

LabVIEW[™]

PDA Module User Manual

Worldwide Technical Support and Product Information

ni.com

National Instruments Corporate Headquarters

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

Worldwide Offices

Australia 1800 300 800, Austria 43 0 662 45 79 90 0, Belgium 32 0 2 757 00 20, Brazil 55 11 3262 3599,
Canada (Calgary) 403 274 9391, Canada (Ottawa) 613 233 5949, Canada (Québec) 450 510 3055,
Canada (Toronto) 905 785 0085, Canada (Vancouver) 514 685 7530, China 86 21 6555 7838,
Czech Republic 420 224 235 774, Denmark 45 45 76 26 00, Finland 385 0 9 725 725 11,
France 33 0 1 48 14 24 24, Germany 49 0 89 741 31 30, Greece 30 2 10 42 96 427, India 91 80 51190000,
Israel 972 0 3 6393737, Italy 39 02 413091, Japan 81 3 5472 2970, Korea 82 02 3451 3400,
Malaysia 603 9131 0918, Mexico 001 800 010 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 7 095 783 68 51, Singapore 65 6226 5886, Slovenia 386 3 425 4200, South Africa 27 0 11 805 8197,
Spain 34 91 640 0085, Sweden 46 0 8 587 895 00, Switzerland 41 56 200 51 51, Taiwan 886 2 2528 7227,
Thailand 662 992 7519, United Kingdom 44 0 1635 523545

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

© 2003–2004 National Instruments Corporation. All rights reserved.

Important Information

Warranty

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

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

National Instruments believes that the information in this 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 THEREOF PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

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

Trademarks

LabVIEW™, National Instruments™, NI™, and ni.com™ are trademarks of National Instruments Corporation.

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

Patents

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

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

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

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

Contents

About This Manual

Conventions	vii
Related Documentation.....	viii

Chapter 1

Introduction

PDA VIs and Applications	1-1
PDA Emulators and Simulators	1-1

Chapter 2

Building PDA Applications

Selecting the Execution Target	2-1
LabVIEW for Windows	2-2
Palm OS Device	2-2
Palm OS Emulator.....	2-3
Pocket PC 2003 Device	2-3
Pocket PC 2003 Emulator	2-3
Building PDA VIs into PDA Applications	2-3
Transferring Data to and from a PDA Device	2-4
(Palm OS) File I/O Using the LabVIEW Conduit.....	2-4
Transferring Files to the Host Computer	2-4
Transferring Files to the PDA Device.....	2-5
(Palm OS) File I/O Using Expansion Cards	2-5
(Palm OS) Using the DatalogMgr Utility.....	2-5
(Pocket PC 2003) File I/O Using ActiveSync	2-5
Serial I/O	2-6
Infrared	2-6
Wireless TCP.....	2-6
Bluetooth	2-6

Chapter 3

Debugging PDA Applications

Enabling Debugging for PDA Applications	3-1
Debugging PDA Application.....	3-1

Chapter 4

Building Wireless PDA Applications

TCP, UDP, and Bluetooth Support.....	4-1
Clients, Services, and Arbitrators	4-1
Types of Wireless Applications.....	4-2
Direct Connection Applications.....	4-2
Connecting over TCP or UDP	4-2
Connecting Using Bluetooth Wireless Technology	4-2
Calling by Reference	4-2
Distributed Connection Applications.....	4-3
Configuring Services for PDA Devices.....	4-4
Setting up Servers	4-4

Chapter 5

Programming Techniques

Palm OS and Pocket PC 2003 VI Templates	5-1
Functionality Differences in PDA VIs	5-2
PDA Operating System Considerations	5-2
Palm OS Considerations	5-3
Pocket PC 2003 Considerations	5-4
Front Panel Design	5-4
Window Appearance.....	5-4
Hierarchical Limitations	5-5
Controls and Indicators	5-5
Showing and Hiding Controls and Indicators.....	5-6
Unsupported Front Panel Objects.....	5-6
Block Diagram Programming.....	5-7
Using Events	5-7
Unsupported Block Diagram Objects	5-7
Application Design and Performance Considerations.....	5-8

Appendix A

Technical Support and Professional Services

About This Manual

This manual contains information about and describes how to use the LabVIEW PDA Module.

Refer to the *Getting Started with the LabVIEW PDA Module* user guide for installation instructions, getting started information, and a tutorial that guides you through the basics of building, running, and debugging PDA VIs.

This manual assumes you are familiar with Microsoft Windows and basic LabVIEW concepts. This manual also assumes you have basic knowledge of Palm OS and/or Pocket PC 2003.

Conventions

The following conventions appear in this manual:

»

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



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



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

bold

Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names, controls and buttons on the front panel, dialog boxes, sections of dialog boxes, menu names, and palette names.

italic

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

monospace

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

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

Platform Text in this font denotes a specific platform and indicates that the text following it applies only to that platform.

tap A tap on a PDA device is the equivalent to a mouse click on a desktop computer.

Related Documentation

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

- *Getting Started with the LabVIEW PDA Module*
- *LabVIEW User Manual*
- *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**
- Documentation for the PDA device, emulator, or Palm OS simulator you use

Introduction

The LabVIEW PDA Module is an add-on module for LabVIEW you can use to build applications that run on personal digital assistant (PDA) devices.

The PDA Module extends the capabilities of LabVIEW so you can develop and build executable PDA applications on a host computer for a Palm OS or Pocket PC 2003 device. The Palm OS or Pocket PC 2003 device downloads and runs the PDA application you build on the host computer.

PDA VIs and Applications

PDA VIs are host VIs you create to build into executable PDA applications. You can run these PDA applications on a PDA device, emulator, or Palm OS simulator. Each PDA VI has a corresponding host VI that includes a front panel and block diagram. You use the PDA Module to build PDA VIs into PDA applications on the host computer and download the PDA application to the PDA device, emulator, or Palm OS simulator.

PDA Emulators and Simulators

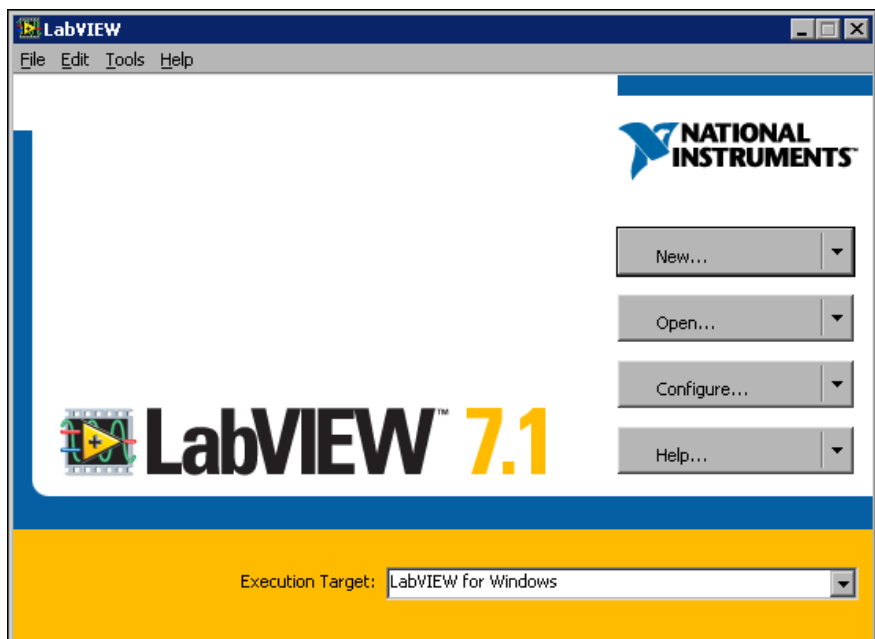
PDA emulators and simulators are tools you can use during development to quickly run and test PDA applications without having to download the PDA application to the actual PDA device. The PDA emulator or simulator you use depends on the PDA platform you install.

Building PDA Applications

This chapter describes how to create applications that run on PDA devices, including building the executable PDA application and downloading it to the PDA device.

Selecting the Execution Target

When you launch LabVIEW after installing the PDA Module, LabVIEW contains an extra option for you to select the execution target, as shown in the following figure.



The PDA Module includes three different types of execution targets—LabVIEW for Windows, PDA devices, and PDA emulators. If you are using Palm OS 5.0 and later, you also can use a Palm OS simulator as an execution target. Select **Palm OS Device** as an execution target if you are using a Palm OS simulator.

The type of device or emulator depends on the PDA Module platform you installed. You can install the Palm OS platform and the Pocket PC platform on the same host computer.

The execution target you select enables the LabVIEW functionality available for that execution target. For example, because Palm OS devices do not support some front panel and block diagram objects—such as DAQ I/O name controls and ActiveX functions—the **Palm OS Device** execution target includes only the front panel and block diagram objects the Palm OS devices support.

The execution target also determines where LabVIEW runs the VIs you build. When you select **LabVIEW for Windows** as the execution target and click the **Run** button, the VI runs on the host computer. When you select a PDA device or emulator as the execution target and click the **Run** button, LabVIEW executes actions based on the settings in the **Run Button Action for PDA Targets** page of the **Options** dialog box. Select **Tools»Options** to display the **Options** dialog box.

To switch execution targets, select an execution target from the **Execution Target** pull-down menu on the **LabVIEW** dialog box. You also can switch execution targets at any time during development by selecting **Operate»Switch Execution Target**.

LabVIEW for Windows

Select **LabVIEW for Windows** from the **Execution Target** pull-down menu to build VIs for use on a host computer. The **LabVIEW for Windows** execution target uses the default **Controls** and **Functions** palette view. **LabVIEW for Windows** is the default execution target.

Palm OS Device

Select **Palm OS Device** from the **Execution Target** pull-down menu to build VIs for a Palm OS device or (**Palm OS 5.0 and later**) Palm OS simulator. When you select a Palm OS device as the execution target, LabVIEW uses the PDA-4x **Controls** and **Functions** palette view, which contains the front panel and block diagram objects that Palm OS devices and simulators support.



Note You must install `Mathlib.prc` on the Palm OS device to run PDA applications built for a Palm OS device or simulator. Refer to the *Getting Started with the LabVIEW PDA Module* user guide for information about installing the `Mathlib.prc` file on the Palm OS device.

Palm OS Emulator

Select **Palm OS Emulator** from the **Execution Target** pull-down menu to build VIs for a Palm OS emulator. When you select **Palm OS Emulator** as the execution target, LabVIEW uses the PDA-4x **Controls** and **Functions** palette view, which contains the front panel and block diagram objects that Palm OS emulators support.



Note You must install `Mathlib.prc` on the Palm OS emulator to run PDA applications built for a Palm OS emulator. Refer to the *Getting Started with the LabVIEW PDA Module* user guide for information about installing the `Mathlib.prc` file on a Palm OS emulator.

Pocket PC 2003 Device

Select a Pocket PC 2003 device from the **Execution Target** pull-down menu to build VIs for a Pocket PC 2003 device. The **Execution Target** pull-down menu lists each Pocket PC 2003 device that is connected to the host computer and a generic **Pocket PC 2003 (Default Device)** option. When you select a Pocket PC 2003 device as the execution target, LabVIEW uses the PDA-2x **Controls** and **Functions** palette view, which contains the front panel and block diagram objects that Pocket PC 2003 devices support.



Note PDA applications built for Pocket PC 2003 devices do not run on Pocket PC 2003 emulators.

Pocket PC 2003 Emulator

Select **Pocket PC 2003 Emulator** from the **Execution Target** pull-down menu to build VIs for a Pocket PC 2003 emulator. When you select a Pocket PC 2003 emulator as the execution target, LabVIEW uses the PDA-2x **Controls** and **Functions** palette view, which contains the front panel and block diagram objects that Pocket PC 2003 emulators support.



Note PDA applications built for Pocket PC 2003 devices do not run on Pocket PC 2003 emulators.

Building PDA VIs into PDA Applications

After you develop the VI on the host computer, you build the PDA VI into a PDA application by clicking the **Run** button if you configure the **Run** button to build the PDA VI into a PDA application. You also can select **Tools»Build for PDA**. The **Build PDA Application** dialog box appears.



Note The **Tools»Build for PDA** menu item is disabled when you select **LabVIEW for Windows** or any other non-PDA target as the execution target.

The PDA Module builds the block diagram code into an executable PDA VI you can run on the PDA target. On Palm OS, PDA VIs are `.prc` files. On Pocket PC 2003, PDA VIs are `.exe` files. You then download the PDA application to the PDA device by selecting **Operate»Download Application** or by clicking the **Run** button. You also can use the software the PDA device provides to download a PDA application to a PDA device as you do any other PDA application.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help** and selecting the **PDA Module** book in the **Contents** tab, for more information about building PDA VIs into PDA applications.

Transferring Data to and from a PDA Device

You have several options when transferring data between a PDA device and the host computer. Some methods require a physical connection between the PDA and the computer, and others are wireless.

(Palm OS) File I/O Using the LabVIEW Conduit

The LabVIEW Conduit is a utility that allows you to use Palm HotSync to copy files that LabVIEW recognizes to and from a Palm OS device.

To customize the behavior of the LabVIEW Conduit, right-click the Palm HotSync icon in the Windows system tray and select **Custom** from the shortcut menu. Select **LabVIEW Conduit** and click the **Change** button. The HotSync **LabVIEW Conduit** dialog box appears. Refer to the *HotSync Manager Help* for more information about the different HotSync options and transferring files between the host computer and PDA device.

Transferring Files to the Host Computer

To copy files from the Palm OS device to the host computer, select **Handheld overwrites Desktop** in the HotSync **LabVIEW Conduit** dialog box. When you sync the Palm OS device, LabVIEW places the files in the location you specify in the path input of the file I/O function you use to create the file in the PDA VI.

Transferring Files to the PDA Device

To copy files to the Palm OS device, click the **Browse** button in the **LabVIEW Conduit** dialog box to select the files. Click the **Add File** button to add a file to the list, select **Desktop overwrites handheld**, and click the **OK** button. LabVIEW copies the files you select to the Palm OS device the next time you sync it.

(Palm OS) File I/O Using Expansion Cards

You can perform file I/O operations using Palm OS expansion cards. To write to or read from files on an expansion card, specify a file path of `Card#://name of file`, where # is the volume reference number of the expansion card. If you have only one expansion card, the file path usually is `Card1://name of file`.

(Palm OS) Using the DatalogMgr Utility

Use the DatalogMgr utility to view a list of files you transferred to the Palm OS device from the host computer or created on the Palm OS device. On the PDA device, tap **DatalogMgr** to view a list of all LabVIEW files on the PDA device and the size of each file in number of records or in bytes. The DatalogMgr displays where the LabVIEW Conduit copies the files on the host computer when you sync. You also can delete files from the DatalogMgr.



Note You must install the `DatalogMgr.prc` on the Palm OS device to transfer files to and from the host computer. The LabVIEW Conduit uses the DatalogMgr to detect the presence of LabVIEW files on the PDA device. Refer to the *Getting Started with the LabVIEW PDA Module* user guide for information about installing the `DatalogMgr.prc` file.

(Pocket PC 2003) File I/O Using ActiveSync

Use ActiveSync to transfer files between the host computer and the Pocket PC 2003 device. When you connect the Pocket PC 2003 device to the host computer, you can use ActiveSync to browse the files on the Pocket PC 2003 device and copy files between the host computer and Pocket PC 2003 device.

Serial I/O

You can use a serial connection to transfer data between the PDA device and the host computer just as you use the serial I/O VIs to transfer data between a computer and an instrument or between two computers.

Use the NI Example Finder, available in LabVIEW by selecting **Help»Find Examples**, to find an example that uses serial I/O to communicate with a host VI.

Infrared

You can use the IrDA functions to transfer data between the PDA device and the host computer or another PDA device.

Use the NI Example Finder, available in LabVIEW by selecting **Help»Find Examples**, to find an example that uses IrDA I/O to communicate with a host VI.

Wireless TCP

If you use a PDA device that supports wireless TCP communication, you can use the TCP VIs and functions to transfer data between the PDA device and the host computer or another PDA device. Refer to Chapter 4, [Building Wireless PDA Applications](#), for information about creating wireless PDA applications.

Use the NI Example Finder, available in LabVIEW by selecting **Help»Find Examples**, to find examples that use TCP to communicate with a host VI.

Bluetooth

You can use the Bluetooth VIs and functions to communicate with devices that use the Bluetooth communication protocol and the Microsoft Bluetooth driver on Windows XP Service Pack 1 or later.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about the Bluetooth VIs and functions.

Debugging PDA Applications

This chapter describes how to debug a PDA application on a PDA device. Debugging PDA applications is significantly different from debugging VIs on the host computer because PDA applications do not include a block diagram.

Enabling Debugging for PDA Applications

By default, the LabVIEW PDA Module does not build PDA applications you can debug on the PDA device. To enable debugging, select **Tools» Build for PDA** and click the **Code Generation Settings** tab. Place a checkmark in the **Generate debug info** checkbox and build the PDA VI into a PDA application. Enabling debugging increases the size of the PDA application. Rebuild the PDA VI without debugging before deploying it.

Debugging PDA Application

When you debug a Palm OS PDA application, the PDA application uses a serial connection to connect the corresponding PDA VI on the host computer. However, when you debug a Pocket PC 2003 application, the PDA application can use a serial or a USB connection. You use the block diagram of the PDA VI on the host computer as a conduit between the PDA application running on the PDA device and the PDA VI running on the host computer.



Note (Palm OS) You must use a serial cradle or serial cable to debug a PDA application on a Palm OS device, emulator, or simulator. You cannot use a USB connection.

The front panel of the PDA application is fully functional during debugging. However, the front panel controls of the host VI have no effect on the PDA application, and the indicators of the PDA VI do not reflect the execution of the PDA application.

You can use all the debugging tools except **Execution Highlighting** on the block diagram of the PDA VI when you debug a PDA application. You also can insert probes and breakpoints as you do in any other VI.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help** and selecting the **PDA Module** book in the **Contents** tab, for more information about debugging PDA applications.

Building Wireless PDA Applications

This chapter describes how you can create services on remote servers and use wireless TCP communication, UDP communication, and Bluetooth wireless technology on a PDA device to access the services.

You must have a TCP-capable PDA device to use TCP communication. Refer to the documentation for the PDA device to determine if the PDA device supports TCP networking to build wireless applications.

TCP, UDP, and Bluetooth Support

The LabVIEW PDA Module supports all LabVIEW TCP, UDP, and Bluetooth VIs and functions. You also can use the Call By Reference Node to call a VI remotely using a wireless PDA device over TCP. To use the Call By Reference Node, you must call the Open Application Reference function and enter `TCP::address` in the **machine name** input, where *address* is the IP address of the remote computer that is running the VIs you want to reference.



Note PDA devices support all UDP functions except the UDP Multicast Open function.

Clients, Services, and Arbitrators

Wireless PDA applications consist of a client, services, and, in some cases, an arbitrator.

The client is the PDA application that monitors or controls remote services over the network.

Services are VIs or other applications that perform tasks that the client accesses. For example, a service might be a VI that monitors temperature or tank level, or a VI that controls a valve or fan. You build services as you do any other VI. The way you connect to these services depends on the type of wireless application you use.

You use an arbitrator in some wireless applications to facilitate communication among a client and the services on the subnet, which is a portion of a network that shares a common component, such as IP address prefix.

Types of Wireless Applications

There are two main types of wireless PDA applications—direct connection applications and distributed connection applications.

Direct Connection Applications

Use direct connection applications when you know the location of the remote services to which you need to connect.

Direct connection wireless applications consist of a PDA client communicating directly with a service on a server or another device. An example of a direct connection application is a client running on a PDA device that uses the TCP Open Connection function to connect to a remote service on the network and then uses the TCP Read function to read the data that the remote service broadcasts over TCP.

Connecting over TCP or UDP

You can use the TCP or UDP functions to communicate between the client and the service as you do in any other VIs that use TCP or UDP communication. Refer to Chapter 18, *Networking in LabVIEW*, of the *LabVIEW User Manual* for more information about using TCP or UDP communication.

Connecting Using Bluetooth Wireless Technology

You can use the Bluetooth VIs and functions to communicate with devices that use the Bluetooth wireless technology.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about the Bluetooth VIs and functions.

Calling by Reference

You also can call a service by reference using the Call By Reference Node or the PDA Request Service VI. VI Server must run on the server to support the Call By Reference Node, and the services available on a server must be in memory so you can reference them.

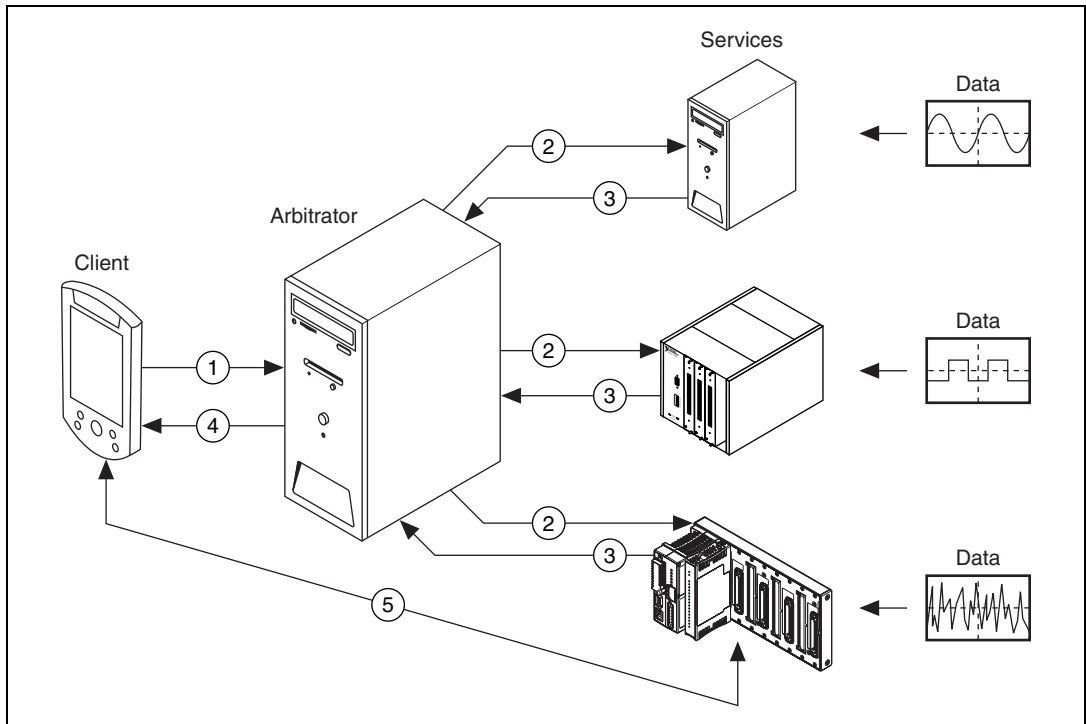
The PDA Request Service VI provides an easy way to connect a PDA client to services by reference. When you place the PDA Request Service VI on the block diagram, a dialog box appears that prompts you to enter a path to a service. When users run the client on the PDA device, the PDA VI prompts them to enter or retrieve information from a service that the client calls by reference on a server. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about the PDA Request Service VI.

Distributed Connection Applications

Use distributed connection applications when you want to have flexible access to several services distributed over a network.

Distributed connection wireless applications consist of a PDA client, remote servers that run service VIs, and an arbitrator VI that facilitates connections among the services and the client. The arbitrator VI can reside on a separate server or on one of the servers where a service resides.

The following figure shows a distributed connection PDA application.



1. The client on the PDA device connects to the server using TCP.
2. The arbitrator sends a UDP broadcast to the local subnet that contains services the PDA client wants to reference. The arbitrator must request services on behalf of the PDA client because some PDA devices do not support UDP broadcasts.
3. When the server that contains a service receives the UDP broadcast, the server checks its available services and replies if it has the services the PDA client requested.
4. The arbitrator uses TCP to reply to the PDA client with the servers of the local subnet that replied as having the services the PDA client requested.
5. The PDA client connects to one of the servers to call a particular service.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about building distributed connection wireless applications. You also can refer to the `DeviceDiscoveryExample.llb` located in the `\examples\PDA` directory for an example of a distributed connection wireless application.

Configuring Services for PDA Devices

Use the **Configure Services for PDA** dialog box to build distributed connection wireless applications. Select **Tools»Configure Services for PDA** to launch the **Configure Services for PDA** dialog box and create an index VI that contains the available services for a server.

You can have more than one index VI on a server. When you create an index VI using the **Configure Services for PDA** dialog box, you must use the dialog box to edit the index VI.



Caution Directly editing an index VI can corrupt an index VI.

Setting up Servers

In distributed connection applications, you must run the Service Response VI, located in `vi.lib\PDA\Service Response.llb`, on each server that provides services to the client. In the **Path to Index VI** text box, enter the path to the index VI you create using the **Configure Services for PDA** dialog box. Enter the computer name in the **Name of Machine** text box before you run the Service Response VI.

You must run the Arbitrator Service Query VI, located in `vi.lib\PDA\Arbitrator Service Query.llb`, on the arbitrator. The Arbitrator Service Query VI facilitates the communication among the client and the services.

When you create the PDA VI, include the PDA Service Query VI as a subVI. The PDA Service Query VI, located in `vi.lib\PDA\DeviceDiscovery.llb`, connects to the Arbitrator Service Query VI to initiate the service request.

Programming Techniques

This chapter describes limitations you might encounter while creating PDA VIs and ways to work around these limitations. This chapter also describes techniques you can use to create effective PDA VIs and PDA applications.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help** and selecting the **PDA Module** book in the **Contents** tab, for more information about front panel and block diagram features that are not supported or that behave differently on PDA execution targets than they do on LabVIEW for Windows.

Palm OS and Pocket PC 2003 VI Templates

Because PDA devices have smaller screens and a limited set of controls and indicators, the LabVIEW PDA Module includes templates for creating PDA VIs. Select **New»Palm OS Screen** or **Pocket PC Screen** from the **LabVIEW** dialog box, or select **File»New** and select **Palm OS Screen** or **Pocket PC Screen** to use these template VIs as the starting point for developing PDA VIs.

(Palm OS) The Palm OS Screen template VI uses 48-point fonts, and the overall size of the front panel is 640 × 640 pixels. When you build a PDA application for Palm OS devices, emulators, or simulators, LabVIEW reduces the front panel by a scale factor of four, so fonts appear as 12-point type, and the front panel is 160 × 160 pixels.

The Palm OS Screen template VI also includes a standard Palm OS title bar. Do not place controls or indicators in this region.

(Pocket PC 2003) The Pocket PC Screen template VI uses 28-point fonts, and the overall size of the front panel is 480 × 640 pixels. When you build a PDA application for Pocket PC 2003 devices or emulators, LabVIEW reduces the front panel by a scale factor of two, so fonts appear as 14-point type, and the front panel is 240 × 320 pixels.

Functionality Differences in PDA VIs

PDA VIs you create and build into PDA applications do not support the full range of functionality as VIs you create for a desktop computer because of limitations inherent to the smaller size of the PDA device, memory restrictions, and operating system differences.

The following PDA device considerations can affect the way you design a PDA VI.

- PDA devices have a smaller screen and a significantly slower CPU than desktop computers.
- The main input device is the pen, which makes inputting text slower.
- Many PDA devices do not have floating-point coprocessors, so floating-point operations can be slow because the software performs all floating-point calculations.

PDA Operating System Considerations

Because Palm OS and Pocket PC 2003 are designed for small size and mobility, both environments have fewer features than a desktop computer. As a result of this design, you usually cannot create a VI for a desktop computer and expect it to run on a PDA device without any modifications.

The following operating system and file system limitations can affect the way you design a PDA VI for Palm OS and Pocket PC 2003.

- PDA devices allocate all the memory a PDA application needs when the PDA application runs or when another PDA application calls the PDA application. The PDA operating system frees memory when the PDA application stops running or returns. As a result, memory usage can depend on the run-time behavior and be difficult to predict. However, the PDA operating system allocates uninitialized shift registers the first time it encounters them and frees them only when the PDA application stops. PDA devices reset controls to the default value each time you run or call a PDA application, but the PDA devices do not reset uninitialized shift registers.
- PDA VIs ignore all options in the **VI Properties** dialog box, with the following exceptions:
 - On the **Execution** page, PDA VIs ignore all priority settings except **subroutine**. PDA VIs set to run in the **subroutine** priority execute one by one.
 - On the **Execution** page, you can place a checkmark in the **Reentrant execution** checkbox to create reentrant PDA VIs.

- On the **Execution** page, you can place a checkmark in the **Allow debugging** checkbox to enable debugging on PDA devices.
- On the **Windows Appearance** page, you can click the **Customize** button and place a checkmark in the **Show front panel when called** checkbox in the **Customize Window Appearance** dialog box.

Palm OS Considerations

The following considerations can affect the way you design a PDA VI for Palm OS devices, emulators, and simulators.

- Palm OS is not multithreaded and has limited memory for programs.
- Palm OS does not use a traditional file system. It stores data in memory chunks called records that are grouped into databases. Expansion cards use a traditional file system.
- Palm OS implements all files as databases and can have a maximum of 64,000 records in each database. Each record can be a maximum of 64 KB in size.
- Each VI within a PDA VI cannot be larger than 64 KB because of the way Palm OS links the applications together. However, a top-level PDA VI can contain several subVIs whose total file size is greater than 64 KB.

The size of a PDA VI on the host computer is not an accurate indication of how large it will be when you build it as a PDA application. When you build a PDA VI into a PDA application, LabVIEW notifies you if the file size is too large. Break up the VI into smaller subVIs to reduce the file size.

- There is a total limit of 64 KB on all front panel array data.
- Applications are limited to approximately 160 KB of heap, or dynamic memory space, for everything except code. The application uses some of this space to maintain state, so not all of it is available for data. PDA VIs immediately terminate if they run out of memory. You can use the PDA Memory VIs to manage the dynamic memory space.

Pocket PC 2003 Considerations

PDA applications are limited to about 32 MB of space for everything except code. The application uses some of this space to maintain state, so not all of it is available for data. PDA applications immediately terminate if they run out of memory.

Front Panel Design

There are several differences between developing a front panel for a host VI and developing a front panel for a PDA VI. The following sections describe considerations for window appearance, VI hierarchy, and available controls and indicators.

Window Appearance

Front panels and objects on the front panel behave and appear differently on PDA devices than they do on a desktop computer. Consider the following issues when you develop PDA VIs.

- PDA VIs display only one front panel at a time. If you set a subVI to show its front panel when called, the subVI front panel obscures the front panel of the calling VI.
- PDA VIs do not include toolbar icons or standard desktop menus. You can use custom run-time menus.
- You cannot use the <Tab> key to navigate the controls or indicators or use keyboard shortcuts in PDA VIs. Minimize the need for keyboard input on the front panel.
- When you build the PDA VI on the host computer, LabVIEW scales all positions, sizes, and fonts by the factor you enter in the **Front panel scale factor** control on the **Code Generation** tab of the **PDA Build Settings** dialog box. Refer to the *Palm OS and Pocket PC 2003 VI Templates* section of this chapter for information about Palm OS and Pocket PC 2003 default fonts and front panel sizes. The PDA-4x palette view contains controls and indicators with large fonts and sizes you can use to create front panels that work well on Palm OS. The PDA-2x palette view contains controls and indicators with large fonts and sizes you can use to create front panels that work well on the Pocket PC 2003.

Hierarchical Limitations

Because of the different application structure on PDA devices, PDA VIs always start from one top-level VI and complete when that top-level VI completes. This behavior is different from a typical LabVIEW application in which you can start and run multiple top-level VIs simultaneously.

Controls and Indicators

Most controls and indicators on the PDA-4x and PDA-2x palettes are based on the built-in controls in Palm OS and Pocket PC 2003 to save memory. The appearance of these controls makes it easy to create a standard user interface, but the controls might not look the same on the PDA device as they do on the host computer.

- The following controls and indicators are built in to Palm OS and Pocket PC 2003:
 - Numeric
 - Button
 - Checkbox
 - String and path
 - Enumerated
 - Ring
 - Listbox
 - Table
 - Tab
- Array and cluster controls and indicators do not appear on the front panel of PDA VIs. If you want the controls and indicators that are in an array or cluster visible on the front panel, you must remove the elements from the array or cluster.
- Boolean controls have only the **Latch When Released** and **Switch When Released** mechanical actions. LabVIEW treats the default mechanical action, **Switch When Pressed**, as **Switch When Released**, and the other mechanical actions as **Latch When Released** or **Switch When Released**.
- If the PDA device supports color, the PDA VI uses the nearest color match. Most built-in controls use the default system colors, and you cannot change them.
- You can place bitmaps on front panels of PDA VIs. You also can use the Draw 1-bit Pixmap function, Draw 4-bit Pixmap function, or Draw 8-bit Pixmap function. However, LabVIEW does not scale bitmap

images when you build the PDA VI into a PDA application. The bitmaps retain their original size to avoid the aliasing effects that can result if they are reduced in size.

Showing and Hiding Controls and Indicators

The way you programmatically show and hide controls and indicators in a PDA VI is different than on LabVIEW for Windows. Use the Show or Hide Control VI to programmatically control the visibility of controls and indicators on the front panel of a PDA VI.

Unsupported Front Panel Objects

LabVIEW discards any unsupported front panel controls when you build the PDA VI into a PDA application. Also, PDA VIs ignore unsupported features in supported controls. PDA devices do not support the following front panel objects.

- Multicolumn listboxes
- Intensity graphs and charts
- Digital waveform graphs
- 3D graphs
- Polar plots
- Smith plots
- Minmax plots
- Radar plots
- Picture rings
- Text and picture rings
- Refnums
- Transparent controls
- I/O controls and indicators
- ActiveX controls and indicators
- Variant controls and indicators

Block Diagram Programming

You can use the following techniques for developing block diagrams for PDA VIs you want to build into PDA applications.

- You cannot view the block diagram on the PDA device because PDA applications are compiled applications for a PDA device. You view and edit the block diagram in the PDA VI from which you built the PDA application.
- The PDA Module supports most basic LabVIEW functions. However, the underlying behavior, particularly with respect to memory usage, is completely different. Refer to the [Application Design and Performance Considerations](#) section of this chapter for information about memory usage.
- You can use the host computer to remotely debug a PDA device in its cradle. Refer to Chapter 3, [Debugging PDA Applications](#), for information about debugging techniques.

Using Events

The PDA Module supports the Mouse Up, Mouse Down, Mouse Move, Value Change, and Menu Selection events. The mouse events work only with picture controls.



Note (Palm OS) To use coordinates from the Mouse Move event on Palm OS devices in high-density display mode, you must convert the mouse coordinates to the low-density equivalent before you wire them to a picture control. For example, you must convert the high-density coordinates of 240×200 to 120×100 .

Refer to Chapter 9, *Event-Driven Programming*, of the *LabVIEW User Manual* for more information about events.

Unsupported Block Diagram Objects

PDA devices do not support the following block diagram objects.

- Waveform data types and all related waveform VIs and functions
- **(Palm OS)** Data acquisition and instrument I/O features
- **(Pocket PC 2003 Emulator)** TCP Open Connection and UDP Open functions

- Probes with pictures, paths, or extended-precision numbers
- Custom probes
- Variant data

Application Design and Performance Considerations

All PDA devices run on battery power. A user typically runs PDA applications for short intervals and then expects the device to switch to low power mode until the next time the user needs it.

To conserve battery resources, include a Wait For Front Panel Activity function or an Event structure in a PDA VI to create a wait state to save battery power. Adding a wait state allows the PDA device to go to sleep if the user does not tap the screen for the duration the user specifies in the preferences settings of the PDA device.

LabVIEW optimizes PDA VIs for data size, speed, and code size, in that order. Design PDA VIs with wide, shallow VI hierarchies. Nested VI calls use large amounts of memory and might cause the PDA VI to fail. Wide, shallow hierarchies have only a few VIs running at any time and use less memory.

Even though Palm OS does not support multiple threads, it uses cooperative multitasking. Therefore, you can design PDA VIs to have parallel loops on the block diagrams. One loop runs for a time slice of approximately 50 ms, and then another loop runs for a time slice. PDA VIs automatically use cooperative multitasking when you use two or more nodes that a wire does not connect. However, PDA VIs with parallel loops run slower than VIs with serial loops.

If you set the execution priority of a PDA VI to **subroutine**, LabVIEW serializes execution of the PDA VI. The PDA VI runs faster, but the user interface does not get any CPU time, and the application might become unresponsive. Place a Wait Until Next ms Multiple function on the block diagram to accommodate this PDA device limitation.

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 immediate 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 Application Engineers worldwide in the NI Developer Exchange at ni.com/exchange. National Instruments Application Engineers make sure every question receives an answer.
- **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, NI Alliance Program members can help. To learn more, call your local NI office or visit ni.com/alliance.

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