NI serial hardware supports additional advanced features:

- Adjustable FIFO levels
- RS-485 transceiver control
- RS-485 socketed bias resistors
- RS-485 programmatically controlled bias resistors
- RS-232 transceiver state
- RS-232 DTE/DCE transceiver control

To determine which features your product supports, refer to the following table.

Serial Hardware Features

Hardware	<u>Adjustable FIFO Settings</u>	<u>Get Interface Type</u>	<u>RS-485 Transceiver Control</u>	<u>RS-485 Socketed Bias Resistors</u>	<u>RS-485 Programmatically Controlled Bias Resistors</u>
PCI/PXI-8430, PCI/PXI-8432	X	X			
PCI-232, PXI-8420, PXI-8422	X	X			
PCI/PXI-8431 eight port, PCI-485 eight port, PXI-8421 eight port	X	X	X		
All other PCI/PXI-8431, PCI/PXI-8433, PCI-485, PXI-	X	X	X	X	

8421, and PXI-8423					
USB-232 one port		X			
USB-232 two and four port		X			
USB-485 one port		X	X		X
USB-485 two and four port		X	X	X	X
ENET-232					
ENET-485			X	X	
ExpressCard- 8420		X			
ExpressCard- 8421		X	X		X
PCMCIA-232	X	X			
PCMCIA-485	X	X	X		

FIFO Buffers

FIFO buffers are present on the 16550-compatible UARTs—one for the transmitter and one for the receiver. The Receive Buffer control sets the number of characters received in the FIFO before the PC is interrupted to read the data. The Transmit Buffer control sets the maximum number of bytes written to the FIFO in a block when the PC is interrupted to write the data.

The built-in ports on PXI RT controllers and the PCI, PXI, and PCMCIA plug-in serial boards have configurable FIFO settings. Use the **Advanced** tab in MAX to configure your FIFO settings. When you configure FIFO settings, consider the following points:

- You can select larger FIFO buffer sizes to reduce the number of interrupts your PC receives and thereby minimize system overhead.
- If transfer rates are high, you can lower the Receive Buffer value to prevent overrun errors due to interrupt latency.
- If your data transfer sizes are small, and your Receive Buffer value is above your data sizes, your system is less efficient. For maximum efficiency, set your Receive Buffer value such that your data transfer size is a multiple of that value.

Retrieving the Serial Interface Type (RS-232 or RS-485)

There are two ways to retrieve the serial interface type (RS-232 or RS-485):

- Read the NI-VISA wire mode attribute.
- The NI-Serial software uses programmatic control codes and the DeviceIoControl Windows function to retrieve the serial interface type.

To retrieve the serial interface type using the DeviceIoControl Windows function, complete the following steps:

1. Add the following lines to your source code:

```
#include <winioctl.h>  
#include <NiSerial.h>
```



Note The header file NiSerial.h is included on your NI-Serial software CD. You also can find it in the NI-Serial folder where you installed your National Instruments software (typically, C:\Program Files\National Instruments\NI-Serial).

2. The SERIAL_INTERFACE attribute can have the following values:
 - RS485_INTERFACE
 - RS232_INTERFACE
3. To retrieve the serial interface type, use NISERIAL_GET_INTERFACE_TYPE and DeviceIoControl with the following code:

```
SERIAL_INTERFACE l_SerialInterface;  
DWORD          l_ByteCount;
```

```
DeviceIoControl (  
    PortHandle,  
    NISERIAL_GET_INTERFACE_TYPE,  
    (LPVOID) NULL,
```

```
0,  
(LPVOID) &l_SerialInterface,  
sizeof (l_SerialInterface),  
(LPDWORD) &l_ByteCount,  
NULL  
);
```

RS-485 Transceiver Control

The RS-485 hardware supports four modes of hardware transceiver control. You can use hardware transceiver control to enable and disable your transmitters and receivers so that they function on different bus topologies. The following table lists the status of the transmitters and receivers under each transceiver control mode.

RS-485 Transceiver Control Modes

Mode	Transmitter	Receiver
Four-wire mode	Always enabled	Always enabled
Two-wire mode: DTR controlled with echo	Enabled when DTR asserted	Always enabled
Two-wire mode: DTR controlled	Enabled when DTR asserted	Enabled when DTR unasserted
Two-wire mode: Auto Control	Enabled when transmitting data	Enabled when not transmitting data

Four-Wire Mode

Use the four-wire mode for most full-duplex systems. In this mode, the transmitter and receiver are always enabled. This mode is the default.

Two-Wire Mode: DTR Controlled with Echo

You can use this mode in half-duplex systems where you need to control the transmitter programmatically. In the DTR-with-echo mode, the transmitter is tri-stated when the DTR signal of the UART (Universal Asynchronous Receiver/Transmitter) is unasserted. To transmit, your application first must enable the transmitter by asserting DTR. After the data is fully transmitted, your application unasserts DTR to disable the transmitter. Because the receiver is always enabled in this mode, you receive packets not only from other devices, but also your transmitter. Thus, your receiver echoes all data you transmit.

Two-Wire Mode: DTR Controlled

This mode is similar to the two-wire, DTR-with-echo mode. Although this mode uses the same method as the DTR-with-echo mode to control the transmitter, the hardware automatically disables the receiver whenever the transmitter is enabled. Thus, you do not receive the packets sent from your transmitter.

Two-Wire Mode: Auto Control

In this mode, the serial hardware transparently enables the transmitter and receiver in a two-wire system. Use this mode to remove the burden of transceiver control from your application. The hardware automatically enables the transmitter for each byte to be transmitted. Also, the hardware disables the receiver whenever the transmitter is enabled, so you do not receive the packets sent from your transmitter.



Note When you are communicating with a two-wire device, National Instruments recommends that you use the two-wire Auto Control mode. Because this mode handles the transmitter/receiver enabling for a two-wire connection in your hardware, it reduces the software overhead required to perform this operation in your application program.

For more information about serial communication in two or four-wire modes, refer to [Serial Communication Issues](#).

Setting the RS-485 Transceiver Control Mode in Windows

Setting the Default Mode of Operation

To set the default mode of operation:

- Use the **Advanced** tab in Measurement & Automation Explorer (MAX).

Modifying the RS-485 Transceiver Mode Programmatically

To modify the RS-485 transceiver mode programmatically:

- Change the NI-VISA wire mode attribute.
- The NI-Serial software uses programmatic control codes and the DeviceIoControl Windows function for programming the RS-485 transceiver control mode.

Setting the RS-485 Transceiver Control Mode on PCI, PXI, USB, ExpressCard, and PCMCIA Hardware

To set and retrieve the RS-485 transceiver control mode using the DeviceIoControl Windows function on PCI, PXI, USB, ExpressCard, and PCMCIA RS-485 hardware, complete the following steps:

1. Add the following lines to your source code:

```
#include <winioctl.h>  
#include <NiSerial.h>
```



Note You can find the header file NiSerial.h in the NI-Serial folder where you installed your National Instruments software (typically, C:\Program Files\National Instruments\NI-Serial). It is also included on your NI-Serial software CD.

2. The TRANSCEIVER_MODE attribute can have the following values:
 - RS485_MODE_4WIRE
 - RS485_MODE_2W_ECHO
 - RS485_MODE_2W_DTR
 - RS485_MODE_2W_AUTO
3. To set the RS-485 transceiver control mode, use NISERIAL_SET_RS485_MODE and DeviceIoControl. For

example, to set to two-wire Auto Control mode, use the following code:

```
TRANSCEIVER_MODE l_TransceiverMode = RS485_MODE_2  
DWORD          l_ByteCount;
```

```
DeviceIoControl (  
    PortHandle,  
    NISERIAL_SET_RS485_MODE,  
    (LPVOID) &l_TransceiverMode,  
    sizeof (l_TransceiverMode),  
    (LPVOID) NULL,  
    0,  
    (LPDWORD) &l_ByteCount,  
    NULL  
);
```

4. To retrieve the current RS-485 transceiver control mode, you can use NISERIAL_GET_RS485_MODE and DeviceIoControl with the following code:

```
TRANSCEIVER_MODE l_TransceiverMode;  
DWORD            l_ByteCount;
```

```
DeviceIoControl (  
    PortHandle,  
    NISERIAL_GET_RS485_MODE,  
    (LPVOID) NULL,  
    0,  
    (LPVOID) &l_TransceiverMode,  
    sizeof (l_TransceiverMode),  
    (LPDWORD) &l_ByteCount,  
    NULL  
);
```

Setting the RS-485 Transceiver Control Mode on ENET Hardware

To set and retrieve the RS-485 transceiver control mode using the

DeviceIoControl Windows function on ENET RS-485 hardware, complete the following steps:

1. Add the following lines to your source code:

```
#include <winioctl.h>
#define SERIAL_TM_4WIRE    0x80
#define SERIAL_TM_2W_ECHO 0x81
#define SERIAL_TM_2W_DTR  0x81
#define SERIAL_TM_2W_AUTO 0x83
#define IOCTL_SERIAL_SET_TRANSCEIVER_MODE
    CTL_CODE(FILE_DEVICE_SERIAL_PORT,37,METHOD_BI
```

2. To set the RS-485 transceiver control mode, use IOCTL_SERIAL_SET_TRANSCEIVER_MODE and DeviceIoControl. For example, to set to two-wire Auto Control mode, use the following code:

```
ULONG bytecount;
TRANSCEIVER_MODE transceiver_mode = SERIAL_TM_2W_
DeviceIoControl(
    comhandle,
    IOCTL_SERIAL_SET_TRANSCEIVER_MODE,
    (PVOID) &transceiver_mode,
    sizeof(transceiver_mode),
    (PVOID) NULL, 0,
    &bytecount,
    NULL
);
```

Setting the RS-485 Transceiver Control Mode in LabVIEW Real-Time

Setting the Default Mode of Operation

To set the default mode of operation:

- For PXI:

Use the **Advanced** tab in Measurement & Automation Explorer (MAX) to define the default transceiver control mode.

- For FieldPoint devices supporting multiple transceiver control modes, follow these steps:



Note Without modification, the default mode is set to four-wire.

1. Using an FTP client, download the `/ni-rt/system/niserial.dbs` file from the LabVIEW Real-Time target.
2. Locate the entry in the `niserial.dbs` file that says `SerialPortName = "COMx"`, where `COMx` corresponds to the port you want to modify.
3. In the next line, add the following entry to the `niserial.dbs` file:

`TransceiverMode = <Transceiver Mode>`

where `<Transceiver Mode>` is:

<Transceiver Mode>	Description
128	Four-wire mode
129	Two-wire mode: DTR controlled with echo
130	Two-wire mode: DTR controlled

4. Copy the file back to its original location on the LabVIEW Real-Time target, replacing the original file. Reboot your LabVIEW Real-Time target for your changes to take effect.

For example, to configure COM4 to use two-wire Auto Control mode, the following would appear in the `niserial.dbs` file:

```
SerialPortName = "COM4"  
TransceiverMode = 131
```

Modifying the Transceiver Control Mode Programmatically

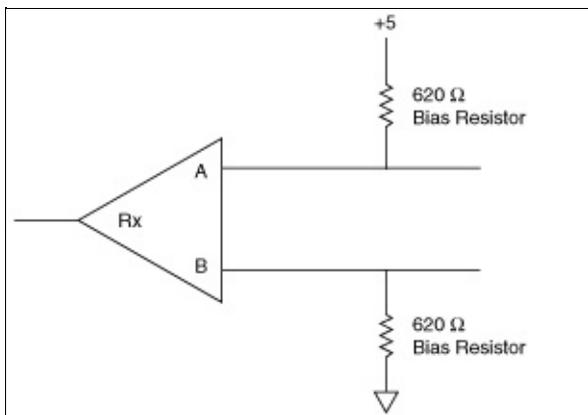
To modify the transceiver control mode programmatically:

- In LabVIEW Real-Time, you must use the VISA wire mode attribute to change the RS-485 transceiver control mode programmatically.

Bias Resistors

An RS-485 transmission line enters an indeterminate state if no nodes are transmitting on it. This indeterminate state can cause the receivers to receive invalid data bits from noise picked up on the cable. To prevent a line from receiving these data bits, the transmission line can be forced into a known state by installing two bias resistors at one node on the transmission line. Doing so creates a voltage divider that forces the voltage between the differential pair to be greater than 200 mV, the threshold voltage for the receiver.

The following figure shows a transmission line using bias resistors.



Transmission Line Using Bias Resistors

On the PCI RS-485 two and four-port serial cards, PXI RS-485 two and four-port serial cards, and ENET RS-485 two and four-port serial cards, there are four user-configurable bias resistors in front of each connector. These resistors are socketed and pre-loaded with 620 Ω resistors. They are connected to the receive signals of each port to maintain a known state when the bus is idle. The connections are made as follows:

- RXD+ and CTS– are pulled up to +5 V
- RXD– and CTS+ are pulled down to GND

Rather than using two 620 Ω resistors at one node, you can increase the value of the resistors and put them at every node. For instance, if there are eight nodes in a system, you can use 4.7 kΩ resistors at each node to effectively achieve the same result.

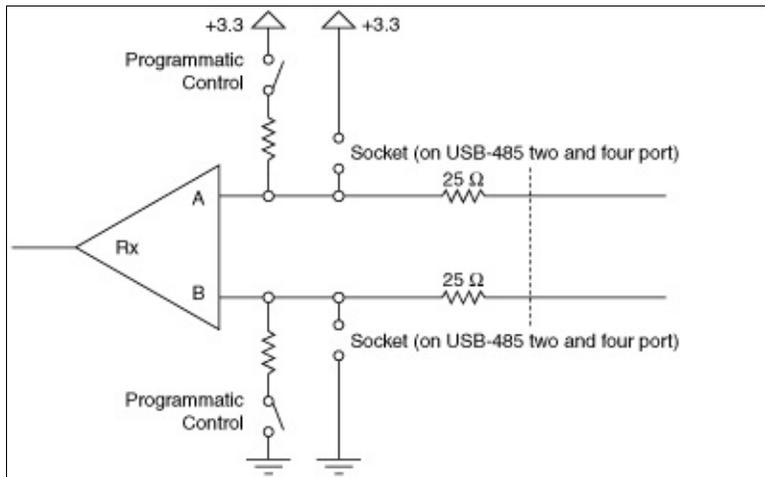
If your application needs termination, an optional external RS-485 DB-9 terminator that provides 120 Ω is available.

USB-485 Programmatically Controlled Bias Resistors

On the USB-485 hardware, there are programmatically controlled onboard bias resistors. In addition, the USB-485 two and four-port hardware has user-configurable socketed bias resistors. By default, the USB-485 hardware uses the programmatically controlled bias resistors, which are connected to the receive signals of each port to maintain a known state when the bus is idle. The connections are made as follows:

- RXD+ and CTS– are pulled up to +3.3 V
- RXD– and CTS+ are pulled down to GND

The following figure shows a USB transmission line using bias resistors.



USB-485 Transmission Line Using Bias Resistors

The values of the programmatically controlled bias resistors have been calculated such that there is a voltage of at least 200 mV between the differential pair. Rather than using the programmatically controlled bias resistors, you can load custom values of resistors into the sockets in front of each connector or use external bias resistors. If these resistors are installed, make sure the programmatically controlled bias resistors are disabled.

-  **Note** Using custom bias resistors on the USB-485 two port may violate the USB suspend current specification.
-  **Note** The series 25 Ω resistors in the figure above are for short-circuit protection and do not impact serial communication.

Setting the USB-485 Programmatically Controlled Bias Resistor Mode

The USB-485 hardware can programmatically control the onboard (nonsocketed) bias resistors. By default, these resistors are enabled during normal device operation and disabled during USB suspend. There are two ways to control these resistors:

- Use the **Advanced** tab in MAX. If the **Enable Dynamic Bias Resistors** box is checked, the resistors are enabled automatically whenever a serial port is open during normal operation. If the box is not checked, the resistors are disabled.
- Your software can enable or disable the bias resistors programmatically by making DeviceIoControl calls to the serial driver.

Setting the RS-485 Programmatically Controlled Bias Resistors with DeviceloControl

The NI-Serial software uses programmatic control codes and the DeviceloControl Windows function for programming the RS-485 bias resistor mode. To set and retrieve the RS-485 bias resistor mode, complete the following steps:

1. Add the following lines to your source code:

```
#include <winioctl.h>
#include <NiSerial.h>
```



Note The header file NiSerial.h is included on your NI-Serial software CD. You also can find it in the NI-Serial folder where you installed your National Instruments software (typically, C:\Program Files\National Instruments\NI-Serial).

2. The SERIAL_RS485_BIAS attribute can have the following values:

- RS485_BIAS_OFF
- RS485_BIAS_ON

3. To set the RS-485 bias resistor mode, use NISERIAL_SET_RS485_BIAS and DeviceloControl. For example, to enable programmatically controlled RS-485 bias resistors, use the following code:

```
SERIAL_RS485_BIAS l_SerialRs485Bias = RS485_BIAS_ON;
DWORD          l_ByteCount;
```

```
DeviceIoControl (
    PortHandle,
    NISERIAL_SET_RS485_BIAS,
    (LPVOID) &l_SerialRs485Bias,
    sizeof (l_SerialRs485Bias),
    (LPVOID) NULL,
    0,
```

```
(LPDWORD) &l_ByteCount,  
NULL  
);
```

4. To retrieve the current RS-485 bias resistor mode, you can use `NISERIAL_GET_RS485_BIAS` and `DeviceIoControl` with the following code:

```
SERIAL_RS485_BIAS l_SerialRs485Bias;  
DWORD l_ByteCount;
```

```
DeviceIoControl (  
    PortHandle,  
    NISERIAL_GET_RS485_BIAS,  
    (LPVOID) NULL,  
    0,  
    (LPVOID) &l_SerialRs485Bias,  
    sizeof (l_SerialRs485Bias),  
    (LPDWORD) &l_ByteCount,  
    NULL  
);
```

Retrieving the RS-232 Transceiver State

The NI-Serial software returns which mode the RS-232 transceivers are currently operating in, and whether the serial connection is valid.



Note *Transceiver mode* refers to the programmed mode. *Transceiver state* refers to the physical state of the transceivers (actual operating mode and whether valid or invalid).

There are two ways to retrieve the RS-232 transceiver state and transceiver mode:

- Read the NI-VISA wire mode attribute to retrieve the transceiver state and the NI-VISA IsPortConnected attribute to retrieve the transceiver mode.
- The NI-Serial software uses programmatic control codes and the DeviceIoControl Windows function to retrieve the RS-232 transceiver state and transceiver mode.

To retrieve the RS-232 transceiver state using the DeviceIoControl Windows function, complete the following steps:

1. Add the following lines to your source code:

```
#include <winioctl.h>  
#include <NiSerial.h>
```



Note The header file NiSerial.h is included on your NI-Serial software CD. You also can find it in the NI-Serial folder where you installed your National Instruments software (typically, C:\Program Files\National Instruments\NI-Serial).

2. The RS-232 transceiver state attribute is contained within a struct of type SERIAL_RS232_STATE. The struct members and possible values are shown below:

Member Name	Values
PortMode	RS232_MODE_DTE RS232_MODE_DCE RS232_MODE_AUTO

Connection SERIAL_CONNECTION_INVALID
SERIAL_CONNECTION_VALID

3. To retrieve the current RS-232 transceiver state, use NISERIAL_GET_RS232_STATE and DeviceIoControl with the following code:

```
SERIAL_RS232_STATE l_SerialRs232State;  
DWORD             l_ByteCount;
```

```
DeviceIoControl (  
    PortHandle,  
    NISERIAL_GET_RS232_STATE,  
    (LPVOID) NULL,  
    0,  
    (LPVOID) &l_SerialRs232State,  
    sizeof (l_SerialRs232State),  
    (LPDWORD) &l_ByteCount,  
    NULL  
);
```

USB-232 Transceiver Control

The USB-232 two and four-port hardware supports both DTE and DCE transceiver modes. By default, the RS-232 transceiver mode is forced to DTE, but you can adjust this in two ways:

- Use the **Advanced** tab in MAX. To set the transceiver mode to be used whenever a port is opened, select the desired mode from the transceiver mode pull-down menu.
- Change the transceiver mode programmatically by making a `DeviceloControl` function call to the serial driver.

The following table describes the RS-232 modes.

RS-232 Modes

RS-232 Mode	Description
Force DTE (Default)	The transceiver is forced into DTE mode
Force DCE	The transceiver is forced into DCE mode
Auto 232	The transceiver automatically switches between DTE and DCE modes about four times a second until a valid RS-232 signal is detected



Note When using Auto232, if you connect two serial ports set to Auto232, or if valid RS-232 signals are present in both modes, the final mode may be either DTE or DCE.



Note A valid RS-232 signal is defined as at least one valid RS-232 voltage detected on any receiver pin (RXD, CTS, DSR, RI, or DCD).

The USB-232 hardware also allows you to set and retrieve the state of the of the Ring Indicate (RI) and Data Carrier Detect (DCD) lines along with the Transceiver Mode.

[Setting the RS-232 Transceiver Mode with DeviceloControl](#)

[Setting Ring Indicate \(RI\) and Data Carrier Detect \(DCD\) with DeviceloControl](#)

Setting the RS-232 Transceiver Mode with DeviceloControl

There are two ways to set and retrieve the RS-232 transceiver state:

- Set and read the NI-VISA wire mode attribute.
- The NI-Serial software uses programmatic control codes and the DeviceloControl Windows function to retrieve the RS-232 transceiver state.

To set and retrieve the RS-232 transceiver state using the DeviceloControl Windows function, complete the following steps:

1. Add the following lines to your source code:

```
#include <winiioctl.h>  
#include <NiSerial.h>
```



Note The header file NiSerial.h is included on your NI-Serial software CD. You also can find it in the NI-Serial folder where you installed your National Instruments software (typically, C:\Program Files\National Instruments\NI-Serial).

2. The TRANSCEIVER_MODE attribute can have the following values:

- RS232_MODE_DTE
- RS232_MODE_DCE
- RS232_MODE_AUTO

3. To set the RS-232 transceiver, use NISERIAL_SET_RS232_MODE and DeviceloControl. For example, to set to Force DCE mode, use the following code:

```
TRANSCEIVER_MODE l_TransceiverMode = RS232_MODE_DCE;  
DWORD           l_ByteCount;
```

```
DeviceIoControl (
```

```
PortHandle,  
NISERIAL_SET_RS232_MODE,  
(LPVOID) &l_TransceiverMode,  
sizeof (l_TransceiverMode),  
(LPVOID) NULL,  
0,  
(LPDWORD) &l_ByteCount,  
NULL  
);
```

4. To retrieve the current RS-232 transceiver mode, you can use NISERIAL_GET_RS232_MODE and DeviceIoControl with the following code:

```
TRANSCEIVER_MODE l_TransceiverMode;  
DWORD l_ByteCount;
```

```
DeviceIoControl (  
PortHandle,  
NISERIAL_GET_RS232_MODE,  
(LPVOID) NULL,  
0,  
(LPVOID) &l_TransceiverMode,  
sizeof (l_TransceiverMode),  
(LPDWORD) &l_ByteCount,  
NULL  
);
```

Setting Ring Indicate (RI) and Data Carrier Detect (DCD) with DeviceloControl

There are two ways to set and retrieve these RS-232 signals:

- Set and read the NI-VISA Line RI State and Line DCD State attributes.
- The NI-Serial software uses programmatic control codes and the DeviceloControl Windows function to retrieve these RS-232 signals.

To retrieve these RS-232 signals using the DeviceloControl Windows function, complete the following steps:

1. Add the following lines to your source code:

```
#include <winioctl.h>
#include <NiSerial.h>
```



Note The header file NiSerial.h is included on your NI-Serial software CD. You also can find it in the NI-Serial folder where you installed your National Instruments software (typically, C:\Program Files\National Instruments\NI-Serial).

2. The SERIAL_RI_OUT attribute can have the following values:

- SERIAL_RI_ON
- SERIAL_RI_OFF

3. To set the RS-232 signal RI when the transceiver is in DCE mode, use the following code:

```
DWORD l_ByteCount;
```

```
DeviceIoControl (  
    PortHandle,  
    NISERIAL_SET_RI,  
    (LPVOID) NULL,  
    0,
```

```
(LPVOID) NULL,  
0,  
(LPDWORD) &l_ByteCount,  
NULL  
);
```

4. To clear the RS-232 signal RI when the transceiver is in DCE mode, use the following code:

```
DWORD l_ByteCount;
```

```
DeviceIoControl (  
    PortHandle,  
    NISERIAL_CLR_RI,  
    (LPVOID) NULL,  
    0,  
    (LPVOID) NULL,  
    0,  
    (LPDWORD) &l_ByteCount,  
    NULL  
);
```

5. To retrieve the current state of the RS-232 signal RI when the transceiver is in DCE mode, use the following code:

```
SERIAL_RI_OUT l_SerialRiOut;  
DWORD l_ByteCount;
```

```
DeviceIoControl (  
    PortHandle,  
    NISERIAL_GET_RI,  
    (LPVOID) NULL,  
    0,  
    (LPVOID) &l_SerialRiOut,  
    sizeof (l_SerialRiOut),  
    (LPDWORD) &l_ByteCount,  
    NULL  
);
```

6. The SERIAL_DCD_OUT attribute can have the following values:

- SERIAL_DCD_ON
- SERIAL_DCD_OFF

7. To set the RS-232 signal DCD when the transceiver is in DCE mode, use the following code:

```
DWORD l_ByteCount;
```

```
DeviceIoControl (
    PortHandle,
    NISERIAL_SET_DCD,
    (LPVOID) NULL,
    0,
    (LPVOID) NULL,
    0,
    (LPDWORD) &l_ByteCount,
    NULL
);
```

8. To clear the RS-232 signal DCD when the transceiver is in DCE mode, use the following code:

```
DWORD l_ByteCount;
```

```
DeviceIoControl (
    PortHandle,
    NISERIAL_CLR_DCD,
    (LPVOID) NULL,
    0,
    (LPVOID) NULL,
    0,
    (LPDWORD) &l_ByteCount,
    NULL
);
```

9. To retrieve the current state of the RS-232 signal DCD when the transceiver is in DCE mode, use the following code:

```
SERIAL_DCD_OUT l_SerialDcdOut;  
DWORD         l_ByteCount;
```

```
DeviceIoControl (  
    PortHandle,  
    NISERIAL_GET_DCD,  
    (LPVOID) NULL,  
    0,  
    (LPVOID) &l_SerialDcdOut,  
    sizeof (l_SerialDcdOut),  
    (LPDWORD) &l_ByteCount,  
    NULL  
);
```

Status LEDs

The ENET serial and USB serial two and four-port hardware uses bicolor light-emitting diodes (LEDs) to indicate device and port status.

[USB LED Descriptions](#)

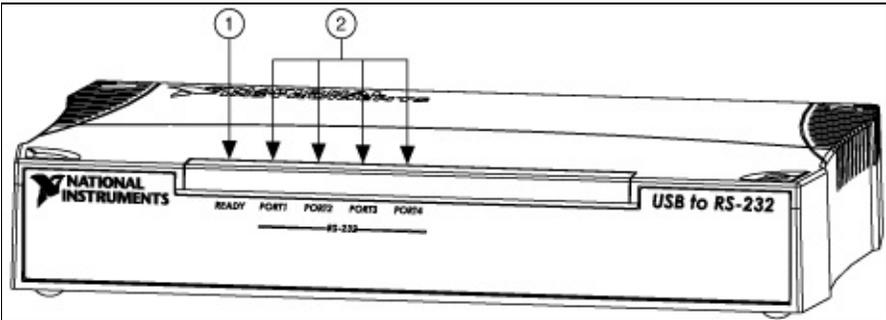
[ENET LED Descriptions](#)

[ENET PWR/RDY LED Signaling](#)

USB LED Descriptions

The USB serial two and four-port hardware uses bicolor LEDs to indicate device and port status. The following table describes these LEDs; the following figure shows their location.

LED	Description
Ready	<p>Dim Red—Powered, but not connected to USB (USB-485/4 only)</p> <p>Red—Powered and connected to USB, but not fully configured</p> <p>Yellow—Device is ready (normal operation)</p> <p>Blinking Red or Red-Yellow—Device error. Contact NI.</p>
Port x	<p>Solid Red—Port is open, but no valid signals detected (USB-232 only)</p> <p>Solid Green—Port is open</p> <p>Blinking Yellow—Port is transmitting</p> <p>Blinking Green—Port is receiving</p> <p>Alternated Blinking Green/Yellow—Port is transmitting and receiving</p> <p>Blinking Red—Port error (framing error, FIFO overrun, or parity error)</p>



1 READY LED

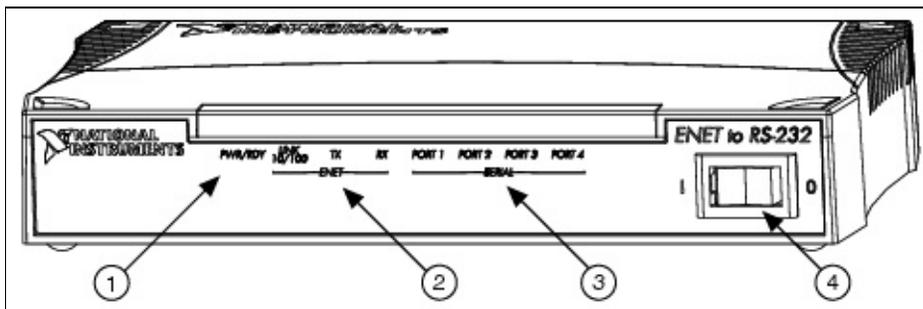
2 PORT LEDs

USB-Serial Hardware LEDs

ENET LED Descriptions

The ENET serial hardware uses bicolor LEDs to indicate device and port status. The following table describes these LEDs; the following figure shows their location.

LED	Description
PWR/RDY	Flashes rapidly at start-up while performing self tests and when acquiring network parameters. A steady yellow state indicates the box is ready for operation. Slow flashing pattern indicates an error occurred.
LINK 10/100	Indicates the serial ENET interface detected a twisted pair (10Base-T or 100Base-TX) link. The color indicates the connection speed. If yellow, the speed is 10 Mbits/s. If green, the speed is 100 Mbits/s.
TX	Indicates the serial ENET interface is transmitting to the Ethernet network.
RX	Indicates the serial ENET interface is receiving Ethernet network traffic.
PORT <i>x</i>	Indicates which serial ports are open.



- 1 Power/Ready LED
- 2 Ethernet LEDs
- 3 Serial Port LEDs
- 4 Power Switch

Four-Port Serial ENET Hardware LEDs

ENET PWR/RDY LED Signaling

This topic describes how to interpret the **PWR/RDY** LED error codes.

The **PWR/RDY** LED has several purposes on the serial ENETs. When you first power on the unit, the **PWR/RDY** LED alternates rapidly between red and yellow while it completes its power-on self-tests and acquires network parameters. When the tests complete successfully and the IP address is assigned from either nonvolatile memory or the network, the **PWR/RDY** LED remains steady yellow, indicating that the unit is ready to operate.

The **PWR/RDY** LED also alternates rapidly between red and yellow while the device is in network configuration mode. At other times, the **PWR/RDY** LED blinks slowly in a recognizable pattern to alert you of internal errors. Use the following steps to interpret and record the pattern that the **PWR/RDY** LED flashes, and then contact National Instruments.



Note By recording the **PWR/RDY** LED error messages before calling National Instruments, you can save yourself time, and customer support can answer your questions more accurately and efficiently. Do not switch off power to your serial ENET before recording the flashing **PWR/RDY** LED pattern.

PWR/RDY LED signaling can report up to 81 different errors. The errors are numbered from 11 to 99 and are reported through sequences of **PWR/RDY** LED flashes.



Note There is no zero in any error message. This means that error message numbers 0–10, 20, 30, 40, 50, 60, 70, 80, and 90 are not possible.

Step 1. Count the Long Flashes

A three-second interval, during which the **PWR/RDY** LED is yellow, separates each repetition of the sequence. The sequence begins with a series of long one-second flashes—that is, one second red, one second yellow. These long flashes represent the digit in the tens column. There can be one to nine long flashes, which represent digits 1 through 9. For example, one long flash represents the digit 1 in the tens column, and nine long flashes represent the digit 9 in the tens column.

Step 2. Count the Short Flashes

The long flashes are followed by shorter flashes; each short flash lasts about one-fifth of a second—that is, one-fifth of a second red, one-fifth of a second yellow. These short flashes represent the digit in the ones column. Again, there can be one to nine flashes, which represent the digits 1 through 9. For example, one short flash represents the digit 1 in the ones column, and nine short flashes represent the digit 9 in the ones column.

Using this method, the **PWR/RDY** LED flashes the following sequence to represent error message 11:

<three seconds yellow> <one long red flash> <one short red flash>
<three seconds yellow>...

The **PWR/RDY** LED flashes the following sequence to represent error message 31:

<three seconds yellow> <three long red flashes> <one short red flash>
<three seconds yellow>...

Step 3. Record Your Error Message Number

When you have computed your error message number, write it down and also note the ON/OFF state of the **LINK**, **TX**, and **RX** LEDs. Have this information available when calling National Instruments.

Connector Descriptions

The following topics describe serial hardware connectors.

[Connecting Cables](#)

[Connecting Two-Wire Devices](#)

[DB-9 Connector](#)

[Modular Jack Connector](#)

[DB-25 Connector](#)

[68-Pin Connector](#)

[100-Pin Connector](#)

Connecting Cables

For the PCMCIA and ExpressCard hardware, two-port PCI and PXI serial hardware, and all USB and ENET hardware, you can use the standard DB-9 connector found on most serial cables. To use the DB-9 connector with the four-port PCI and PXI serial hardware, you need the 10-position modular jack to DB-9 cable, which is available from National Instruments. You can also use a DB-25 connector with the four-port serial boards by ordering the 10-position modular jack to DB-25 converter cable from National Instruments. The eight-port serial boards include a pig tail cable adapter, providing eight standard DB-9 connectors. The 16-port boards include a breakout box, providing 16 standard DB-9 connectors.



Note To achieve the specified isolation voltage for four-port isolated PCI or PXI serial boards, use *only* the 10-position modular jack to DB-9 cable included in your four-port isolated PCI or PXI serial board kit.

For more information about cable connectors, see the connector description topics.

Connecting Two-Wire Devices

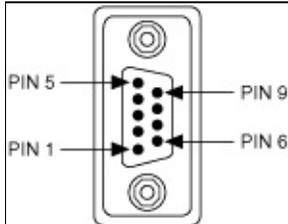
The NI RS-485 hardware is designed to work with either two or four-wire devices. If you are using a two-wire device, refer to the device documentation for specific wiring instructions.

In general, half-duplex networks use a single twisted pair of wires for communication in both directions, so you must connect both the transmitter and the receiver at each end of the same pair of wires. For example, to connect an RS-485 data acquisition device to a port on your board using half-duplex communication, you need a single twisted pair of wires. At the board, connect the TXD+ and RXD+ signals (pins 8 and 4 on a DB-9 connector, pins 5 and 20 on a DB-25 connector) together and to one wire. Connect the other end of this wire to both the TXD+ and RXD+ signals on the data acquisition device. Use the same method to connect the TXD– and RXD– signals (pins 9 and 5 on a DB-9 connector, pins 22 and 7 on a DB-25 connector) to the second wire.

For information about setting the transceiver mode for two-wire communication, refer to [RS-485 Transceiver Control](#). For more information about duplex architectures, refer to [Serial Port Information](#).

DB-9 Connector

The following figure and table give the pin locations and descriptions of the DB-9 connector, the 10-position modular jack to DB-9 cable, the cable adapter for the eight-port board, and the DB-9 connectors to the 16-port breakout box.



DB-9 Connector Pin Locations

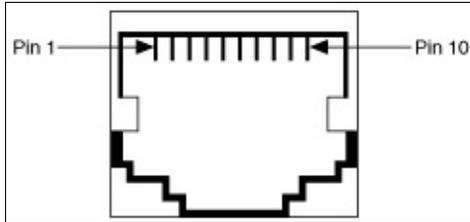
DB-9 Pin Descriptions

DB-9 Pin	232 Signal DTE	485 Signal
1	DCD*	GND
2	RXD	CTS+ (HSI+)
3	TXD	RTS+ (HSO+)
4	DTR*	RXD+
5	GND	RXD-
6	DSR*	CTS- (HSI-)
7	RTS	RTS- (HSO-)
8	CTS	TXD+
9	RI*	TXD-

* These signals are "No Connect" on the isolated 232 boards or ports 9–16 of the PCI-232/16 legacy board.

Modular Jack Connector

The following figure and table give the pin locations and descriptions of the 10-position modular jack.



10-Position Modular Jack Pin Locations

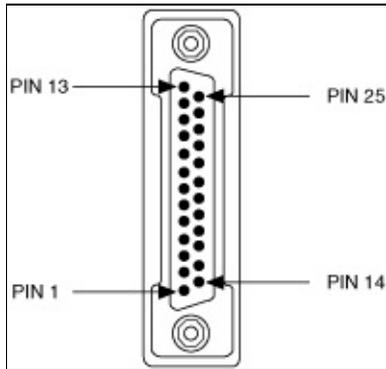
10-Position Modular Jack Pin Descriptions

10-Position Modular Jack Pin	232 Signal	485 Signal
10	DCD*	GND
9	RXD	CTS+ (HSI+)
8	TXD	RTS+ (HSO+)
7	DTR*	RXD+
6	GND	RXD-
5	DSR*	CTS- (HSI-)
4	RTS	RTS- (HSO-)
3	CTS	TXD+
2	RI*	TXD-
1	No Connect	No Connect

* These signals are "No Connect" on the isolated 232 ports.

DB-25 Connector

The following figure and table give the pin locations and descriptions of the DB-25 connector, which is on the optional 10-position modular jack to DB-25 cable.



DB-25 Connector Pin Locations

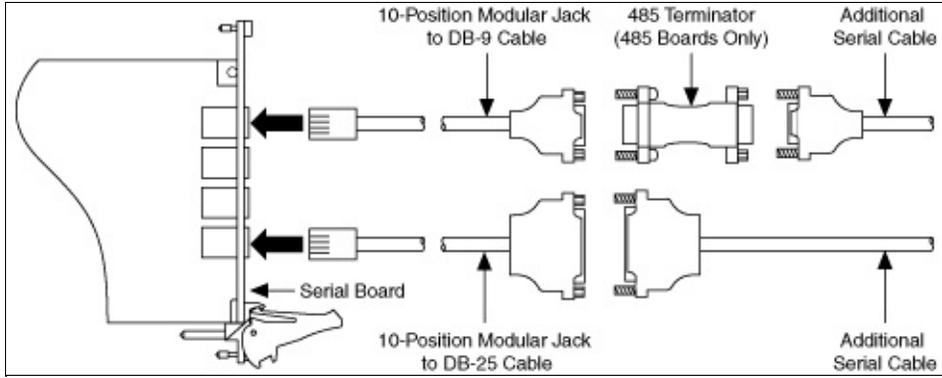
DB-25 Pin Descriptions

DB-25 Pin	232 Signal	485 Signal
2	TXD	RTS+ (HSO+)
3	RXD	CTS+ (HSI+)
4	RTS	RTS- (HSO-)
5	CTS	TXD+
6	DSR*	CTS- (HSI-)
7	GND	RXD-
8	DCD*	GND
20	DTR*	RXD+
22	RI*	TXD-

Pins not listed in this table are No Connect.

* These signals are "No Connect" on the isolated 232 ports.

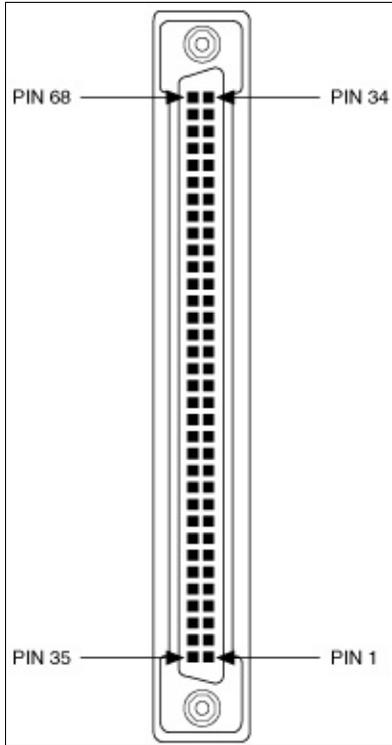
The following figure shows how to connect the cables when you install a four-port PXI serial board.



Connecting the Cables to a Four-Port PXI Serial Board

68-Pin Connector

The following figure and table give the pin locations and descriptions of the 68-pin connector.



68-Pin Connector Pin Locations

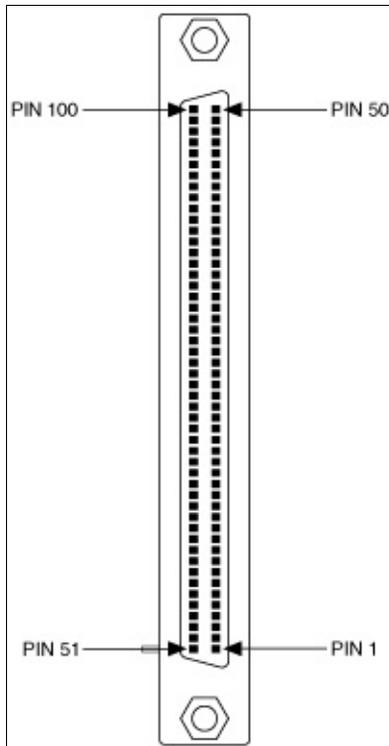
68-Pin Connector Pin Descriptions

68-Pin Connector Port											
1	2	3	4	5	6	7	8	485 Signal	485 D-Sub 9 Connector	232 Signal	232 D-Sub 9 Connector
66	57	49	40	32	23	15	6	RXD-	5	DCD	1
68	59	51	42	34	25	17	8	CTS+	2	RXD	2
65	56	48	39	31	22	14	5	RTS+	3	TXD	3
64	55	47	38	30	21	13	4	RXD+	4	DTR	4
60	60	43	43	26	26	9	9	GND	1	GND	5
63	54	46	37	29	20	12	3	CTS-	6	DSR	6
62	53	45	36	28	19	11	2	RTS-	7	RTS	7
61	52	44	35	27	18	10	1	TXD+	8	CTS	8

67	58	50	41	33	24	16	7	TXD-	9			RI	9
----	----	----	----	----	----	----	---	------	---	--	--	----	---

100-Pin Connector

The following figure and table give the pin locations and descriptions of the 100-pin connector.



100-Pin Connector Pin Locations

100-Pin Connector Pin Descriptions

100-Pin Connector Port																232 Signal	232 D 9 Conn
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
2	46	6	42	10	38	14	34	—	—	—	—	—	—	—	—	DCD*	1
3	47	7	43	11	39	15	35	18	32	20	30	22	28	24	26	RXD	2
4	48	8	44	12	40	16	36	19	33	21	31	23	29	25	27	TXD	3
5	49	9	45	13	41	17	37	—	—	—	—	—	—	—	—	DTR*	4
1	1	1	1	50	50	50	50	51	51	51	51	100	100	100	100	GND	5
52	96	56	92	60	88	64	84	—	—	—	—	—	—	—	—	DSR*	6
53	97	57	93	61	89	65	85	68	82	70	80	72	78	74	76	RTS	7
54	98	58	94	62	90	66	86	69	83	71	81	73	79	75	77	CTS	8

55	99	59	95	63	91	67	87	—	—	—	—	—	—	—	—	—	RI*	9
----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	-----	---

* These signals are not supported on ports 9–16 of the PCI-232/16 and PXI-8420/16 serial boards.

Serial Port Information

The following topics describe the RS-232, RS-422, and RS-485 standards and explain some of the issues involved with these types of serial communication.

[RS-232](#)

[RS-422](#)

[RS-485](#)

[Serial Communication Issues](#)

[Duplex Architectures](#)

[Full Duplex](#)

[Half Duplex](#)

[RS-485 Termination](#)

[Bias Resistors](#)

[RS-232 DTE versus DCE](#)

The following table lists the features of the RS-232, RS-422, and RS-485 standards.

RS-232, RS-422, and RS-485 Features

Feature	RS-232	RS-422	RS-485
Type of transmission lines	Single ended	Differential	Differential
Maximum number of drivers	1	1	32
Maximum number of receivers	1	10	32
Maximum cable length	50 ft	4,000 ft	4,000 ft
Maximum data rate	20 kbytes/s	10 Mbytes/s	10 Mbytes/s
Maximum CMV	± 25 V	± 7 V	+12 to -7 V
Driver output	5 to 25 V	2 to 6 V	1.5 to 6 V
Driver load	<3 k Ω	100 Ω	60 Ω

RS-232

As specified in the ANSI/EIA-232-D Standard, *Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange*, RS-232 standardizes serial communication between computers and between computer terminals and modems. Many applications use the RS-232 standard to interface peripherals to personal computers. RS-232 uses transmission lines in which the state of each signal is represented by referencing the voltage level of a single line to ground. RS-232 was designed for serial communication up to distances of 50 ft and with data rates up to 20 kbytes/s. However, because of improvements in line drivers and cabling, it is possible to increase the actual performance of the bus past the limitations on speed and distance recommended in the specification.

RS-422

As specified in the EIA/RS-422-A Standard, *Electrical Characteristics of Balanced Voltage Digital Interface Circuits*, RS-422 defines a serial interface much like RS-232. However, RS-422 uses balanced (or differential) transmission lines. Balanced transmission lines use two transmission lines for each signal. The state of each signal is represented, not by a voltage level on one line referenced to ground as in RS-232, but rather by the relative voltage of the two lines to each other. For example, the TX signal is carried on two wires, wire A and wire B. A logical 1 is represented by the voltage on line A being greater than the voltage on line B. A logical 0 is represented by the voltage on line A being less than the voltage on line B. Differential voltage transmission creates a signal that is more immune to noise as well as voltage loss due to transmission line effects. Thus, you can use RS-422 for longer distances (up to 4,000 ft) and greater transmission speeds (up to 10 Mbytes/s) than RS-232.

RS-485

As specified in the EIA-485 Standard, *Standard for Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems*, RS-485 expands on the RS-422 standard by increasing the number of devices you can use from 10 to 32 and by working with half-duplex bus architectures. Unlike the RS-422 standard, RS-485 addresses the issue of using multiple transmitters on the same line. RS-485 defines the electrical characteristics necessary to ensure adequate signal voltages under maximum load, as well as short-circuit protection. RS-485 can also withstand multiple drivers driving conflicting signals at the same time.

Serial Communication Issues

The following topics explain some serial communication issues, including duplex architectures, termination methods, bias resistors, and types of connecting equipment.

[Duplex Architectures](#)

[Full Duplex](#)

[Half Duplex](#)

[RS-485 Termination](#)

[Bias Resistors](#)

[RS-232 DTE versus DCE](#)

Duplex Architectures

Duplex refers to the means of bandwidth usage in a serial system. The two common means of bi-directional serial communication are full duplex and half duplex. Half-duplex communication involves a transmitter and a receiver connected to each end of the same wire or pair of wires.

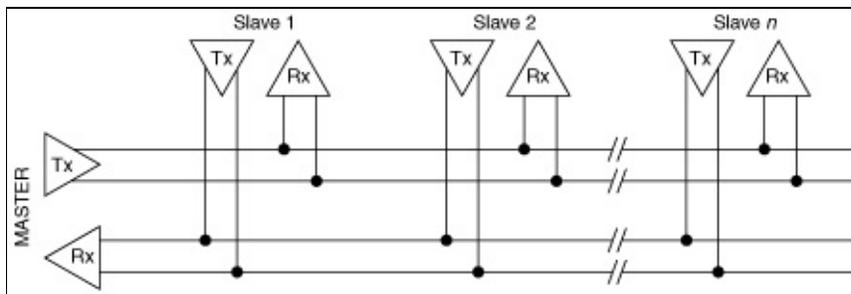
Because the same transmission line both sends and receives data, devices cannot send data in both directions at the same time. First, one device transmits over the wire(s) to the receiver of the second device. When the first device finishes transmitting, both devices switch the connections from their transmitter to their receiver, or vice versa. The device that was receiving data can then transmit over the line.

In full-duplex communication, the devices use a separate wire (or pair of wires) for simultaneous transmission in each direction. Thus, the devices do not switch between transmitting and receiving.

In a differential serial bus (such as RS-422 or RS-485), a half-duplex system transmits and receives over the same twisted pair of wires. Thus, half-duplex communication is often referred to as *two-wire* communication. Likewise, full-duplex communication is often referred to as *four-wire* communication, because the full-duplex system uses a separate pair of wires for communication in each direction.

Full Duplex

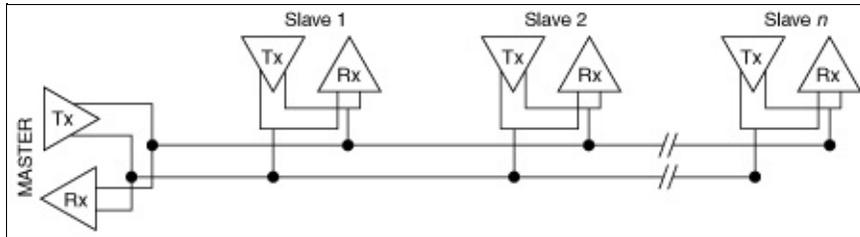
A typical full-duplex multidrop bus architecture involves a master-slave protocol. Only one device, the master, can control access to the bus. All other devices are slaves. Slave devices must wait for the master to give them access to the bus. In a typical full-duplex system, one transmission line connects the bus master transmitter to all of the slave receivers. A second transmission line connects all of the slave transmitters to the bus master receiver. Because each transmission line has two separate wires, a full-duplex system is often referred to as a four-wire system. The master uses four-wire mode, and the slaves use two-wire mode. The following figure shows a typical full-duplex system.



Typical Full-Duplex System

Half Duplex

A typical half-duplex multidrop bus architecture also involves a master-slave protocol. However, in a half-duplex system, all transmitters and receivers are connected to the same transmission line. A half-duplex system is often referred to as a two-wire system. The following figure shows a typical half-duplex system.

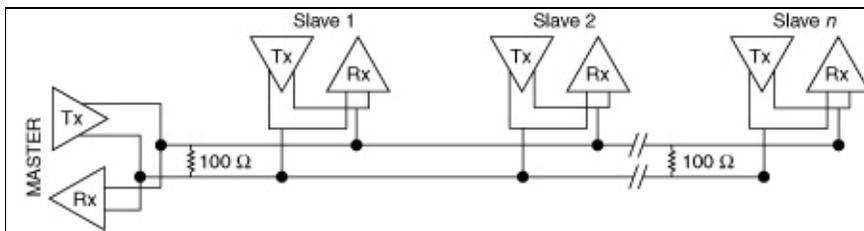


Typical Half-Duplex System

RS-485 Termination

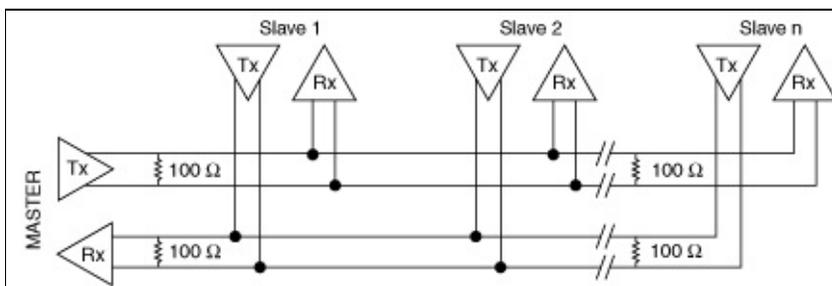
Because each differential pair of wires is a transmission line, you must properly terminate the line to prevent reflections. A common method of terminating a two-wire multidrop RS-485 network is to install terminating resistors at each end of the multidrop network. If you daisy-chain multiple instruments together, you need a terminating resistor at only the first and last instruments. The terminating resistor should match the characteristic impedance of the transmission line (typically 100 to 120 Ω). You can order an optional DB-9 RS-485 termination connector that contains embedded terminating resistors for easy termination from National Instruments. For ordering information, contact National Instruments.

The following figure shows a multidrop network using terminating resistors.



Multidrop Network Using Terminating Resistors

The following figure shows a full-duplex network using terminating resistors.



Full-Duplex Network Using Terminating Resistors

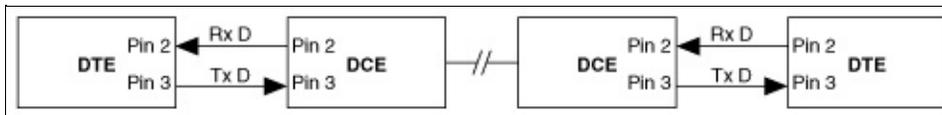
Bias Resistors

An RS-485 transmission line enters an indeterminate state if no nodes are transmitting on it. This indeterminate state can cause the receivers to receive invalid data bits from noise picked up on the cable. To prevent a line from receiving these data bits, the transmission line can be forced into a known state by installing two bias resistors at one node on the transmission line. Doing so creates a voltage divider that forces the voltage between the differential pair to be greater than 200 mV, the threshold voltage for the receiver. For more information about RS-485 bias resistors, refer to [Using Your Serial Hardware](#).

RS-232 DTE versus DCE

In the RS-232 specification, DTE (Data Terminal Equipment) and DCE (Data Communications Equipment) refer to the types of equipment on either end of a serial connection. In general, DTE and DCE refer to computer equipment and modems, respectively. Because the RS-232 specification mainly involves connecting a DTE directly to a DCE and vice versa, the pinouts are defined so that cabling is simple. That is, a cable connected a computer to a modem by wiring pin 1 to pin 1, pin 2 to pin 2, and so on. This method is known as *straight-through* cabling.

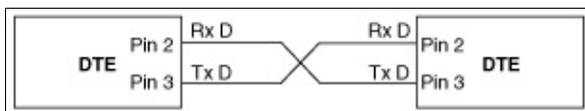
The following figure shows straight-through cabling in a DTE-to-DCE interface.



Straight-Through Cabling in a DTE-to-DCE Interface

Straight-through cabling is still the standard method to connect a modem to your PC. However, because many applications use serial communication to connect two or more DTEs without modems, the cabling becomes more complicated. If two DTEs are wired together using a straight-through cable, one transmitter is connected to the other transmitter, and one receiver is connected to the other receiver. In this setup, no transmissions can occur. Thus, these applications must use a cabling scheme that connects the transmitter on one device to the receiver on the other device and vice versa. This method is known as *null-modem* cabling, because it replaces the two modems that traditional RS-232 applications would require between the two DTEs. To communicate from one DTE serial port to another, use a null-modem cable.

The following figure shows null-modem cabling in a DTE-to-DTE interface.



Null-Modem Cabling in a DTE-to-DTE Interface

PCI, PXI, USB (one port only), ENET, ExpressCard, and PCMCIA RS-

232 ports are DTE serial ports.

USB-232 (two and four port only) can be configured to DTE or DCE.
Refer to [USB-232 Transceiver Control](#) for more information.

Uninstalling the Hardware and Software

The following topics describe how to uninstall your serial hardware and the NI-Serial software.

[Delete a Serial Interface](#)

[Uninstall Software from Your LabVIEW RT Target](#)

[Uninstall the Windows Software](#)

Delete a Serial Interface

Before you physically remove a serial interface from your Windows system, follow these steps to remove the hardware information:

1. [Launch MAX](#).
2. Expand the **Devices and Interfaces** directory by clicking the + next to the folder.
3. Right-click on your serial interface and select **Delete** from the drop-down menu that appears.
4. When prompted, click the **Yes** button to confirm the removal of your interface.
5. MAX automatically updates the list of installed serial interfaces. You can also select **View»Refresh** to update the list.

On LabVIEW RT, it is not necessary to programmatically delete hardware from the system. Simply power down your RT controller and remove the hardware. When you power on your RT controller again, the ports are automatically removed from the system. You may need to refresh MAX to see the changes.

Uninstall Software from Your LabVIEW RT Target

To uninstall the NI-Serial software from your LabVIEW RT target, complete the following steps:

1. [Launch MAX](#).
2. Expand **Remote Systems** by clicking the + next to it.
3. Find your remote system in the list and expand it.
4. Right-click **Software** and select **Add/Remove Software**.
5. In the **Add/Remove Software** window, click **NI-Serial** and select **Uninstall the Feature**.
6. Click **Next** to start the software download. Click **Finish** when the download is complete.

Uninstall the Windows Software

Before you uninstall the NI-Serial software, you must remove the hardware information from your system, as described in [Delete a Serial Interface](#).

To uninstall the NI-Serial software, complete the following steps:

1. Select **Start»Control Panel**.
2. On Windows 2000/XP/Server 2003, double-click the **Add/Remove Programs** icon. On Windows Vista, double-click the **Programs and Features** icon.
3. Select **National Instruments Software** in the list and click the **Change/Remove** button on Windows 2000/XP/Server 2003 or the **Uninstall/Change** button on Windows Vista.
4. Select the software components you want to remove and click the **Remove** button. The uninstallation program removes all folders, utilities, and registry entries associated with the NI-Serial software.
5. Shut down Windows, power off your computer, and physically remove the serial interfaces from your system.

The uninstallation program removes only items that the setup wizard installed. If you added anything to a directory that the setup wizard created, the uninstallation program does not delete that directory. You must remove any remaining components yourself.

To reinstall the hardware and software, refer to [Install the Hardware](#) and [NI-Serial Software Installation](#).

Troubleshooting and Frequently Asked Questions

The following topics describe how to troubleshoot problems and answer some common questions.

[Verify Installation and Troubleshoot NI-Serial Problems](#)

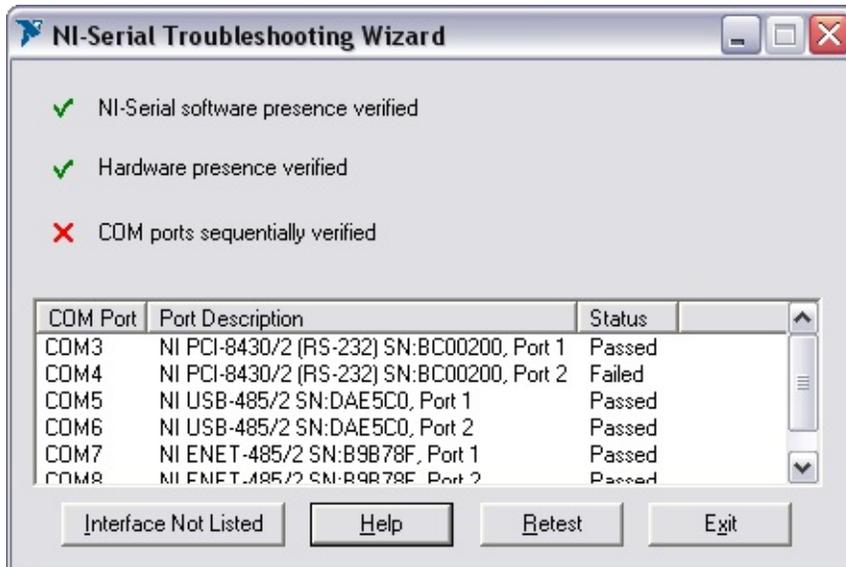
[Frequently Asked Questions](#)

Verify Installation and Troubleshoot NI-Serial Problems

The NI-Serial Troubleshooting Wizard supports ports only on your local Windows system. It does not test your LabVIEW Real-Time serial ports.

To verify the hardware and software installation and troubleshoot NI-Serial problems, follow these steps to run the Troubleshooting Wizard:

1. [Launch MAX](#).
2. Select **Help»Troubleshooting»NI-Serial Troubleshooting Wizard**. The Troubleshooting Wizard tests your serial interface and displays the results as shown below.



NI-Serial Troubleshooting Wizard

The Troubleshooting Wizard verifies that your serial driver is installed properly, that the configuration of your hardware does not conflict with anything else in your system, and that the serial driver can communicate with your hardware correctly.

To view online help for the Troubleshooting Wizard, click the **Help** button.

Frequently Asked Questions

[How do I force Windows to detect my PCI, PXI, or PCMCIA hardware?](#)

[How do I force Windows to detect my USB hardware?](#)

[How do I force Windows to detect my ExpressCard hardware?](#)

[How can I determine which type of serial hardware I have installed?](#)

[Why does my PCI/PXI serial card display the wrong serial number?](#)

[How can I determine which version of the NI-Serial software I have installed?](#)

[How do I troubleshoot problems?](#)

[How can I determine whether my hardware and software are installed properly?](#)

[How can I repair my NI-Serial software?](#)

[How can I determine which port is associated with COMx?](#)

[How can I change the COM numbers of my National Instruments serial ports?](#)

[Why can't I make changes to my settings in MAX on Windows Vista?](#)

[Why does my hardware keep reappearing after I delete it on Windows Vista?](#)

[How many serial ports can I have on my machine?](#)

[How many interrupts are required for my serial interface?](#)

[How do I resolve serial USB power conflicts?](#)

[What is the difference between USB 2.0 and USB 1.1 and can I use the USB serial hardware in USB 2.0 networks?](#)

[Why isn't the Auto232 transceiver mode on my USB-232 working?](#)

[Why am I seeing poor performance on Windows XP when I use multiple USB serial interfaces?](#)

[Why did my USB serial ports enumerate backward?](#)

[How do I access my serial ENET interface using the hostname?](#)

[How do I communicate with serial ENET interfaces located on other subnets?](#)

Why does the NI Ethernet Device Configuration utility list my serial ENET interface as *BUSY* when I know my network uses DHCP?

Why doesn't my serial ENET interface show up in the NI Ethernet Device Configuration utility?

Why did my application work with a plug-in interface but times out with a serial ENET interface?

What is the relationship between bits per second, throughput, and bus utilization?

What happens to data sent to a closed serial port?

Why do loopback tests of large data transfers sometimes have problems?

Does the serial hardware have built-in support for protocols such as SDLC and ModBus?

How do I use 9 data bit framing?

Can I use NI serial hardware with a non-Windows platform?

Can I use USB serial hardware, Ethernet serial hardware, and plug-in serial hardware in the same computer at the same time?

How do I force Windows to detect my PCI, PXI, or PCMCIA hardware?

If your serial hardware does not show up in the device manager, Windows did not detect your hardware. To solve this problem, complete the following steps:

1. Select **Start»Control Panel** and double-click the **System** icon.
2. Select the **Hardware** tab and click the **Device Manager** button.
3. Double-click the **Multifunction Adapters** icon.
4. From the **Multifunction adapters** list, right-click the National Instruments interface that was not detected, as shown in the following figure. (This example shows a PCI interface.)



Selecting an Interface to Uninstall

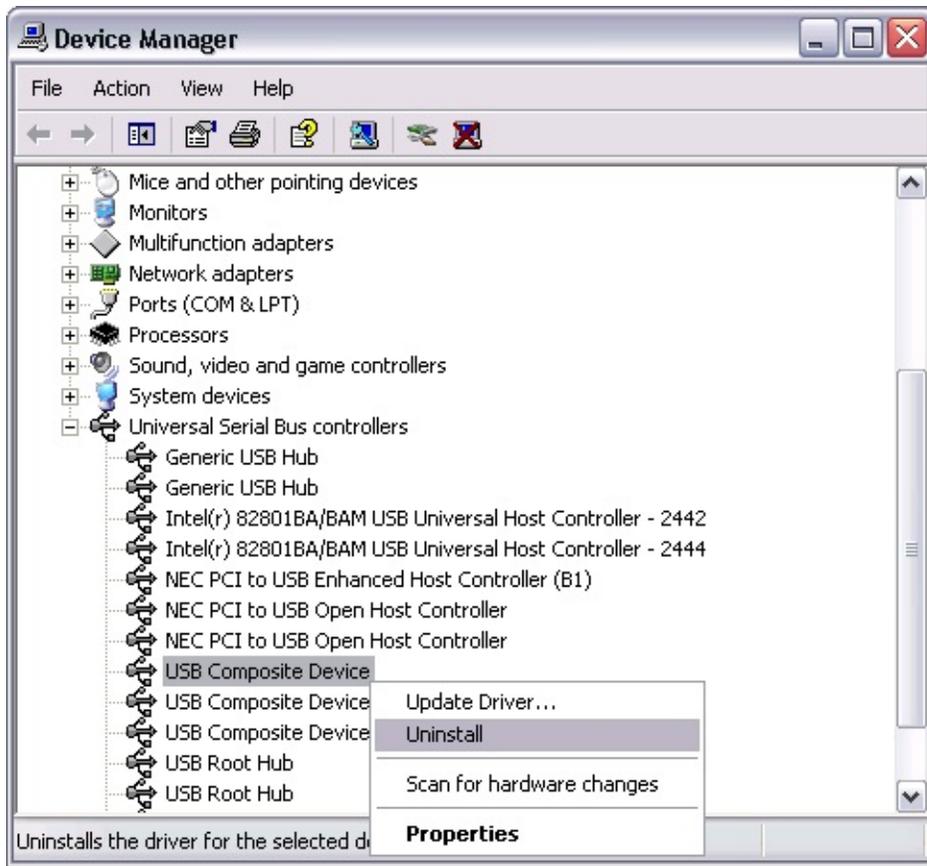
5. Choose the **Uninstall** option.
6. In the **Confirm Device Removal** dialog box, click the **OK** button to remove the hardware information.

7. Click the **Scan for Hardware Changes** icon at the top of the Device Manager window.
8. Windows should automatically detect your hardware.

How do I force Windows to detect my USB hardware?

If your USB hardware does not show up in the device manager, Windows did not detect your hardware. To solve this problem, complete the following steps:

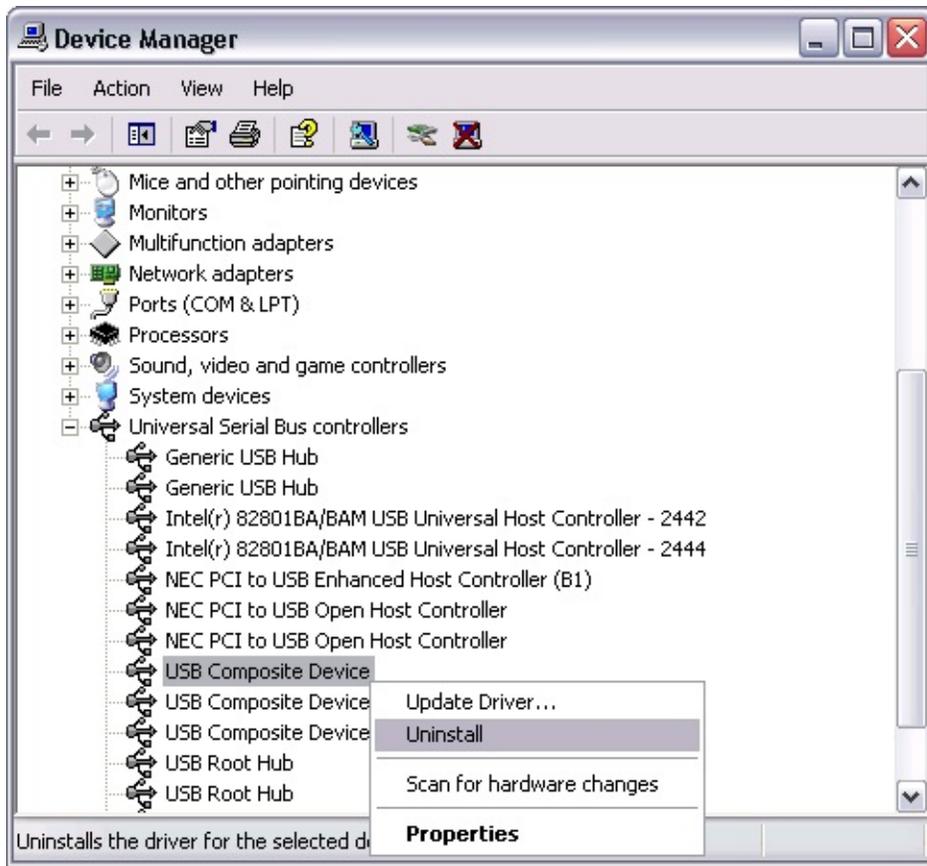
1. Select **Start»Control Panel** and double-click the **System** icon.
2. Select the **Hardware** tab and click the **Device Manager** button. The Device Manager window should appear as shown below.
3. Double-click the **Ports (COM & LPT)** icon. Verify that all USB serial communication ports are displayed without exclamation marks (!). If any USB serial port has an exclamation mark, right-click the port and select **Uninstall**.
4. Double-click the **Universal Serial Bus Controllers** icon. Verify that no USB Composite Devices have exclamation marks (!). If any USB Composite Device has an exclamation mark, right-click the port and select **Uninstall**.
5. Click the **Scan for Hardware Changes** icon at the top of the Device Manager window.
6. Windows should automatically detect your hardware.
7. If any of your serial ports are still not visible, disconnect and reconnect the USB serial hardware to a different USB port and return to step 2.



How do I force Windows to detect my ExpressCard hardware?

If your serial hardware does not show up in the device manager, Windows did not detect your hardware. To solve this problem, complete the following steps:

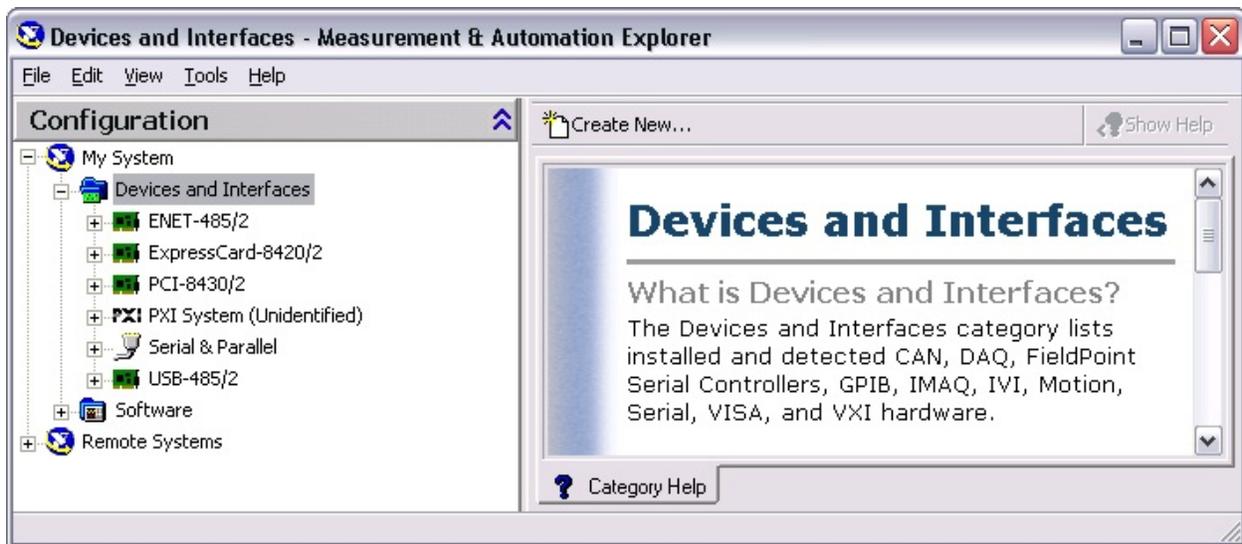
1. Select **Start»Control Panel** and double-click the **System** icon.
2. Select the **Hardware** tab and click the **Device Manager** button. The Device Manager window should appear as shown below.
3. Double-click the **Ports (COM & LPT)** icon. Verify that all ExpressCard serial communication ports are displayed without exclamation marks (!). If any ExpressCard serial port has an exclamation mark, right-click the port and select **Uninstall**.
4. Double-click the **Universal Serial Bus Controllers** icon. Verify that no USB Composite Devices have exclamation marks (!). If any USB Composite Device has an exclamation mark, right-click the port and select **Uninstall**.
5. Click the **Scan for Hardware Changes** icon at the top of the Device Manager window.
6. Windows should automatically detect your hardware.



How can I determine which type of serial hardware I have installed?

To view the serial hardware installed on your Windows system, complete the following steps:

1. [Launch MAX](#).
2. Under **My System**, expand the **Devices and Interfaces** directory by clicking on the + next to the folder. MAX lists your installed serial hardware under **Devices and Interfaces**, as shown in the following figure.



To view the serial hardware installed on your LabVIEW Real-Time controller, complete the following steps:

1. [Launch MAX](#).
2. Expand **Remote Systems** by clicking the + next to the folder.
3. Find your RT controller in the list and expand it.
4. Expand the **Devices and Interfaces** directory.
5. Expand the **Serial** entry. MAX lists your installed serial hardware under **Serial**, as shown in the following figure.

Serial - Measurement & Automation Explorer

File Edit View Tools Help

Configuration

Create New Port... Save All Ports Show Help

My System

- Remote Systems
 - PxiController
 - Devices and Interfaces
 - PXI PXI System (NI PXI-8175)
 - Serial
 - COM1
 - COM2
 - PXI-8431/8 (Chassis 1, Slot 2)
 - PXI-8431/4 (Chassis 1, Slot 3)
 - PXI-8430/2 (Chassis 1, Slot 4)
 - Software

Resource Name	Interface Name	Setting
COM1	ASRL1 (COM1 - Built-In)	9600, 8N1
COM2	ASRL2 (COM2 - Built-In)	9600, 8N1
COM3	ASRL3 (COM3 - PXI-8431/8)	9600, 8N1
COM4	ASRL4 (COM4 - PXI-8431/8)	9600, 8N1
COM5	ASRL5 (COM5 - PXI-8431/8)	9600, 8N1
COM6	ASRL6 (COM6 - PXI-8431/8)	9600, 8N1
COM7	ASRL7 (COM7 - PXI-8431/8)	9600, 8N1
COM8	ASRL8 (COM8 - PXI-8431/8)	9600, 8N1
COM9	ASRL9 (COM9 - PXI-8431/8)	9600, 8N1
COM10	ASRL10 (COM10 - PXI-8431/8)	9600, 8N1
COM11	ASRL11 (COM11 - PXI-8431/4)	9600, 8N1
COM12	ASRL12 (COM12 - PXI-8431/4)	9600, 8N1
COM13	ASRL13 (COM13 - PXI-8431/4)	9600, 8N1
COM14	ASRL14 (COM14 - PXI-8431/4)	9600, 8N1
COM15	ASRL15 (COM15 - PXI-8430/2)	9600, 8N1
COM16	ASRL16 (COM16 - PXI-8430/2)	9600, 8N1

Attributes

Why does my PCI/PXI serial card display the wrong serial number?

If you swap one PCI/PXI serial card for another of the exact same type, the serial number shown in the device manager and MAX may not change. This happens because Windows cannot distinguish the old hardware from the new. To fix this problem, delete the interface from the system and either scan for new hardware in the device manager or restart your computer. This forces Windows to reinstall your hardware, and the correct serial number should be displayed.

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

You can use MAX to [view your NI-Serial software version](#).

How do I troubleshoot problems?

Use the [NI-Serial Troubleshooting Wizard](#). This wizard verifies that your serial hardware and the NI-Serial software are installed correctly and working properly.

How can I determine whether my hardware and software are installed properly?

Use the [NI-Serial Troubleshooting Wizard](#). This wizard verifies that your serial hardware and the NI-Serial software are installed correctly and working properly.

How can I repair my NI-Serial software?

To repair the NI-Serial software, complete the following steps:

1. Select **Start»Control Panel**.
2. On Windows 2000/XP/Server 2003, double-click the **Add/Remove Programs** icon. On Windows Vista, double-click the **Programs and Features** icon.
3. Select **National Instruments Software** in the list and click the **Change/Remove** button on Windows 2000/XP/Server 2003 or the **Uninstall/Change** button on Windows Vista.
4. Select the software components you want to repair and click the **Repair** button. The repair checks all utilities and registry entries associated with the NI-Serial software and repairs or replaces files as necessary.
5. Restart the computer to complete the repair process.
6. Use the [NI-Serial Troubleshooting Wizard](#) to verify that your software has been repaired and is working properly.

How can I determine which port is associated with COMx?

To determine which physical port is associated with COMx, complete the following steps:

1. [Launch MAX](#).
2. Expand the **Devices and Interfaces** directory by clicking on the + next to the folder. MAX lists your installed serial hardware under **Devices and Interfaces**.
3. Click your serial interface in the device tree. The right window displays information on what COM number is associated with each physical port on your serial interface.

How can I change the COM numbers of my National Instruments serial ports?

Select a new COM number from the COM Port Number control of the **Advanced** MAX tab and click **Save**.

Why can't I make changes to my settings in MAX on Windows Vista?

To make changes in MAX on Windows Vista, follow these steps to launch MAX with administrative privileges:

1. If MAX is open, close it.
2. Find the MAX icon on the desktop or by going to **Start»Programs»National Instruments»Measurement & Automation.**
3. Right-click on the MAX icon and select **Run as administrator.** You must enter an administrative password if you are not an administrator.

You should now be able to modify and save settings in MAX.

Why does my hardware keep reappearing after I delete it on Windows Vista?

Microsoft has changed the hardware removal behavior on Windows Vista. When you delete a piece of hardware, Windows Vista automatically rescans for new hardware. If you delete more than one piece of hardware at the same time, the first piece is added back to the system when you delete the second piece. To prevent this behavior, you must remove each piece of hardware after you delete it from MAX or the device manager.

How many serial ports can I have on my machine?

You can have up to 256 serial ports.

How many interrupts are required for my serial interface?

If you have a shared IRQ board or an isolated board, you need one interrupt per board.

Serial ENET and serial USB interfaces do not require interrupts.

How do I resolve serial USB power conflicts?

The USB-232, USB-232/2, USB-232/4, USB-485, and USB-485/2 are *high- power* bus-powered devices that require more than 100 mA during normal operation. Thus, they must be plugged into an externally powered hub that can deliver up to 500 mA per USB port. If the USB network does not have enough power to support the USB serial hardware, and a system dialog box informs you that there is not enough power for the device, reconnect the hardware to an externally powered hub.

The USB-485/4 is a self-powered device that requires external power. If the device is plugged into a USB port, and the device does not have power, it will not enumerate and the ports will not show in the Device Manager. To ensure proper port enumeration, always make sure the USB-485/4 has external power before connecting it to the USB network.

What is the difference between USB 2.0 and USB 1.1, and can I use the USB serial hardware in USB 2.0 networks?

The USB 2.0 specification replaces the USB 1.1 specification and adds a high-speed mode of operation (480 Mbit/s) to the previously existing full-speed (12 Mbit/s) and low-speed (1.5 Mbit/s) modes of operation.

Because USB 2.0 expands on the features of USB 1.1 while preserving backward device compatibility, you can use most full-speed and low-speed devices with either USB 2.0 high-speed networks or USB 1.1 full-speed networks.

USB-232/485 devices are USB 2.0 full-speed compliant, so you can use them in either high-speed (USB 2.0 hubs) or full-speed (USB 1.1 hubs) networks.

Why isn't the Auto232 transceiver mode on my USB-232 working?

The Auto232 mode of operation allows you to connect to either a RS-232 DTE or DCE device using either a null-modem or straight-through cable. The USB-232 port will automatically switch approximately every 250 ms between DTE and DCE operating modes until at least one RS-232 receiver line (RX, CTS, DSR, RI, or DCD) has a valid voltage. Auto232 is available on the USB-232/2 and USB-232/4, and is not available on the USB-232 (single port).

Not all devices and applications can take advantage of this feature:

- Some devices can generate valid voltages on receive lines for both modes, even though the device itself is only DTE or DCE. Thus, these devices may not operate correctly when using Auto232 mode.
- Some applications immediately start communicating after a COM port is open, thus not allowing enough time for the proper operating mode to be selected.
- Some cabling schemes short RI, DCD, and DSR together. While this does not present a problem in DTE mode, these signals are driven when in DCE mode. Thus, by shorting the signals together, the outputs will be double (or triple) driven which may cause signal noise or other artifacts. Use a straight-through cable instead.

If problems with Auto232 can not be resolved, set the USB-232 to the correct mode (DTE or DCE) for your device and application. By default, the USB-232 ports are set to DTE mode.

Why am I seeing poor performance on Windows XP when I use multiple USB serial interfaces?

Windows XP system performance may be reduced when multiple USB-232 and USB-485 devices are installed. This can cause slow screen updating (mouse and window movements) and reduced task switching performance. This problem is addressed in Windows XP Service Pack 1. To obtain Windows XP Service Pack 1, use Windows Update, or contact Microsoft at www.microsoft.com.

Why did my USB serial ports enumerate backward?

If you use a dual-boot system (Windows XP and Windows 2000), the COM ports may enumerate out of order after switching operating systems. This is because the device firmware is downloaded when the device is first powered. Because the firmware images for operation under Windows 2000 and Windows XP are different, and the USB bus power may not be cycled during a reboot, it is possible to run the incorrect firmware for the current operating system. You can fix this by disconnecting and reconnecting the USB-232 or USB-485 from the USB port or hub (and disconnecting power on the USB-485/4) to force the appropriate firmware to be downloaded.

How do I access my serial ENET interface using the hostname?

If you want to use the same hostname that the device is configured with and you know its IP address, skip to the next step. Otherwise, run the NI Ethernet Device Configuration utility. (You must use a computer on the same subnet as the device.) Refer to [Viewing Network Settings](#) for more information about the NI Ethernet Device Configuration utility.

- Verify the device is present on the network by pinging the IP address of your device. *Ping* is a network utility that sends a packet to a device/host on the network, then waits for it to be echoed back, which indicates an active device/host. Find the directory where ping is installed and type in the following command:

```
ping IPaddress
```

where *IPaddress* is the IP address assigned to your serial device server. When your device responds to the ping requests, repeat this procedure, but this time using the hostname:

```
ping hostname
```

- If your device fails to respond when using the hostname, the DNS entry for the device is incorrect.
- If the network parameters were statically (manually) assigned, the hostname must be manually entered in your network's Domain Name Server (DNS) table before you can use the hostname to communicate with your device. Have your network administrator enter this information.
- You also can manually check that the device's hostname has been entered in the DNS table by using NSLookup. This network utility queries the DNS for entries in the DNS table. You can use NSLookup to determine the hostname associated with any particular IP address listed in the DNS table. Find the directory where NSLookup is installed and type in the following command:

```
nslookup IPaddress
```

where *IPaddress* is the IP address you want to look up. Alternately, to determine the IP address associated with a particular hostname, type in the following command instead:

`nslookup hostname`

If you still cannot communicate with your device, contact National Instruments.

How do I communicate with serial ENET interfaces located on other subnets?

- Verify that the device is present on the network by pinging the IP address of your device. Find the directory where ping is installed and type in the following command:

`ping IPaddress`

where *IPaddress* is the IP address assigned to your serial device server.

- Confirm that the gateway settings of the device are set properly. To do this, run the NI Ethernet Device Configuration utility on a computer on the same subnet as the device.
- Confirm that the gateway settings of the host machine are set properly.

To view your network settings, find the directory where ipconfig is installed and run it.

If you need more information, you can run ipconfig with the /all option by typing `ipconfig /all`. This shows all settings for your machine. Be sure to use the settings for the Ethernet adapter you are using to communicate with the serial device server.

Why does the NI Ethernet Device Configuration utility list my serial ENET interface as *BUSY* when I know my network uses DHCP?

For up to one minute, the serial device server attempts to retrieve network parameters from a DHCP server. During this time, *BUSY* appears in the utility display. If this status does not change, there are several reasons why DHCP may fail to respond. One of the most likely reasons is that your subnet has no more available addresses to assign using DHCP. Consult your network administrator to determine the specific reason for the problem.

Why doesn't my serial ENET interface show up in the NI Ethernet Device Configuration utility?

Allow at least 90 seconds for the serial device server to complete its power-on self tests before launching device configuration or clicking **Refresh**.

Communication between the NI Ethernet Device Configuration utility and the serial device server is not guaranteed. Though unlikely, it is possible that the communication gets lost in the network. Click **Refresh** more than once before concluding that the device is not discovered.

Verify that the serial device server shares the same subnet as the computer running the NI Ethernet Device Configuration utility. The NI Ethernet Device Configuration utility cannot communicate with devices through gateways on the network and will not discover devices on another subnet.

Why did my application work with a plug-in interface but times out with a serial ENET interface?

Because the serial device server is a network device, it is dependent on network conditions for timely communications. An unpredictable delay is associated with using the remote serial ports. Applications may require longer serial timeouts to compensate for this delay. However, different applications handle serial timeouts differently, and each must be adjusted individually to account for your network conditions.

What is the relationship between bits per second, throughput, and bus utilization?

The term *bits per second* (bps) refers to the bit rate of data transfer, with $1/\text{bps}$ being the length of time required to transmit one bit.

Throughput is the actual data transfer measured by *amount of data transmitted* *time required to transmit*

Bus utilization is the percentage of time that the serial bus is actively transmitting data. Because a wide range of factors such as flow control, processor latency, and device architecture can affect serial performance, bus utilization often is not 100 percent. In addition, there is overhead associated with serial communication in the form of start bits, stop bits, and parity bits. This overhead reduces the throughput by 20–30 percent. Thus, the actual data throughput can be considerably less than the bits per second setting.

What happens to data sent to a closed serial port?

Because serial communication ports are active only when the port is open (for example, by using `CreateFile`), data may be received and made available only when the port is open. Assume that any data received by a closed com port may be lost or corrupted.

Why do loopback tests of large data transfers sometimes have problems?

In a typical loopback test, an external wrap plug is used to loop back transmit signals to receive signals. In this configuration, data transmitted is read back on the same serial port to test port communication.

Loopback programs often perform this data transfer using a single write followed by a single read, and thus rely on driver software and serial hardware buffering to hold the data between the two function calls.

However, these buffers are of finite size, and thus any data transfer that exceeds the buffer size either times out, blocks (due to flow control), or drops data. This behavior is a result of the test design and does not represent a hardware or driver error. Buffer sizes vary between operating systems and hardware, but typically range from 4–8 KB in size.

Does the serial hardware have built-in support for protocols such as SDLC and ModBus?

Because the serial hardware uses UARTs for communication, protocols such as SDLC and ModBus are not supported in hardware. However, these protocols (or their derivatives) can be supported by software applications that handle the protocol translation.

How do I use 9 data bit framing?

While NI-Serial software and hardware directly supports only 5, 6, 7, and 8 data bits, it may be possible to use the parity function to create a ninth data bit, depending on your application. To create and control a ninth data bit, enable parity and set parity mark or parity space for a ninth bit of either a 1 or 0, respectively. However, you must do this every time the ninth bit changes state, and thus it may not work for high-throughput applications.

Can I use NI serial hardware with a non-Windows platform?

Refer to the following table for operating system support for NI serial hardware.

NI-Serial Hardware Operating System Support

Bus	Windows Vista 32	Windows Vista 64	Windows 2000/XP/Server 2003	Windows XP x64/Server 2003 x64	Linux x86	Solaris 2.x	L
PCI-843x	X✓	X✓	X✓	X✓			
PCI-232/485	X✓	X✓	X✓	X✓	X✓		
PXI-843x	X✓	X✓	X✓	X✓			
PXI-842x	X✓	X✓	X✓	X✓			
USB	X✓		X✓				
ENET	X✓	X	X✓	X	X	X	
ExpressCard	X✓		X✓				
PCMCIA	X✓		X✓	X✓	X✓		

X NI-VISA Support

✓ Native OS Support

Can I use USB serial hardware, Ethernet serial hardware, ExpressCard serial hardware, and plug-in serial hardware in the same computer at the same time?

As long as your operating system supports all your NI serial hardware, you can mix different types of serial interfaces in your computer. All NI serial hardware and software is designed and tested for interoperability.

Glossary

Prefixes Symbols A B C D E F G H I K L M
P R S T U V

Prefixes

Symbol	Prefix	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}

Symbols

°	degrees
Ω	ohms
%	percent

A

A amperes

ANSI American National Standards Institute

B

b bits

B bytes

baud bits per second

bps bits per second

C

C Celsius

COM Computer Output Microform; used in reference to a communication port

CTS clear to send

D

- DB-xx subminiature D connector, where xx is the number of pins
- DCD data carrier detect
- DCE data communications equipment or data circuit-terminating equipment
- DLL dynamic link library
- DMA direct memory access
- DSR data set ready
- DTE data terminal equipment
- $\overline{\text{DTR}}$ data terminal ready—the overscore denotes that the signal is active low
- duplex the means of bandwidth usage in a serial system

E

EIA Electronic Industries Association

EMI electromagnetic interference

ENET Ethernet

F

FCC Federal Communications Commission

FIFO First-In-First-Out

ft feet

G

G ground

H

HSI handshake input

HSO handshake output

Hz Hertz

I

I/O input/output

IEEE Institute of Electrical and Electronic Engineers

in. inches

IRQ interrupt request

ISA Industry Standard Architecture

K

Kbaud kilobits per second

L

LED light-emitting diode

M

m meters

MB megabyte

P

PC personal computer

PCI Peripheral Components Interconnect

R

RAM random-access memory

RI ring indicator

RTS request to send

RX receive

$\overline{\text{RXD}}$ receive data—the overscore denotes that the signal is active low

S

s seconds

SCSI Small Computer Systems Interface

T

TX transmit

$\overline{\text{TXD}}$ transmit data—the overscore denotes that the signal is active low

$\overline{\text{TXRDY}}$ transmit ready—the overscore denotes that the signal is active low

U

UART Universal Asynchronous Receiver/Transmitter

V

V volts

VDC volts direct current

VXI VME eXtensions for Instrumentation

Important Information

[Warranty](#)

[Copyright](#)

[Trademarks](#)

[Patents](#)

[Warning Regarding Use of NI Products](#)

[Compliance](#)

Warranty

The serial hardware is 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 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 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.

National Instruments respects the intellectual property of others, and we ask our users to do the same. NI software is protected by copyright and other intellectual property laws. Where NI software may be used to reproduce software or other materials belonging to others, you may use NI software only to reproduce materials that you may reproduce in accordance with the terms of any applicable license or other legal restriction.

Trademarks

National Instruments, NI, ni.com, and LabVIEW are trademarks of National Instruments Corporation. Refer to the *Terms of Use* section on ni.com/legal for more information about [National Instruments trademarks](#).

FireWire® is the registered trademark of Apple Computer, Inc.

Handle Graphics®, MATLAB®, Real-Time Workshop®, Simulink®, Stateflow®, and xPC TargetBox® are registered trademarks, and TargetBox™ and Target Language Compiler™ are trademarks of The MathWorks, Inc.

Tektronix® and Tek are registered trademarks of Tektronix, Inc.

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

Members of the National Instruments Alliance Partner Program are business entities independent from National Instruments and have no agency, partnership, or joint-venture relationship with National Instruments.

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 with FCC/Canada Radio Frequency Interference Regulations

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). All National Instruments (NI) products are FCC Class A products.

Depending on where it is operated, this Class A 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.

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

Consult the FCC Web site at 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 marking 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 NI 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 is required to correct the interference at their 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.

Compliance with EU Directives

Users in the European Union (EU) should refer to the Declaration of Conformity (DoC) for information* pertaining to the CE marking. Refer to the Declaration of Conformity (DoC) for this product for any additional regulatory compliance information. To obtain the DoC for this product, visit ni.com/hardref.nsf, search by model number or product line, and click the appropriate link in the Certification column.

* The CE marking Declaration of Conformity contains important supplementary information and instructions for the user or installer.

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 at ni.com/support include the following:
 - **Self-Help Resources**—For answers and solutions, visit the award-winning National Instruments Web site for software drivers and updates, a searchable [KnowledgeBase](#), [product manuals](#), step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on.
 - **Free Technical Support**—All registered users receive free Basic Service, which includes access to hundreds of Applications Engineers worldwide in the [NI Developer Exchange](#) at ni.com/exchange. National Instruments Applications Engineers make sure every question receives an answer.

For information about other [technical support options](#) in your area, visit ni.com/services or [contact](#) your local office at ni.com/contact.

- **Training and Certification**—Visit ni.com/training for self-paced training, eLearning virtual classrooms, interactive CDs, and Certification program information. 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, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit ni.com/alliance.
- **Declaration of Conformity (DoC)**—A DoC is our claim of compliance with the Council of the European Communities using the manufacturer's declaration of conformity. This system affords the user protection for electronic compatibility (EMC) and product safety. You can obtain the DoC for your product by visiting ni.com/certification.
- **Calibration Certificate**—If your product supports calibration, you can obtain the calibration certificate for your product at

ni.com/calibration.

If you searched ni.com and could not find the answers you need, contact your [local office](#) or NI corporate headquarters. 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.

Branch Offices

Office	Telephone Number
Australia	1800 300 800
Austria	43 662 457990-0
Belgium	32 (0) 2 757 0020
Brazil	55 11 3262 3599
Canada	800 433 3488
China	86 21 5050 9800
Czech Republic	420 224 235 774
Denmark	45 45 76 26 00
Finland	385 (0) 9 725 72511
France	33 (0) 1 57 66 24 24
Germany	49 89 7413130
India	91 80 41190000
Israel	972 0 3 6393737
Italy	39 02 413091
Japan	81 3 5472 2970
Korea	82 02 3451 3400
Lebanon	961 (0) 1 33 28 28
Malaysia	1800 887710
Mexico	01 800 010 0793
Netherlands	31 (0) 348 433 466
New Zealand	0800 553 322
Norway	47 (0) 66 90 76 60
Poland	48 22 3390150
Portugal	351 210 311 210
Russia	7 495 783 6851
Singapore	1800 226 5886
Slovenia	386 3 425 42 00

South Africa	27 0 11 805 8197
Spain	34 91 640 0085
Sweden	46 (0) 8 587 895 00
Switzerland	41 56 2005151
Taiwan	886 02 2377 2222
Thailand	662 278 6777
Turkey	90 212 279 3031
United Kingdom	44 (0) 1635 523545
United States (Corporate)	512 683 0100