Warranty and Repair Policy Statement

General

Seller warrants that its products furnished hereunder will, at the time of delivery, be free from defects in material and workmanship and will conform to Seller's published specifications applicable at the time of sale. Seller's obligation or liability to Buyer for products which do not conform to above stated warranty shall be limited to Seller, at Seller's sole discretion, either repairing the product, replacing the product with a like or similar product, or refunding the purchase price of the nonconforming product, provided that written notice of said nonconformance is received by Seller within the time periods set forth below:

a. For all software products, including licensed programs, ninety (90) days from date of initial delivery to Buyer;

b. For all hardware products (excluding batteries), including complete systems, fifteen (15) months from date of initial delivery to Buyer, subject to the additional conditions of paragraph c) below;

c. In the event that Buyer's returned product is a Discontinued product and is not repairable for any reason, Seller may elect to replace it with like or similar product that is, in Seller's sole judgment, the closest equivalent to the returned product. Seller does not warrant that such replacement product will be an exact functional replacement of the returned product.

Further, all products warranted hereunder for which Seller has received timely notice of nonconformance must be returned FOB Seller's plant no later than thirty (30) days after the expiration of the warranty periods set forth above.

These warranties provided herein shall not apply to any products which Seller determines have been subjected, by Buyer or others, to operating and/or environmental conditions in excess of the limits established in Seller's published specifications or otherwise have been the subject of mishandling, misuse, neglect, improper testing, repair, alteration or damage. THESE WARRANTIES EXTEND TO BUYER ONLY AND NOT TO BUYER'S CUSTOMERS OR USERS OF BUYER'S PRODUCT AND ARE IN LIEU OF ALL OTHER WARRANTIES WHETHER EXPRESS, IMPLIED OR STATUTORY INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SELLER BE LIABLE FOR INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES. Seller's liability for any claim of any kind shall in no case exceed the obligation or liability specified in this Warranty clause.

Technical Assistance and Service

Seller's warranty as herein set forth shall not be enlarged, diminished or affected by, and no obligation or liability shall arise or grow out of, Seller's rendering of technical advice, facilities or service in connection with Buyer's order of the goods furnished hereunder. Products returned for warranty service, but which are found to be fully functional and in conformance with specifications may be subject to a nominal service charge and return freight charges. Periodic re-calibration of products, if required, is the responsibility of Buyer and is not provided under this Warranty.

Online Support

Online support is available through technical support links on Intelligent Instrumentation's 24-hour World Wide Web site at http://www.lanpoint.com. The site contains information on Intelligent Instrumentation's products, new developments, announcements, application notes, application examples, and other useful information. The site and support areas continue to grow as new products, updates, and features are added.

Email Support

Intelligent Instrumentation's technical support can be reached via email. When sending an email message, be sure to include complete contact information, the product model/part number with third-party accessory information, and a detailed description of the problem to support@lanpoint.com
**Static Sensitivity**

Seller ships all static-susceptible products in anti-static packages. Seller’s Warranty as herein set forth shall not cover warranty repair or replacement for products damaged by static due to Buyer’s failure to use proper protective procedures when handling, storing, or installing products.

**Trademarks**

LANpoint Computers®, LANpoint®, Intelligent Instrumentation® are trademarks of Intelligent Instrumentation, Inc. Other products or brand names are trademarks or registered trademarks of their respective companies.

**Use of Equipment**

Intelligent Instrumentation Inc. assumes no responsibility for any direct, indirect or consequential loss or damages resulting from misuse of the equipment or for improper or inadequate maintenance of the equipment or for any such damage or loss resulting from the use of other equipment, attachments, accessories, and repairs at any time made to or placed upon the equipment or any replacement thereof. Furthermore, Intelligent Instrumentation Inc. makes no representations or warranties, either expressed or implied, in connection with the use of the equipment in the event it is improperly used, repaired or maintained.

**FCC Radio Frequency Interference Statement**

This equipment generates and uses radio frequency energy, and may cause interference to radio or television reception.

Per FCC rules, Part 15, Subpart J, operation of this equipment is subject to the conditions that no harmful interference is caused and that interference must be accepted that may be caused by other incidental or restricted radiation devices, industrial, scientific or medical equipment, or from any authorized radio user.

The operator of a computing device may be required to stop operating his device upon a finding that the device is causing harmful interference and it is in the public interest to stop operation until the interference problem has been corrected.

The user of this equipment is responsible for any interference to radio or television reception caused by the equipment. It is the responsibility of the user to correct such interference.

**European CE Certification**

European CE certification is as described on the Declaration of Conformity that ships with each LANpoint Power X terminal.

**Revision History for the LANpoint Power X Developer’s Manual**

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>032212</td>
<td>Pre-release</td>
</tr>
<tr>
<td>1.0</td>
<td>083112</td>
<td>Initial release</td>
</tr>
</tbody>
</table>
# Table of Contents

## Chapter 1  Setup & System Configuration

1.1 About this Manual ........................................ 9  
1.2 Hardware Features .................................... 10  
1.2.1 Circuit Boards .................................. 10  
  1.2.1.1 CPU Board .................................. 10  
  1.2.1.2 Optional Digital I/O Termination Panel .......... 10  
1.2.2 Power ........................................ 11  
1.2.3 Ethernet ....................................... 11  
1.2.4 Keyboard and USB ................................. 11  
1.2.5 Serial Communications Port ....................... 12  
1.2.6 Bar Code/AutoID input ........................... 12  
1.2.7 Audio Input and Output ........................... 12  
1.2.8 Display ...................................... 13  
1.2.9 Status LEDs .................................... 13  
1.2.10 Screen Navigation and Operation.................. 13  
1.2.11 External Keyboard, Mouse, and other USB or Bluetooth Accessory .... 13  
1.2.12 Navigation Using the Touchscreen ............... 15  
1.2.13 Using Combination Keystrokes .................... 15  
1.3 Operation and Configuration .......................... 15  
  1.3.1 Windows CE Operating System .................... 16  
  1.3.2 Initial Power-up Sequence ....................... 16  
  1.3.3 Configure TCP/IP Parameters, or Disable Connection .......... 16  
  1.3.4 File Transfer Methods .......................... 16  
  1.3.5 Accessing Shared Directories ................... 17  
  1.3.6 Instructions to Create a Shortcut in Windows CE ..... 17  

## Chapter 2  Developing Custom Programs

2.1 Selecting the Correct Development Tools for your Environment ........ 19  
2.1.1 Visual Studio 2005 ................................ 19  
2.2 Development Overview ................................ 19  
  2.2.1 Application & Data Transfer Methods ............. 19  
2.3 Development with Microsoft Visual Studio 2005 .................. 20  
  2.3.1 Installing Microsoft Visual Studio.NET 2005 ..... 20  
  2.3.2 Installing the LANpoint SDK .................... 20  
  2.3.3 Sample C++ development session .................. 23  
2.4 ActiveSync Reference ................................ 27  
  2.4.1 Terminal Setup - ActiveSync ...................... 28  
    2.4.1.1 Preparing the terminal for ActiveSync ........ 28  
    2.4.1.2 New Hardware Wizard ....................... 31  
    2.4.1.3 Deploying an Application to a terminal via ActiveSync .... 33
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.3</td>
<td>Return Values</td>
<td>61</td>
</tr>
<tr>
<td>4.2.3.1</td>
<td>When Errors Occur</td>
<td>61</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Digital I/O Input Functions</td>
<td>62</td>
</tr>
<tr>
<td>4.2.4.1</td>
<td>Return the State of a Single Channel</td>
<td>62</td>
</tr>
<tr>
<td>4.2.4.2</td>
<td>Return the Current State of All Input Channels</td>
<td>63</td>
</tr>
<tr>
<td>4.2.4.3</td>
<td>Configure the Mode of an Input Channel</td>
<td>64</td>
</tr>
<tr>
<td>4.2.4.4</td>
<td>Obtain the Current Configuration Mode of a Channel</td>
<td>65</td>
</tr>
<tr>
<td>4.2.4.5</td>
<td>Store Input Modes and Output States</td>
<td>65</td>
</tr>
<tr>
<td>4.2.4.6</td>
<td>Retrieve Input Modes and Output States</td>
<td>66</td>
</tr>
<tr>
<td>4.2.4.7</td>
<td>Return Data from a Counter Bit</td>
<td>67</td>
</tr>
<tr>
<td>4.2.5</td>
<td>Digital I/O Output Functions</td>
<td>67</td>
</tr>
<tr>
<td>4.2.5.1</td>
<td>Control a Single Output Channel</td>
<td>68</td>
</tr>
<tr>
<td>4.2.5.2</td>
<td>Open or Close All Outputs</td>
<td>69</td>
</tr>
<tr>
<td>4.2.6</td>
<td>Digital I/O Error Codes</td>
<td>70</td>
</tr>
<tr>
<td>4.3</td>
<td>Reboot API</td>
<td>71</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Reboot Calls</td>
<td>72</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Reboot Error Codes</td>
<td>72</td>
</tr>
</tbody>
</table>

Index
Page Intentionally Blank
Chapter 1  Setup & System Configuration

This manual provides information on how to program the LANpoint Power X terminal and is written for developers having knowledge of the Microsoft Windows CE operating system and Microsoft Visual Studio.

In addition to the LANpoint Power X Developer’s Manual, there is a LANpoint Power X Quick Start Guide, and the LANpoint Power X Installation & Maintenance Manual. There are also manuals for specific mounting accessories, which are shipped with those products.

All manuals are available as Adobe Acrobat files located on Intelligent Instrumentation’s web site at www.lanpoint.com. Sample programs are also provided on this web site.

1.1 About this Manual

The LANpoint Power X Developer’s Manual contains the following information:

Chapter 1  Setup & System Configuration

This chapter provides a brief overview of the terminal’s hardware features and general software features including: operation of the terminal from the initial power-up sequence, configuring the terminal, transferring files, creating shared directories, creating shortcuts, and permanently saving configuration settings.

Chapter 2  Developing Custom Programs

This chapter contains installation information for the Microsoft eMbedded Visual Tools software, Microsoft Visual Studio, and the LANpoint SDK (Software Development Kit). It includes information on writing and compiling custom programs. It provides instructions on launching eMbedded Visual Tools, Microsoft Visual Studio, configuring the platform manager, and file transfer methods.

Chapter 3  Utility Programs

This chapter explains the following utility programs: CEPad text editor, keyboard wedge, programmable function keys, time synchronization, Remote Manager, SerialSocket, and the optional lock-down kiosk Web browser. Several terminal emulation utilities for cloud computing and thin-client deployment are also described in this chapter.

Chapter 4  APIs

This chapter contains information about the digital I/O API and the reboot API.
1.2 Hardware Features

1.2.1 Circuit Boards

The standard terminal contains a CPU board. The Digital I/O termination panel board is an option. This section includes a brief description of these boards. For more information about hardware, see the LANpoint Power X Installation and Maintenance Manual.

1.2.1.1 CPU Board

The CPU board includes an embedded microcontroller, Flash and RAM memory, power circuitry, and supporting hardware for the display, keypad, external connectors, and options/accessories.

1.2.1.1.1 Embedded Microcontroller

The CPU board features the Intel XScale PXA310 624MHz embedded microcontroller. The Windows CE 6 (or other) operating system and boot code run on this CPU and oversee all of the terminal’s functions.

1.2.1.1.2 Memory: RAM

The terminal contains 128MB of DRAM

1.2.1.1.3 Memory: FLASH, SD/MMC Slot

FLASH memory provides at least 512MB of non-volatile solid-state memory for the operating system, utilities, application program(s), and application data storage

The terminal ships from the factory with approximately 390MB free space in FLASH for application programs and local application data storage.

An SD card of virtually any size can be added to provide additional non-volatile data storage that will act the same as the hard drive of a PC, but with no moving parts. SD cards can be hot-swapped. The SD slot is on the main board, toward the top right side when viewing the terminal from the rear with the cover off.

Applications to be executed on start-up just on a desktop PC by placing them or shortcuts to them in the \Windows\Startup folder.

LANpoint Power X models have persistent flash memory. That is, changes in configuration, files, upgrades and such is available after reboot without special action. NOTE: While this seems normal, this is unlike Windows CE.NET 4.2 and earlier OS releases, where everything except the \flash folder was in the RAM filesystem and therefore only persisted until it is rebooted.

1.2.1.2 Optional Digital I/O Termination Panel

The digital I/O termination panel option provides easy connection to the eight optically-isolated inputs and eight digital outputs. For more information, see the LANpoint Power X Installation and Maintenance Manual.
1.2.2 Power
The main board has a barrel connector for application of 12VDC power from an external source. It also has a 2-pin header for temporarily powering from an optional backup battery pack - instructions for installing the backup battery pack can be found in the LANpoint Power X Installation Manual. Each terminal has a 4-pin auxiliary power connector that provide ground, +3.3 V, +5 V, and +12 V for auxiliary equipment such as serial scanners. See the Appendix of the LANpoint Power X Installation and Maintenance Manual for more information.

1.2.3 Ethernet
Every terminal has an RJ45 10/100 Base-T Ethernet port. The terminals can be ordered with an IEEE802.11b/g wireless radio frequency (WiFi) Ethernet option; these models come with an external antenna. See the LANpoint Power X Installation and Maintenance Manual for configuration, and additional information.

1.2.4 Keyboard and USB
The integral 69-key QWERTY keypad is implemented as a USB device and features a long-life industrial design. It is waterproof and features a special clear epoxy coating on the keys to protect the legends from wear and tear. There are eight programmable function keys for use as menu selectors and other operator shortcuts. Each of the eight keys (as well as almost any key or key combination) can be programmed to launch an application or insert a user-defined string of characters into the keyboard buffer. By using modifier keys (ctrl, alt, shift), up to 32 individual functions can be programmed using these 8 keys. See the programmable function key utility in section 3 of this manual for more information.

An external USB - and Bluetooth, if terminal is so equipped - keyboard, mouse, Auto-ID device, thumb drive and/or other USB devices can be connected to the terminal if desired. USB on this terminal supports hot-swap of most devices - they become operational as soon as plugged-in and do not require a reboot.

![FIGURE 1 USB Connectors](image-url)
1.2.5 Serial Communications Port

COM5 is used for normal RS232 serial communications. The COM port factory default communications parameters are 9600 baud, no parity, 8 data bits, 1 stop bit, and no flow control.

A serial AutoID device like a laser scanner, can be connected to this port - see more information immediately below. The WedgeCE program can be used to wedge input data into the keyboard data stream from the COM5 port.

1.2.6 Bar Code/AutoID Input

The terminal supports Auto-ID readers from many manufacturers via several connectivity technologies.

- Standard USB devices (magnetic stripe and barcode badge slot readers, wands, lasers, imagers, etc.) can be connected via the USB ports. Power is supplied through the USB port. The AutoID data is placed directly in the keyboard data stream by most USB HID devices.
- Serially connected devices such as 2D scanners, slot readers, biometric readers, proximity readers and the like, can be connected via COM5. A jumper associated with the port can be set to supply +5VDC or +12VDC to pin 9 of the standard DB9 connector.
- A data "wedge" software utility included with each terminal may be used to "wedge" serial input data into the keyboard buffer. See the Chapter 3 for more information.

To power an autoID device (or any accessory, for that matter), each terminal has a connector with 4 pins that can be used to provide ground, 3.3VDC, 5VDC, and/or 12VDC via a jumper wire. Alternatively, pin 9 of the COM port can be configured to provide +5VDC or +12VDC, common on many serial scanners, via a jumper. See the appendix of the LANpoint Power X Installation and Maintenance Manual, Auxiliary Power Outputs section for more information.

1.2.7 Audio Input and Output

The terminals come with a small integral speaker. They also include audio input and output jacks for attaching separate speakers and microphone, as shown below. Jack connectors are described in more detail in the Installation Manual

Use the Windows CE controls to adjust volume. Detailed instructions are included in the Installation Manual.
1.2.8 Display

The terminals come with a 640 x 240 resolution (1/2 VGA) 64K color display with 350-nit backlight and wide viewing angle. Touchscreen is available as an option.

By default, the backlight is configured to turn off after a period of inactivity - 15 minutes is factory default. The display is blank/black when the backlight is off. Any action - touchscreen touch, key touch, mouse movement, and so on, will turn it back on. To turn it off, or turn it on continuously, or adjust the length of the inactivity timer, use the Backlight tab in the Display applet in the Control Panel. For instance, if the battery backup accessory is installed, you may wish to select the Backlight tab, select the Automatically turn off backlight while on battery power checkbox, select the desired interval, and click the OK button.

1.2.9 Status LEDs

Two LEDs are visible on the face of the terminal:

- Power - green for external power, yellow for battery power
- Ethernet Link - illuminated when linked to wired or wireless network

1.2.10 Screen Navigation and Operation

The pointer can be manipulated by using an external mouse, the optional touch-screen, or the keyboard, either built-in or external.

1.2.11 External Keyboard, Mouse, and other USB or Bluetooth Accessory

To use an external USB keyboard, mouse, scanner, or other USB accessory simply connect the cable to one of the USB ports. Most devices will be operational immediately. Some may require a reboot.

To use an external bluetooth keyboard, mouse, scanner, or other bluetooth accessory, pair the accessory with the terminal. To do this:

Press the Start button, select Control Panel and double-click on Bluetooth Device Properties. The Bluetooth Manager window opens:
Press the *Scan Device* button. Any bluetooth devices in range will show up in the *Untrusted* window. You may have to press a *RESET* button on the bluetooth device if it does not show up within a few minutes. Select the one you are interested in and press the --> button to move it to the *Trusted* window. You may then see an authenticate request like this:

Usually you will not need to authenticate the device - if not, press the *No* button. If you press the *Yes* button, you will be asked for the device’s PIN number. After this action, double-click on the device in the *Trusted* window and select the appropriate checkboxes - particularly *Active*:
Most devices will be operational immediately. Some may require a reboot. You can now add another bluetooth device or simply close the Bluetooth Manager dialog window.

1.2.12 Navigation Using the Touchscreen

The optional touchscreen on the terminal works like a mouse. Tapping the screen once on a menu item causes the same response as left-clicking a mouse. Two consecutive rapid taps generate a double left-click response. Pressing and holding the Alt key while delivering a single tap to the screen, or pressing on an icon on the screen for more than one second, results in the same response as right-clicking a mouse.

Calibration

If the touchscreen does not respond to the location touched, touchscreen calibration should be executed. You may use the combination keystrokes listed immediately below if the screen is far out of calibration. The touchscreen calibration screen can be selected in the control panel found by pressing the Start button. Select the control panel’s Stylus icon and its Calibration tab. Press the Recalibrate button and follow the instructions: Simply follow the on-screen instructions and press and release on the ’+’ graphic each time it is displayed.

1.2.13 Using Combination Keystrokes

Navigation and operation of the terminal can take place without a mouse or touch screen. Developers will most likely want to at least attach an external mouse. If that is not desired, the most commonly needed key commands are listed in the LANpoint Power X Installation and Maintenance Manual or the LANpoint Power X Quick Start Guide.

1.3 Operation and Configuration

The topics in this section include:

- the Microsoft Windows CE version 6.0 operating system
- the initial power-up sequence
1.3.1 **Windows CE Operating System**

The terminals run the Microsoft Windows CE version 6.0 operating system. The operating system configuration includes support for the following hardware devices, among others:

- Ethernet controller
- SD/MMC Card for expanding data and program storage space.
- XScale PXA310 processor

1.3.2 **Initial Power-up Sequence**

The fixed-mount terminal does not have a power switch. The terminal receives power from an external power source via an adapter when you connect to the DC power input connector.

The `Power LED` on the front of the terminal immediately illuminates green when the terminal is powered-up.

The system boots up and performs hardware verification. The speaker emits the startup tone and the LANpoint Power X desktop will be displayed when done.

1.3.3 **Configure TCP/IP Parameters, or Disable Connection**

To change the IP address, subnet mask, and gateway, use the Control Panel in Windows.

Set the IP address and other Ethernet parameters (WINS, DNS, DHCP, etc.) or disable a wired or wireless Ethernet connection using the *Network and Dial-up Connections* applet. The dialog is the same or very similar to a Windows desktop computer’s network/dialup dialog.

For additional parameters associated with wireless Ethernet (SSID, WEP, etc.), there are additional dialogs described in the *LANpoint Power X Installation and Maintenance Manual*.

1.3.4 **File Transfer Methods**

To transfer files from a development computer or any computer to the terminal, the following methods can be used:

- Use Build/Deploy Solution option in Microsoft Visual Studio.NET 2005 and later.
- Use the Remote Manager to upload the application using a Web interface. (see section 3.6, *Remote Manager Utility*)
- Define shared network folders (see section 1.3.5, *Accessing Shared Directories*).
- Use Microsoft *ActiveSync* (see Section 2.4, *ActiveSync Reference*) to transfer the desired files
- Use a USB thumb drive. The drive shows-up as a folder named "Hard Drive" in Windows Explorer.
• FTP

For permanent non-volatile or one-time file transfer, the SD card slot can be utilized. This slot allows hot-swapping of the cards. The slot needs to be accessible, so covers and/or mountings must be removed or partially disassembled in most cases. The card shows up as a folder named "Storage Card" in Windows Explorer.

1.3.5 Accessing Shared Directories

Accessing a shared directory on a PC on the Ethernet network allows you to copy files between the terminal and a PC, as well as to update files in the shared directory on the PC, as if that shared directory were stored locally on the terminal.

1. It is a good idea to assign each terminal a unique name on your network. If this has already been done, skip to step 7.
2. Click on the Start button and select Control Panel.
3. Double-click the System icon to open the System Properties dialog.
4. Click on the Device Name tab. Enter a unique identifier in the Device name: field. Enter a description in the Device description: field if you wish.
5. Press the OK button in the System Properties dialog.
6. Restart the terminal - either disconnect the power, press the hard-restart button through the back of the case, or press the Start button icon, select Programs and Software Reboot.
7. In the Command Prompt window on the terminal (Start button, Programs, Command Prompt), enter the command:

```
net use Test \ComputerName\SharedDirectoryName
```

where ComputerName is the computer with which you want to share data. The ComputerName and SharedDirectoryName is defined on the PC beforehand; these values and the methods to set them (and look them up) are operating system dependent. See your system administrator if either or both are unknown to you.

8. A dialog asking for a User name and password may open. Enter the user name and password for the host computer. Some computers may require an entry in the Domain field. Leave this field empty if your computer does not require an entry.
9. Click the OK button.
10. A folder named Test now exists on the terminal under the Network folder. Below that folder is displayed the contents of the host computer’s shared folder.
11. Use Windows Explorer or any other program to copy, modify, delete, and so on, as if the files were all stored locally.

1.3.6 Instructions to Create a Shortcut in Windows CE

1. Select (highlight) the executable program.
2. Copy the file using the Edit menu Copy item.
3. Go to the directory location where you want to place the shortcut and select *Edit* and select the *Paste Shortcut* item.

**Note:** To have an application automatically activate at startup, you can either:

1) create a shortcut to the application in a \Flash\Startup folder (this method is provided so as to be backward compatible to the LANpoint 7 and earlier terminals), or
2) simply place it in the \Windows\Startup as you would with a desktop PC.

**WARNING:** UNDER NO CIRCUMSTANCES SHOULD THE OPTIONAL LOCK-DOWN BROWSER UTILITY be placed in the \Windows\Startup folder - there is no way to stop it from launching at startup if this is done. This means that no future configuration, software change, utility program, troubleshooting, or any other software will ever be able to be used without returning the terminal to the factory.

**WARNING:** Be careful not to duplicate startup programs, as both methods are built into the units.

4. To add parameters to the end of the shortcut command line, right click on the *shortcut* and select *Properties*.

5. Select the *Shortcut* tab. The *Target* field contains the command line for the program file which includes the path to the directory location and the name of the program.

6. Type the *parameters* at the end of the shortcut command line and click the *OK* button.
Chapter 2  Developing Custom Programs

This chapter provides instructions for installing the development software and contains information on developing custom programs for the LANpoint Power X terminal. Sample programs are provided on the Intelligent Instrumentation Web site www.lanpoint.com

2.1 Selecting the Correct Development Tools for your Environment

Developers have several options when making a decision regarding which development tools to use during Windows CE 6.0 application development including Visual Studio 2005.

In general, the following table summarizes the tools that can be used with Windows CE v6.0 which includes .NET Compact Framework v3.5.

2.1.1 Visual Studio 2005

These tools are suitable for native and managed C++ development, C#, and Visual Basic. The end product of the development process is an executable (.exe) which can be transferred to the unit and executed. They can also build a cabinet (.cab) file that can be transferred to the unit for easy multi-terminal deployments - users can double click on the .cab file and installation occurs as an automated process, similar as to what users see when they install new programs on a desktop PC.

2.2 Development Overview

In general, the development process using either development tool set is the same:

• The developer installs the appropriate Microsoft Visual Studio tools
• The developer installs the Intelligent Instrumentation SDK. The SDK must be installed AFTER any development tools.
• The developer may use the Configuration Manager found in Visual Studio to establish communications between the development PC and the target device for downloading code, setting breakpoints in the code, examination of current variables, etc.
• The developer writes a new application and debugs it on the target platform to ensure that it performs as expected.
• The developer distributes the application using one of the methods below.

2.2.1 Application & Data Transfer Methods

The application can be distributed to the terminal via one of the following methods:
Developing Custom Programs

- Use Build/Deploy Solution in Microsoft Visual Studio, to deploy copies of the application.
- Use the Remote Manager to upload the application using a Web interface. (see section 3.6 Remote Manager Utility)
- Define shared network folders (see section 1.3.5 Accessing Shared Directories) and copy the application from the shared folder to the target device.
- Use Microsoft ActiveSync to transfer the desired files. See Section 2.4 ActiveSync Reference
- Use a removable media such as an SD card or USB thumb drive.
- FTP

2.3 Development with Microsoft Visual Studio 2005

A simple example using the Visual C++ programming language is shown below.

2.3.1 Installing Microsoft Visual Studio.NET 2005

The system requirements for supporting the version of Microsoft Visual Studio being used should be reviewed in order to ensure that the development PC meets the minimum acceptable requirement.

Insert the installation disk for Microsoft Visual Studio into the CD ROM drive on the development computer. Follow the installation instructions provided by Microsoft Corporation to install.

**Note:** Some beta developers have noted that if they have a version of Visual Studio.NET installed that is later than Visual Studio.NET 2005 and then install Visual Studio.NET 2005, that they have experienced errors, especially later when loading the LANpoint Power X SDK. To solve the problem, do NOT install the documentation - this is a selection in the "custom" options of Visual Studio.NET 2005.

If the installation program does not automatically start, open Windows Explorer and navigate to the CD-ROM drive icon and right-click on the drive icon. In the pop-up menu, select AutoPlay. The installation default settings are recommended.

2.3.2 Installing the LANpoint SDK

- Install the LANpoint Power X SDK (Software Development Kit) for Microsoft Visual Studio after completing the installation of the Microsoft Visual Studio software on the development computer.

**Note:** Use the same installation directory that you chose for any other SDK tools installed with the Microsoft Visual Studio for the installation of the LANPointPowerX SDK which can be downloaded at http://www.lanpoint.com/downloads.html
The SDK is included in the downloadable LANpoint Power X SDK, utilities and drivers zip file that contains the files listed in the table below.

<table>
<thead>
<tr>
<th>Folder</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveSync_USB_Host_Driver</td>
<td>Folder containing the USB driver for use on the development PC when downloading an application via ActiveSync</td>
</tr>
<tr>
<td>LANpointPowerX.msi</td>
<td>This is the LANpoint Power X SDK for WinCE 6.0 to use with Microsoft development environments.</td>
</tr>
<tr>
<td>cerhost.exe</td>
<td>This is the remote display viewer for the PC host and/or development PC. See section 3.3.1 Using the Remote Display Utility for usage.</td>
</tr>
</tbody>
</table>

These files, including the SDK is available as a CD and can be ordered with the purchase of a terminal, or can be downloaded from www.lanpoint.com/downloads.html. Double-click the appropriate .msi file to perform the SDK installation.

If you currently have a version of Microsoft Visual Studio installed on your development machine you may receive a message telling you download some utilities from Microsoft. Follow those instructions if you wish.

The SDK setup dialog screens below will be shown.
FIGURE 2.2 SDK Setup dialogs

Note: Some beta developers have noted that if they have a version of Visual Studio.NET installed that is later than Visual Studio.NET 2005 and then install Visual Studio.NET 2005, that they have experienced errors when loading the LANPoint Power X SDK. To solve the problem, do NOT install the Visual Studio.NET 2005 documentation - this is a selection in the "custom" options of Visual Studio.NET 2005.
2.3.3 Sample C++ development session

The section below shows the major steps to develop a simple application and download it to the terminal directly from Visual Studio, one of the several methods that can be used to download programs to the terminal. This example is specific to Microsoft Visual Studio.NET 2005.
Back in the main Visual Studio 2005 window, press the New Project button in the Projects folder in order to begin the development of a custom application program. A New Project dialog will open (shown as an insert superimposed on screen shot below). Select the language you wish to develop with in the Project Types: window (C++ in this example) and pick the desired template. In this example we selected Win32 Smart Device Project.
Give the application a name (HelloWorld in this example) in the Name: window, and specify where on the development PC to store the project in the Location: window. Press the OK button.

Press the Next button.

The LANpointPowerX_SDK platform will be displayed in the left Installed SDKs window. Highlight it and press the right-arrow button to move it to the Selected SDKs window. Then press the Next button and back in
the main Visual Studio 2005 window, select the *Build* menu item. Set the *Application Settings* properties in the *Project Settings* window as shown:

Press the *Finish* button now that the demo program is successfully built.
In this example, the basic skeleton Demo program was created by the Project Wizard.

Fine - now you have a demo program ready to go. How do you get it to the terminal to test or deploy it? There are several ways as mentioned in the File Transfer Methods paragraph in Chapter 1. The easiest method is via USB thumb drive, but some applications are better loaded ActiveSync.

2.4 ActiveSync Reference

Microsoft ActiveSync is a popular transfer mechanism for developers deploying applications to Windows CE devices. This reference section outlines the steps for using ActiveSync with the LANpoint PowerX terminal.
Microsoft ActiveSync can be downloaded from www.microsoft.com. It will be installed on the development PC.

**NOTE:** The features and use of ActiveSync are beyond the scope of this document and users are encouraged to read the documentation that comes as part of the Microsoft ActiveSync product.

**Note:** Microsoft ActiveSync is available for download from the Microsoft website and is free of charge. In addition to the Microsoft ActiveSync software, users will need an ActiveSync cable from the OTG port on the terminal to a USB port on the development computer. This cable can be ordered from Intelligent Instrumentation - Part #LPA076. This cable can also be ordered from Philmore (LKG Industries, Rockford, IL, USA) - Part #70-8028

### 2.4.1 Terminal Setup - ActiveSync

#### 2.4.1.1 Preparing the terminal for ActiveSync

There are two things that must be configured when using ActiveSync - 1) the terminal must be changed from USB Host to USB Device via the control panel, and 2) the USB ports must be rendered inoperative temporarily while ActiveSync uses the OTG port; this is done via hardware jumper changes. **NOTE THAT THE INTEGRAL KEYBOARD IS ONE OF THE USB DEVICES THAT WILL BE INOPERATIVE.**

IT IS VERY IMPORTANT THAT YOU USE ONE OF THE METHODS BELOW (remote display, touchscreen, bluetooth) SO YOU CAN RESET THE USB PORTS AFTER USING ACTIVESYNC. REMEMBER, THE KEYBOARD WILL BE INACTIVATED UNTIL THE USB PORTS ARE RESTORED.
That said, there IS a failsafe way to reset the USB ports involving moving a jumper on the main board and rebooting the terminal twice. Instructions to do that are given in section 2.4.1.4 Restore Keyboard and USB functionality on the terminal.

**NOTE:** Configuring the terminal as a USB Device, the terminal’s USB ports and integral keyboard are inoperative. You may still access the terminal by:
- Remote Display - this is preferred method of many because it is very easy to setup and it is also a very powerful testing and debugging tool. The development computer shows the terminal’s desktop display and allows use of the development computer’s keyboard and mouse device remotely. The terminal must be running the remote display utility, cerdisp. See section 3.3 for instructions.
- Touchscreen, if the terminal is so equipped, and SIP (Soft Input Panel in bottom right corner of system tray)
- Bluetooth keyboard and mouse if terminal is so equipped

If you choose the Bluetooth method, it must be setup before use - see Bluetooth accessory information in section 1.2 of this manual for setup/pairing instructions. The touchscreen is always active on those terminals with the touchscreen option.

Go to Control Panel and click on the USB Functionality Chooser icon.

![USB Functionality Chooser](image)

and then select USB 2.0 Device and press the Save button.
IMPORTANT: it will take time to write the changes to registry and you will have to reboot the unit for changes to take effect, so do not hard reboot right away. Either

1) wait a full 60 seconds and hard-boot - or -

2) do a soft-reboot (note, later, when changing from 2.0 device back to 1.1 host, the keyboard will not be active, so shift-F8 will not soft-reboot).

Now, disconnect power to the terminal. Activate the OTG port by moving the 3 OTG/USB jumpers at the P16 jumper block as shown below. You will need to remove the rear cover to do this. Then re-connect power.

When the terminal has rebooted, use the ActiveSync cable to connect the OTG port of the terminal to the development PC.
2.4.1.2 New Hardware Wizard

If this terminal is an already known device (you’ve attached to it before), you will not see the New Hardware Wizard screens - the next 7 screens on the PC. If it is truly new hardware, make the choices as shown on these screens:

The file name to enter in the dialog shown on the right is wceusbsbsh.sys. It can be found on the SDK CD or downloaded from www.lanpoint.com.
You may see the warning as shown on the right below. This is normal.

You are now finished with the New Hardware Wizard:
2.4.1.3 Deploying an Application to a terminal via ActiveSync

Now that the terminal has been attached to the development PC, you will be asked if you want to set up a partnership. Select *No* and the development PC will connect to the terminal:
In Microsoft Visual Studio, select the project to deploy. In this example, it is the sample project built above in Section 2.3:
The following window will open. Specify where the application will be stored on the terminal. In this case in the folder \Flash\Tests.
In the Build menu where you had the successful build in section 2.3 above, select Deploy Solution to transfer the application to the terminal:

The application will now be on the terminal where you can test it:

2.4.1.4 Restore Keyboard and USB functionality on the terminal

Follow the same steps performed in Section 2.4.1.1 Preparing the terminal for ActiveSync above, resetting USB Functionality from USB2.0 Device to USB1.1 Host, and returning the three OTG jumpers on P16 to their original settings. This will return the terminal to normal operational status.

If you did not take one of the steps necessary to restore the keyboard and USB functionality as listed in the NOTE box in section 2.4.1.1 Preparing the terminal for ActiveSync, do not despair. The failsafe method for restoring the terminal to USB1.1 Host is to remove power from the terminal and move the jumper to connect
pins 13 and 14 of the P16 jumper block as shown in the following graphic. You can do this at the same time you move the three OTG jumpers to the USB setting. Then re-power the terminal. It will reboot twice and after the second reboot, the USB ports - and integral keyboard - will be activated. Test that the keyboard is active, remove power and move the pin 13-14 jumper block back to it’s stored position on only one of the pins.
2.5 Uninstalling the SDK

To uninstall the LANpoint SDK, click on the *Change or Remove Programs* icon in the Control panel, select *LANpointPowerX SDK*, and press the *Remove* button. Then answer *Yes* to the verification prompt.

![FIGURE 2.3 - Un-Install the SDK](image-url)
Chapter 3  Utility Programs

3.1  CEPad Text Editor Utility

The terminal includes a text editor application called CEPad. CEPad operates similarly to Windows Notepad, enabling you to edit text files on the terminal.

The basic File menu allows you to create a New file, Open an existing file, Save the file, use Save As to save an existing file with a new filename, and Exit the application.

3.2  Wedge Utility

The WedgeCE utility allows the wedging of COM port data into the keyboard input stream of the active application. WedgeCE simplifies the process of writing an application that handles bar code input data, or data streams from RS-232 devices attached to the terminal, by making the data from those sources appear as though it was typed on the keyboard.

NOTE: USB and Blue Tooth barcode reading devices have - for the most part - "HID" (Human Interface Device) interfaces that mimic keyboards so as to insert data into the keyboard input stream automatically; hence they do not rely on WedgeCE

This application can wedge data from the COM5 serial port to the keyboard input stream. The WedgeCE.exe application is located in the folder \\Flash.

WedgeCE uses COM5 default communications parameters. To use other other parameters, create a shortcut (see section 1.3.6, Instructions to Create a Shortcut in Windows CE) to WedgeCE and include the appropriate command line parameters. Starting WedgeCE with no parameters will default to the default parameters shown in section 3.2.2 below.

TIP: To have the shortcut start automatically, place the shortcut file in the \\Flash\Startup folder or the \Windows\Startup folder.

3.2.1  Configuring WedgeCE Parameters

Application Command Line Format

Parameters must be entered in the order shown. The format of the application command line is:

WedgeCE <baud> <parity> <data bits> <stop bits> <delay> <prefix>

where:

- <baud>  Baud rate
- <parity> Parity
- <data bits> Data bits
- <stop bits> Stop bits
- <delay> Delay
- <prefix> Prefix
<baud> is the *baud rate* for the serial port.

<parity> indicates the *parity* for the serial port. Use n (no parity), o (odd parity) or e (even parity).

<data bits> is the number of *data bits* for serial communications. For eight data bits, use a value of 8.

<stop bits> is the number of *stop bits* for serial communications. Use 1, 2, or 15. A value of 15 indicates the use of 1.5 stop bits.

<delay> WedgeCE uses a *delay* parameter to determine when input from the serial port is complete and the data accumulated should be wedged into the keyboard input stream. A delay parameter of 50 ms is adequate for most bar code scanners. The amount of time required for serial input varies, depending on the type of device that is sending the serial data.

- If the value in the inter-character delay field is too small, data that should arrive in the keyboard buffer at one time may arrive in several packets containing a few characters.
- If the value in this field is too large, there may be an excessive delay after the end of the input before the wedging occurs.

<prefix> is a string which is prepended to the data each time a packet is wedged into the keyboard buffer. This parameter is useful in identifying the source of the input data when there are several data sources. The maximum length of the prefix is 32 characters.

**Example**

For instance, to use WedgeCE with a serial device that is connected to the COM port and has its communications parameters set to 2400 baud, even parity, 7 data bits, 2 stop bits, you wish to delay a little longer than normal between characters being retrieved by the port, say 100ms, because it is a slow device, and you wish to have "AUTOID:" prepended to the data that is retrieved from the device connected to the COM port, use:

```
WedgeCE 2400 e 7 2 100 AUTOID:
```

Input data will be wedged into the keyboard input stream as AUTOID:input data
3.2.2 WedgeCE Default Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>baud</td>
<td>9600</td>
</tr>
<tr>
<td>parity</td>
<td>n (none)</td>
</tr>
<tr>
<td>data bits</td>
<td>8</td>
</tr>
<tr>
<td>stop bits</td>
<td>1</td>
</tr>
<tr>
<td>inter-character delay</td>
<td>50 milliseconds (ms)</td>
</tr>
<tr>
<td>Prefix</td>
<td>None</td>
</tr>
</tbody>
</table>

3.3 Remote Display Utility

The LANpoint Power X allows the developer to access almost all of the functions of the operating system remotely from a standard PC with the Remote Display Utility. This utility allows remote manipulation of the terminal via the familiar Windows Graphical User Interface. This utility has two parts, a client portion (\Windows\cerdisp.exe) on the LANpoint Power X, and a host portion (CERHOST.EXE) on the PC.

3.3.1 Using the Remote Display Utility

1. First, copy and run the program CERHOST.EXE on a development PC connected to the network. CERHOST.EXE can be found on the SDK CD, or downloaded from the LANpoint Power X section of page www.lanpoint.com/downloads.html.

2. Next, press the Start button, select Run, and enter cerdisp -c. It can also be started using the Remote Manager utility - see Section 3.6 below for instructions. The LANpoint Power X Graphical User Interface will appear in the CERHOST window on the development PC. Click on File/Connect and a window similar to the following appears as shown in the left image below. Usually one or more terminal names will populate the Active target devices window. Click on the device name as shown in the middle...
Utility Programs

image and the IP address will be provided, or you can type-in the IP address in the lower box. Press OK
and you will see the display of the LANpoint Power X:

![LANpoint Power X display as shown on remote host PC](image.png)

You can now operate the terminal, with keyboard and mouse control, remotely from the PC host

### 3.4 Programmable Function Key Utility

The terminal has eight function keys, F1 - F8, located on the top row of the keyboard. The Programmable
Function Key (PFK) utility allows soft keys to be configured to either translate the keystrokes into text strings
fed to the keyboard buffer or to launch programs. Using combinations of the key with alt-, ctrl- or shift-, up to
32 functions can be programmed using only the 'F' keys. Most of the other keys can also be assigned
different definitions using the PFK utility.

There are two PFK programs that perform two different tasks:
- The `PFKConfig.exe` program is used to create new PFK definitions, edit existing PFK definitions, and
delete PFK definitions.
- The PFK program executes the PFK definitions.

#### 3.4.1 Creating a New Programmable Function Key File

The terminal may have one PFK definition file with one or more definitions in it, or multiple definition files
with one or more definitions in it, at the discretion of the developer. The PFKConfig dialog is used to create
the PFK definition file.

**Note:** Terminals shipped from the factory with a default PFK definition file (default.pfk). You may want to edit this file instead of creating a new file. Since only one copy of PFK.exe may be executed at a time, you must halt the copy
loaded at startup before using a different key definition file.
To create a new PFK definition file:

1. Launch the `PFKConfigCE.exe` application. The default location for this file is located in \Flash.
2. In the `PfkConfig - Untitled` dialog, click the **New** button.
3. The New Function key dialog opens.
   
   In the **Function** field, press the key or the key combination for the new PFK. A key combination consists of one or more modifier keys (Control, Alt, and Shift) in combination with almost any other key.

   **Note:** The Caps Lock, Print Screen, and Pause keys may not be used. There are some reserved function keys that cannot be used in a PFK file: Ctrl+Alt+Shift+X (which is used to abort PFK) and Ctrl+Alt+Shift+D (which is used to disable/enable PFK).

   The key or key combination appears in the Function field.

4. To have the PFK ***launch an application***, select the **Launch application** radio button and specify the application path and name in the **Output** field. Command line arguments can be added after inserting a semi-colon - like ***xxx.exe;+1*** where +1 is the command line parameter.

5. Otherwise, if the PFK hot key is to wedge a string into the keyboard stream, select the **Wedge into keyboard stream** radio button and type the text string (up to 256 characters long) into the **Output** field.

   **Note:** To include a literal \ character in the text, type \\ in the string. For example, to launch \Flash\Any File.exe when the hot key is pressed, type the text string \Flash\Any File.exe. Failure to do so may result in apparently erroneous messages, such as "unknown key" when trying to edit the .pfk file at a later date, or the graying-out of the "save" button when trying to save the .pfk file. You could alternately use the forward slash (/) character instead of \\ in path definitions.

6. Click the **OK** button. The New Function Key dialog closes and the new function key appears in the list in the `PfkConfig-Untitled` dialog.

7. To create another PFK, repeat steps 2 through 6.

8. Save the new PFK file by clicking the **Save As** button. Save under \Flash\... for permanent storage.

9. Click the **Exit** button.

10. Use the terminal’s keyboard or an external keyboard to test the operation of the PFKs, using the instructions in section 3.4.3, **Using a Programmed Key**.

### 3.4.2 Editing a Programmable Function Key File

1. Run the `PFKConfigCE` program.

2. In the `PFKConfigCE-Untitled` dialog, click the **Open** button and locate the PFK file to edit.
Utility Programs

3. Select the desired function key file from the list of functions shown and click the Edit button. The New Function Key dialog opens, displaying the contents of the selected PFK file.

4. Make the desired changes to the PFK file and click the OK button to close the New Function Key dialog.

5. Click the Save button in the PfkConfigCE dialog to save the changes.

6. Click the Exit button to close PFKConfigCE.

7. Test the changes.

3.4.3 Using a Programmed Key

To use programmed key(s):

1. If PFK.exe is already running (factory default), you will see its icon in the system tray and you can skip to step 8 below.

2. If PFK.exe is not currently running, create a shortcut (see section 1.3.6, Instructions to Create a Shortcut in Windows CE) to PFK.exe and right-click on it.

3. Select Properties from the menu.

4. In the Properties dialog, select the Shortcut tab.

5. In the Target field, enter the path and filename of the function key file to be tested.

   For example:
   "\Flash\PFK.EXE"   "\Flash\Folder containing your PFK file\File name.pfk"

6. Click the OK button to exit the dialog.

7. Double-click the shortcut. The PFK icon appears in the system tray. If you do not see this icon in the system tray, repeat steps 2-5 and verify the path is correct.

8. To see which PFKs are currently running on the PFK program, double-click the icon in the system tray. The Programmable Function Keys dialog opens, displaying a list of the PFKs in the file specified in the path in step 4.

   To close this dialog and have the program continue running, click the Hide button.

   To close the dialog and shut down this program, click the Exit button.

9. Press the key(s) to be tested. Verify that the key functions correctly by noting that the correct program was launched or that text shows-up in an active text window (to test the keys that output text, open a text editor such as CEPad.exe before pressing the key).

3.4.4 Deleting a Programmed Key

1. To delete a programmed key definition, run PfkConfigCE.exe.

2. In the PfkConfigCE dialog, click the Open button and select the .pfk configuration file.
3. Select the function key definition from the list and click the *Delete* button. The listed entry disappears.

**Note:** If you select the wrong function to delete and have not clicked the *Save* button, you can recover the configuration. Click the *Exit* button and a warning pop-up dialog opens asking *Save changes to YourFileName.pfk?* Click the *No* button to retain the previous settings and close the application.

4. Double-check before clicking the *Save* button

5. Click the *Exit* button, then stop/restart PFK.exe if it is running.

### 3.5 Time Synchronization

The synchronization of the LANpoint terminal’s clock with an external time standard is useful for all types of applications, especially labor tracking and time-and-attendance applications. The terminal supports clock synchronization via RFC 2030, also known as the SNTP (Simple Network Time Protocol) protocol.

You can find a list of U.S. Government and other time servers that support SNTP at http://www.boulder.nist.gov/timefreq/service/time-servers.html

#### 3.5.0.1 Time Synchronization Configuration Program

Time synchronization functionality is provided by a service running under the Windows CE 6.0 operating system. The properties which control it are stored in the registry. Intelligent Instrumentation has provided a configuration program to eliminate the need to edit the registry directly.

The configuration program is located in `\Flash\SNTPServiceConfig.exe`. Details follow.

1. Run `SNTPServiceConfig`. The `SNTPServiceConfig` dialog will open.

The *Time service status* will generally be "Running". As configured from the factory, it is not given a server with which to synchronize, so no operations will be performed. To configure the server for synchronization operation, use the dialog’s menu:
2. Click on the Service | Properties menu item to open the dialog shown below.

![Service Properties](image)

3. Set the appropriate server name(s) and/or server IP address(es). The Servers item in the dialog is simply an EDIT field to which you can add additional servers, one per line, or from which you can remove servers by simply deleting their corresponding lines.

4. Select the Update Time from Servers checkbox.

5. Enter a suitable Refresh Interval value of 300000 or greater. This value indicates how often the server(s) will be contacted to synchronize the local clock, in milliseconds. The default value in the application, 600000, corresponds to updating the synchronization every 10 minutes which is usually far too often. More common would be a value like 86400000 to synchronize once per day, or perhaps even larger (86400000 = 60000 milliseconds per minute x 60 minutes per hour x 24 hours per day).

6. Enter a suitable Max Correction value. This is the maximum number of milliseconds by which the local clock can be changed to match the server clock when the Trust Local Clock checkbox is marked. The default value, indicating that any amount of adjustment is reasonable, is usually the correct value to use. If the number of milliseconds to be adjusted is larger than Max Correction, the clocks will not be synchronized.

7. Enter the Recovery Interval after failure value. Following an unsuccessful attempt to reach the time server(s), this is the number of milliseconds to wait before retrying, with a minimum value of 300000. For wired Ethernet, the default 300000 is usually adequate. For wireless networks, this value may need to be adjusted.

8. Select the Trust Local Clock checkbox if you wish to have the local clock treated as at least marginally accurate. When this checkbox is unset and a synchronization operation occurs, the local clock will always be updated with the server's time, no matter what the amount of unsynchronization. When the checkbox is set, the local clock is presumed to be reasonably accurate and the time will only be updated to match the server time when the amount of unsynchronization is less significant than the Max Correction value.

9. Press the OK button to notify the time synchronization service that the new parameters should now be used and to exit the dialog.
3.5.0.1.1 Other pull-down menu items

Update: After configuration, the server must be sent an indication that the parameters have been changed. This occurs automatically when you select the OK button to accept the new configuration settings. If a setting is changed in some other way, via a registry editor, for example, the Server | Update menu item can be used to trigger the server to reload its settings from the registry and begin using them.

Immediate Sync: The application provides a means to force an immediate synchronization operation. This is accomplished by choosing the Service | Sync menu item. This sends a message to the service telling it to contact a time server and perform a synchronization operation. Note that this synchronization operation forces the local clock to be updated, regardless of the settings for Trust Local Clock and Max Correction. It also unschedules the next synchronization time and recalculates the next sync time based on the time of the forced sync.

Immediate Set: The application provides a means to force an immediate clock operation. This is accomplished by choosing the Service | Set menu item. This sends a message to the service telling it to contact a time server and perform a synchronization operation but, unlike the Sync operation, it does not force the local clock to be updated, but uses the Trust Local Clock and Max Correction settings to determine whether the local clock should be updated or not.

Start/Stop: The application can be used to control the service, starting or stopping it using the Service | Start and Service | Stop menu items. There is seldom any reason to do this, however.

3.6 Remote Manager Utility

The Remote Manager utility provides basic terminal management functionality, including access to the file system, application management, remote reboot capability, and security settings. The Remote Manager is a set of Web pages served-up by a Web server on the terminal, called WebDevice, to a browser on a remote computer attached to the network. WebDevice has server-side scripting capability that presents the content of the Remote Manager pages.

Note: WebDevice consumes resources and may affect the response time of other applications running on the terminal.

Note: The terminal’s IP address must be set before you can use Remote Manager. For information on setting the IP address see Chapter 1, Setup & System Configuration, section 1.3.3, Configure TCP/IP Parameters, or Disable Connection.

An administrator who knows the user names and passwords for the terminals can perform the following tasks remotely through a Web browser:

• Assign the terminal a unique name, a description, and location.
Utility Programs

• Perform file management on the terminal: uploading, downloading, copying, and deleting files, creating new directories and deleting existing ones.
• Managing applications running on the terminal, including viewing a list of applications currently executing, launching applications, and terminating applications.
• Rebooting the terminal.
• Controlling the security settings, user name and password, for the above capabilities.

3.6.1 Enabling WebDevice for Remote Management

If WebDevice is running on the terminal, you will see the WebDevice icon in the system tray. If not, run `\Flash\webdevice\WDCE.exe`.

Double-clicking on the WebDevice icon in the system tray causes the WebDevice window to open. The window contains four buttons that perform the following tasks:

<table>
<thead>
<tr>
<th>Button</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hide</td>
<td>Closes the dialog window, leaving WebDevice running.</td>
</tr>
<tr>
<td>Options</td>
<td>Opens a dialog displaying the port setting. The default port number is 80. Only one session of WebDevice can run on any given port.</td>
</tr>
<tr>
<td>Close</td>
<td>Closes WebDevice and removes the WebDevice icon from the system tray, disabling Remote Manager capabilities.</td>
</tr>
<tr>
<td>Clear</td>
<td>Clears the text in the WebDevice message window.</td>
</tr>
</tbody>
</table>

Password Protection

Two sections of the Remote Manager are not password protected: the Home page and Help. All of the other sections of the Remote Manager are password protected. Attempting to access any of the password-protected sections causes the browser to open a pop-up dialog asking for a user name and password. See section 3.6.2.2, Security Manager: Allowing Access to Folders for more information.

Help Links

The Remote Manager web site contains context-sensitive help. The help link is located in the upper right-hand corner of every page.

General information about the Remote Manager is presented in this manual. For specifics and additional information, refer to the Remote Manager Help pages.
3.6.2 Using the Remote Manager

To use the Remote Manager, run a Web browser on a computer and point the browser to the desired terminal by entering the terminal’s IP address in the browser’s URL window.

The home page shown in FIGURE 3.4, Remote Manager Home Page, is the first page shown. All other remote management function pages can be accessed from the home page.

![Remote Manager Home Page](image)

FIGURE 3.4 Remote Manager Home Page
Troubleshooting Tips

If a 404 page not found or a timeout error occurs:

1. Enter http://nnn.nnn.nnn.nnn in the URL window, where nnn.nnn.nnn.nnn is the terminal’s *IP address*.
   
   Some browsers require the leading http:// in the URL window.

2. Verify the WebDevice program is running by looking for the WebDevice icon in the system tray.

3. Verify the network connectivity by pinging the terminal from the browser computer.
   
   a. If the computer is running Windows, open a DOS or Command line window.
      
      b. Type ping nnn.nnn.nnn.nnn where nnn.nnn.nnn.nnn is the *IP address* of the terminal.

   If a time-out occurs, this indicates a network problem with the PC, the terminal, or the network connecting them. See section 1.3.3, *Configure TCP/IP Parameters, or Disable Connection* for more information.

4. If the IP address shown on the home page is not the IP address entered in the browser’s in step 1, it is because more than one Ethernet port is enabled on the LANpoint Power X - maybe both wired Ethernet and WiFi. This does not hamper the usage of the Remote Manager utility, but probably is not a good idea for most applications. Disable the unused ports in the control panel - see section 1.3.3, *Configure TCP/IP Parameters, or Disable Connection*

### 3.6.2.1 Management

The terminal information section of the home page displays the terminal’s IP address, name, location, and description. The management section provides links to four management sections: File Manager, Application Manager, System Manager, and Security Manager. Viewing or modifying any of these items requires a user name and password. Factory default user names and passwords are shown in the table below.

<table>
<thead>
<tr>
<th>Section</th>
<th>User Name</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terminal Information (view)</td>
<td>none (home page)</td>
<td>none (home page)</td>
</tr>
<tr>
<td>Terminal Information (modify)</td>
<td>admin</td>
<td>admin</td>
</tr>
<tr>
<td>File Manager (view)</td>
<td>user</td>
<td>user</td>
</tr>
<tr>
<td>File Manager (modify)</td>
<td>admin</td>
<td>admin</td>
</tr>
<tr>
<td>Application Manager</td>
<td>admin</td>
<td>admin</td>
</tr>
</tbody>
</table>

*TABLE 3.1 Default User Names and Passwords*
Most of the pages are self-explanatory. The Security Manager pages require additional descriptions as described below.

### 3.6.2.2 Security Manager: Allowing Access to Folders

The Security Manager provides a means for setting user names and passwords to allow access to various folders that contain the Remote Manager Web pages. Password protection applies at the folder level, not the individual Web page (file) level. Different user names and passwords can be assigned for each folder, restricting access to the web pages within.

To access the Security Manager:

1. Select **Security Manager** on the Remote Manager Home Page. A **Password** dialog will open.

<table>
<thead>
<tr>
<th>Section</th>
<th>User Name</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Manager</td>
<td>admin</td>
<td>admin</td>
</tr>
<tr>
<td>Security Manager</td>
<td>admin</td>
<td>admin</td>
</tr>
</tbody>
</table>

**TABLE 3.1 Default User Names and Passwords**

**Note:** It is highly recommended that the default user name and passwords be changed with the **Security Manager** as soon as possible.

![FIGURE 3.5 Network User Name and Password dialog](image)
2. Enter the user name *admin*, password *admin*, and click the *OK* button. The *Security Management main page* will open.

![Image of Security Management main page](https://via.placeholder.com/150)

**FIGURE 3.6 Security Management main page**

To modify security, click on the *Edit/Remove* link which opens the web page shown in **FIGURE 3.7, WebDevice Server System Configuration page**.

![Image of WebDevice Server System Configuration page](https://via.placeholder.com/150)

**FIGURE 3.7 WebDevice Server System Configuration page**

When you click the links *Add New Protected Folder* or *Edit*, the pop-up dialog shown in **FIGURE 3.8, WebDevice Server Protected Folder opens**.
If you no longer want a folder to be password protected, click Remove on the appropriate folder line. The line item is deleted from the list of protected folders. The folder remains on the terminal, though not protected.

![WebDevice Server Protected Folder](image)

**FIGURE 3.8 WebDevice Server Protected Folder**

### 3.6.2.2.1 Editing the User Name or Password for a Protected Folder

If you are editing an existing folder, the Folder to Protect field contains the path to the protected folder.

From the page shown in FIGURE 3.8, WebDevice Server Protected Folder, change the user name and/or password for the folder, then click the Save Settings button to save the settings and exit this page.

**Note:** Do not change the folder name in the Folder to Protect field.

### 3.6.2.2.2 Adding a New Password Protected Folder

When adding new security to a folder, enter the path to the folder, relative to the terminal’s `\flash\webdevice\hroot` folder, in the Folder to Protect field. For instance, to add security to a folder `\flash\webdevice\hroot\newfolder`, type `\newfolder` in the field. Then enter a user name and a password. Click the Save Settings button to save the settings. Close this page.

### 3.7 SerialSocket Utility

The SerialSocket utility allows a remote computer to access the devices connected to the serial port on the terminal. When this application is executed, it listens on several TCP/IP sockets. When a remote host connects to one of the sockets, the application opens the serial port and establishes a connection between the socket and the serial port. Any data sent to the socket is sent out the serial port. Data that is received through the serial port is sent to the host over the socket connection.
How it Operates

The SerialSocket utility is multi-threaded and creates one thread for the serial port on the terminal. The thread listens on the corresponding TCP/IP socket and handles the connection and opening of the serial port, processing data to and from the port/socket. The socket is numbered 3005.

The SerialSocket utility is located at `\Flash\SerialSocket.exe`. There are no command line parameters. When you run SerialSocket, the SerialSocket icon is displayed in the Windows CE system tray.

3.7.1 Port Configuration for SerialSocket Utility

The default parameter for the COM port is 9600 baud, no parity, 8 data bits, one stop bit, and no flow control.

3.7.2 Viewing the Status of the SerialSocket Utility

To view the status of the SerialSocket utility:

1. Double-click the SerialSocket icon in the system tray. The SerialSocket Status dialog opens.
   This dialog displays the status of the serial port on the terminal. The status is either Serial port open or Waiting for connect.
   • The Serial port open status indicates a remote host opened the socket corresponding to the serial port and the SerialSocket utility successfully opened the serial port.
   • The Waiting for connect status indicates a remote host has not opened the socket corresponding to the serial port.

2. To close the dialog while the utility continues to run, click the Hide button.
3. To end the SerialSocket utility, click the Terminate button.
4. To update the status of the serial port in the dialog, click the Refresh button.

3.7.3 Testing SerialSocket with HyperTerminal

You can use HyperTerminal or another similar program to open a socket connection to SerialSocket and send and receive data through the terminal’s COM5 port.

1. Use LCmdSet or the Control Panel to configure the COM port setting to match the serial device connected to the port as mentioned in 3.7.1, Port Configuration for SerialSocket Utility.

2. On a computer with TCP/IP connectivity to the terminal, run Hyper Terminal and configure a connection as described in section 3.7.1, Port Configuration for SerialSocket Utility, EXCEPT configure the Connect To dialog to TCP/IP (Winsock), enter the terminal’s IP address in the Host address field, and set the port number to the port number for the desired COM port (e.g., 3002 for COM2).

3. Strings typed into HyperTerminal should show on the device connected to the COM port and/or data entered through the COM port should show in the HyperTerminal window.
3.8 Lock-down / kiosk Web browser Utility

Every LANpoint Power X terminal comes with the standard Internet Explorer Version 6 for Windows CE browser. In addition, an optional "lock-down" browser - sometimes used in kiosk applications - can be ordered. The lock-down browser, iesimple.exe, has no URL address box, hides the toolbar and system tray, ignores function and special keystrokes like ctl-alt-del and alt-tab, and generally limits web page access to only those pages that are specifically included by the developer in the target application. Non-productive Web browsing is therefore eliminated. With the lock-down browser, an application developer can include web-based applications with complete control of the screens that are presented.

**WARNING:** UNDER NO CIRCUMSTANCES SHOULD THE OPTIONAL LOCK-DOWN BROWSER UTILITY be placed in the Windows\Startup folder - there is no way to stop it from launching at startup if this is done. This means that no future configuration, software change, utility program, troubleshooting, or any other software will ever be able to be used without returning the terminal to the factory.

**Setting the homepage for the lock-down browser**

To set the home page for iesimple, launch the standard Internet Explorer browser and set the home page using its toolbar.

**Launching the lock-down browser at boot-up**

---

**VERY VERY IMPORTANT!**

Do NOT launch the lockdown browser at startup by using the Windows\startup folder! There is no way to stop the launch of the browser, no way to exit the browser, AND NO WAY TO PERFORM ANY OTHER ACTION, INCLUDING SOFTWARE UPDATES, DEBUGGING, AND ADMINISTRATION TASKS. Instead, launch the lockdown browser by using the flash\startup folder. Intelligent Instrumentation Applications Engineering (support@lanpoint.com) can then be contacted for special instructions about how to stop the launch of the lockdown browser at startup.

The lock-down browser can be found at \flash\iesimple\iesimple.exe. Many times application developers will put a shortcut to this browser in the startup folder so that it has sole control of the application interface. With this in place, the lockdown browser launches at bootup.

**Note:** Future software updates and configuration of the terminal requires special knowledge when the lock-down browser is launched at startup. Contact support@lanpoint.com for bootup launching alternatives and/or instructions on how to stop the lock-down browser from launching at startup.

Launching the lockdown browser at startup eliminates any further configuration or modification of software on the terminal, since all other applications are unavailable.
3.9 Thin Client, Cloud Computing, Virtualization

The terminal can be run as a thin client or standalone with custom application software. Thin client and cloud computing architectures allow the same applications to run on diverse types of hardware, regardless of the operating system and processor selection. For organizations wanting more flexible deployment of applications and to more easily control management costs, thin client architecture and cloud computing architecture offers an important enhancement to the traditional client-server architecture that is based on PC’s, servers and mainframe computers.

Pre-loaded thin-client/cloud computing software includes several text based thin clients; cloud services, application hosting, and other virtualization technologies are available.

Text based clients include VT/ANSI/HP terminal emulation, IBM5250 terminal emulation, and IBM3270 terminal emulation and are produced by Connect, Inc.

3.9.1 Text-based Terminal Emulation

With the Connect text-based thin client running, the terminal appears to the server as a dumb terminal. All application execution, processing, and data storage occur on the server, midrange-computer, or mainframe computer as it would with a native "dumb" terminal. You can install the version you wish - emulating ASCII terminals such as VT100, VT220, and HP92 terminals, emulating IBM 5250 terminals, or emulating IBM3270 terminals.

You can install and run this client-side software at any time, in evaluation mode - it will have an extra "nuisance" message and will time-out every 30 minutes, but it is a full-featured install in every other respect. The assignment of an authorization number, keyed to the terminal’s Ethernet Address, removes these limitations for production use.

3.9.1.1 Text-Based Terminal Emulation Installation Instructions

Before you start, you may want to note the terminal's hardware address (MAC) address and contact apps_eng@lanpoint.com, or call Intelligent Instrumentation Application Engineering for an authorization code. You will need the Sales Order number from your purchase of the terminal emulation software, or the terminal emulation software serial number that is shown on the cover of the "Terminal Emulation QuickStart Guide" provided with each license.

1. Set terminal's IP address and install the Terminal Emulation Software
   - set the terminal's IP address according to instructions found in section 1.3.3, Configure TCP/IP Parameters, or Disable Connection.
   - run the program \flash\twinclient\install.exe. An install window with several buttons will be displayed. Click on the button to install the terminal emulation type of your choice, or press the corresponding Function key (F1 = VT/HP; F2 = 3270; F3 = 5250, F4 = remove emulation software, etc). After you verify that you wish to install the software, a command window will open where you will see the appropriate files copied as necessary. When done, a success message will be displayed in the install window (i.e. "5250 Emulation software installed").
reboot the terminal by re-powering or pressing the F6 (or clicking the Reboot button on the install screen) - you will see a screen that shows "TwinClient TN xx". This is the initial terminal emulation screen.

Note: you no longer have the toolbar and the desktop environment on the screen; see section 3.9.1.2, Text-Based Terminal Emulation Un-Install Instructions for exiting back to WindowsCE

2. Terminal Emulation Configuration - assign Host Computer IP addresses

- at the "TwinClient TN xx" screen, press shift-C on the terminal's keyboard. This takes you into the emulator's "Edit Functions" configuration menu.
- highlight "Edit Host List" and hit enter. You will see a default host name (this is a valid host belonging to the emulation software’s manufacturer, Connect, Inc. It works with whichever "flavor" of terminal emulation you choose. You can use this for demos if you wish - contact Intelligent Instrumentation’s Technical Support for a valid user ID and password) and the cursor will be positioned for you to enter your host computer's IP address. Enter this value and hit enter.
- At the prompt for Port (number), enter the port number (usually 23 for Telnet sessions) and press F3 to save the changes.
- arrow down to the "Run Emulator" selection and hit enter to exit the configuration menu.
- reboot the terminal for the IP addresses and other menu configuration items to take effect.
- When the "TwinClient TNxx" screen is shown, simply press the Enter key to attach to your host.

Note: These are minimal instructions. Full instructions can be found in the downloadable manual "855M523, LANpoint CE.NET & LANpoint7 QuickStart Guide" at the overview page for the terminal being used, at www.lanpoint.com. That manual goes into much more detail with instructions to use DNS, DHCP, multiple (failover) hosts, and advanced configuration utilities.

3.9.1.2 Text-Based Terminal Emulation Un-Install Instructions

In the configuration menu (see step 2 immediately above), scroll down to the "exit" selection. This will exit you to the Windows CE operating system's desktop. Then run the install program again(\Flash\twinclient\install.exe).

Choose the "Remove Terminal Emulation" (F4 key) selection. You will see a Command window open briefly and some files being deleted, as the software is being un-installed. When done, reboot the terminal.
Chapter 4  LANpoint Power X APIs

This chapter contains information about the APIs (Application Program Interfaces) for the LANpoint Power X terminals. APIs are provided for the languages C++, C# and Visual Basic, but not all APIs are available for all languages. Microsoft Visual Studio includes many languages, including C++, C# and Visual Basic.

Function calls are shown in this chapter, as well as error codes. Download sample code for the APIs from Intelligent Instrumentation’s web site www.lanpoint.com at the overview page for the terminal being used.

4.1 Overview

Two APIs are provided: Digital I/O API and the Reboot API. Each API includes function calls for C++. Most functions are also available for C# and Visual Basic, though some cannot be provided for various reasons.

Digital I/O API

The digital I/O API enables the configuration, monitoring, and control of optional digital inputs and outputs - eight digital input channels and eight output channels.

Reboot API

This API enables you to reboot the terminal via the application. Note that for simplicity’s sake, this single call routine is NOT in a separate API file - it is included in the Digital I/O API.

4.1.1 API Header Files and Import Libraries

The LANpoint Power X SDK installed in Chapter 2, Developing Custom Programs, contains the header files for each of these APIs. Include the header file for each API in your program and link the appropriate import library with your program when using the eMbedded Visual C++ function calls. There are three required files: LP4dio.h, LP4dioerr.h, and LP8_dio.lib. Some example Microsoft Visual Studio screen shots of their usage is shown immediately below, followed by descriptions of the supported Digital I/O functions.
4.2 Digital I/O API

This digital I/O API enables the configuration, monitoring, and control of digital inputs and outputs, which are optional.
4.2.1 Input and Output Channels

The digital I/O option provides eight digital input channels and eight TTL output channels. Digital input channels can be used as event counters for signals with frequencies up to 50 Hz. The valid channels for inputs and digital outputs are shown below.

4.2.1.1 Input Channels

The input channels can be configured to:

- Return the current state of the input.
- Count low-to-high transitions of the input signals.

<table>
<thead>
<tr>
<th>Number of Input Channels</th>
<th>Valid Channel Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>0, 1, 2, 3, 4, 5, 6, 7</td>
</tr>
</tbody>
</table>

4.2.1.2 Output Channels

The output channels are set through software to the desired state:

- Either continuously on or continuously off.
- Configured to pulse the channel closed for a short, programmable period of time. This type of setting is useful for an access control application to temporarily unlock a door, and for applications requiring an indicator light or buzzer for a momentary period of time.

<table>
<thead>
<tr>
<th>Number of Output Channels</th>
<th>Valid Channel Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>0, 1, 2, 3, 4, 5, 6, 7</td>
</tr>
</tbody>
</table>

4.2.2 Required Files

The programming functions are defined in the header file LP4dio.h (LP4dio.cs for C# and LP4dio.vb for Visual Basic), which contains data constant definitions and API function prototypes. The library file LP8_dio.lib provides linkage between the C/C++ application program and the API functions.

4.2.3 Return Values

A return value of NO_ERROR_LPCE indicates the function completed successfully. A non-zero return value indicates a failure of the requested functionality. Different non-zero numbers indicate specific errors. Numbers greater than zero represent fatal errors and numbers less than zero represent warnings. Error descriptions are located in Section 4.2.6, Digital I/O Error Codes.

4.2.3.1 When Errors Occur

Errors indicate the hardware failed to respond to a given request.
If a critical error occurs:
• Examine the source code to determine if function parameters are causing the error.
• Verify the Digital I/O hardware is properly installed.
• Contact technical support.

4.2.4 Digital I/O Input Functions

There are two output functions and six input functions described below. Each includes the name of the function, a brief description, the parameter list, and a return value.

The digital input functions are:

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Function Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LpCEReadBit</td>
<td>Reads the state of one input bit</td>
</tr>
<tr>
<td>LpCEReadByte</td>
<td>Reads the state of all eight input bits</td>
</tr>
<tr>
<td>LpCEConfigureInputBit</td>
<td>Configures the bit for a requested mode</td>
</tr>
<tr>
<td>LpCEGetInputBitMode</td>
<td>Queries the system for the current configuration mode of a bit</td>
</tr>
<tr>
<td>LpCEStoreConfig</td>
<td>Stores the present input modes and present output states of all output channels to EEPROM</td>
</tr>
<tr>
<td>LpCERetrieveConfig</td>
<td>Reads the EEPROM and configures the input modes and output states using the most recent settings stored to the EEPROM</td>
</tr>
<tr>
<td>LpCEReadCounter</td>
<td>Returns data from the requested counter bit and optionally resets counts to zero after returning data</td>
</tr>
</tbody>
</table>

4.2.4.1 Return the State of a Single Channel

The current digital state (high, value 1, or low, value 0) of any input channel bit can be read by this function. The digital state of a bit is valid, regardless of the configuration - it applies even if the bit is configured as a counter. Regardless of whether the bit was previously configured as BIT_DIGITAL or BIT_COUNTER, the data value reflects the voltage at the digital input pin.

To obtain the count of transitions when a channel is configured as BIT_COUNTER, use the LpCEReadCounter(...) function instead.
4.2.4.2 Return the Current State of All Input Channels

The lower eight bits of data return a digital representation of the current states of all eight input channels. The digital states of all bits are valid, regardless of the configuration. This applies even if the bit is configured as a counter, the current state of the input, on or off, is returned.

To obtain the count of transitions when a channel is configured as BIT_COUNTER, use the LpCEReadCounter(...) function instead.

Function Parameters and Return Status Values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*data</td>
<td>Returned data value is 0-255.</td>
</tr>
</tbody>
</table>
Function call in eMbedded Visual C++

```c
int LpCEReadByte(int *data);
```

Function call in C#

```c
int LpCEReadByte ( ref int data );
```

Function call in Visual Basic

```c
LpCEReadByte (ByRef data As Integer) As Integer
```

4.2.4.3 Configure the Mode of an Input Channel

One of three modes can be assigned to each input channel (bit) - digital (on/off), trigger, or counter.

Function Parameters and Return Status Values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bit</code></td>
<td>0 to 7 bit position in digital input byte. Configure this bit by using one of the three modes.</td>
</tr>
<tr>
<td><code>mode</code></td>
<td>Normal input query i.e., On (1) or Off (0). See the example for LpCEReadBit.</td>
</tr>
<tr>
<td>BIT_DIGITAL</td>
<td><code>bit</code> counts low to high transitions. See the example for LpCEReadCounter.</td>
</tr>
</tbody>
</table>

Return value error code:

- 0   NO_ERROR_LPCE
- 40 ERROR_INVALID_READ_NULL_BYTE
- 41 ERROR_W_BYTE_1
- 42 ERROR_W_BYTE_2

Command successfully completed. `data` parameter is a null pointer.

Critical error: Command byte write timeout.

Critical error: Data byte read timeout.

Parameter `bit` must be between 0 and 7. One of three `modes` are acceptable.

Function call in eMbedded Visual C++

```c
int LpCEConfigureInputBit(int bit, int mode)
```

Function call in C#

```c
int LpCEConfigureInputBit(int bit, LP4DIOMode mode);
```

Function call in Visual Basic
LpCEConfigureInputBit
(ByVal bit As Integer, ByVal mode As LP4DIOMode) As Integer

4.2.4.4 Obtain the Current Configuration Mode of a Channel

To obtain the current configuration mode of an input channel, use this call to query the system. Each channel operates in one mode at any given time. Other threads and processes that use this API may have affected the mode.

The factory-default power-up setting for all channels is BIT_DIGITAL mode. Subsequent power-up settings may be different, if modified by the LpCEStoreConfig function.

Function Parameters and Return Status Values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit</td>
<td>Returns the last configured mode. mode returns BIT_DIGITAL, BIT_TRIGGER, or BIT_COUNTER depending on the most recent configuration calls.</td>
</tr>
<tr>
<td>*mode</td>
<td>bit to return mode data.</td>
</tr>
</tbody>
</table>

Return value error code:
- 0   NO_ERROR_LPCE
- 30 ERROR_INVALID_READ_BIT
- 31 ERROR_INVALID_READ_NULL_BIT

Function call in eMbedded Visual C++

```
int LpCEGetInputBitMode(int bit, int *mode);
```

Function call in C#

```
int LpCEGetInputBitMode(int bit, ref LP4DIOMode mode);
```

Function call in Visual Basic

```
LpCEGetInputBitMode Lib (ByVal bit As Integer, ByRef mode As LP4DIOMode) As Integer
```

4.2.4.5 Store Input Modes and Output States

Execute LpCEStoreConfig to store the present input modes and present output states of all output bits to the EEPROM on the digital I/O board. When the system reboots, the digital I/O sub-system reads and configures the input modes and output states from those stored values.

Function Parameters and Return Status Values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit</td>
<td>Parameter bit must be between 0 and 7.</td>
</tr>
<tr>
<td>*mode</td>
<td>Data parameter is a null pointer.</td>
</tr>
</tbody>
</table>
4.2.4.6 Retrieve Input Modes and Output States

The digital I/O sub-system reads the EEPROM on the digital I/O board and configures the input modes and output states using the most recent settings stored by the LpCEStoreConfig function. Use LpCERetrieveConfig to return the values stored.

Function Parameters and Return Status Values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>NO_ERROR_LPCE</td>
</tr>
<tr>
<td>90</td>
<td>ERROR_SC_W1</td>
</tr>
<tr>
<td>91</td>
<td>ERROR_SC_W2</td>
</tr>
<tr>
<td>92</td>
<td>ERROR_SC_R1</td>
</tr>
<tr>
<td>93</td>
<td>ERROR_EEPROM_WRITE_PROTECTED</td>
</tr>
<tr>
<td>100</td>
<td>ERROR_RC_W1</td>
</tr>
<tr>
<td>101</td>
<td>ERROR_RC_W2</td>
</tr>
<tr>
<td>102</td>
<td>ERROR_RC_R1</td>
</tr>
<tr>
<td>103</td>
<td>ERROR_RC_TRIGS</td>
</tr>
</tbody>
</table>
4.2.4.7 Return Data from a Counter Bit

This call returns the data from the requested counter bit and optionally resets counts to zero after returning the data. Configure the bit as BIT_COUNTER before using this function.

**Function Parameters and Return Status Values**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit</td>
<td>bit to read, 0 thru 7. Return data.</td>
</tr>
<tr>
<td>*data</td>
<td>1 resets the counter to zero after read. 0 leaves the counter value unchanged after this read operation.</td>
</tr>
<tr>
<td>reset</td>
<td></td>
</tr>
</tbody>
</table>

Return value error code:

- 0 NO_ERROR_LPCE  Command successfully completed.
- 60 ERROR_INVALID_CONT_BIT  Parameter bit must be between 0 and 7.
- 61 ERROR_INVALID_CONT_NULL  Parameter data must not be NULL.
- 62 ERROR_INVALID_CONT_MODE  bit property must be properly configured before read as a counter.
- 63 ERROR_INVALID_CONT_OVER  Counts rolled over. If input is at the maximum rate of 50 Hz, overrun will occur in 2.6 years.
- 64 ERROR_INVALID_CONT_RESET  Parameter reset must be 0 or 1.

**Function call in eMbedded Visual C++**

```c++
int LpCEReadCounter(int bit, long *data, int reset);
```

**Function call in C#**

```csharp
int LpCEReadCounter(int bit, ref int data, bool reset);
```

**Function call in Visual Basic**

```vbnet
LpCEReadCounter
(ByVal bit As Integer, ByRef data As Integer, ByVal reset As Boolean) As Integer
```

4.2.5 Digital I/O Output Functions

There are two output functions that interact with the output hardware, determining whether the channel(s) are open (normal state) or closed (activated).

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Function Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LpCEWriteBit</td>
<td>Controls one channel’s output state.</td>
</tr>
</tbody>
</table>
4.2.5.1 Control a Single Output Channel

This function provides control over one output channel at a time. Specify the bit (from bit 0 thru bit 7) to activate the corresponding channel. The moment parameter specifies the amount of time the channel is activated.

Function Parameters and Return Status Values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit</td>
<td>0 to 7 bit position in digital output byte.</td>
</tr>
<tr>
<td>moment</td>
<td>0 deactivates the bit (opens circuit). 255 activates the bit (closes circuit). The output state of the bit remains until you make another call to LpCEWriteBit or LpCEWriteByte. Values 1 thru 254 close the circuit for 100 to 25400 ms before the circuit is again opened.</td>
</tr>
</tbody>
</table>

Return value error code:
- 0   NO_ERROR_LPCE
- 10 ERROR_INVALID_WRITE_BIT
- 11 ERROR_INVALID_WRITE_MOMENT
- 12 ERROR_W_BIT_1
- 13 ERROR_W_BIT_2

For instance,

In the function LpCEWriteBit(2, 0):

The channel associated with bit 2 is opened and remains open until you make another call and redefine the moment.

In the function LpCEWriteBit(7, 24):

The channel associated with bit 7 activates (closes the circuit) for 2400 ms, then becomes inactive (circuit opens).

In the function LpCEWriteBit(5, 255):

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Function Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LpCEWriteByte</td>
<td>Controls eight channels’ output states.</td>
</tr>
</tbody>
</table>
The channel associated with bit 5 activates (closes the circuit) and remains closed until you make another call and redefine the moment.

**Note:** The state of the channels may change when using the LpCEWriteByte function.

Function call in eMbedded Visual C++

```c++
int LpCEWriteBit(int bit, int moment);
```

Function call in C#

```csharp
int LpCEWriteBit(int bit, int moment);
```

Function call in Visual Basic

```vbnet
LpCEWriteBit (ByVal bit As Integer, ByVal moment As Integer) As Integer
```

### 4.2.5.2 Open or Close All Outputs

This function uses the lower eight bits of the parameter *data* to open or close all channels. To set a bit for momentary output, use the LpCEWriteBit function.

Function Parameters and Return Status Values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>data</em></td>
<td>Data bit pattern is output to the 8-bit data output port.</td>
</tr>
</tbody>
</table>

Return value error code:

- 0  NO_ERROR_LPCE  Command successfully completed.
- 20 ERROR_INVALID_WRITE_BYTE  Parameter *data* must be between 0 and 255.
- 21 ERROR_W_BYTE_1  Critical error: Command byte write timeout.
- 22 ERROR_W_BYTE_2  Critical error: Data byte write timeout.

Function call in eMbedded Visual C++

```c++
int LpCEWriteByte(int data);
```

Function call in C#

```csharp
int LpCEWriteByte(int data);
```

Function call in Visual Basic

```vbnet
LpCEWriteByte (ByVal data As Integer) As Integer
```
## 4.2.6 Digital I/O Error Codes

This table contains all of the Digital I/O API error codes.

<table>
<thead>
<tr>
<th>Error Value</th>
<th>Return Value</th>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Applies to All Functions</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>NO_ERROR_LPCE</td>
<td>Command successfully completed.</td>
<td></td>
</tr>
<tr>
<td><strong>Output Functions</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>ERROR_INVALID_WRITE_BIT</td>
<td>Parameter bit must be between 0 and 7.</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>ERROR_INVALID_WRITE MOMENT</td>
<td>Parameter moment must be between 0 and 255.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>ERROR_W_BIT_1</td>
<td>Critical error: Command write timeout.</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>ERROR_W_BIT_2</td>
<td>Critical error: Moment bit write timeout.</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>ERROR_INVALID_WRITE_BYTE</td>
<td>Parameter moment must be between 0 and 255.</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>ERROR_W_BYTE_1</td>
<td>Critical error: Command byte write timeout.</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>ERROR_W_BYTE_2</td>
<td>Critical error: Data byte write timeout.</td>
<td></td>
</tr>
<tr>
<td><strong>Input Functions</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>ERROR_INVALID_READ_BIT</td>
<td>Parameter bit must be between 0 and 7.</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>ERROR_INVALID_READ_NULL_BIT</td>
<td>Parameter data is null pointer.</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>ERROR_INVALID_READ_MODE</td>
<td>Bit requested was not previously configured as BIT_DIGITAL_CONFIGURED by LpCEConfigureInputBit.</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>ERROR_R_BIT_1</td>
<td>Critical error: Command bit read timeout.</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>ERROR_R_BIT_2</td>
<td>Critical error: Data bit read timeout.</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>ERROR_INVALID_READ_NULL_BYTE</td>
<td>Parameter data is a null pointer.</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>ERROR_R_BYTE_1</td>
<td>Critical error: Command byte write timeout.</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>ERROR_R_BYTE_2</td>
<td>Critical error: Data byte read timeout.</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>ERROR_INVALID_CONF_BIT</td>
<td>Parameter bit must be between 0 and 7.</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>ERROR_INVALID_CONF_MODE</td>
<td>Parameter mode is not acceptable.</td>
<td></td>
</tr>
</tbody>
</table>
4.3 Reboot API

The Reboot API enables you to add the capability to reboot the terminal via your application. Please NOTE: the single reboot call is included in the Digital I/O API for simplicity sake; there is not a separate API to reboot.

The programming functions are defined in the header file ExitWin.h (ExitWin.cs for C#, ExitWin.vb for VB) which contains data constant definitions and API function prototypes. The library file ExitWin.lib provides linkage between the application program and the API functions for C/C++.

<table>
<thead>
<tr>
<th>Error Value</th>
<th>Return Value Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>ERROR_INVALID_CONT_BIT</td>
<td>Parameter bit must be between 0 and 7.</td>
</tr>
<tr>
<td>61</td>
<td>ERROR_INVALID_CONT_NULL</td>
<td>Parameter data must not be NULL.</td>
</tr>
<tr>
<td>62</td>
<td>ERROR_INVALID_CONT_MODE</td>
<td>bit property must be properly configured as BIT_COUNTER_CONFIGURED by LpCEConfigureInputBit before it is read as a counter.</td>
</tr>
<tr>
<td>63</td>
<td>ERROR_INVALID_CONT_OVER</td>
<td>Counts rolled over. Given input at the maximum rate of 50 Hz, overrun will occur in 2.6 years.</td>
</tr>
<tr>
<td>64</td>
<td>ERROR_INVALID_CONT_RESET</td>
<td>Parameter reset must be 0 or 1.</td>
</tr>
<tr>
<td>80</td>
<td>ERROR_INVALID_GET_BIT</td>
<td>Parameter bit in LpCEReadBit must be between 0 and 7.</td>
</tr>
<tr>
<td>81</td>
<td>ERROR_INVALID_GET_NULL</td>
<td>Parameter data in LpCEReadBit is a NULL pointer.</td>
</tr>
</tbody>
</table>

### Storing Input Modes and Output States

<table>
<thead>
<tr>
<th>Error Value</th>
<th>Return Value Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>ERROR_SC_W1</td>
<td>Timeout writing SAVE command.</td>
</tr>
<tr>
<td>91</td>
<td>ERROR_SC_W2</td>
<td>Timeout writing ReadStatus command.</td>
</tr>
<tr>
<td>92</td>
<td>ERROR_SC_R1</td>
<td>Timeout reading status byte.</td>
</tr>
<tr>
<td>93</td>
<td>ERROR_EEPROM_WRITE_PROTECTED</td>
<td>EEPROM presumed write-protected.</td>
</tr>
</tbody>
</table>

### Retrieving Input Modes and Output States

<table>
<thead>
<tr>
<th>Error Value</th>
<th>Return Value Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>ERROR_RC_W1</td>
<td>Timeout writing RST.</td>
</tr>
<tr>
<td>101</td>
<td>ERROR_RC_W2</td>
<td>Timeout writing GetStatus command.</td>
</tr>
<tr>
<td>102</td>
<td>ERROR_RC_R1</td>
<td>Timeout reading status byte.</td>
</tr>
<tr>
<td>103</td>
<td>ERROR_RC_TRIGS</td>
<td>Timeout reading hardware setup.</td>
</tr>
</tbody>
</table>
4.3.1 Reboot Calls
There is one Reboot call.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Function Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IIExitWindows</td>
<td>Reboots the terminal</td>
</tr>
</tbody>
</table>

**Function call in eMbedded Visual C++**

```c
void IIExitWindows();
```

**Function call in C#**

```c
void IIExitWindows();
```

**Function call in Visual Basic**

```vb
ExitWindows()
```

4.3.2 Reboot Error Codes
There are no error codes for the reboot function.
Index

Numerics
3270 Terminal Emulation - see also Terminal Emulation Software 56
5250 Terminal Emulation - see also Terminal Emulation Software 56

A
ActiveSync
  cable, part number & how to procure 28
  Deployment method 16
  how to procure software from Microsoft 16, 28
  Setup 27
API
  Digital I/O functions 60
    channel numbering 61
    error codes 70
    input functions 62
      all channels, returning state of 63
      channel, return mode of 65
      channel, set mode of 64
      counter, returning count/optional reset to zero 67
      input channel, returning state of 62
      input modes, retrieving 66
      input modes, storing 65
    output functions 67
      all channels, controlling 69
      output states, retrieving 66
      output states, storing 65
      single channel, controlling 68
    return values 61
header files and import libraries
  all APIs 59
  Digital I/O 61
  Reboot 71
overview of all APIs 59
Reboot function
  files required 71
  list 72
application hosting 56
application software
  custom
    bar code input data 39
    file transfer methods 16, 19
      ActiveSync Reference 27
    general how-to steps 19
    installing application on LANpoint Power X 19
    selecting the correct software development tools 19
    Visual Studio
      development example 23
      installing development environment 20
    audio input and output connectors 12
    audio volume adjustment 12
    Auto-ID
      devices 12
      bluetooth wireless device pairing 13
      powering via auxiliary/accessory power connector 12
      powering via COM port pin 9 12
  battery backup 11
  bluetooth
    Auto-ID wireless scanner pairing 13
    optional bluetooth connections
      keyboard, external 13
      mouse, external 13
  browser
    IE for Windows CE 55
    lock-down kiosk-style 55
    lock-down kiosk-style critical warning 55
C
  C# - see also application software, custom 20
  cable
    accessory power 11, 12
    audio input and output 12
    Auto-ID 12
    battery backup 11
    COM (serial) 12
    keyboard, external 13
    mouse, external 13
    optional digital I/O termination panel 10
    power input 11
    USB devices 11
    WiFi antenna - see also LANpoint Power X Installation Manual 11
    wired Ethernet 11
  CEPad text editor utility 39
  circuit boards 10
    see also features, hardware, circuit boards 10
  clock, see also time synchronization 45
  Cloud Computing 56
  COM port 12
    default settings 12
    configuring
      Network address parameters or disable connection - see also IP address 16
  CPU board 10
  custom programs
Index

see also - application software, custom 19

D

data
  bar code input, wedging 39
developing software
  see also - application software, custom 19
devices
  USB devices 11
DHCP address, configuring - see also IP address 16
Digital I/O - see also API, Digital I/O functions 60
Digital I/O - termination panel option 10
directories, creating shared 17
disabling
  wired or wireless network connection - see also IP address 16
display
  backlight control - blank/black display 13
  backlit color LCD 13
  touch screen option 13
DNS address, configuring - see also IP address 16
downloading files
  see also file transfer methods 16
  via shared directories, how to 17

E
error codes
  Digital I/O API 70
  Reboot API 72
Ethernet
  wired 11
  wireless 11

F
features
  Auto-ID - see also Auto-ID, devices 12
  backlit color LCD display 13
  barcode - see also Auto-ID devices 12
hardware
  circuit boards 10
    CPU board 10
    COM DB9 serial 12
    Ethernet, wired 11
    Ethernet, wireless 11
    power 11
    RAM 10
    SD/MMC 10
    USB 11
  Digital I/O termination panel option 10
    optional battery backup 11
    integral keyboard 11
    LEDs, front panel status 13
  software
    operating system 15
    utilities - see also utility programs 39
    file transfer methods 16, 19
    list of ways to upload and download files 19
      via shared directories, how to 17
    FLASH memory 10
    function keys, programmable 42

H
hardware
  circuit boards
    CPU board
      audio input and output connectors 12
  CPU board
  see also - features, hardware 10
HP Terminal Emulation - see also Terminal Emulation Software 56
HyperTerminal, testing the SerialSocket utility 54

I
IBM3270 Terminal Emulation - see also Terminal Emulation Software 56
IBM5250 Terminal Emulation - see also Terminal Emulation Software 56
IEsimple lock-down browser critical warning 55
IEsimple lock-down kiosk-style browser utility option 55
Input 12
installation
  application software - see also application software, custom, installing application on LANpoint Power X 19
  barcode and other Auto-ID devices 12, 13
cable connections
  keyboard, external 13
  mouse, external 13
optional bluetooth connections
  keyboard, external 13
  mouse, external 13
terminal, power, networking, common accessories, etc. installation topics and instructions - see also LANpoint Power X Installation Manual 9

K
keyboard
  external 11, 13
  standard integral 11
    check OTG/USB settings if non-operative 28
    re-enable if inoperative after using OTG port 36
    wedging, see WedgeCE utility 39
kiosk
  lock-down Web Browser utility 55
  lock-down Web Browser utility, critical warning 55
L
launch a program with Remote Manager 48
LCD
backlight control 13
backlit color 13
touchscreen 15
touchscreen calibration 15
touchscreen navigation 13
LED, front panel status
link 13
power 13
library files
Digital I/O API 61
Reboot API 71
Link LED 13
link status LED 13
M
memory
non-volatile, FLASH 10
non-volatile, SD/MMC card slot 10
RAM 10
Microsoft Visual Studio - see also application
software, custom 20
mouse
external 13
N
navigation
keyboard 15
touchscreen option 15
O
operating system
Windows CE version 10, 16
OTG port - activating for ActiveSync use 30
P
PFK utility 42
port, serial RS232 communication 12
power status LED 13
power, external voltage 11
power, optional battery backup 11
power-up sequence 16
processor, Intel XScale model 10, 16
Programmable Function Key - see PFK 42
programming, software development - see also
application software, custom 19
R
RAM 10
real-time clock
see also time synchronization 45
Reboot
via API 71
with Remote Manager 48
Remote Display utility 41
Remote Manager (WebDevice) 16, 47
RS-232
cable connection 12
default settings 12
S
sample programs
Visual Studio development example 23
SD card
using as additional non-volatile memory 10
SD/MMC card slot 16
SDK for LANpoint
how to procure 21
installation example - Visual Studio 20
serial communication (COM) port 12
SerialSocket utility 53
Shortcut, creating 17
software
APIs 59
custom development
see also - application software, custom 19
terminal emulation 56
utilities - see also utility programs 39
software development kit - see also SDK for
LANpoint 20
startup
creating a startup application shortcut 18
sequence of events 16
status LEDs list 13
stop a program with Remote Manager 48
T
TCP/IP
configuring - see also IP address 16
sockets, SerialSocket utility 53
terminal emulation software 56
terminate a program with Remote Manager 48
text editor, CEPad 39
thin-client software
see also, terminal emulation software 56
time synchronization 45
touchscreen option
calibration 15
using 15
U
uploading files
see also file transfer methods 16
via shared directories, how to 17
Index

USB
  Auto-ID reader/writers 11
  external keyboard 11
  re-enable after using OTG port for ActiveSync 36
  temporarily disable to use OTG port for ActiveSync 28
utility programs
  CEPad Text Editor 39
  kiosk lock-down browser 55
  PFK 42
  Programmable Function Key - see also PFK 42
  Remote Display 41
  Remote Manager - see also WebDevice 47
  SerialSocket 53
  Time Synchronization 45
  WebDevice 47
  WedgeCE 39

V
  virtualization 56
  Visual Studio - see also application software, custom 20
  VisualBasic - see also application software, custom 20
  volume adjustment 12
  VTxxx Terminal Emulation - see also Terminal Emulation Software 56

W
  Web Browser - optional lock-down kiosk browser & critical warning 55
  WebDevice remote manager 16, 20, 47
  WedgeCE, Auto-ID keyboard data wedge utility 12, 39
  WIFI, configuring properties or disabling connection 16
  Windows CE
    operating system 16