

The Intermec logo is displayed in a bold, blue, sans-serif font. It is positioned to the left of a square icon containing a white circular graphic with a blue square in the center. The entire logo and icon are set against a background of light gray, overlapping circular lines that resemble an orbital or network diagram.

Intermec



User's Manual



**751G Color Mobile
Computer**

Intermec Technologies Corporation

Worldwide Headquarters
6001 36th Ave.W.
Everett, WA 98203
U.S.A.

Cedar Rapids Technical Communications
550 Second Street SE
Cedar Rapids, IA 52401
U.S.A.

www.intermec.com

The information contained herein is provided solely for the purpose of allowing customers to operate and service Intermec-manufactured equipment and is not to be released, reproduced, or used for any other purpose without written permission of Intermec Technologies Corporation.

Information and specifications contained in this document are subject to change without prior notice and do not represent a commitment on the part of Intermec Technologies Corporation.

© 2004-2006 by Intermec Technologies Corporation. All rights reserved.

The word Intermec, the Intermec logo, Norand, ArciTech, Beverage Routebook, CrossBar, dcBrowser, Duratherm, EasyADC, EasyCoder, EasySet, Fingerprint, i-gistics, INCA (under license), Intellitag, Intellitag Gen2, JANUS, LabelShop, MobileLAN, Picolink, Ready-to-Work, RoutePower, Sabre, ScanPlus, ShopScan, Smart Mobile Computing, TE 2000, Trakker Antares, and Vista Powered are either trademarks or registered trademarks of Intermec Technologies Corporation.

There are U.S. and foreign patents as well as U.S. and foreign patent applications pending.



Wi-Fi is a registered certification mark of the Wi-Fi Alliance.

Microsoft, Windows, and the Windows logo are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Bluetooth is a trademark of Bluetooth SIG, Inc., U.S.A.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (www.openssl.org).

This product includes cryptographic software written by Eric Young (EAY@cryptsoft.com).

This product uses Regex++, Index software during its operational phases. The owner of Regex++ has granted use of the software to anyone provided such use is accompanied by the following copyright and permission notice:

Regex++, Index. (Version 3.31, 16th Dec 2001)

Copyright © 1998-2001 Dr John Maddock

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Dr John Maddock makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Document Change Record

This page records changes to this document. The document was originally released as Revision A.

Revision Letter	Date	Description of Change
B	04/2005	Added AIT-III information
C	12/2006	Updated to include information about assured radio deactivation, SmartSystems, Wistron radio, EA11 imager, Microsoft WordPad, and IrDA and LAN interfaces,

Contents

Before You Begin	xi
Safety Information	xi
Global Services and Support	xi
Who Should Read This Document?	xii
Related Documents	xii
Patent Information	xiii

1 Using the Computer 1

Ambient Light Sensor	2
Audio System	3
Speaker	3
Microphone	3
External Headset Jack	4
Battery	4
Installing and Charging the Battery	5
Removing the Battery	6
Maximizing Battery Life	7
Beeper	7
Enabling the Registry Storage	7
Enabling the Beeper	8
Adjusting the Beeper Volume	8
Disabling the Beeper	9
Intermec Settings Applet	9
Keypad	10
Backlight for Keypad	10
Key Sequences	10
[Orange] Plane Keys	10
Alpha (Green) Plane Keys	11
LEDs	12
Resetting Your Computer	13
Scanning Bar Codes	13
Scanning with the Area Imager	13
Improving the Performance of the Area Imager	14
Software Build Version	14
Software Tools	15
SmartSystems Foundation Console (www.intermec.com/SmartSystems)	15
SmartSystems Platform Bundles (SSPB)	15
Intermec Resource Kits (www.intermec.com/IDL)	15

Storage Media 15
 Accessing the Secure Digital Card Slot 16
 Internal Card Slots and Connector 16
 Attaching a Tab to the Secure Digital Card 16
 Accessing Files Stored on the Secure Digital Card. 17
 Removing the Secure Digital Card 17

Wireless Network Support. 17

Accessories. 18

Physical and Environmental Specifications 18

2 Windows CE .NET 21

Software Builds 22

Where to Find Information 22

Basic Skills. 22
 Desktop Screen. 23
 Programs 23
 Start Menu and Task Bar 23
 Notifications. 24
 Entering Information 24
 Typing With the Onscreen Keyboard 24
 Using Transcriber 25
 Selecting Typed Text. 25
 Finding and Organizing Information 26
 Customizing Your Computer 27
 Adjusting Settings 27
 Adding or Removing Programs 27

Microsoft ActiveSync. 30

Microsoft WordPad. 30
 Creating a Document 31
 Typing Mode 31
 Writing Mode 32
 Synchronizing WordPad Documents 32

Internet Explorer 33
 Viewing Mobile Favorites and Channels. 33
 Browsing the Internet. 33

3	Configuring the Computer	35
	Configuring Parameters	36
	Configuring the Computer With Intermec Settings	36
	Synchronizing the Computer System Time with a Time Server	36
	Configuring the Computer through the Network	37
	Configuring the Computer in a TCP/IP Direct Connect Network	37
	Configuring the Computer in a UDP Plus Network	38
	Using Configuration Parameters	39
	Configuring the Printer	39
	Directly to a Port	39
	Directly to a Generic Serial Port	39
	Configuring the Scanner	40
	Scanner Control and Data Transfer	40
	Data Collection Configuration	40
	Changing Comm Settings	41
	Improving the Performance of the Area Imager	41
	Reading Distances	42
	Installing Applications on the Computer	43
	Using Microsoft ActiveSync	43
	Using a Storage Card	44
	Using the SmartSystems Console	45
	Using Wavelink Avalanche	45
	Installing Cabinet Files	46
	Developing Applications for the Computer	46
	Packaging Applications for the Computer	47
	Launching Your Application Automatically	47
	RunAutoRun	48
	AutoExec	49
	AutoRun	50
	AutoCopy	50
	AutoReg	51
	AutoCab	52
	Creating Cab Files	53
	Creating Device-Specific Cab Files	53
	Creating an .inf File	53
	[CEStrings]	54
	Sample .INF File	59
	Using Installation Functions in Setup.dll	62
	After the CAB File Extraction	62
	Creating Cab Files with CAB Wizard	65
	Troubleshooting the CAB Wizard	66
	Customization and Lockdown	66

Contents

Kernel I/O Controls	67
IOCTL_HAL_GET_DEVICE_INFO	67
IOCTL_HAL_ITC_READ_PARM	68
IOCTL_HAL_ITC_WRITE_SYSPARM	70
IOCTL_HAL_GET_DEVICEID	71
IOCTL_HAL_GET_OAL_VERINFO	72
IOCTL_HAL_GET_BOOTLOADER_VERINFO	73
IOCTL_HAL_WARMBOOT	73
IOCTL_HAL_COLDBOOT	74
IOCTL_HAL_GET_RESET_INFO	74
IOCTL_HAL_GET_BOOT_DEVICE	75
IOCTL_HAL_REBOOT	76
IOCTL_PROCESSOR_INFORMATION	76
IOCTL_GET_CPU_ID	77
Networking APIs	78
Basic Connect/Disconnect Functions	78
RadioConnect()	78
RadioDisconnect()	79
RadioDisassociate()	79
Query Information Functions	79
GetAssociationStatus()	79
GetAuthenticationMode()	80
GetBSSID()	80
GetDiversity()	81
GetLinkSpeed()	81
GetMac()	82
GetNetworkMode()	82
GetNetworkType()	83
GetSSID()	83
GetPowerMode()	84
GetRSSI()	84
GetTXPower()	85
GetWepStatus()	85
GetRadioIpAddress()	86
GetCCXStatus()	86
Set Information Functions	87
AddWep()	87
EnableWep()	87
EncryptionStatus()	88
RemoveWep()	88
SetAuthenticationMode()	89
SetChannel()	89
SetNetworkMode()	90
SetPowerMode()	90
SetSSID()	91
SetCCXStatus()	91
SetMixedCellMode()	91

Helper Functions	92
ConfigureProfile()	92
EnableSuppLogging()	92
EnableZeroConfig()	93
GetCurrentDriverName()	93
isDHCPEnabled()	93
isOrinoco()	94
isSupplicantRunning()	94
isZeroConfigEnabled()	94
RenewDHCP()	95
ResetRadioToSystemSave()	95
StartScanList()	95
StartSupplicant()	96
StopSupplicant()	96
SwitchPacketDriver()	96
Deprecated Functions	97
Notifications	97
NLEDGetDeviceInfo	98
NLEDSetDevice	98
Reboot Functions	98
Reprogramming the 751G Keypad	99
Key Values	99
How Key Values Are Stored in Registry	100
Change Notification	100
Advanced Keypad Remapping	101
Scan Codes	101
Sample View of Registry Keys	102

4 Maintaining the Computer 103

Updating the System Software	104
Using a Storage Card to Upgrade the Computer	104
Using the SmartSystems Console to Upgrade the Computer	105
Troubleshooting Your Computer	106
Cleaning the Scanner	109

5 Network Support 111

802.11b/g Communications	112
Remote Access (Modems)	113
Connecting to an Internet Service Provider	113
Connecting to Work	116
Ending a Connection	117

Contents

Configuring Security 118
 Loading Certificates 118
 Wireless Networks 118
 Choosing Between Microsoft and Funk Security 120
 Configuring Funk Security 120
 Configuring Microsoft Security 126

SmartSystems™ Foundation 127

| Index 129

Before You Begin

This section provides you with safety information, technical support information, and sources for additional product information.

Safety Information

Your safety is extremely important. Read and follow all warnings and cautions in this document before handling and operating Intermec equipment. You can be seriously injured, and equipment and data can be damaged if you do not follow the safety warnings and cautions.

This section explains how to identify and understand dangers, warnings, cautions, and notes that are in this document.



A warning alerts you of an operating procedure, practice, condition, or statement that must be strictly observed to avoid death or serious injury to the persons working on the equipment.



A caution alerts you to an operating procedure, practice, condition, or statement that must be strictly observed to prevent equipment damage or destruction, or corruption or loss of data.



Note: Notes either provide extra information about a topic or contain special instructions for handling a particular condition or set of circumstances.

Global Services and Support

Warranty Information

To understand the warranty for your Intermec product, visit the Intermec web site at www.intermec.com and click **Service & Support**. The **Intermec Global Sales & Service** page appears. From the **Service & Support** menu, move your pointer over **Support**, and then click **Warranty**.

Disclaimer of warranties: The sample code included in this document is presented for reference only. The code does not necessarily represent complete, tested programs. The code is provided “as is with all faults.” All warranties are expressly disclaimed, including the implied warranties of merchantability and fitness for a particular purpose.

Web Support

Visit the Intermec web site at www.intermec.com to download our current manuals in PDF format. To order printed versions of the Intermec manuals, contact your local Intermec representative or distributor.

Visit the Intermec technical knowledge base (Knowledge Central) at intermec.custhelp.com to review technical information or to request technical support for your Intermec product.

Telephone Support

These services are available from Intermec Technologies Corporation.

Service	Description	In the U.S.A. and Canada call 1-800-755-5505 and choose this option
Order Intermec products	<ul style="list-style-type: none"> Place an order. Ask about an existing order. 	1 and then choose 2
Order Intermec media	Order printer labels and ribbons.	1 and then choose 1
Order spare parts	Order spare parts.	1 or 2 and then choose 4
Technical Support	Talk to technical support about your Intermec product.	2 and then choose 2
Service	<ul style="list-style-type: none"> Get a return authorization number for authorized service center repair. Request an on-site repair technician. 	2 and then choose 1
Service contracts	<ul style="list-style-type: none"> Ask about an existing contract. Renew a contract. Inquire about repair billing or other service invoicing questions. 	1 or 2 and then choose 3

You can find information on Intermec telephone support services at www.intermec.com/ait. To find the correct telephone number for your country, click **Contact**.

Who Should Read This Document?

The *751G Color Mobile Computer User's Manual* is written for the person who is responsible for installing, configuring, maintaining, and troubleshooting the product.

Before you install and configure your product, you should be familiar with your network and general networking terms, such as IP address.

Related Documents

This table contains a list of related Intermec documents and part numbers.

Document Title	Part Number
<i>Model 751G Mobile Computer Quick Start Guide</i>	962-054-093
<i>Intermec Computer Command Reference Manual</i>	073529
<i>Important 2610C Radio Information!</i>	075494

The Intermec web site contains Intermec documents (in PDF) that you can download for free.

To download documents

- 1 Browse to www.intermec.com.
- 2 Click Service & Support > Manuals.
- 3 In the Select a Product field, choose the product whose documentation you want to download.

To order printed versions of the Intermec manuals, contact your local Intermec representative or distributor.

Patent Information

This product is protected by one or more of the following patents:

4,882,476; 4,894,523; 4,953,113; 4,961,043; 4,970,379; 4,988,852;
5,019,699; 5,021,642; 5,038,024; 5,081,343; 5,095,197; 5,144,119;
5,144,121; 5,182,441; 5,187,355; 5,187,356; 5,195,183; 5,195,183;
5,195,183; 5,216,233; 5,216,550; 5,218,191; 5,227,614; 5,233,172;
5,241,488; 5,243,602; 5,258,606; 5,278,487; 5,288,985; 5,308,966;
5,322,991; 5,331,136; 5,331,580; 5,342,210; 5,349,678; 5,359,185;
5,371,858; 5,373,478; 5,389,770; 5,397,885; 5,410,141; 5,414,251;
5,416,463; 5,442,167; 5,464,972; 5,468,947; 5,468,950; 5,477,044;
5,486,689; 5,488,575; 5,500,516; 5,502,297; 5,504,367; 5,508,599;
5,514,858; 5,530,619; 5,534,684; 5,536,924; 5,539,191; 5,541,419;
5,548,108; 5,550,362; 5,550,364; 5,565,669; 5,567,925; 5,568,645;
5,572,007; 5,576,529; 5,592,512; 5,594,230; 5,598,007; 5,608,578;
5,616,909; 5,619,027; 5,627,360; 5,640,001; 5,657,317; 5,659,431;
5,671,436; 5,672,860; 5,684,290; 5,719,678; 5,729,003; 5,793,604;
5,742,041; 5,761,219; 5,764,798; 5,777,308; 5,777,309; 5,777,310;
5,786,583; 5,798,509; 5,798,513; 5,804,805; 5,805,807; 5,811,776;
5,811,777; 5,818,027; 5,821,523; 5,828,052; 5,831,819; 5,834,749;
5,834,753; 5,837,987; 5,841,121; 5,842,070; 5,844,222; 5,854,478;
5,862,267; 5,869,840; 5,873,070; 5,877,486; 5,878,395; 5,883,492;
5,883,493; 5,886,338; 5,889,386; 5,895,906; 5,898,162; 5,902,987;
5,902,988; 5,912,452; 5,923,022; 5,936,224; 5,949,056; 5,969,321;
5,969,326; 5,969,328; 5,979,768; 5,986,435; 5,987,192; 5,992,750;
6,003,775; 6,012,640; 6,016,960; 6,018,597; 6,024,289; 6,034,379;
6,036,093; 6,039,252; 6,064,763; 6,075,340; 6,095,422; 6,097,839;
6,102,289; 6,102,295; 6,109,528; 6,119,941; 6,128,414; 6,138,915;
6,149,061; 6,149,063; 6,152,370; 6,155,490; 6,158,661; 6,164,542;
6,164,545; 6,173,893; 6,195,053; 6,234,393; 6,234,395; 6,244,512;
6,249,008; 6,328,214; 6,330,975; 6,345,765; 6,356,949; 6,367,699;
6,375,075; 6,375,076; 6,431,451; 6,435,411; 6,484,944; 6,488,209;
6,497,368; 6,532,152; 6,538,413; 6,539,422; 6,621,942; 6,641,046;
6,681,994; 6,687,403; 6,688,523; 6,732,930; Des. 417445

Docking Station/Device: 5,052,943; 5,195,183; 5,317,691; 5,331,580;
5,544,010; 5,644,471

There may be other U.S. and foreign patents pending.

Before You Begin

1 Using the Computer

This chapter introduces the 751G Color Mobile Computer, developed by Intermec to enhance wireless connectivity needs. This chapter contains hardware and software configuration information to assist you in making the most out of your 751G.

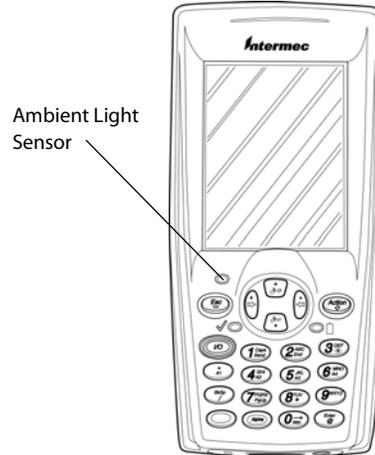


Note: Desktop icons and applet icons are shown to the left. Any place that **Start** is mentioned, tap the following Windows icon in the bottom, left corner of your desktop.

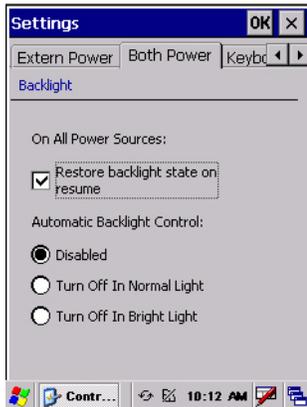


Ambient Light Sensor

The ambient light sensor turns on the display lighting when conditions warrant but automatically turns off again as surrounding light increases. This conserves your 751G battery power.



 To adjust the ambient light sensor, tap **Start > Settings > Control Panel**. Double-tap the **Backlight** icon, then tap the right arrow to move to and tap the **Both Power** tab. Make your selections, then tap **OK** to exit this applet.

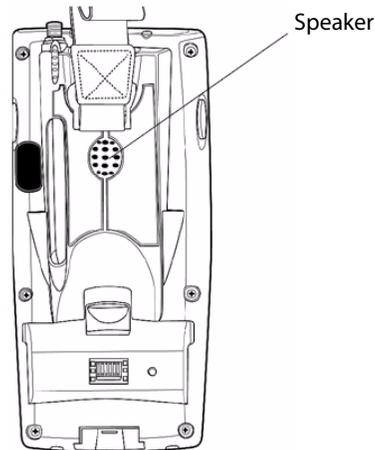


Audio System

The audio system consists of the speaker, internal microphone, and the external headset jack.

Speaker

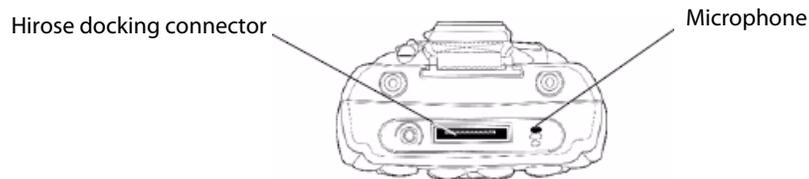
The speaker, which is capable of variable volume levels, is located on the back of the 751G. This speaker has a transducer volume of 85 dB min at 10 cm (3.9") and a frequency range of 1-8 KHz.



Warning: Do not place the speaker next to your ear when the speaker volume is set to “Loud” (maximum), or you may damage your hearing.

Microphone

The built-in microphone is located on the bottom of the unit next to the Hirose docking connector.



This is the bottom of the 751G. Note that the keypad is to the bottom in this illustration.

External Headset Jack

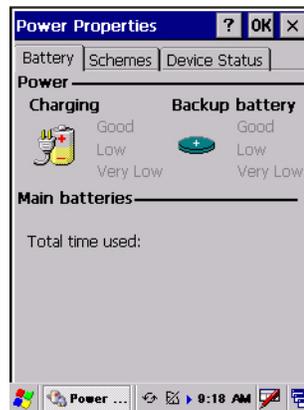
The external headset jack connects a mobile phone style headset to the 751G for use in noisy environments. The jack is a 2.5 mm, three-conductor jack, with autosensing of the headset jack insertion which disables the internal speaker and microphone. The external headset jack is located on the bottom of the 751G next to the Hirose docking connector.



Battery



The 751G comes with a 14.4 Watt-hour, 7.2V, replaceable Lithium-Ion (LiIon) battery. To view the status of this battery from the 751G, tap **Start** > **Settings** > **Control Panel**. Double-tap the **Power** icon, then tap the **Battery** tab. Tap **OK** to exit this applet.



If your computer shuts down because of low battery conditions, your computer does not operate. This is done to ensure that data is protected. Although the battery does protect the data against loss for several hours, you should connect your computer to a power source when you first detect a low battery condition.



Note: Your computer has an internal backup super capacitor, a temporary power storage device, that protects data for up to ten minutes. It also shuts down the 751G if the main battery suddenly goes away (removed from the computer). Depending on the processes running, it may not have adequate power for a graceful shutdown. If so, the 751G performs a cold-boot the next time power is applied.

In short, put the 751G into a suspend (sleep) mode before you remove the main battery.

If you have at least one device in your 751G (radio, scanner, or imager), the battery power fail level is set so that after the system shuts down in a low battery condition, there is still sufficient charge to allow the unit to remain configured, keep proper time, and maintain DRAM (Dynamic Random Access Memory) for at least 72 hours at room temperature *if* the main battery remains in the mobile computer.

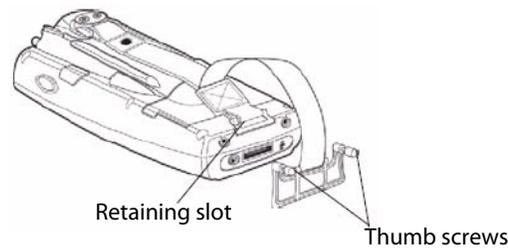
The configuration and time are lost if:

- The battery discharges beyond this level.
- The battery is removed when the computer is *not in suspend mode*.
- A cold-boot (reset) is performed on the computer.

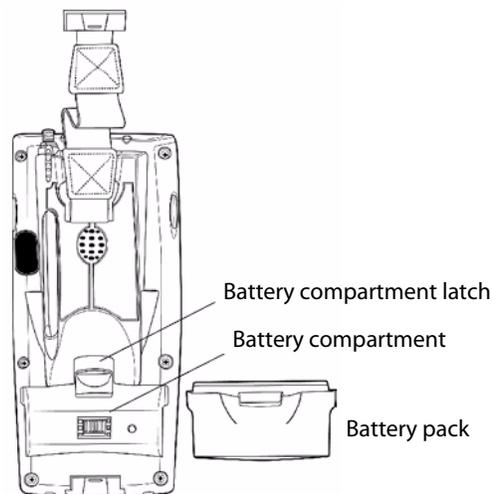
Installing and Charging the Battery

Make sure you fully charge the battery before you use your 751G. To charge the battery, you need to install it in the 751G.

- 1 Remove the two thumb screws on the connector cover to release the hand strap and back cover.
- 2 Slide the bottom of the strap forward to release it from the retaining slot.



- 3 Tilt, insert, and place the battery into the compartment. Make sure the battery compartment latch clicks in place to ensure the battery is secure.



- 4 Insert your 751G into its single dock for charging.

- 5 Charge the battery pack for three hours before using. However, to ensure proper charging, perform the remaining steps first, with the AC adapter or dock connected:
 - a The first time you turn on your 751G, it boots to the operating system. After a few seconds, you see the Windows CE .NET Desktop screen. Tap your stylus to advance to the next display on the screen.
 - b You will be prompted through the several screens to complete the setup process. Read the display messages and follow the instructions. When you reach the Windows CE .NET Desktop screen, you have completed the setup.



Caution

You must use only the Intermec power supply approved for use with the 751G. Using any other power supply will damage the 751G.



Note: For help installing and using the single dock, see the *700 Series Single Dock Quick Start Guide* (P/N 962-040-009) shipped with the dock.

Removing the Battery

Follow these instructions to remove the battery from the 751G.



Caution

Only use the battery compartment latch to dislodge and remove the battery. Using any other tool or method to remove the battery may damage the battery or the 751G.



Caution

Removing the main battery when the backup battery low or critically low icon appears on the status bar may cause your 751G to cold boot and you may lose data.

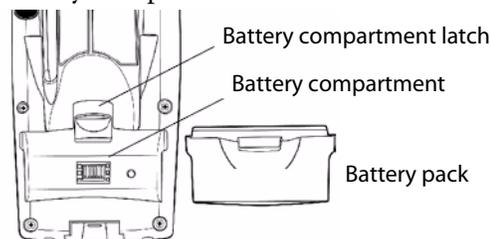


Caution

If you fail to replace the battery immediately, you may lose important data or applications.

To remove the battery

Pull up on the battery compartment latch, then lift the battery out of the battery compartment.



Maximizing Battery Life

- Set the **Backlight Timeout** to 10 seconds.
- Verify that **Radio Power Management** is enabled (Fast PSP). Enabling radio power management allows your radio to switch between awake and sleep modes based on network traffic.
- Verify that each setting under **Power Management** has a value of 1 minute for a combined automatic shutoff time of 3 minutes.

Beeper



Note: Each time a cold-boot is performed on the 751G, all default settings are restored unless registry storage is enabled.

To learn how to set volume levels for screen taps, ActiveSync alert noises, etc., tap **Start > Help > Windows CE Basics**.

Enabling the Registry Storage



For Windows CE .NET, the Flash File System (PSM) is the only medium available for saving the registry data. Tap **Start > Settings > Control Panel**. Double-tap the **Utilities** icon, then tap the **Registry Save** tab. Check **Enable Registry Storage** to enable this function, then tap **OK**.

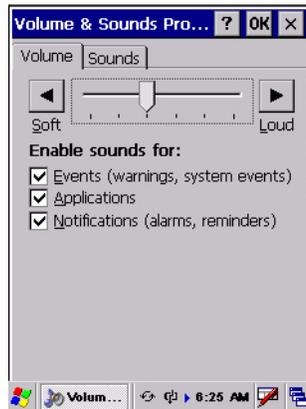


Enabling the Beeper



To enable the beeper:

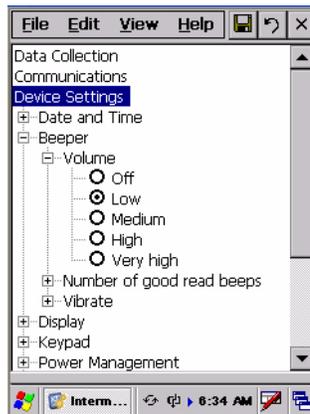
- 1 Tap **Start > Settings > Control Panel**. Double-tap the **Volume & Sounds** icon, then tap the **Volume** tab.
- 2 Drag the slider bar to the right, away from the “Soft” position.
- 3 Tap **OK** to exit this applet.



Adjusting the Beeper Volume



To select a beeper volume for the 751G, tap **Start > Intermec Settings**, then tap the **Device Settings** option. Tap (+) to expand the **Beeper** option, then tap (+) to expand the **Volume** option. Select an item, then tap (+) to close this option.



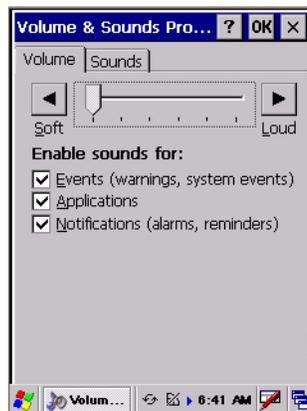
Note: Information about the Intermec Settings applet is found in the *Intermec Computer Command Reference Manual* (P/N 073529). See your Intermec representative for information.

Disabling the Beeper

To disable the beeper:



- 1 Tap **Start > Settings > Control Panel**. Double-tap the **Volume & Sounds** icon, then tap the **Volume** tab.
- 2 Tap the **Soft** button to drag the slider bar all the way to the left.
- 3 Tap **OK** to exit this applet.



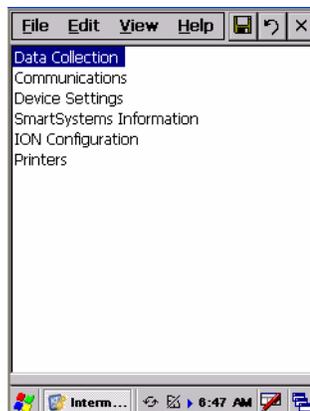
Intermec Settings Applet

Use the Intermec Settings applet to gather, view, and update device configuration settings. Information about the settings you can configure with the Intermec Settings applet is in the *Intermec Computer Command Reference Manual* (P/N 073529) available online at www.intermec.com.

See the Data Collection Resource Kit in the Intermec Developer Library (IDL) for information about data collection functions. The IDL is available as a download from the Intermec web site at www.intermec.com/idl. Contact your Intermec representative for more information.



Tap **Start > Settings > Control Panel**, then double-tap the **Intermec Settings** icon to access the applet.



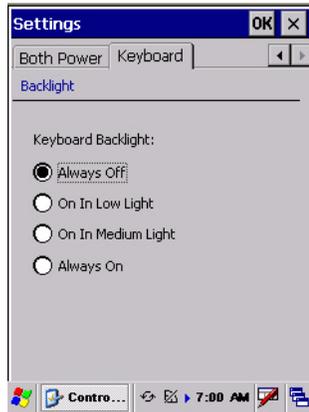
Keypad

Instructions for the keypad include the backlight and keypress sequences.

Backlight for Keypad



You can configure your keypad to turn on a backlight to assist you when you are working in low lighting. To adjust the backlight for the keypad, tap **Start > Settings > Control Panel**, then double-tap the **Backlight** icon. Tap the right arrow to move to and tap the **Keyboard** tab. Make your selection, then tap **OK** to exit this applet.



Key Sequences

Use the following key sequences to enter characters into your 751G using the numeric keypad.

[Orange] Plane Keys

The [orange] plane key provides you access to display controls, special characters, and CE .NET options.

Press the [orange] key for each orange plane key stroke you wish to make. For example, to turn on the front light, press and hold the [orange] key plus the [3] key. To turn the front light off, press these keys again.

The following table lists sequences that use the [orange] plane key. See [Chapter 2, “Windows CE .NET”](#) for information about Windows CE .NET applications.

[Orange] Plane Keys

Press the Keys	To Do This
[orange] [3]	Toggle backlight on/off, goes through backlight power levels
[orange] [.]	
[orange] [4]	
[orange] [5]	

[Orange] Plane Keys (continued)

Press the Keys	To Do This
[orange] [6]	
[orange] [7]	Move up one page.
[orange] [8]	Enter an asterisk (*).
[orange] [9]	Move down one page.
[orange] [0]	Access the CE .NET Start menu.
[orange] [Enter]	Enter an at symbol (@).
[orange] [BkSp]	Enter a backslash (/).
[orange] [Esc]	Enter a minus sign (-).
[orange] [Action]	Enter a plus sign (+).
[orange] [right arrow]	Tab to the right.
[orange] [left arrow]	Tab to the left.
[orange] [up arrow]	Increase volume
[orange] [down arrow]	Decrease volume

Alpha (Green) Plane Keys

You can enter the alphabet using the Alpha (green) plane keys. Below and on the next page are the key sequences.

When you press **[Alpha]**, the Scanning/Alpha LED shows red for the Alpha mode. The keypad stays in Alpha mode until you press **[Alpha]** again.

To type a lowercase “c,” press **[Alpha] [2] [2] [2]**. To type a letter on the same key as the last letter entered, wait two seconds, then enter the correct series of keystrokes to create the next letter.

While in the Alpha mode and you press **[1]** to initiate the CAPS mode, you will render a CAPS LOCK until you press **[1]** again. Once you are in CAPS mode, you stay in CAPS until it is pressed again. Press **[0]** to enter a space.

Alpha (Green) Plane Keys

To Enter	Press the Keys	To Enter	Press the Keys
a	[Alpha] [2]	A	[Alpha] [1] [2]
b	[Alpha] [2] [2]	B	[Alpha] [1] [2] [2]
c	[Alpha] [2] [2] [2]	C	[Alpha] [1] [2] [2] [2]
d	[Alpha] [3]	D	[Alpha] [1] [3]
e	[Alpha] [3] [3]	E	[Alpha] [1] [3] [3]
f	[Alpha] [3] [3] [3]	F	[Alpha] [1] [3] [3] [3]
g	[Alpha] [4]	G	[Alpha] [1] [4]
h	[Alpha] [4] [4]	H	[Alpha] [1] [4] [4]
i	[Alpha] [4] [4] [4]	I	[Alpha] [1] [4] [4] [4]
j	[Alpha] [5]	J	[Alpha] [1] [5]

Alpha (Green) Plane Keys (continued)

To Enter	Press the Keys	To Enter	Press the Keys
k	[Alpha] [5] [5]	K	[Alpha] [1] [5] [5]
l	[Alpha] [5] [5] [5]	L	[Alpha] [1] [5] [5] [5]
m	[Alpha] [6]	M	[Alpha] [1] [6]
n	[Alpha] [6] [6]	N	[Alpha] [1] [6] [6]
o	[Alpha] [6] [6] [6]	O	[Alpha] [1] [6] [6] [6]
p	[Alpha] [7]	P	[Alpha] [1] [7]
q	[Alpha] [7] [7]	Q	[Alpha] [1] [7] [7]
r	[Alpha] [7] [7] [7]	R	[Alpha] [1] [7] [7] [7]
s	[Alpha] [7] [7] [7] [7]	S	[Alpha] [1] [7] [7] [7] [7]
t	[Alpha] [8]	T	[Alpha] [1] [8]
u	[Alpha] [8] [8]	U	[Alpha] [1] [8] [8]
v	[Alpha] [8] [8] [8]	V	[Alpha] [1] [8] [8] [8]
w	[Alpha] [9]	W	[Alpha] [1] [9]
x	[Alpha] [9] [9]	X	[Alpha] [1] [9] [9]
y	[Alpha] [9] [9] [9]	Y	[Alpha] [1] [9] [9] [9]
z	[Alpha] [9] [9] [9] [9]	Z	[Alpha] [1] [9] [9] [9] [9]

LEDs

The battery status LED and the scanning/keypad shift and notification LED turn red, green, or yellow.

Battery Status LED

LED Color and Action	Description
Steady Green	Battery is more than 95% charged and the 751G is on charger.
Blinking Red	Battery is low. The blinking speed increases as the battery's power gets increasingly lower.
Red	Main battery is low; or if charging, remains red until the 95% charge status is reached.
Yellow	The 751G is on a charging source and there is no battery pack installed. The mobile computer may also be out of the charging range of 32° to 122° F (0° to 50° C). When back in range, charging resumes and the LED changes to red or green.
Alternating Red/Yellow	Replace the battery pack.

Scanning/Keypad Shift and Notification LED

LED Color/Action	Description
Momentary Green	Indicates the scanner has initialized and had a good scan.
Blinking Green	Indicates the scanner is initializing.
Steady Red	Indicates the keypad is shifted to the Alpha plane and the 751G is turned on.
Blinking Red	Indicates the radio is on when in suspend mode and when the radio is initialized.
Yellow	When the keypad is in Alpha mode, the LED temporarily switches from red to yellow to indicate a good scan.

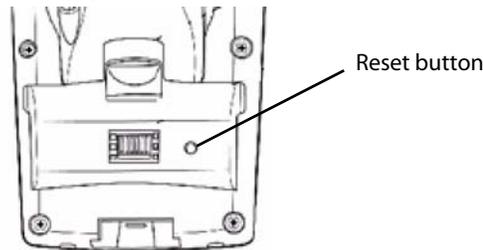
Resetting Your Computer

In some cases where the 751G completely stops responding, it may be necessary to perform a cold reset. Because cold resetting may result in data loss, only use this method if all other recovery methods have failed.



Note: Cold resetting deletes all programs and data stored in RAM including the Object Store. Make sure data is backed up to your host computer or a storage card before performing a cold reset.

To reset your computer, release the lower clip of the hand strap, remove the battery pack, press the **Reset** button, then reinstall the battery pack.



This illustration shows the back of the 751G inside the battery compartment.

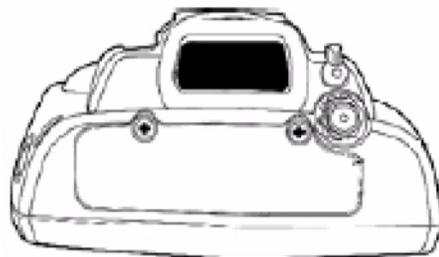
Scanning Bar Codes

Use the area imager to scan and enter bar code data. The 751G supports reading 1D and 2D images. These bar code symbologies are enabled by default on the 751G: Code 39, Code 128, UPC-A, UPC-E, EAN-8, EAN-13, and Datamatrix.

If you are using bar code labels that are encoded in a different symbology, you need to enable the symbology on the computer. Use the Intermec Settings applet to enable and disable symbologies. See the *Intermec Computer Command Reference Manual* available from the Intermec web site at www.intermec.com.

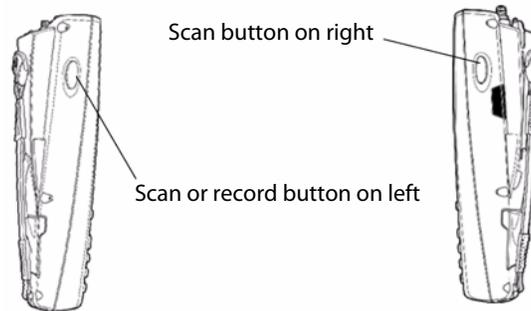
Scanning with the Area Imager

The 751G has an area imager on the top of the unit that can scan 1D and 2D bar code symbologies. It also supports omni-directional (360°) scanning where you can position the unit in any orientation to scan a bar code label. Using the 2D imager is like taking a digital picture.



To use the area imager

- 1 Press the power switch to turn on the 751G, point the scanner window a few inches from the bar code label, and hold steady.
- 2 Press either **Scan** button and center the red aiming beam over the bar code label. The aiming beam is smaller when the imager is closer to the bar code and larger when it is further away.
- 3 When a bar code label is successfully read and a high beep is emitted, release the **Scan** button you pressed.



Improving the Performance of the Area Imager

If you have problems scanning a bar code with the 2D imager, go to [“Improving the Performance of the Area Imager” on page 41](#) for tips on improving its performance.

Software Build Version

The Persistent Storage Manager (PSM) is an area of storage which is embedded in a section of the system’s FLASH memory. This storage area is *not* erased during a cold-boot. It may, however, be erased during the reflashing process. In addition to storing applications and data files, you do have the option to store a persistent registry to the PSM region.



To check to see if your 751G has the latest PSM build or the latest CE build, double-tap the **Internet Explorer** icon from the desktop, then scroll down for the latest information displayed beneath the **751G Version Information** title.



Software Tools

The following Intermec software tools are available as free downloads:

SmartSystems Foundation Console (www.intermec.com/SmartSystems)

This tool includes a management console that provides a default method to configure and manage Intermec devices “out-of-the-box,” without the purchase of additional software licenses. This is for anyone who must configure and deploy multiple devices or manage multiple licenses.

SmartSystems Platform Bundles (SSPB)

The SmartSystems Platform Bundle (SSPB) is a bundle of software that contains the Data Collection Engine (DCE), SmartSystems, Funk Supplicant, Intermec Settings, and Intermec Developer Library (IDL) runtime.

The SSPB is stored in the “\Flash File Store” folder off the root of your 751G and automatically installed on the device when it is initially started up. Updated bundles are available as software downloads from the Intermec web site at www.intermec.com/SmartSystems. Click **Downloads** on the left to access the latest.

Intermec Resource Kits (www.intermec.com/IDL)

Resource Kits provide tools that build applications using the features of Intermec devices. Resource kits include: Bluetooth, Communications, Data Collection, Device Settings, Mobile Gadgets, Printing, and RFID.

This is for anyone who develops software for the 751G.

Storage Media



Note: MultiMediaCards (MMCs) and CompactFlash (CF) storage cards are not supported in 751G.

Note: The 751G currently supports Delkin Devices Secure Digital cards only. Intermec Technologies cannot guarantee that other SD cards will work with the 751G.

The 751G supports the Secure Digital storage card. Use the following procedures to insert a Secure Digital card, access the files on a Secure Digital card, and remove a Secure Digital card.

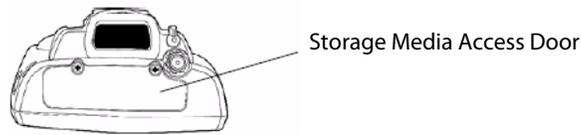


Warning

Warning: Before installing a Secure Digital card, inspect the gasket on the door for any damage or wear, and replace the door if any damage or wear is found. Otherwise, use of this terminal in a hazardous environment may cause loss of life.

Accessing the Secure Digital Card Slot

To access the card slot, locate the access door at the top of the 751G, loosen its two screws, then remove the door. *Note that the screws to this door are to be torqued to 1.5 in-lbs.* See the *Model 751G Mobile Computer Quick Start Guide* (P/N 962-054-093) for more information.

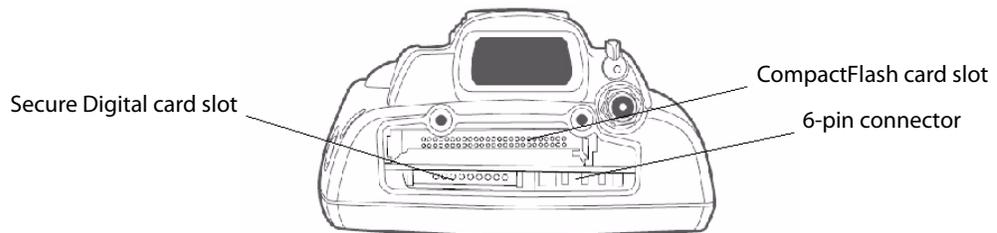


This illustration shows the top of the 751G. Note the keypad is to the bottom.

Internal Card Slots and Connector

Below is a view of the various card slots within your 751G. Note that you only have access to the Secure Digital card slot. The other two slots are embedded into the unit and cannot be removed.

- A radio is embedded in the CompactFlash card slot and is not accessible.
- The Secure Digital card goes into the bottom left card slot.
- The SmartCard adapter plugs into the 6-pin connector in the bottom right.



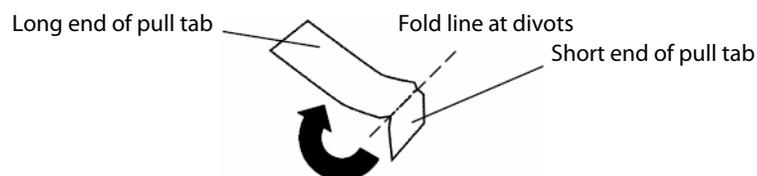
This illustration shows the top of the 751G. Note the keypad is to the bottom.

Attaching a Tab to the Secure Digital Card

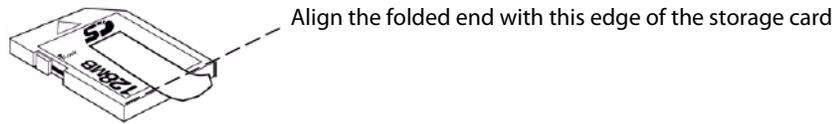
The Secure Digital storage card, as ordered from Intermec, come with acrylic adhesive pull tabs. If you are using a storage card that you plan to remove from the 751G, this tab can make its removal easier.

Do the following to attach the tab to your storage card. Note that the pull tab has divots cut into either side, towards the shorter end. Use these divots as a guide.

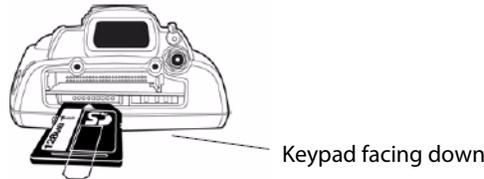
- 1 Completely peel the paper off the short end of the tab. Partially pull the paper off the long end of the tab away from the divots. Fold the short end under, at the divots, to stick to itself.



- 2 Align the folded edge of the pull tab where there is no adhesive with the bottom end of the storage card. Peel away the rest of the paper from the long end, then firmly press down the remaining adhesive area of the tab onto the storage card.



- 3 Insert the storage card, with the contacts facing the keypad, into your 751G to ensure that no adhesive is exposed once the tab is placed.



- 4 Press on the storage card until you hear a click. If needed, close the storage media access door.

Accessing Files Stored on the Secure Digital Card

When inserted in the 751G, the Secure Digital card inserted in your 751G, it appears as the “\SDMMC Disk” folder. To access this folder, select **My Computer**, then tap the “\SDMMC Disk” folder.

Removing the Secure Digital Card

- 1 Press the **Power** key for seconds, and then release the **Power** key to turn off the 751G. Remove the storage media access door.
- 2 Gently depress the Secure Digital card to release the card, then pull the card from its slot.
- 3 Replace the storage media access door.
- 4 Press the **Power** key for two to three seconds, and then release the **Power** key to turn on the 751G.

Wireless Network Support

Radios are installed at the factory and cannot be installed by a user. The 751G must be serviced to install or replace radios. Contact your Intermec representative for more information. See [Chapter 5, “Network Support”](#) for information about supported radios.



Note: Changes or modifications not expressly approved by Intermec could void the user’s authority to operate the equipment.

Accessories

The following accessories are available for the 751G. *Note that this is not a complete list.* Contact your Intermec representative for information about these and other accessories that are not in this list.

Communications and charging dock
Single bay communications cradle with serial/USB/O interface
USB through multipurpose connector at base of unit or RS232 serial adapters
Serial/USB cables
Snap-on module for button memory
CAC (Common Access Card) adapter
Plug-in interface CAC reader
4-slot battery charger
Pistol grip scanning handle
Holster
Dust cover

Physical and Environmental Specifications

Use these specifications to locate technical information about the 751G and its available features and options.

Display

1/4 VGA Transflective, software-controlled backlight
 Pixels: 240x320
 Diagonal: 97 mm (3.8 in)
 Colors: 256 K

Environmental

Operating Temperature: -10° to 50°C (14° to 122°F)
 Storage Temperature: -20° to 60°C (-4° to 140°F)
 Relative Humidity: 5% to 95% noncondensing
 Rain and Dust Resistance: IP64 compliant
 Drop Specifications: 1.2 m (4 ft) drop

Secure Digital Expansion Slots

The 751G supports the Delkin Devices Secure Digital storage card.

Integrated Scanner Options

EA11 Linear Imager

Integrated Wireless

802.11b/g (Wi-Fi® certified): WLAN (802.11b/g)
 Bluetooth™ compatible module

Keypad Option

22-key layout with one-touch numerics and shifted alpha with 4-way navigation buttons

Memory and Storage

RAM Memory: 64 MB

Flash ROM: 64 MB, includes ROM folder for application storage

Microprocessor

Intel® XScale™ PXA255 Application Processor, 400 MHz

Operating System

Microsoft® Windows® CE .NET (4.2)

Physical Dimensions

Length: 191 mm (7.53 in)

Width: 50 mm (1.97 in)

Height: 90 mm (3.50 in)

Weight: 460 g (16.0 oz)

Power

Battery Type: Lithium-Ion (LiIon), 7.2V, (1x2000 mAh cells), customer-replaceable

Battery Capacity: 14.4 Watt-hours

Battery Life: 8+ hours, application-dependent

Recharging Time: 4 hours

Charging Range: 0° to 40°C (32° to 104°F)

Regulator Approvals

UL and cUL Listed, UL 60950 and UL 1604 and CSA 22.2 #157, FCC Part 15, TUV, CE mark

Standard Communications

RS232; USB

2 Windows CE .NET

This chapter introduces Microsoft Windows CE .NET. While using your 751G Mobile Computer, keep this key point in mind:

Tap **Start** on the task bar, located at the bottom, left corner of the screen, to quickly move to programs, files, and settings. Use the task bar at the bottom of the screen to perform tasks in programs. The task bar includes menus, buttons, and the onscreen keyboard.



Note: Desktop icons and applet icons are shown to the left. Any place that **Start** is mentioned, tap the following Windows icon in the bottom, left corner of your desktop.



Software Builds

Go to “[Software Build Version](#)” on page 14 to determine which Intermecc build is on your unit.

Where to Find Information

This chapter describes your 751G CE .NET applications, and explains how to connect your 751G to a PC, a network, or the Internet. Below is a guide to assist you in using your 751G.

For information on:	See this source:
Programs on your mobile computer.	This chapter and mobile computer Help. To view Help, tap Start > Help .
Additional programs that can be installed on the mobile computer.	The <i>Windows CE .NET Companion CD</i> .
Connecting to and synchronizing with a PC.	The Quick Start Guide or <i>ActiveSync Help</i> on your PC.
Last-minute updates and detailed technical information.	The Read Me files, located in the Microsoft ActiveSync folder on the PC and on the <i>Windows CE .NET Companion CD</i> .
Up-to-date information on your Windows CE .NET device.	msdn.microsoft.com/embedded/downloads/ce/default.aspx

Use these URLs for additional information about Microsoft Windows CE .NET:

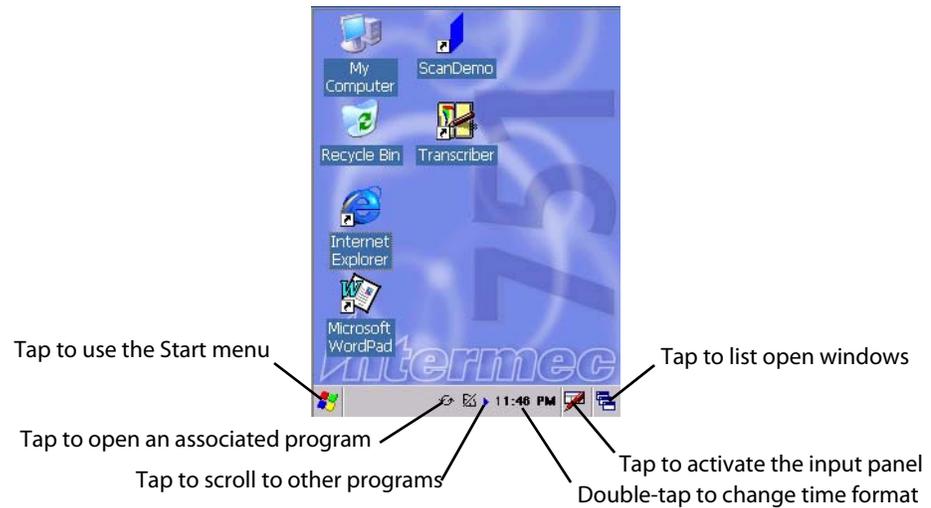
- msdn.microsoft.com/support
- support.microsoft.com
- www.microsoft.com/technet/community/newsgroups/security/default.mspx (*a free support option*)

Basic Skills

Learning to use your 751G is easy. This section describes the basic concepts of using and customizing your 751G.

Desktop Screen

When you turn on your 751G for the first time each day, you see the Desktop screen.



To customize what is displayed on the Desktop screen, including the background image, tap **Start > Settings > Control Panel**, then double-tap the **Display** icon.

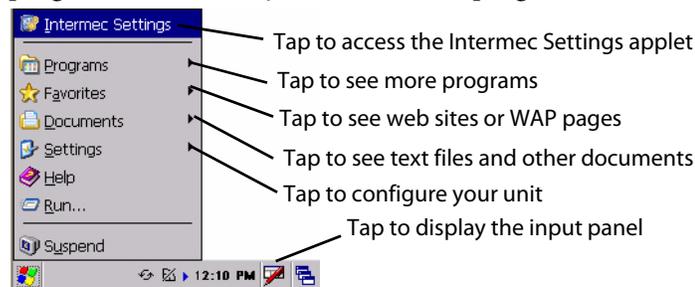
Status icons display information such as low batteries or when the 751G is connected to a PC or to the Internet. You can tap an icon to open the associated setting or program.

Programs

You can switch from one program to another by selecting it from the **Start** menu. (You can customize which programs you see on this menu. For information, see [“Adjusting Settings” on page 27.](#)) To access some programs, tap **Start > Programs**, and then the program name.

Start Menu and Task Bar

The Start Menu is located at the bottom of the screen. It displays the active program, and allows you to switch to programs and close screens.



The task bar, which displays the current time, is at the bottom of the screen. The task bar includes menu names, buttons, and the **Input Panel** icon. Use this task bar to perform tasks in programs.

Notifications

When you have something to do, your device notifies you in any of the following ways. You can choose notification types.

- A message box appears on the screen.
- A sound, which you can specify is played.
- A light flashes on your 751G.

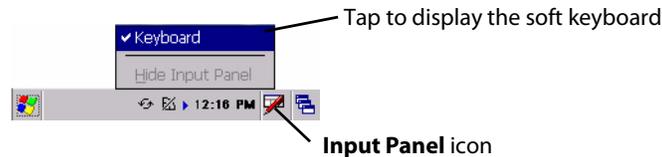
Entering Information

You can enter information on your 751G in several ways, depending on the type of device you have and the program you are using:

Typing	Enter typed text into the 751G. You can do this by tapping keys on the onscreen keyboard or by using handwriting recognition software.
Writing	Using the stylus, write directly on the screen.
Drawing	Using the stylus, draw directly on the screen.

Use the input panel to enter information in any program on your 751G. You can either type using the onscreen keyboard or write using **Transcriber**. The characters appear as typed text on the screen.

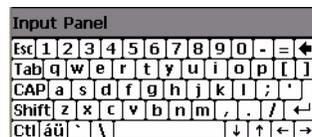
To show the input panel, tap the **Input Panel** icon, then tap **Keyboard**. To hide the input panel, tap the **Keyboard** icon, then tap **Hide Input Panel**.



Typing With the Onscreen Keyboard

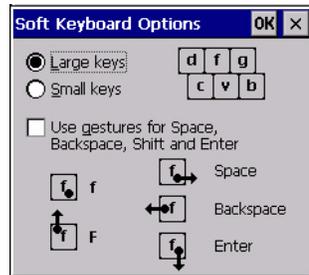
Tap the stylus input icon, then tap **Keyboard**. On the soft keyboard that is displayed, tap the keys with your stylus.

- To type lowercase letters, tap the keys with the stylus.
- To type a single uppercase letter or symbol, tap the **Shift** key. To tap multiple uppercase letters or symbols, tap the **CAP** key. *Note that the CAP key only appears if the keyboard is set to small keys.*
- To convert to uppercase, tap and hold the stylus on a letter and drag up.
- To add a space, drag the stylus to the right across at least two keys.
- To backspace, drag the stylus to the left across at least two keys.
- To insert a carriage return, tap and hold the stylus anywhere on the keyboard and drag down.





If you want to use larger keys, tap **Start > Settings > Control Panel**, then double-tap the **Input Panel** icon. Tap **Options**, then select **Large keys**. Tap **OK**, then **OK** again to close the Input Panel properties.

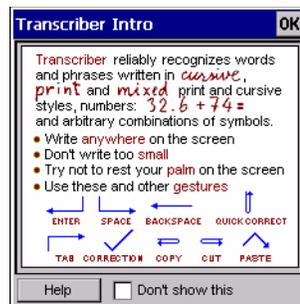


Using Transcriber

With Transcriber, you can write on the screen with the stylus just as you would on paper. You can write a sentence or more of information, then pause and let Transcriber change written characters to typed characters.



For specific instructions on using Transcriber, double-tap the **Transcriber** shortcut on the desktop screen, then tap **Help**. Tap **OK** to close the Transcriber Intro box.



To enable the Transcriber feature, tap the **Transcriber** icon on the task bar, then write anywhere on the screen. *Note the gray box behind the icon.* The input then appears in the active window. To disable the Transcriber, tap the **Transcriber** icon again. *This removes the gray box in the background.*



Transcriber icon

Selecting Typed Text

If you want to edit or format typed text, you must select it first.

- Drag the stylus across the text you want to select.

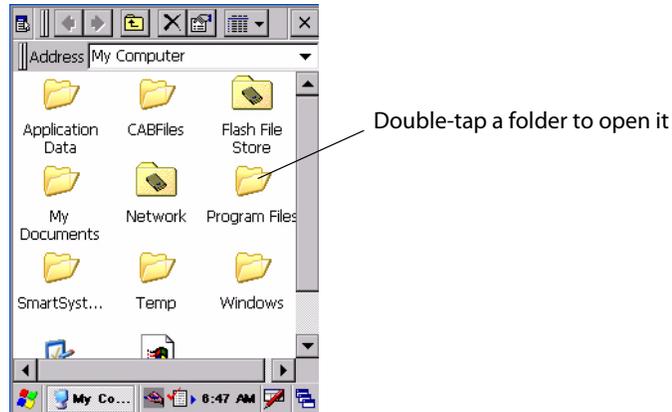
You can cut, copy, and paste text by tapping and holding the selected words and then tapping an editing command on the pop-up menu, or by tapping the command on the **Edit** menu.

Finding and Organizing Information

Use Windows Explorer to find and organize files into folders on the 751G.

To open Windows Explorer

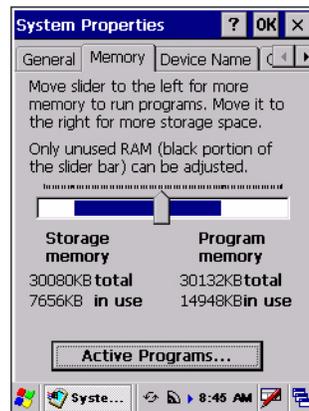
- 1 Tap **Start** > **Programs** > **Windows Explorer**.
- 2 Double-tap any folder to open it.
- 3 Move files by tapping and holding the items you want to move, then tap either **Cut** or **Copy** and **Paste** on the pop-up menu.



You can also use the System applet to pull up a list of active programs currently running on your 751G.

To start Task Manager

- 1 Tap **Start** > **Settings** > **Control Panel**, then double-tap the **System** icon.
- 2 Tap the **Memory** tab, then tap **Active Programs** for the Task Manager.



To use a different application, select an application, then tap **Switch To**. To stop an application, select that application, then tap **End Task**.



Customizing Your Computer

You can customize your 751G by adjusting settings and installing software.

Adjusting Settings

You can adjust settings to suit the way you work. To see available options, tap **Start > Settings > Control Panel**, then double-tap any of the applets. You might want to adjust the following:



Date/Time

Date/Time

To change the time or calendar.



Display

Display

To customize the look of the desktop.



Owner

Owner

To enter your contact information.



Password

Password

To limit access to your 751G.



Power

Power

To maximize battery life.

Adding or Removing Programs

Programs added to your 751G at the factory are stored in ROM (Read Only Memory). You cannot remove this software, and you cannot accidentally lose ROM contents. All other programs and data files added to your 751G after factory installation are stored in RAM (Random Access Memory).

You can install any program created for your 751G, as long as your 751G has enough memory. The most popular place to find software for your 751G is on the Windows CE .NET web site (msdn.microsoft.com/embedded/downloads/ce/default.aspx).

Adding Programs Using Microsoft ActiveSync

Install applicable software on your PC before installing it on the 751G.



System

- 1 Determine your 751G and processor type so that you know which version of the software to install. Tap **Start** > **Settings** > **Control Panel**, then double-tap the **System** icon. Note the processor information on the **General** tab beneath the **Computer** heading.



- 2 Download the program to your PC (or insert the CD or disk that contains the program into your PC). You may see a single *.EXE or *.ZIP file, a SETUP.EXE file, or several versions of files for different 751G types and processors. Be sure to select the program designed for the Windows CE .NET and your 751G processor type.
- 3 Read any installation instructions, Read Me files, or program documentation. Many programs provide special installation instructions.
- 4 Connect your 751G and PC.
- 5 Double-click the *.EXE file.
 - If the file is an installer, the installation wizard begins. Follow the directions on the screen. Once the software is installed, the installer automatically transfers the software to your 751G.
 - If the file is not an installer, an error message stating that the program is valid but it is designed for a different type of computer is displayed. Move this file to your 751G. If you cannot find any installation instructions for the program in the Read Me file or documentation, use Microsoft ActiveSync Explore to copy the program file to the “My Computer\Program Files” folder on your 751G. For information on copying files using Microsoft ActiveSync, see *ActiveSync Help*.

Once installation is complete, tap **Start** > **Programs**, and then the program icon to switch to it.

Adding a Program Directly from the Internet



System

- 1 Determine your 751G and processor type so that you know which version of the software to install. Tap **Start > Settings > Control Panel**, then double-tap the **System** icon. Note the processor information on the **General** tab beneath the **Computer** heading.
- 2 Download the program to your 751G straight from the Internet using Internet Explorer. You may see a single *.EXE or *.ZIP file, a SETUP.EXE file, or several versions of files for different 751G types and processors. Be sure to select the program designed for the Windows CE .NET and your 751G processor type.
- 3 Read program installation instructions, Read Me files, or other documentation. Many programs provide installation instructions.
- 4 Tap the file, such as *.EXE file to start the installation wizard. Follow the directions on the screen.

Adding a Program to the Start Menu

You can either use Windows Explorer on the 751G to move the program to the “\My Computer\Windows\Start Menu” folder, or use Microsoft ActiveSync on the PC to create a shortcut to the program and place the shortcut in the “\My Computer\Windows\Start Menu” folder.

Using Windows Explorer on the Computer

Tap **Start > Programs > Windows Explorer**, and locate the program. Tap and hold the program and tap **Cut** on the pop-up menu. Open the “\My Computer\Windows\Start Menu” folder, tap and hold a blank area of the window, and tap **Paste** on the pop-up menu. The program now appears on the **Start** menu. For more information on using Windows Explorer, see [“Finding and Organizing Information” on page 26](#).

Using Microsoft ActiveSync on the PC

Use the Explore in Microsoft ActiveSync to explore your 751G files and locate the program. Right-click the program, and then click **Create Shortcut**. Move the shortcut to the “\My Computer\Windows\Start Menu” folder. The shortcut now appears on the **Start** menu. For more information, see *ActiveSync Help*.

Removing Programs



Remove Programs

Tap **Start > Settings > Control Panel**, then double-tap the **Remove Programs** icon.

If the program does not appear in the list of installed programs, use Windows Explorer on your 751G to locate the program, tap and hold the program, and then tap **Delete** on the pop-up menu.

Microsoft ActiveSync



Tap **Start** > **Settings** > **Control Panel**, then double-tap the **PC Connection** icon. Tap **Change Connection**, then select the baud rate from the drop-down list.



Visit the following Microsoft web site for the latest in updates, technical information, and samples:

msdn.microsoft.com/embedded/downloads/ce/default.aspx

With Microsoft ActiveSync, you can back up and restore your 751G data, and copy files between your 751G and PC.

Install Microsoft ActiveSync on the desktop of your PC from the following URL. For more information, on installing Microsoft ActiveSync, see your Quick Start card. ActiveSync is already installed on your 751G.

msdn.microsoft.com/downloads/

After installation is complete, the Microsoft ActiveSync Setup Wizard helps you connect your 751G to your PC or set up a partnership so you can browse for or move information between your 751G and your PC.

Disconnect the 751G from your PC and you are ready to go!



Note: While Microsoft ActiveSync does synchronize files between your PC and your 751G, the 751G does not support applications such as Calendar, Contacts, Tasks, Inbox, Channels, and Pocket Access.

Microsoft WordPad

WordPad works with Microsoft Word on your desktop to access copies of your documents. You can create new documents on your 751G, or you can copy documents from your desktop to your 751G. Synchronize documents between your desktop and your 751G to have up-to-date content in both locations.



To access WordPad, either double-tap the Microsoft WordPad icon on your desktop, or select **Start** > **Programs** > **Microsoft WordPad**.

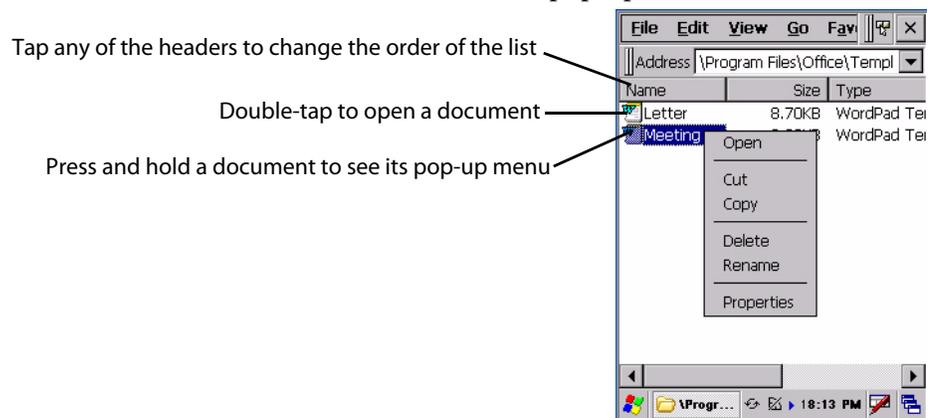
For more information on using Microsoft WordPad, select **Start** > **Help** > **WordPad**.

Creating a Document

Use WordPad to create documents, such as letters or meeting minutes. To create a new file, tap **File > New**, then select either a blank document or a template, depending on what you have selected in the **Tools > Options** dialog box. Select an input mode from the **View** menu.

You can open only one document at a time; when you open a second document, you have to save the first. Documents you create or edit are usually saved as WordPad (.WPD), but you can also save documents in other formats such as Word (.DOC) or Rich Text Format (.RTF).

Windows Explorer contains a list of files stored on your 751G. Double-tap a file to open it. To delete, make copies of, and rename files, tap and hold a file in the list, then, select the action on the pop-up menu.



You can change the zoom magnification by tapping **View > Zoom**, then select the percentage you want. Select a higher percentage to enter text and a lower one to see more of your document.

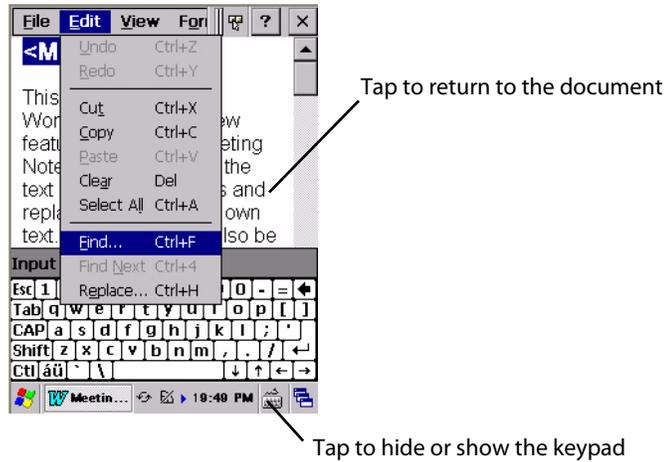
If you are opening a Word document created on a desktop, you may select **View > Wrap to Window** so that you can see the entire document.

To check spelling, select text, then tap **Tools > Spell Check**. To use your new document as a template, move the document to the Templates folder.

Typing Mode

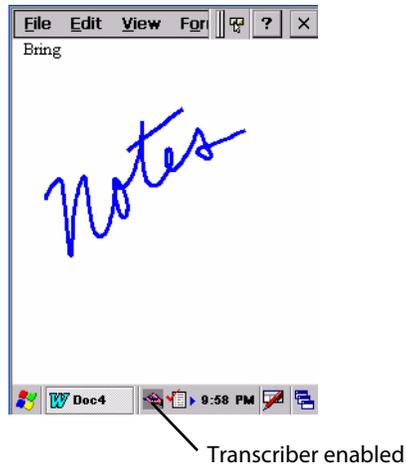
Using the input panel, enter typed text into the document. For more information on entering typed text, see [“Entering Information” on page 24](#).

To format existing text and to edit text, first select the text. You can select text as you do in a Word document, using your stylus instead of the mouse to drag through the text you want to select. You can search a document to find text by tapping **Edit > Find**.



Writing Mode

With Transcriber enabled, use your stylus to write directly on the screen. The zoom magnification is greater than in typing mode to allow you to write more easily. For more information on writing and selecting writing, see [“Entering Information” on page 24](#).



Synchronizing WordPad Documents

WordPad documents can be synchronized with Word documents on your desktop. To synchronize files, first select the Files information type for synchronization in ActiveSync. When you select Files, the My Documents folder for the 751G is created on your desktop. Place all files you want to synchronize with the device in this folder. Password-protected files cannot be synchronized.

All WordPad files stored in My Documents and its subfolders are synchronized with the desktop. ActiveSync converts documents during synchronization.

For more information on synchronization or file conversion, see *ActiveSync Help* on the desktop.



Note: When you delete a file on either your desktop or your 751G, the file is deleted in the other location when you synchronize.

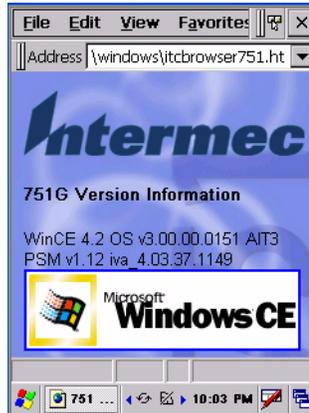
Internet Explorer

Use Microsoft Internet Explorer to view web sites or WAP pages. To do this, create the connection first via an ISP or network, as described in [“Connecting to an Internet Service Provider” on page 113](#).

When connected to an ISP or network, you can also download files and programs from the Internet or intranet.



To switch to Internet Explorer on your 751G, double-tap the **Internet Explorer** icon on your desktop or select **Start > Programs > Internet Explorer** to access the application



Viewing Mobile Favorites and Channels

- 1 Tap **Favorites** from the tool menu to display your list of favorites.
- 2 Tap the page you want to view.



Tap to add or delete a folder or favorite link

Tap the favorite to view

Browsing the Internet

- 1 Set up a connection to your ISP or corporate network using information as described in [“Connecting to an Internet Service Provider” on page 113](#).

- 2 To connect and start browsing, either tap **Favorites** from the toolbar, then tap the favorite to view, or in the **Address** bar that appears at the top of the screen, enter the web address you want to visit using the input panel, then tap the [**Enter**] key on the panel to go to that web site.



Tap the drop-down arrow to select from previously entered addresses.



Tap to bring down a list of addresses



Note: To add a favorite link while using the 751G, go to the page you want to add, select **Favorites > Add to Favorites**.

3 Configuring the Computer

There are multiple ways to get an application to your 751G Mobile Computer; like there are multiple ways to package the application for delivery.



Note: Desktop icons and applet icons are shown to the left. Any place that **Start** is mentioned, tap the following Windows icon in the bottom, left corner of your desktop.



Configuring Parameters

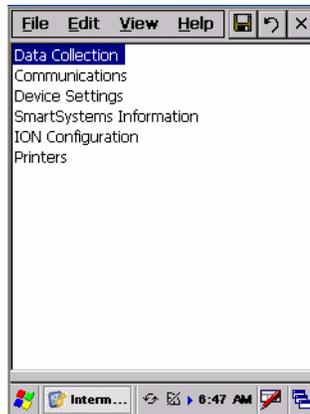
You can configure many parameters on the 751G, such as the bar code symbologies it decodes or the network settings. These characteristics are controlled by configuration parameters. The values you set for these configuration parameters determine how the computer operates. Use configuration commands to configure the 751G.

Configuring the Computer With Intermec Settings

Use the Intermec Settings applet to configure the 751G and view system information. You can access the Intermec Settings applet while running any application.



From the 751G desktop, select **Start > Settings > Control Panel**, then double-tap the **Intermec Settings** icon.



For detailed information on most of the commands available in the Intermec Settings applet, see the *Intermec Computer Command Reference Manual* (P/N 073529) via the Intermec web site. Go to “[Before You Begin](#)” for access information.

Synchronizing the Computer System Time with a Time Server

It is important that the time on all of your 751Gs be synchronized with a network time server to ensure real-time communications and updates. Network time servers acquire Coordinated Universal Time (UTC) from an outside source such as the U.S. Naval Observatory (USNO). The 751G uses Simple Network Time Protocol (SNTP) to synchronize with a network time server.

The default reference time server is the USNO (tock.usno.navy.mil). To synchronize the time on your 751G with this time server, you must have a valid connection to the Internet. You can also synchronize the 751G system time with a corporate network server within your firewall that is SNTP-capable. To use an internal corporate network server, you need to set the command name in the registry.

Configuring the Computer through the Network

You can change the configuration parameters of the 751G by sending commands through a host computer or through the network. If you are using a network, you can configure one or more 751Gs at a time. You can remotely configure the wireless 751G by sending a command from an application on the host computer. *Note that you cannot set all parameters through the network. You can only set those commands that have a syntax in the Intermec Computer Command Reference Manual.*



Note: You can continue running an application on the 751G while configuring it from the host computer.

Configuring the Computer in a TCP/IP Direct Connect Network

Use the host computer to configure a wireless 751G in a TCP/IP network. To send and receive configuration data, write a host application that can communicate with the 751G directly through an access point or through the network. Use the TMF protocol to send and receive transactions between the host application and the 751G.

To set up the host computer, verify communication with the 751G. To set up the application, prepare and write a host application that can communicate with the IAS and send transactions to and receive transactions from the 751G in this format:

```
TMF field  commands
```

where:

TMF field is a 2-byte field containing one of these values:

- CG Configuration Get request sent from the host application.
- Cg Configuration Get response sent from the 751G to the host computer.
- CS Configuration Set request sent from the host application.
- Cs Configuration Set response sent from the 751G to the host computer.

commands are the reader and configuration commands to set on the 751G or the current value to retrieve from the 751G. To save configuration changes in flash memory, send the . +1 reader command as the last command. See the *Intermec Computer Command Reference Manual* for a list of commands.

Example

In the host application, you want to get the current values of two configuration commands from the 751G. Send the CG\$+NABV transaction from the host application.



Note: The transaction header is not shown in this example. You do not need a transaction header for a host application in a TCP/IP network, but you do for a UDP Plus network.

where:

- CG is a TMF Configuration Get request.
- \$+ is the Change Configuration reader command.
- BV is the Beeper Volume configuration command.

The computer returns the CgS+BV4 transaction to the host application:

- Cg is a TMF Configuration Get response.
- \$+ is the Change Configuration reader command.
- BV4 means the Beeper Volume configuration command is set to a value of 4, which is a very high beeper volume.

Configuring the Computer in a UDP Plus Network

Use the host computer to configure a 751G in your wireless network. To send and receive configuration data or files, write a host application that can communicate with an Intermec Application Server (IAS) (*formerly Gateway or DCS 30X*).

For help, see the appropriate Gateway or DCS 30X User's Manual. Use the Terminal Message Format (TMF) protocol to send and receive transactions between the host application and the 751G.

To set up the IAS, configure a peer-to-peer destination name for the host application. Create a \$NGCFGRSP transaction ID that routes to this destination name. The IAS uses the transaction ID to route responses from the 751G back to the host application. \$NGCFGRSP is a special transaction ID that the server uses to forward configuration response data from a 751G.

All configuration responses are routed with the \$NGCFGRSP transaction ID. The IAS cannot track multiple applications sending reader or configuration commands. If you have two host applications sending reader or configuration commands, they must both be configured to receive the \$NGCFGRSP transactions, and receive all 751G responses.

To set up the host computer, verify host computer-to-IAS communication.

To set up the application, prepare and write a host application that can communicate with the IAS and send transactions to and receive transactions from the 751G in this format.

```
transaction header          TMF field  commands
```

where:

transaction header is a 96-byte field with message number, date, time, source application ID, destinations application ID, transaction ID, and other. Set the system message (SYS\$MSG) flag to E in the transaction header.

TMF field is a 2-byte field containing one of these values:

- CG Configuration Get request sent from the host application.
- Cg Configuration Get response sent from the 751G to host computer.
- CS Configuration Set request sent from the host application.
- Cs Configuration Set response sent from the 751G to the host computer

commands are the reader and configuration commands to set on the 751G or the current value to retrieve from the 751G. To save configuration changes in flash memory, send the .+1 reader command as the last command. See the *Intermec Computer Command Reference Manual* for supported commands.

Using Configuration Parameters

A configuration parameter changes the way the 751G operates. Use either of these methods to execute configuration parameters:

Scan EasySet bar code labels:

Use the EasySet application from Intermec Technologies Corporation to print configuration labels. Scan labels to change imager configuration and data transfer settings. See the EasySet online help for information.

Send Reader Commands through the Network or from an Application

See the *Intermec Computer Command Reference Manual* for information.

Configuring the Printer

The 751G works with a Zebra PT403 Portable Printer, which interfaces through an I/O adapter (P/N 074143). Contact an Intermec representative for information about this printer.

Methods for printing using Windows CE at this time is as follows:

- Add port drivers to print ASCII directly to the port.
- Use LinePrinter ActiveX Control from the Software Developer's Kit (SDK) - see the *SDK User's Manual* for more information.

Directly to a Port

Printing directly to the port sends RAW data to the printer. The format of this data depends upon your application and the printer capabilities.

You must understand the printer commands available for your specific printer. Generally, applications just send raw ASCII text to the printer. Since you are sending data to the printer from your application directly to the port you are in complete control of the printers operations. This allows you to do line printing (*print one line at a time*) rather than the page format printing offered by the GDI approach. It is also much faster since data does not have to be converted from one graphics format to the other (display to printer). Most Intermec printers use Epson Escape Sequences to control print format operations.

These commands are available in documentation you receive with your printers or from technical support. Win32 APIs are required to print directly to the port.

Directly to a Generic Serial Port

To print directly to a generic serial port printer (non-Intermec printers):

- Use CreateFile() to open ports - COM1: can be opened on most devices.
- Use WriteFile() to send data directly to the printer.
- Use CloseHandle() when you are finished printing to close the port.

Configuring the Scanner

The 751G comes with a 2D Imager that decodes several stacked 1D and 2D symbologies, including PDF417, Data Matrix, and MaxiCode without “painting.” It can also read 1D codes from any orientation. For example the scan beam does not need to align perpendicular to the symbol in order to read it. Photography is a secondary application; the lens in the device favors bar code reading. Photos are 640x480, 256 gray-scale.

An ImageDemo application is available to demonstrate imager features. See the *ImageDemo User’s Guide* (P/N 934-002-001) for more information.

Scanner Control and Data Transfer

The data server and associated software provide ways to handle scanner control, data transfer between the scanner subsystem and user applications:

- **Automatic Data Collection COM Interfaces:**
These COM interfaces allow user applications to receive bar code data, and configure and control the bar code reader engine.
- **ITCAxBarCodeReaderControl functions:**
These ActiveX controls allow user applications to collect bar code data from the scanner, to configure the scanner, and to configure audio and visual notification when data arrives.
- **ITCAxReaderCommand functions:**
Use these ActiveX controls to modify and retrieve configuration information using the reader interface commands.
- **Scanning EasySet bar code labels:**
You can use the EasySet bar code creation software from Intermec Technologies Corporation to print configuration labels. Scan the labels to change the scanner configuration and data transfer settings.

Use the software to print scannable configuration labels to change your configuration settings. For more information, see the EasySet online help. EasySet is available from the Intermec Data Capture web site.

Data Collection Configuration

From the 751G, tap **Start > Intermec Settings** to configure scanner settings. See the *Intermec Computer Command Reference Manual* online manual for information about the settings you can configure with this applet. Note that these are in alphabetical order.

Codabar	Code 93	MaxiCode	RSS 14
Codablock A	Code 128	Micro PDF417	RSS Expanded
Codablock F	Data Matrix	MSI	RSS Limited
Code 2 of 5	EAN/UCC Composite	PDF417	Telepen
Code 11	Interleaved 2 of 5	Plessey	UPC/EAN
Code 39	Matrix 2 of 5	QR Code	

Changing Comm Settings

Tap **Change Comm Settings** to configure the settings for the COM1 port. Current settings are restored after a warm-boot is performed, but are lost after a cold-boot is performed. When these settings are not changed, the **OK** button is disabled (grayed out). When changes are made, tap **OK** after it is enabled to accept these changes.

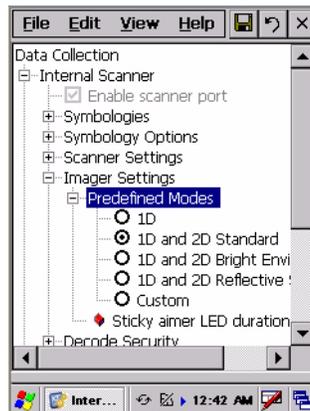
Baud Rate	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
Data Bits	7 or 8
Parity	None, Odd, Even, Mark, Space
Stop Bits	1 or 2
Flow Control	None or Hardware

Improving the Performance of the Area Imager

If you have problems scanning a bar code with the 2D imager, try doing these tips to improve the performance of your imager:



- Tap **Start > Settings > Control Panel**, then double-tap the **Intermec Settings** icon to access the applet. Tap (+) to expand **Data Collection > Internal Scanner > Imager Settings > Predefined Modes**, then select one of the following:



Select this option:	If you are scanning:
1D	only 1D labels
1D and 2D Standard	all other cases
1D and 2D Bright Environment	in high ambient light (like outdoors in sunshine)
1D and 2D Reflective Surface	glossy labels

Select Custom to access all standard imager settings such as “Lighting Goal” or “Lighting Mode.” More information about these settings, commands, and parameters are found in the *Intermec Computer Command Reference Manual* available from the Intermec web site at www.intermec.com.

- Keep your hand as steady as possible while scanning a label.
- Position the imager as close to the bar code as possible while still being able to capture the entire bar code.
- Enable only the bar codes that you need to use every day.

Reading Distances

Typical reading distances are done in an office environment using office lights (4 lux). Minimum distances are measured in the dark (0 lux). Both reading distances are provided in respective scan engine integration guides. Contact your Intermec representative for more information.

The minimum standard reading distances for 751Gs built with integrated scan engines are shown below. When correctly mounted, an exit window reduces reading distances by about 4%.

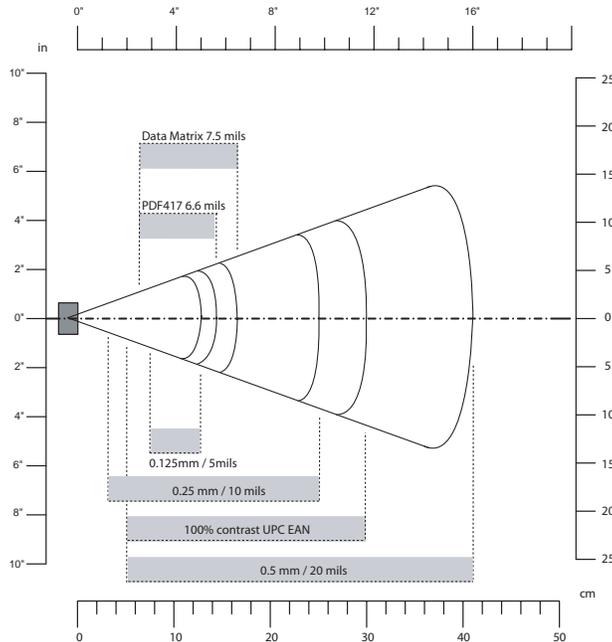
2D Area Imager Reading Distances with 0.04" Setbacks

Symbology	Density	Near Distance	Far Distance
MaxiCode	35 mil	4.98 cm/ 1.96"	33.92 cm/ 12.96"
Data Matrix	15 mil	9.30 cm/ 3.66"	16.41 cm/ 6.46"
PDF417	10 mil	7.77 cm/ 3.06"	22.76 cm/ 8.96"
	8 mil	8.28 cm/ 3.26"	20.22 cm/ 7.96"
	6.6 mil	11.33 cm/ 4.46"	15.77 cm/ 6.21"
Code 39	15 mil	5.23 cm/ 2.06"	29.87 cm/ 11.76"
	10 mil	8.03 cm/ 3.16"	23.27 cm/ 9.16"
	8 mil	8.79 cm/ 3.46"	19.20 cm/ 7.56"
UPC	13 mil	6.25 cm/ 2.46"	31.65 cm/ 12.46"

EA11 Standard Minimum Reading Distances with 0.04" Setbacks

Symbology	Density	Minimum Distance	Maximum Distance
Code 39	0.125 mm/ 5 mil	7.26 cm/ 2.86"	12.09 cm/ 4.76"
	0.20 mm/ 8 mil	3.96 cm/ 1.56"	20.98 cm/ 8.26"
	0.25 mm/ 10 mil	3.45 cm/ 1.36"	25.04 cm/ 9.86"
	0.50 mm/ 20 mil	4.98 cm/ 1.96"	40.28 cm/ 15.86"
UPC/EAN	0.33 mm/ 13 mil	4.98 cm/ 1.96"	29.92 cm/ 11.66"
Datamatrix	0.191 mm/ 7.5mil	3.71 cm/ 2.46"	16.41 cm/ 6.46"
	0.254 mm/ 10 mil	5.98 cm/ 1.96"	20.73 cm/ 8.16"
	0.381 mm/ 15 mil	*	27.58 cm/ 10.86"
PDF417	0.168 mm/ 6.6 mil	6.25 cm/ 2.46"	13.87 cm/ 5.46"
	0.254 mm/ 10 mil	4.47 cm/ 1.76"	21.74 cm/ 8.56"
	0.381 mm/ 15 mil	4.98 cm/ 1.96"	33.43 cm/ 13.16"

* Minimum distance depends on symbology length and scan angle.



EA11 Standard - Minimum Reading Distances

Installing Applications on the Computer

Consider one of the following options to get the package to the preferred location on your 751G:

- Microsoft ActiveSync
- Secure Digital storage cards ([page 44](#))
- SmartSystems Console ([page 45](#))
- Wavelink Avalanche ([page 45](#))

Using Microsoft ActiveSync



Note: These instructions assume the 751G Management Tools were installed on your desktop.

The Microsoft ActiveSync tool is located on the *751G Companion CD*. See [Chapter 2, “Windows CE .NET”](#) for information about this tool as provided by Microsoft Corporation.

This can be a serial, USB, or 802.11i Microsoft ActiveSync connection. Files can be copied using Windows Explorer on a PC or a laptop computer. This is usually good when updating a few 751Gs.

These instructions assume that Microsoft ActiveSync was installed onto your PC and is up and running. If not, go to [Chapter 2, “Windows CE .NET”](#) for an URL from which to download the latest application.

To use Microsoft ActiveSync

- 1 Use an ActiveSync cable to connect the 751G to your PC.
- 2 Wait for a “Connected” message to appear in the Microsoft ActiveSync application to signal a connection to the 751G. If necessary, select **File > Get Connected** to initiate a connection.



- 3 Click **Explore** to access the “Mobile Device” folder on your unit.
- 4 From your PC, select **Start > Windows Explorer**, then browse the “C:\Intermec\751G Mgmt Tools\CabFiles” path for any CAB files needed for your 751G. Right-click the file, then select **Copy**.
- 5 In the “Mobile Device” folder, go to the folder where you want the files located on the 751G, do a right-click for a pop-up menu, select **Paste**.
- 6 When the files are pasted, perform a warm-boot on the 751G, then wait for the LED on the top left of your keypad to stop blinking. Tap **Start > Programs > Windows Explorer** to locate the newly copied executable files, then tap these files to activate their utilities.

Using a Storage Card

Use the following steps to install an application using a storage card:

- 1 Suspend the 751G and remove its Secure Digital storage card.
- 2 Using a Secure Digital adapter, place the Secure Digital storage card in your PC card drive, then create a subdirectory on the Secure Digital drive in which to store your application.
- 3 Copy your application, data files, and all required DLLs and drivers to the subdirectory created on the Secure Digital storage card.
- 4 Add your application to the AutoUser.dat file on the “\SDMMC Disk\2577” directory with the following statement:

```
RUN=\your directory>\<yourapp.exe>
```

 where *your directory* is the directory on the Secure Digital storage card where the application was installed, and *yourapp.exe* is the name of your application. Finish the “RUN=” statement with a CR/LF combination. There may be multiple run statements in the file.

- 5 Add the AuotRun.exe file to the “\SDMMC Disk\2577” folder.
- 6 Remove the Secure Digital storage card from your PC and reinstall it into the 751G, then warm-boot the 751G to add these files to the Secure Digital storage card.

If the AutoUser.dat file is found and the “RUN=” statement is correct, the task manager launches and executes your program on startup.

Using the SmartSystems Console



You can use the SmartSystems Console to drag-and-drop Intermec applications onto your 751Gs. The 751G ships with the SmartSystems client loaded on it. The console is part of SmartSystems Foundation and is available from the Intermec web site.

- 1 Download the file from the Intermec web site, unzip it on your desktop.
- 2 From the SmartSystems Console, drag-and-drop the application onto each 751G discovered in your network.

To download SmartSystems Foundation, go to www.intermec.com/idl and open the Device Management page. For information on using the SmartSystems Console, see its online help.

Using Wavelink Avalanche

You can use the Wavelink Avalanche device management system to install applications on all of your wireless 751Gs.

The wireless 751G ships with the Avalanche Enabler already loaded on it. The Avalanche Enabler is configured to activate automatically (typically on a clean-boot).



Note: If you manually activate the Avalanche Enabler on the 751G, you may be prompted for a password when you exit the Avalanche Enabler. The default password is `leave`.

When the Avalanche Enabler is activated, the 751G attempts to connect to the Avalanche Agent. When the 751G connects to the Agent, the Agent determines whether an update is available and immediately starts the software upgrade, file transfer, or configuration update.

- 1 Use the Avalanche Management Console to install software packages and updates. For help, see its online help, contact an Intermec representative, or visit the Wavelink web site at www.wavelink.com.
- 2 Schedule the 751G updates or manually initiate an update using the Avalanche Management Console.

Installing Cabinet Files

Cab files (*short form of “cabinet” files*) are compressed folders as defined by Microsoft. A “cabinet” file is a single file, usually suffixed with .cab, that stores compressed files in a file library. A compressed file can be spread over several cabinet files. During installation, the setup application decompresses the files stored in a cabinet and copies them to the user’s system.

For the 751G, cab files register DLLs, create shortcuts, modify registry entries, and run custom setup programs. Tap a cab file to extract that file or place the cab file on one of the approved storage devices in the “\CabFiles” folder, then perform a warm-boot on the 751G. There are two methods available to extract a cab file:

- Tap a cab file to extract it. With this method, the cab file is automatically deleted when the extraction process is successful, *unless* the cab file is set with the read-only attribute.
- Use the AutoCab method to extract all files when a cold-boot is performed on the 751G. This method is on the *Intermec Developer Library CD*, see its *Software Tools User’s Manual* for information.

Developing Applications for the Computer

751Gs run applications programmed in Microsoft Visual Studios 2005. Use this chapter to understand what you need to develop a new application for the 751G.



Note: Microsoft eMbedded Visual C++ 4.0 may be used, but some features are not available.

Use Resource Kits from the Intermec Developer Library (IDL) to develop applications for your 751G, which are downloadable from the Intermec web site at www.intermec.com/idl.

You need the following hardware and software components to use the resource kits:

- Pentium desktop, 400 MHz or higher
- Windows 2000 (Service Pack 2 or later) or Windows XP (Home, Professional, or Server)
- For native and managed development, Microsoft Visual Studio 2005
- 128 MB RAM (196 MB recommended)
- 360 MB hard drive space minimum installation (720 MB for complete)
- CD-ROM drive compatible with multimedia desktop specification
- VGA or higher-resolution monitor (Super VGA recommended)
- Microsoft Mouse or compatible pointing device

Packaging Applications for the Computer

- You could package an application as a cabinet (cab) file. *Recommended*
- For simple applications, the application itself may be the file to deliver.
- It could be a directory structure that contains the application, supporting files like ActiveX controls, DLLs, images, sound files, and data files.

Consider any of these when choosing a storage location for applications:

- In the 751G are two built-in storage options: the Object Store and the Persistent Storage Manager (PSM). The Object Store is RAM that looks like a disk. Anything copied here is deleted when a cold-boot is performed on the unit. The PSM is an area of storage embedded in a section of the system's FLASH memory. This storage area is *not* erased during a cold-boot, but it may be erased during the reflash. You also have the option to store a persistent registry to the PSM region.
- If the optional Secure Digital storage card is in the system, then consider this card the primary location to place applications installation files. The "\SDMMC Disk" folder represents the Secure Digital card.
- Use the small non-volatile Flash File Store region to hold CAB files that rebuild the system at cold-boot or install applications from a CAB file *into* the Flash File Store so they are "ready-to-run" when a cold-boot is performed. Since the FLASH in the system has a limited number of write cycles, do not use the Flash File Store for excessive writing purposes; however, reading is okay.

Files copied to any of these locations are safe when a cold-boot is performed on a 751G — *providing the AutoRun system is installed in the appropriate location*. You can find this system in the 751G Management Tools portion of the *Intermec Developer's Library CD*. Copying a cab file to the "\Flash File Store\Persistent Copy\CabFiles" folder automatically extracts that cab file on every cold-boot to ensure that your system is properly set up (see "[Installing Cabinet Files](#)" on page 46).

Launching Your Application Automatically



Note: This describes the system component startup for Intermec-provided components only. It does not describe the bootstrap loader process. It only describes the component installation process provided by Windows. It is assumed that you understand the Microsoft Mobile startup procedures and are familiar with how Microsoft components start up.

You can configure the various media used in the Windows system with a folder name and can change the media in the registry of the system. Many of the startup components rely on folder names to locate information files, applications, or other related data.

The registry keys used by FolderCopy and other startup components to retrieve the folder names are as follows:

Flash File Store

[HKLM\Drivers\BuiltIn\FIshDrv]

“FolderName”=“Flash File Store”

SD Card (Storage Card)

[HKLM\System\StorageManager\Profiles\SDMMC]

“Folder”=“Storage Card”

During normal Windows system startup, there are Intermec-specific and non-Intermec components that require an orderly start to properly function. These non-Intermec components may also need to start themselves so the Windows device can function properly. Since there are possible configurations that come from using one or more optional built-in peripheral devices, the platform components starting on the next page are required to manage startup.

RunAutoRun

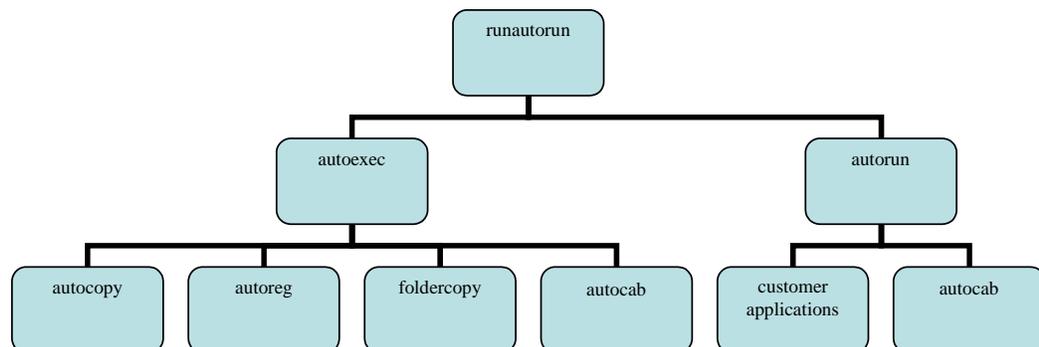
System components are installed and configured during the power up process from a single starting point. RunAutoRun (RunAutoRun.exe), built into the operating system image and located in the “\Windows” folder, checks for AutoExec (AutoExec.exe) in the “\Flash File Store\2577” folder.

Folder names used for the mounted volumes are retrieved from the registry to maintain coherence with the naming of the mounted volumes on the platform. These folder names are not hard-coded.

AutoExec is reserved for Intermec use to configure Intermec-specific applications. It launches the CAB installer, AutoCab (AutoCab.exe), to install platform cab files to the system, such as Intermec Data Collection.

When the AutoExec is complete, RunAutorun then checks for the existence of AutoRun (AutoRun.exe) and executes this program from the first media it is found on. This order is Secure Digital (“\SDMMC Disk\2577”), Object Store (“\2577”), Flash File Store (“\Flash File Store\2577”).

AutoRun is reserved for customer use to configure application launch sequences. It launches the AutoCab installer and any customer programs added to the AutoUser.dat file. Shown is the hierarchy of these files:



AutoExec

AutoExec (AutoExec.exe) automates operations such as pausing, launching processes, or signaling, and is configured through the AutoExec data file (AutoExec.dat). This script file must be in the same directory as the program itself.



Note: Intermec considers the usage of the AutoExec data file as “Intermec Private.” AutoExec installs Intermec applications such as Data Collection, Security Supplicants, Intermec Management, applets, and shortcuts from components found in the Flash File System. Do not modify the AutoExec data file. Instead, use the AutoRun program to add software components.

Usage:

```
AutoExec [-%[W]] [-E=["X"]] [-F=["Y"]] [-LOG=] [-W=[Z]]
```

-%	Passes an ID to use in a call to SignalStarted. This argument is useful only during system startup that relies on a SignalStarted to call. W is an integer value.
-E	Passes a signal event name to use when autoexec completes. X is a string value.
-F	Overrides data file to use. Must be a fully qualified name. Default is “autoexec.dat” in same location as AutoExec.exe program. Y is string value.
-LOG	Set to any value logs activity to AutoExec.txt (in the same location as the AutoExec.exe program). Default is disabled.
-W	Pauses the autoexec process by calling sleep for the number of seconds specified by Z. Z is an integer value.

Process return code uses standard error codes defined in WinError.h.

Keywords that AutoExec supports are:

QUIET	Enables user notification when an error occurs.
LOGGING	Enables logging to a trace file.
SIGNAL	Enables the specified named event and is immediately signaled. Useful for notifying other components of the current status.
CALL	Processes another .dat file. When called file is complete, file is resumed.
RUN	Runs a program with a <i>SW_SHOWNORMAL</i> attribute. Autoexec does not wait for the child process to exit.
LOAD	Runs a program with a <i>SW_HIDE</i> attribute. Autoexec waits for 60 seconds for the child process to exit or <i>EXECWAIT</i> seconds if set.
EXEC	Runs the specified program. AutoExec waits 60 seconds for the child process to exit or <i>EXECWAIT</i> seconds if set.
EXECWAIT	Changes the default EXEC wait time from 60 seconds to the number of seconds specified. There is a maximum 10-minute limit imposed.
WAIT	Forces a sleep for the specified number of seconds to occur.
WAITFOR	Forces a sleep until the named event is signaled.

Examples of keyword usage are as follows:

```
; Allow message pop up if an error occurs.
QUIET 0

; Log any debug output to a trace file.
LOGGING 1
```

```
; Perform a SetEvent on the event name "autoexec_started".
SIGNAL "autoexec_started"

; Include this child data file, childexec.dat.
CALL "\childexec.dat"

; Use autocopy to copy the audio control panel from flash file store to the windows
directory. Wait for up to 60 seconds for it to exit.
EXEC "\Flash File Store\SYSTEM\autocopy.exe" -S"\Flash File Store\System\CPLAudio.cpl"
-D"\Windows\CPLAudio.cpl"

; Change the default EXEC wait time to 90 seconds.
EXECWAIT 90

; Suspend processing any commands for 10 seconds.
WAIT 10

; Suspend processing any commands until event called MyEventName is signaled.
WAITFOR "MyEventName"
```

AutoRun

AutoRun (AutoRun.exe) automates operations such as launching other processes and is configured through the AutoRun data file (AutoRun.dat). This file must be in the same directory as the program itself.

AutoRun supports the following script commands in AutoUser.dat and AutoRun.dat.



Note: If you need to add steps at boot time, add them to AutoUser.dat, not to AutoRun.dat. AutoRun.dat is provided by Intermec and is subject to change. AutoUser.dat is the designated place for the end user to add steps to the boot time process.

EXEC	Launches a specified program, waits for it to complete (up to 10 minutes).
CALL	Processes a specified file of commands and returns.
CHAIN	Processes a specified file of commands and does not return.
RUN	Loads a specified program and executes it.
LOAD	Loads a specified program and executes it.

AutoRun handles quoted file names for the first parameter to allow specifying path names or file names that contain white space. Note only one set of quotes per command is supported. AutoRun.dat entry examples:

RUN	"Flash File Store\Apps\some.exe" arg1, arg2, arg3
CALL	"Flash File Store\2577\usercmds.dat"

AutoCopy

AutoCopy (AutoCopy.exe) copies/moves files between locations. It has no user interface and is configured through command line arguments. It has support for the following parameters, in no particular order:

Usage:

```
AutoCopy [-D["W"]] [-L["X"]] [-M[D]] [-Q[Y]] [-S["Z"]]
```

-D	Indicates destination file name and must be fully qualified. W is a string value.
-L	Indicates qualified file name for logging to enable. Default is disabled, X string value
-M	Moves file to a destination rather than copies the file. Default value is disabled. D is an integer value. D=1 indicates enabled, 0 is disabled.
-Q	Indicates if a message box should appear when an error occurs. Default is disabled. Y is an integer value.
-S	Indicates a source file name and must be fully qualified. Z is a string value.

Process return code uses standard error codes defined in WINERROR.H.

Example:

```
; use AutoCopy to copy the control panel from flash file store to windows.
autocopy.exe -S"\Flash File Store\System\Audio.cpl" -D"\Windows\Audio.cpl"

; use AutoCopy to move the control panel from flash file store to windows.
autocopy.exe -M1 -S"\Flash File Store\System\Audio.cpl" -D"\Windows\Audio.cpl"
```

AutoReg

The AutoReg (AutoReg.exe) component adds registry information to the Windows Mobile registry. It has no user interface and is configured through command line arguments.

Usage:

```
AutoReg [-D] [-HKey] [-Q] "filename"
```

-D	Deletes registry file after a good load, allows systems with implemented hives
-H	Saves registry path, and all child entries, to the specific .REG registry file.
-Q	Indicates whether a message box should appear when a fatal error occurs.
filename	Fully qualified file name to read from or write to, encased in double quotes to support spaces in paths or file names. See examples below.

Process return code uses standard error codes defined in WinError.h.

Example:

```
; use AutoReg to install this registry information.
autoreg.exe "\Flash File Store\install.reg"

; use AutoReg to install this registry information. Delete the file afterwards.
autoreg.exe -D "\Flash File Store\install.reg"

; use AutoReg to extract registry information to a file.
autoreg.exe -HHKEY_LOCAL_MACHINE\Software\Intermec\Version "\version.reg"
```

The format of the input file, in this example, is the standard registry format which should ease the creation of the input file since there are many publicly available utilities to generate a registry file besides Notepad.

AutoCab

AutoCab (Autocab.exe) extracts files, registry settings, and shortcuts from Windows cabinet (.CAB) files. The Windows startup sequence invokes AutoCab as a part of AutoExec and AutoRun. During the Windows Mobile startup sequence, AutoCab processes all cab files in the “\CabFiles” directory relative to the current location of Autocab, unless the location is overridden by command line arguments. AutoCab can run as a stand-alone program to install a cab file or a directory of cab files.

AutoCab only installs the cab file if it was not installed before by AutoCab. To track the installation of a cab file, AutoCab marks the cab file with the System attribute. This attribute is ignored if the device is performing a clean-boot on a non-persistent file system.

AutoCab preserves the cab file after installation if the ReadOnly attribute is set. If not set, the cab file is deleted automatically after installation.

Command line switches are described as follows.

Usage:

```
AutoCab [-ChkRst=][ -File= ][ -Force ][ -Log= ][ -Move= ][ -Quiet= ][ -Show= ][ -Signal= ]
```

-ChkRst=	Set to 1 to configure AutoCab to check for the Reset flag after all cab files are installed. This file is created by cab files that want a clean reset after installation. Default is 0 (do not check for flag).
-File=	Specifies the cab files to extract. Note that the specified files need not end with the .cab extension.
-Force	Forces the specified cab files to extract regardless of whether it was previously extracted.
-Log=	Set to 1 to create log file in same folder AutoCab is running. Debugs cab installation. Default is 0 (disabled).
-Move=	Set to 1 to force source cab file deletion, even when read-only bit set on file. Default is 0 (disabled).
-Quiet=	Set to 0 to allow AutoCab to display user error message box. Debugs cab installation. Default is 1 (quiet).
-Show=	Set to 0 to prevent showing any installation progress interfaces. Also prevents user from canceling installation. Set to 1 to show normal installation. Set to 2 to show Intermec installation progress interface (user can see what is installing but cannot cancel it). Default is 1 (show normal).
-Signal=	Set to string name of signal to use at the completion of cab installation before a reboot occurs (if enabled). AutoCab uses WaitForSingleObject on this name. Default is disabled.

If <PathName> references a single cab file, that file is processed. If <PathName> references a directory, all the .cab files in that directory is processed. If <PathName> is a wild card pattern, all files matching that pattern is processed. If <PathName> is omitted, InstallCab processes all the .cab files in directory “\CabFiles.”

Example:

```
; Install all cab files in the \Flash File Store\XYZ directory, regardless.
AutoCab -FILE="\Flash File Store\XYZ\*.cab" -FORCE
```

```
; Install only one cab file, use Intermec cab installation display
AutoCab -FILE="\myCab\app.cab" -show=2
```

Creating Cab Files

The Windows CE operating system uses a .cab file to install an application on a Windows CE-based device. A .cab file is composed of multiple files that are compressed into one file. Compressing multiple files into one file provides the following benefits:

- All application files are present.
- A partial installation is prevented.
- The application can be installed from several sources, such as a desktop computer or a web site.

Use the CAB Wizard application (CabWiz.exe) to generate a .cab file for your application.

Creating Device-Specific Cab Files

Do the following to create a device-specific .cab file for an application, *in the order provided*:

- 1 Create an .INF file with Windows CE-specific modifications (*below*).
- 2 *Optional* Create a SETUP.DLL file to provide custom control of the installation process ([page 62](#)).
- 3 Use the CAB Wizard to create the .cab file, using the .inf file, the optional Setup.dll file, and the device-specific application files as parameters ([page 65](#)).

Creating an .inf File

An .inf file specifies information about an application for the CAB Wizard. Below are the sections of an .inf file:

[Version]

This specifies the creator of the file, version, other relevant information.

Required?	Yes	
Signature	"signature_name"	"\$Windows NT\$"
Provider	"INF_creator"	Example: RegSettings.All
CESignature	"\$Windows CE\$"	

Example

```
[Version]
Signature = "$Windows NT$"
Provider = "Intermec"
CESignature = "$Windows CE$"
```

[CEStrings]

This specifies string substitutions for the application name and the default installation directory.

Required?	Yes	
AppName	<i>app_name</i>	Name of the application. Other instances of %AppName% in the .inf file are replaced with this string value, such as RP32.
InstallDir	<i>default_install_dir</i>	Default installation directory on the device. Other instances of %InstallDir% in the .inf file are replaced with this string value. Example: \SDMMC_Disk\%AppName%

Example

```
[CEStrings]
AppName="Game Pack"
InstallDir=%CE1%\%AppName%
```

[Strings]

This section is optional and defines one or more string keys. A string key represents a string of printable characters.

Required?	No	
string_key	<i>value</i>	String consisting of letters, digits, other printable characters. Enclose <i>value</i> in double quotation marks "" if corresponding string key is used in an item requiring double quotation marks. No string_keys is okay.

Example

```
[Strings]
reg_path = Software\Intermec\My Test App
```

[CEDevice]

Describes the platform for the targeted application. All keys are optional. If a key is nonexistent or has no data, Windows CE does not perform any checking except the *UnsupportedPlatforms*. If the *UnsupportedPlatforms* key exists but no data, the previous value is not overridden.

Required?	Yes	
ProcessorType	<i>processor_type</i>	The value that is returned by SYSTEMINFO.dwProcessorType. For example, the value for the ARM CPU is 2577
UnsupportedPlatforms	<i>platform_family_name</i>	Lists known unsupported platform family names. If name specified in [CEDevice.xxx] section is different from in [CEDevice] section, both <i>platform_family_name</i> values are unsupported for microprocessor specified by xxx. The list of unsupported platform family names is appended to the previous list of unsupported names. Application Manager will not display the application for an unsupported platform. User will be warned during the setup process if the .cab file is copied to an unsupported device.

Example

```
[CEDevice]
UnsupportedPlatforms = pltfrm1 ; pltfrm1 is unsupported
[CEDevice.SH3]
UnsupportedPlatforms = ; pltfrm1 is still unsupported
```

VersionMin	<i>minor_version</i>	Numeric value returned by OSVERSIONINFO.dwVersionMinor. The .cab file is valid for the currently connected device if the version of this device is greater than or equal to VersionMin .
VersionMax	<i>major_version</i>	Numeric value returned by OSVERSIONINFO.dwVersionMajor. The .cab file is valid for the currently connected device if the version of this device is less than or equal to VersionMax .
BuildMin	<i>build_number</i>	Numeric value returned by OSVERSIONINFO.dwBuildNumber. The .cab file is valid for the currently connected device if the version of this device is greater than or equal to BuildMin .
BuildMax	<i>build_number</i>	Numeric value returned by OSVERSIONINFO.dwBuildNumber. The .cab file is valid for the currently connected device if the version of this device is less than or equal to BuildMax .

Example

The following code example shows three [CEDevice] sections: one that gives basic information for any CPU and two that are specific to the SH3 and the MIPS microprocessors.

```
[CEDevice] ; A "template" for all platforms
UnsupportedPlatforms = pltfrm1; Does not support pltfrm1

; The following specifies version 1.0 devices only.
VersionMin = 1.0
VersionMax = 1.0

[CEDevice.ARM]; Inherits all [CEDevice] settings
; This will create a .CAB file specific to ARM devices.
ProcessorType = 2577; ARM .cab file is valid for ARM microprocessors.
UnsupportedPlatforms = ; pltfrm1 is still unsupported

; The following overrides the version settings so that no version checking is
performed.
VersionMin =
VersionMax =

[CEDevice.MIPS] ; Inherits all [CEDevice] settings
; This will create a .CAB file specific to "MIPS" devices.
ProcessorType = 4000; MIPS .CAB file is valid for MIPS microprocessor.
UnsupportedPlatforms =pltfrm2; pltfrm1, pltfrm2 unsupported for MIPS .CAB
file.
```



Note: To create the two CPU-specific .cab files for the Setup.inf file in the previous example, run the CAB Wizard with the "/cpu arm mips" parameter.

[DefaultInstall]

This describes the default installation of your application. Note that under this section, you will list items expanded upon later in this description.

Required?	Yes	
Copyfiles	<i>copyfile_list_section</i>	Maps to files defined later in the .inf file, such as Files.App, Files.Font, and Files.Bitmaps.
AddReg	<i>add_registry_section</i>	Example: RegSettings.All
CEShortcuts	<i>shortcut_list_section</i>	String that identifies one more section that defines shortcuts to a file, as defined in the [CEShortcuts] section.
CESetupDLL	<i>setup_DLL</i>	Optimal string that specifies a SETUP.DLL file. It is written by the Independent Software Vendor (ISV) and contains customized functions for operations during installation and removal of the application. The file must be specified in the [SourceDisksFiles] section.
CESelfRegister	<i>self_reg_DLL_filename</i>	String that identifies files that self-register by exporting the DllRegisterServer and DllUnregisterServer Component Object Model (COM) functions. Specify these files in the [SourceDiskFiles] section. During installation, if installation on the device fails to call the file's exported DllRegisterServer function, the file's exported DllUnregisterServer function will not be called during removal.

Example

[DefaultInstall]

```
AddReg = RegSettings.All
CEShortcuts = Shortcuts.All
```

[SourceDiskNames]

This section describes the name and path of the disk on which your application resides.

Required?	Yes	
disk_ordinal	<i>disk_label,path</i>	1=,"App files", C:\Appsoft\RP32\... 2=,"Font files",C:\RpTools\... 3=,"CE Tools",C:\windows ce tools...
CESignature		"\$Windows CE\$"

Example

[SourceDisksNames]; Required section

```
1 = ,"Common files",,C:\app\common; Using an absolute path
```

[SourceDisksNames.SH3]

```
2 = ,"SH3 files",,sh3; Using a relative path
```

[SourceDisksNames.MIPS]

```
2 = ,"MIPS files",,mips; Using a relative path
```

[SourceDiskFiles]

This describes the name and path of files in which your application resides.

Required?	Yes	
filename	<i>disk_number[subdir]</i>	RPM.EXE = 1,c:\appsoft\... WCESTART.INI = 1 RPMCE212.INI = 1 TAHOMA.TTF = 2



Note: [,subdir] is relative to the location of the .inf file.

Example

```
[SourceDisksFiles]; Required section
begin.wav = 1
end.wav = 1
sample.hlp = 1
[SourceDisksFiles.SH3]
sample.exe = 2; Uses the SourceDisksNames.SH3 identification of 2.
[SourceDisksFiles.MIPS]
sample.exe = 2; Uses the SourceDisksNames.MIPS identification of 2.
```

[DestinationDirs]

This describes the names and paths of the destination directories for the application on the target device. *Note: Windows CE does not support directory identifiers.*

Required?	Yes
file_list_section	<p><i>0,subdir</i> String that identifies the destination directory. The following list shows the string substitutions supported by Windows CE. Use these only for the beginning of the path.</p> <ul style="list-style-type: none"> \ %CE1% \Program Files %CE2% \Windows %CE3% \My Documents %CE4% \Windows\Startup %CE5% \My Documents %CE6% \Program Files\Accessories %CE7% \Program Files\Communication %CE8% \Program Files\Games %CE9% \Program Files\Pocket Outlook %CE10% \Program Files\Office %CE11% \Windows\Start Menu\Programs %CE12% \Windows\Start Menu\Programs\Accessories %CE13% \Windows\Start Menu\Programs\Communications %CE14% \Windows\Start Menu\Programs\Games %CE15% \Windows\Fonts %CE16% \Windows\Recent %CE17% \Windows\Start Menu %InstallDir% <p>Contains the path to target directory selected during installation - declared in [CEStrings]</p> <p>%AppName%</p> <p>Contains the application name defined in the [CEStrings] section.</p>

Example

```
[DestinationDirs]
Files.Common = 0,%CE1%\My Subdir; \Program Files\My Subdir
Files.Shared = 0,%CE2%; \Windows
```

[CopyFiles]

This section, under the **[DefaultInstall]** section, describes the default files to copy to the target device. Within the **[DefaultInstall]** section, files were listed that must be defined elsewhere in the .inf file. This section identifies that mapping and may contain flags.

Required?	Yes	
copyfile_list_section	<i>destination_filename,[source_filename]</i>	The <i>source_filename</i> parameter is optional if it is the same as <i>destination_filename</i> .
copyfile_list_section	<i>flags</i>	The numeric value that specifies an action to be done while copying files. The following table shows values supported by Windows CE.

Flag	Value	Description
COPYFLG_WARN_IF_SKIP	0x00000001	Warn user if skipping a file is attempted after error.
COPYFLG_NOSKIP	0x00000002	Do not allow a user to skip copying a file.
COPYFLG_NO_OVERWRITE	0x00000010	Do not overwrite files in destination directory.
COPYFLG_REPLACEONLY	0x00000400	Copy the source file to the destination directory only if the file is already in the destination directory.
CE_COPYFLG_NO_DATE_DIALOG	0x20000000	Do not copy files if the target file is newer.
CE_COPYFLG_NODATECHECK	0x40000000	Ignore date while overwriting the target file.
CE_COPYFLG_SHARED	0x80000000	Create a reference when a shared .dll is counted.

Example

```
[DefaultInstall.SH3]
CopyFiles = Files.Common, Files.SH3
[DefaultInstall.MIPS]
CopyFiles = Files.Common, Files.MIPS
```

[AddReg]

This section, under the **[DefaultInstall]** section, is optional and describes the keys and values the .cab file adds to the device registry. Within the **[DefaultInstall]** section, a reference may be made to this section, such as “AddReg=RegSettings.All”. This section defines options for that setting.

Required?	No	
add_registry_section	<i>registry_root_string</i>	String that specifies the registry root location. The following list shows the values supported by Windows CE. <ul style="list-style-type: none"> • HKCR Same as HKEY_CLASSES_ROOT • HKCU Same as HKEY_CURRENT_USER • HKLM Same as HKEY_LOCAL_MACHINE
add_registry_section	<i>value_name</i>	Registry value name. If empty, the “default” registry value name is used.
add_registry_section	<i>flags</i>	Numeric value that specifies information about the registry key. The following table shows the values that are supported by Window CE.

Flag	Value	Description
FLG_ADDREG_NOCLOBBER	0x00000002	If the registry key exists, do not overwrite it. Can be used with any of the other flags in this table.
FLG_ADDREG_TYPE_SZ	0x00000000	REG_SZ registry data type.
FLG_ADDREG_TYPE_MULTI_SZ	0x00010000	REG_MULTI_SZ registry data type. Value field that follows can be a list of strings separated by commas.
FLG_ADDREG_TYPE_BINARY	0x00000001	REG_BINARY registry data type. Value field that follows must be a list of numeric values separated by commas, one byte per field, and must not use the 0x hexadecimal prefix.
FLG_ADDREG_TYPE_DWORD	0x00010001	REG_DWORD data type. The noncompatible format in the Win32 Setup .INF documentation is supported.

Example

```
AddReg = RegSettings.All
```

[RegSettings.All]

```
HKLM,%reg_path%,,0x00000000,alpha; <default> = "alpha"
HKLM,%reg_path%,test,0x00010001,3; Test = 3
HKLM,%reg_path%\new,another,0x00010001,6; New\another = 6
```

[CEShortCuts]

This section, a Windows CE-specific section under the **[DefaultInstall]** section, is optional and describes the shortcuts that the installation application creates on the device. Within the **[DefaultInstall]** section, a reference may have been made to this section, such as “ShortCuts.All”. This section defines the options for that setting.

Required?	No	
shortcut_list_section	<i>shortcut_filename</i>	String that identifies the shortcut name. It does not require the .LNK extension.
shortcut_list_section	<i>shortcut_type_flag</i>	Numeric value. Zero or empty represents a shortcut to a file; any nonzero numeric value represents a shortcut to a folder.
shortcut_list_section	<i>target_file_path</i>	String value that specifies the destination location. Use the target file name for a file, such as MyApp.exe, that must be defined in a file copy list. For a path, use a <i>file_list_section</i> name defined in the [DestinationDirs] section, such as <i>DefaultDestDir</i> , or the <i>%InstallDir%</i> string.
shortcut_list_section	<i>standard_destination_path</i>	Optional string value. A standard <i>%CEx%</i> path or <i>%InstallDir%</i> . If no value is specified, the <i>shortcut_list_section</i> name of the current section or the <i>DefaultDestDir</i> value from [DestinationDirs] is used.

Example

```
CEShortcuts = Shortcuts.All
```

[Shortcuts.All]

```
Sample App,0,sample.exe; Uses the path in DestinationDirs. Sample
App,0,sample.exe,%InstallDir%; The path is explicitly specified.
```

Sample .INF File

```
[Version]; Required section
Signature = "$Windows NT$"
Provider = "Intermec Technologies Corporation"
CESignature = "$Windows CE$"
```

Chapter 3 — Configuring the Computer

```
;[CEDevice]
;ProcessorType =

[DefaultInstall]; Required section
CopyFiles = Files.App, Files.Fonts, Files.BitMaps, Files.Intl,
Files.TelecomNcsCE, Files.Windows, Files.Import, Files.Export, Files.Work,
Files.Database, Files.WinCE AddReg = RegSettings.All ;CEShortcuts =
Shortcuts.All

[SourceDisksNames]; Required section
1 = ,"App files" ,,c:\appsoft\...
2 = ,"Font files" ,,c:\WinNT\Fonts
3 = ,"CE Tools" ,,c:\windows ce tools\wce400\700ie\mfc\lib\x86

[SourceDisksFiles]; Required section
rpm.exe = 1,C:\Appsoft\program\wce400\WCEX86Rel700
wcestart.ini = 1
rpmce212.ini = 1
intermec.bmp = 1
rpmlogo.bmp = 1
rpmname.bmp = 1
import.bmp = 1
export.bmp = 1
clock.bmp = 1
printer.bmp = 1
filecopy.bmp = 1
readme.txt = 1
lang_eng.bin = 1
rpmdata.dbd = 1,database\wce1
tahoma.ttf = 2
mfcce212.dll = 3
olece212.dll = 3
olece211.dll = 1,c:\windows ce tools\wce400\NMSD61102.11\mfc\lib\x86
rdm45wce.dll = 1,c:\rptools\rdm45wce\4_50\lib\wce400\wcex86rel
picfmt.dll = 1,c:\rptools\picfmt\1_00\wce400\wcex86rel6110
fmtctrl.dll = 1,c:\rptools\fmtctrl\1_00\wce400\wcex86rel6110
ugrid.dll = 1,c:\rptools\ugrid\1_00\wce400\wcex86rel6110
simple.dll = 1,c:\rptools\pspbm0c\1_00\wce400\wcex86rel
psink.dll = 1,c:\rptools\psink\1_00\wce400\WCEX86RelMinDependency
pslpwce.dll = 1,c:\rptools\pslpm0c\1_00\wce400\WCEX86RelMinDependency
npcpport.dll = 1,c:\rptools\cedk\212_03\installable drivers\printer\npcp
;dexcom.dll = 1,c:\rptools\psdxm0c\1_00\x86
ncsce.exe = 1,c:\rptools\ncsce\1_04
nrinet.dll = 1,c:\rptools\ncsce\1_04

[DestinationDirs];Required section
;Shortcuts.All = 0,%CE3% ; \Windows\Desktop
Files.App= 0,%InstallDir%
Files.DataBase= 0,%InstallDir%\DataBase
Files.BitMaps= 0,%InstallDir%\Bitmaps
Files.Fonts= 0,%InstallDir%\Fonts
Files.Intl= 0,%InstallDir%\Intl
Files.TelecomNcsCE= 0,%InstallDir%\Telecom\NcsCE
Files.Windows= 0,%InstallDir%\Windows
Files.Import= 0,%InstallDir%\Import
Files.Export= 0,%InstallDir%\Export
Files.Work= 0,%InstallDir%\Work
Files.WinCE= 0,\storage_card\wince
```

```

[CEStrings]; Required section
AppName = Rp32
InstallDir = \storage_card\%AppName%

[Strings]; Optional section
;[Shortcuts.All]
;Sample App,0,sample.exe; Uses the path in DestinationDirs.
;Sample App,0,sample.exe,%InstallDir%; The path is explicitly specified.

[Files.App]
rpm.exe,,,0
rpm.ini,rpmce212.ini,,0
mfcce212.dll,,,0
olece212.dll,,,0
olece211.dll,,,0
rdm45wce.dll,,,0
picfmt.dll,,,0
fmtctrl.dll,,,0
ugrid.dll,,,0
simple.dll,,,0
psink.dll,,,0
pslpwce.dll,,,0
npcppport.dll,,,0
;dexcom.dll,,,0

[Files.DataBase]
rpmdata.dbd,,,0

[Files.Fonts]
tahoma.ttf,,,0

[Files.BitMaps]
intermec.bmp,,,0
rpmlogo.bmp,,,0
rpmname.bmp,,,0
import.bmp,,,0
export.bmp,,,0
clock.bmp,,,0
printer.bmp,,,0
filecopy.bmp,,,0

[Files.Intl]
lang_eng.bin,,,0

[Files.TelecomNcsCE]
ncsce.exe,,,0
nrinet.dll,,,0

[Files.Windows]
readme.txt,,,0

[Files.Import]
readme.txt,,,0

[Files.Export]
readme.txt,,,0

[Files.Work]

```

```
readme.txt,,,0
```

```
[Files.WinCE]  
wcestart.ini,,,0
```

```
[RegSettings.All]  
HKLM,"SOFTWARE\Microsoft\Shell\AutoHide",,0x00010001,1; Autohide the taskbar  
HKLM,"SOFTWARE\Microsoft\Shell\OnTop",,0x00010001,0; Shell is not on top  
HKLM,"SOFTWARE\Microsoft\Clock",SHOW_CLOCK,0x00010001,0  
; Clock is not on taskbar
```

Using Installation Functions in Setup.dll

Setup.dll is an optional file that enables you to perform custom operations during installation and removal of your application. The following list shows the functions that are exported by Setup.dll.

Install_Init	Called before installation begins. Use this function to check the application version when reinstalling an application and to determine if a dependent application is present.
Install_Exit	Called after installation is complete. Use this function to handle errors that occur during application installation.
Uninstall_Init	Called before the removal process begins. Use this function to close the application, if the application is running.
Uninstall_Exit	Called after the removal process is complete. Use this function to save database information to a file and delete the database and to tell the user where the user data files are stored and how to reinstall the application.



Note: Use [DefaultInstall] > CESelfRegister (page 56) in the .inf file to point to Setup.dll.

After the CAB File Extraction

Cab files that need to cause a warm reset after cab extraction will need to create the __resetmeplease__.txt file in the “\Windows” directory. The preferred method to create this file is within the DllMain portion of the Setup.dll file. It looks like this:

```
#include <windows.h>  
#include <Tlhelp32.h>  
#include <winioctl.h>  
#include <ce_setup.h> // in the public SDK dir  
  
#define IOCTL_TERMINAL_RESET CTL_CODE (FILE_DEVICE_UNKNOWN, FILE_ANY_ACCESS,  
2050, METHOD_NEITHER)  
  
BOOL APIENTRY DllMain( HANDLE h, DWORD reason, LPVOID lpReserved )  
{  
return TRUE;  
} // DllMain  
  
//*****  
// $DOCBEGIN$  
// BOOL IsProcessRunning( TCHAR * pname );  
//
```

```

// Description: Get process table snapshot, look for pname running.
//
// Arguments: pname - pointer to name of program to look for.
// for example, app.exe.
//
// Returns: TRUE - process is running.
//          FALSE - process is not running.
// $DOCEND$
//*****
BOOL IsProcessRunning( TCHAR * pname )
{
HANDLE hProcList;
PROCESSENTRY32 peProcess;
DWORD thDeviceProcessID;
TCHAR lpname[MAX_PATH];

if ( !pname || !*pname ) return FALSE;

_tcscpy( lpname, pname );
_tcslwr( lpname );

hProcList = CreateToolhelp32Snapshot( TH32CS_SNAPPROCESS, 0 );

if ( hProcList == INVALID_HANDLE_VALUE ) {
return FALSE;
} // end if

memset( &peProcess, 0, sizeof(peProcess) );
peProcess.dwSize = sizeof(peProcess);

if ( !Process32First( hProcList, &peProcess ) ) {
CloseToolhelp32Snapshot( hProcList );
return FALSE;
} // end if

thDeviceProcessID = 0;

do {
_tcslwr( peProcess.szExeFile );

if ( _tcscstr( peProcess.szExeFile, lpname ) ) {
thDeviceProcessID = peProcess.th32ProcessID;
break;
} // end if
} while ( Process32Next( hProcList, &peProcess ) );

if ( ( GetLastError() == ERROR_NO_MORE_FILES ) && ( thDeviceProcessID == 0 ) )
{
CloseToolhelp32Snapshot( hProcList );
return FALSE;
} // end if

CloseToolhelp32Snapshot( hProcList );
return TRUE;
} // IsProcessRunning

codeINSTALL_INIT Install_Init(
HWND hwndParent,
BOOL fFirstCall,

```

Chapter 3 — Configuring the Computer

```
BOOL fPreviouslyInstalled,
LPCTSTR pszInstallDir )
{
return codeINSTALL_INIT_CONTINUE;
}

codeINSTALL_EXIT Install_Exit (
HWND hwndParent,
LPCTSTR pszInstallDir,
WORD cFailedDirs,
WORD cFailedFiles,
WORD cFailedRegKeys,
WORD cFailedRegVals,
WORD cFailedShortcuts )
{
HANDLE h;
TCHAR srcfile[MAX_PATH];
TCHAR dstfile[MAX_PATH];

if (cFailedDirs || cFailedFiles || cFailedRegKeys ||
cFailedRegVals || cFailedShortcuts)
return codeINSTALL_EXIT_UNINSTALL;

if ( IsProcessRunning( L"autocab.exe" ) )
{
h = CreateFile( L"\\Windows\\__resetmeplease__.txt",
(GENERIC_READ | GENERIC_WRITE), 0, NULL, CREATE_ALWAYS,
FILE_ATTRIBUTE_HIDDEN, NULL );

if ( h != INVALID_HANDLE_VALUE )
CloseHandle( h );
else
{
// Couldn't create the file. If it failed because the file already
// exists, it is not fatal. Otherwise, notify user of the inability to
// reset the device and they will have to perform it manually after all of
// the installations are complete.
} // end if
}
else
{
DWORD dret;

h = CreateFile( L"SYSIO:",
(GENERIC_WRITE | GENERIC_READ), 0, NULL, OPEN_EXISTING,
FILE_ATTRIBUTE_NORMAL, NULL );

// Force a warm start NOW.
if ( h != INVALID_HANDLE_VALUE )
{
DeviceIoControl( h, IOCTL_TERMINAL_RESET, NULL, 0, NULL, 0, &dret, NULL);

// Won't return, but we'll show clean up anyway
CloseHandle( h );
}
else
{
// Couldn't access SYSIO. Notify user.
} // end if
}
```

```

} // end if

return codeINSTALL_EXIT_DONE;
}

codeUNINSTALL_INIT
Uninstall_Init(
HWND hwndParent,
LPCTSTR pszInstallDir ) {

// TODO: Perform the reverse of INSTALL_INIT here
return codeUNINSTALL_INIT_CONTINUE;
}

codeUNINSTALL_EXIT
Uninstall_Exit(HWND hwndParent) {

// TODO: Perform the reverse of INSTALL_EXIT here
return codeUNINSTALL_EXIT_DONE;
}

```

The system software looks for this directory structure and its files on the installed media card storage card or embedded flash file system. No other folders need exist.

```

\2577\autorun.exe
\2577\autorun.dat
\2577\autocab.exe
\2577\autocab.dat
\cabfiles\*.cab

```

Creating Cab Files with CAB Wizard

After you create the .inf file and the optional Setup.dll file, use the CAB Wizard to create the .cab file. Below is the command-line syntax:

```
cabwiz.exe "inf_file" [/dest dest_directory] [/err error_file] [/cpu cpu_type
[cpu_type]]
```

A batch file in <program> directory, with these commands, works well:

```
cabwiz.exe c:\appsoft\\<inf_file_name>
cd \appsoft\

```

<i>inf_file</i>	The Setup.inf file path.
<i>dest_directory</i>	The destination directory for the .cab files. If no directory is specified, the .cab files are created in the "inf_file" directory.
<i>error_file</i>	File name for a log file that contains all warnings and errors that are encountered when the .cab files are compiled. If no file name is specified, errors are displayed in message boxes. If a file name is used, the CAB Wizard runs without the user interface (UI); this is useful for automated builds.
<i>cpu_type</i>	Creates a .cab file for each specified microprocessor tag, which is a label used in the Win32 SETUP.INF file to differentiate between different microprocessor types. The /cpu parameter, followed by multiple <i>cpu_type</i> values, must be the last qualifier in the command line.

Example

This example creates .cab files for the ARM and MIPS microprocessors, assuming the Win32 Setup.inf file contains the ARM and MIPS tags:

```
cabwiz.exe "c:\myfile.inf" /err myfile.err /cpu arm mips
```



Note: CabWiz.exe, MakeCab.exe, and CabWiz.ddf (Windows CE files available on the Windows CE Toolkit) must be installed in the same directory on the desktop computer. Call CabWiz.exe using its full path for the CAB Wizard application to run correctly.

Troubleshooting the CAB Wizard

To identify and avoid problems that might occur when using the CAB Wizard, follow these guidelines:

- Use %% for a percent sign (%) character when using this character in an .inf file string, as specified in Win32 documentation. This will not work under the **[Strings]** section.
- Do not use .inf or .cab files created for Windows CE to install applications on Windows-based desktop platforms.
- Ensure the MakeCab.exe and CabWiz.ddf files, included with Windows CE, are in the same directory as CabWiz.exe.
- Use the full path to call CabWiz.exe.
- Do not create a .cab file with the MakeCab.exe file included with Windows CE. You must use CabWiz.exe, which uses MakeCab.exe to generate the .cab files for Windows CE.
- Do *not* set the read-only attribute for .cab files.

Customization and Lockdown

Some customers would prefer that their users not have access to all of the operating system features. Intermec cannot customize the operating system in any way but a custom application can:

- Delete items from the Start menu, and Programs folder. These items are just shortcuts in the file system so the application is not really being deleted. Cold booting the device will bring these items back so the application will need to be run on every cold boot.
- Use the RegFlushKey() API to save a copy of the registry to a storage device. See the IDL for more information on how to do this. Saving a copy of the registry restores most system settings in a cold boot situation.
- Use the SHFullScreen() API with other APIs to have the application take up the entire display and prevent the Start menu from being available.
- Remap keys and disable keys on the keypad; create a custom SIP; or
- Make changes to the registry to configure the device.

Should you want your 751G to display a full screen, use the following links. These give full instructions on how to display full screen.

- Instructions on how to create a full screen application for eVC++ applications using an SHFullScreen() API:
support.microsoft.com/support/kb/articles/Q266/2/44.ASP
- Instructions on how to create a full screen application for eVB applications also using the SHFullScreen() API:
support.microsoft.com/support/kb/articles/Q265/4/51.ASP

Kernel I/O Controls

This describes the KernelIoControl() functions available to application programmers. Most C++ applications will need to prototype the function as the following to avoid link and compile errors.

```
extern "C" BOOL KernelIoControl(DWORD dwIoControlCode, LPVOID lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD nOutBufSize, LPDWORD lpBytesReturned);
```

You can also see the Device Resource Kit in the Intermec Developer Library (IDL) for information about these functions. The IDL is available as a download from the Intermec web site at www.intermec.com/idl. Contact your Intermec representative for more information.

IOCTL_HAL_GET_DEVICE_INFO

This IOCTL returns either the platform type or the OEMPLATFORM name based on an input value.

Syntax

```
BOOL KernelIoControl( IOCTL_HAL_GET_DEVICE_INFO, LPVOID lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD nOutBufSize, LPDWORD lpBytesReturned );
```

Parameters

lpInBuf	Points to a DWORD containing either the SPI_GETPLATFORMTYPE or SPI_GETOEMINFO value.
lpInBufSize	Must be set to sizeof(DWORD).
lpOutBuf	Must point to a buffer large enough to hold the return data of the function. If SPI_GETPLATFORMTYPE is specified in <i>lpInBuf</i> , then the "PocketPC\0" Unicode string is returned. If SPI_GETOEMINFO is specified in <i>lpInBuf</i> , then the "Intermec 700\0" Unicode string is returned.
nOutBufSize	The size of <i>lpOutBuf</i> in bytes. Must be large enough to hold the string returned.
lpBytesReturned	The actual number of bytes returned by the function for the data requested.

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the extended error value.

IOCTL_HAL_ITC_READ_PARM**Usage**

```
#include "oemioctl.h"
```

Syntax

```
BOOL KernelIoControl( IOCTL_HAL_ITC_READ_PARM, LPVOID  
lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD  
nOutBufSize, LPDWORD lpBytesReturned );
```

Parameters

lpInBuf	Points to this structure. See “ <i>ID Field Values</i> ” below. struct PARMS { BYTE id; BYTE ClassId; };
nInBufSize	Must be set to the size of the PARMS structure.
lpOutBuf	Must point to a buffer large enough to hold the return data of the function. If this field is set to NULL and <i>nOutBufSize</i> is set to zero when the function is called the function will return the number bytes required by the buffer.
nOutBufSize	The size of <i>lpOutBuf</i> in bytes.
lpBytesReturned	Number of bytes returned by the function for the data requested.

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the error value. Either ERROR_INVALID_PARAMETER or ERROR_INSUFFICIENT_BUFFER may be returned when this function is used to get the error.

ID Field Values

The *id* field of the PARMS structure may be one of the following values:

ID Field Values	
ITC_NVPARAM_ETHERNET_ID	Returns Ethernet 802.11b or 802.11b/g MAC Address. Six bytes returned in buffer pointed to by <i>lpOutBuffer</i>
ITC_NVPARAM_SERIAL_NUM	Returns serial number of device in BCD format. Six bytes returned in buffer pointed to by <i>lpOutBuffer</i>
ITC_NVPARAM_MANF_DATE	Returns device manufacture date in BCD YYYY/MM/DD format. 4 bytes sent in buffer pointed to by <i>lpOutBuffer</i>
ITC_NVPARAM_SERVICE_DATE	Returns last device service date in BCD YYYY/MM/DD format. Four bytes sent in buffer pointed to by <i>lpOutBuffer</i>
ITC_NVPARAM_DISPLAY_TYPE	Returns device display type. One byte returned in buffer pointed to by <i>lpOutBuffer</i>
ITC_NVPARAM_EDG_IP	Returns device Ethernet debug IP address. Four bytes returned in buffer pointed to by <i>lpOutBuffer</i>

ID Field Values**ITC_NVPARAM_EDBG_SUBNET**

Returns device Ethernet debug subnet mask. Four bytes returned in buffer pointed to by *lpOutBuffer*

ITC_NVPARAM_ECN

Returns ECNs applied to device in bit array format. Four bytes returned in buffer pointed to by *lpOutBuffer*

ITC_NVPARAM_CONTRAST

Returns device default contrast setting. Two bytes returned in buffer pointed to by *lpOutBuffer*

ITC_NVPARAM_MCODE

Returns manufacturing configuration code for device. Sixteen bytes returned in buffer pointed to by *lpOutBuffer*

ITC_NVPARAM_VERSION_NUMBER

Returns firmware version for various system components. These values for *ClassId* field of PARMS structure are allowed when ITC_NVPARAM_VERSION_NUMBER used in *id* field:

- VN_CLASS_KBD Returns five-byte string, including null terminator, with ASCII value representing keypad microprocessor version in system. String format is *x.xx* with terminating null character.
- VN_CLASS_ASIC Returns five-byte string, including null terminator, with ASCII value representing version of FPGA firmware in system. String format is *x.xx* with terminating null character.
- VN_CLASS_BOOTSTRAP Returns five-byte string, including null terminator, with ASCII value representing version of Bootstrap Loader firmware in system. String format is *x.xx* with terminating null character.

ITC_NVPARAM_INTERMEC_SOFTWARE_CONTENT

Reads manufacturing flag bits from non-volatile data store dictating certain software parameters. BOOLEAN DWORD returned in buffer pointed to by *lpOutBuffer* indicating whether Intermec Content enabled in XIP regions. TRUE indicates enabled. FALSE is not enabled.

ITC_NVPARAM_ANTENNA_DIVERSITY

Reads state of antenna diversity flag. BOOLEAN DWORD returned in buffer pointed to by *lpOutBuffer* indicating whether diversity antenna is installed. TRUE indicates installed. FALSE is not installed.

ITC_NVPARAM_WAN_RI

Reads state of WAN ring indicator flag. BOOLEAN DWORD returned in buffer pointed to by *lpOutBuffer* indicating polarity of WAN RI signal. TRUE indicates active high. FALSE is active low.

ITC_NVPARAM_RTC_RESTORE

Reads state of real-time clock restore flag. BOOLEAN DWORD returned in buffer pointed to by *lpOutBuffer*. TRUE indicates RTC is restored on cold-boot. FALSE is RTC is not restored.

ITC_NVPARAM_INTERMEC_DATACOLLECTION_SW

Reads state of data collection software enabled flag. BOOLEAN DWORD returned in buffer pointed to by *lpOutBuffer* indicating data collection software installs at boot time. FALSE is do not install data collection software

ITC_NVPARAM_INTERMEC_DATACOLLECTION_HW

Reads data collection hardware flags. BYTE returned in buffer pointer to by *lpOutBuffer* indicating type of data collection hardware installed. Maximum value returned is ITC_DEVID_SCANHW_MAX:

- ITC_DEVID_SCANHW_NONE No scanner hardware installed.
- ITC_DEVID_OEM2D_IMAGER OEM 2D imager installed.
- ITC_DEVID_INTERMEC2D_IMAGER Intermec 2D imager installed.
- ITC_DEVID_SE900_LASER SE900 laser installed.
- ITC_DEVID_SE900HS_LASER SE900HS laser installed.
- ITC_DEVID_INTERMEC_EVIO EVIO linear imager installed.

High bit non-zero value indicates S6 scanning engine is installed. Bit mask is ITC_DEVID_S6ENGINE_MASK.

ITC_NVPARAM_WAN_INSTALLED

Reads state of WAN radio installed flag. BOOLEAN DWORD is returned in buffer: TRUE = WAN radio installed.

ITC_NVPARAM_WAN_FREQUENCY

Reads state of WAN radio frequency flag. BOOLEAN DWORD is returned in buffer: TRUE indicates WAN radio frequency is United States. FALSE is a European WAN radio frequency.

ITC_NVPARAM_WAN_RADIO_TYPE

Reads WAN radio ID installed by manufacturing. BYTE returned in buffer pointer to by *lpOutBuffer* indicating type of WAN radio hardware installed. Maximum value returned is ITC_DEVID_WANRADIO_MAX:

- ITC_DEVID_WANRADIO_NONE No WAN radio installed
- ITC_DEVID_WANRADIO_SIERRA_SB555 CDMA Sierra Wireless radio
- ITC_DEVID_WANRADIO_XIRCOM_GEM3503 GSM/GPRS Intel (Xircom) radio
- ITC_DEVID_WANRADIO_SIEMENS_MC45 GSM/GPRS Siemens radio
- ITC_DEVID_WANRADIO_SIEMENS_MC46 GSM/GPRS Siemens radio

ID Field Values	
ITC_NVPARAM_80211_INSTALLED	Reads state of 802.11b or b/g radio installed flag. BOOLEAN DWORD returned in buffer: TRUE = radio installed.
ITC_NVPARAM_80211_RADIO_TYPE	Reads 802.11b or b/g radio ID installed by manufacturing. BYTE returned in buffer pointer to by <i>lpOutBuffer</i> indicates type of radio hardware installed. ITC_DEVID_80211RADIO_MAX is maximum value returned: <ul style="list-style-type: none"> • ITC_DEVID_80211RADIO_NONE No 802.11b or 802.11b/g radio installed. • ITC_DEVID_80211RADIO_INTEL_2011B Intel 2011B radio installed.
ITC_NVPARAM_BLUETOOTH_INSTALLED	Reads state of Bluetooth radio installed flag. BOOLEAN DWORD returned in buffer pointed to by <i>lpOutBuffer</i> . TRUE indicates Bluetooth radio installed. FALSE is no Bluetooth radio installed.
ITC_NVPARAM_SERIAL2_INSTALLED	Reads state of serial 2 (COM2) device installed flag. BOOLEAN DWORD returned in buffer pointed to by <i>lpOutBuffer</i> . TRUE indicates serial 2 device is installed. FALSE is no serial 2 device is installed.
ITC_NVPARAM_VIBRATE_INSTALLED	Reads state of vibrate device installed flag. BOOLEAN DWORD is returned in buffer pointed to by <i>lpOutBuffer</i> . TRUE indicates vibrate device is installed. FALSE is no vibrate device is installed.
ITC_NVPARAM_LAN9000_INSTALLED	Reads state of Ethernet device installed flag. BOOLEAN DWORD is returned in buffer pointed to by <i>lpOutBuffer</i> . TRUE indicates Ethernet device is installed. FALSE is no Ethernet device is installed.
ITC_NVPARAM_SIM_PROTECT_HW_INSTALLED	Reads state of SIM card protection hardware installed flag. BOOLEAN DWORD returned in buffer pointed to by <i>lpOutBuffer</i> . TRUE indicates SIM card protection hardware installed. FALSE is no such hardware installed.
ITC_NVPARAM_SIM_PROTECT_SW_INSTALLED	Reads state of SIM card protection software installed flag. BOOLEAN DWORD returned in buffer pointed to by <i>lpOutBuffer</i> . TRUE indicates SIM card protection software is installed. FALSE is no such software installed.

IOCTL_HAL_ITC_WRITE_SYSPARM

Describes and enables the registry save location.

Usage

#include "oemioctl.h"

Syntax

```
BOOL KernelIoControl( IOCTL_HAL_ITC_WRITE_SYSPARM, LPVOID lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD nOutBufSize, LPDWORD lpBytesReturned );
```

Parameters

lpInBuf	A single byte that may be one of the <i>id</i> values. See "ID Field Values" on the next page.
nInBufSize	Must be set to the size of the <i>lpInBuf</i> in bytes.
lpOutBuf	Must point to a buffer large enough to hold the data to be written to the non-volatile data store.
nOutBufSize	The size of <i>lpOutBuf</i> in bytes.
lpBytesReturned	The number of bytes returned by the function.

Return Values

Returns TRUE if the function is successful, returns FALSE if not. GetLastError() may be used to get the error value. When this function is used to get the error, either ERROR_INVALID_PARAMETER or ERROR_INSUFFICIENT_BUFFER is returned.

ID Field Values

The *id* field of *lpInBuf* may be one of the following values:

ITC_REGISTRY_SAVE_ENABLE

Enables or disables the save registry to non-volatile media feature of the RegFlushKey() function. *lpOutBuf* must be set to zero (FALSE) if the feature is to be disabled or one (TRUE) if the feature is to be enabled.

ITC_DOCK_SWITCH

This IOCTL sets a position of the dock switch. The dock switch may be set to either “modem” or “serial” positions. *lpOutBuf* must point to a buffer that contains a byte value of either DOCK_MODEM or DOCK_SERIAL as defined in oemioctl.h; the value specifies the position the switch is to be set. The call appears as follows:

```
// port = DOCK_MODEM or DOCK_SERIAL as defined in oemioctl.h
BOOL SetDockSwitch( BYTE port)
{
    DWORD cmd = ITC_DOCK_SWITCH;
    DWORD cbRet;

    return KernelIoControl( IOCTL_HAL_ITC_WRITE_SYSPARM, &cmd, sizeof( cmd ),
&port, sizeof( port ), &cbRet )
}
```

ITC_WAKEUP_MASK

This IOCTL sets a bit mask that represents the mask for the five programmable wakeup keys. The I/O key is not a programmable wakeup key. By default it is always the system resume key and all other keys are set to disable key wakeup. A zero in a bit position masks the wakeup for that key. A one in a bit position enables wakeup for that key. *lpOutBuf* must point to a buffer that contains a byte value of a wakeup mask consisting of the OR’ed constants as defined in oemioctl.h. Only the following keys are programmable as wakeup events.

```
#define SCANNER_TRIGGER 1
#define SCANNER_LEFT 2
#define SCANNER_RIGHT 4
#define GOLD_A1 8
#define GOLD_A2 0x10
```

ITC_AMBIENT_KEYBOARD

This IOCTL sets the threshold for the keypad ambient sensor. This can be a value from 0 (always off) to 255 (always on). *lpOutBuf* must point to a buffer that contains a byte value of the desired setting.

ITC_AMBIENT_FRONTLIGHT

This IOCTL sets the threshold for the frontlight ambient sensor. This can be a value from 0 (always off) to 255. *lpOutBuf* must point to a buffer that contains a byte value of the desired setting.

IOCTL_HAL_GET_DEVICEID

This returns the device ID. There are two types of device IDs supported, which are differentiated based on the size of the *output* buffer. The UUID is returned if the buffer size is set to *sizeof(UNIQUE_DEVICEID)*, otherwise the oldstyle device ID is returned.

Usage

```
#include “pkfuncs.h”
#include “deviceid.h”
```

Syntax

```
BOOL KernelIoControl( IOCTL_HAL_GET_DEVICEID, LPVOID
lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD
nOutBufSize, LPDWORD lpBytesReturned );
```

Parameters

lpInBuf	Should be set to NULL. STRICT_ID settings are not supported.
lpInBufSize	Should be set to zero.
lpOutBuf	Must point to a UNIQUE_DEVICEID structure as defined by DEVICEID.H if the UUID is to be returned
nOutBufSize	The size of the UNIQUE_DEVICEID in bytes if the UUID is to be returned. A DEVICE_ID as defined by PKFUNCS.H is returned if the size in bytes is greater than or equal to <i>sizeof(DEVICE_ID)</i> .
lpBytesReturned	The number of bytes returned by the function.

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the extended error value.

IOCTL_HAL_GET_OAL_VERINFO

Returns the HAL version information of the Pocket PC image.

Usage

#include "oemioctl.h"

Syntax

```
BOOL KernelIoControl( IOCTL_HAL_GET_OAL_VERINFO, LPVOID
lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD
nOutBufSize, LPDWORD lpBytesReturned );
```

Parameters

lpInBuf	Should be set to NULL.
lpInBufSize	Should be set to zero.
lpOutBuf	Must point to a VERSIONINFO structure as defined by OEMIOCTL.H. The fields should have these values: <ul style="list-style-type: none"> • cboemverinfo sizeof (tagOemVerInfo); • verinfover 1 • sig; "ITC\0" • id; 'N' • tgtcustomer "" • tgtplat SeaRay • tgtplatversion Current build version number • tgtcpu "Intel\0" • tgtpu "PXA255\0"; • tgtcoreversion "" • date Build time • time Build date
nOutBufSize	The size of VERSIONINFO in bytes.
lpBytesReturned	Returns <i>sizeof(PVERSIONINFO)</i> .

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the extended error value.

IOCTL_HAL_GET_BOOTLOADER_VERINFO

Returns the HAL version information of the OS image.

Usage

#include "oemioctl.h"

Syntax

```
BOOL KernelIoControl( IOCTL_HAL_GET_OAL_VERINFO, LPVOID
lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD
nOutBufSize, LPDWORD lpBytesReturned );
```

Parameters

lpInBuf	Should be set to NULL.
nInBufSize	Should be set to zero.
lpOutBuf	Must point to a VERSIONINFO structure as defined by OEMIOCTL.H. The fields should have these values: <ul style="list-style-type: none"> • cboemverinfo Sizeof (tagOemVerInfo); • verinfover 1 • sig; "ITC\0" • id; 'B' • tgtcustomer "" • tgtplat SeaRay • tgtplatversion Current build version # of bootstrap loader • tgtcputype[8]; "Intel\0"; • tgtcpu "PXA255\0" • tgtcoreversion "" • date Build time • time Build date
nOutBufSize	The size of VERSIONINFO in bytes.
lpBytesReturned	The number of bytes returned to <i>lpOutBuf</i> .

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the extended error value.

IOCTL_HAL_WARMBOOT

Causes the system to perform a warm-boot. The object store is retained.

Usage

#include "oemioctl.h"

Syntax

```
BOOL KernelIoControl( IOCTL_HAL_WARMBOOT, LPVOID
lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD
nOutBufSize, LPDWORD lpBytesReturned );
```

Parameters

lpInBuf	Should be set to NULL.
lpInBufSize	Should be set to zero.
lpOutBuf	Should be NULL.
nOutBufSize	Should be zero.

Return Values

None.

IOCTL_HAL_COLDBOOT

Causes the system to perform a cold-boot. The object store is cleared.

Usage

#include "oemioctl.h"

Syntax

```
BOOL KernelIoControl( IOCTL_HAL_COLDBOOT, LPVOID
lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD
nOutBufSize, LPDWORD lpBytesReturned );
```

Parameters

lpInBuf	Should be set to NULL.
lpInBufSize	Should be set to zero.
lpOutBuf	Should be NULL.
nOutBufSize	Should be zero.

Return Values

None.

IOCTL_HAL_GET_RESET_INFO

This code allows software to check the type of the most recent reset.

Usage

#include "oemioctl.h"

Syntax

```
BOOL KernelIoControl( IOCTL_HAL_GET_RESET_INFO, LPVOID
lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD
nOutBufSize, LPDWORD lpBytesReturned );
```

Parameters

lpInBuf	Should be set to NULL.
lpInBufSize	Should be set to zero.
lpOutBuf	Must point to a HAL_RESET_INFO structure. See sample below.
nOutBufSize	The size of HAL_RESET_INFO in bytes.
lpBytesReturned	The number of bytes returned by the function.

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the extended error value.

Sample

```
typedef struct {
DWORD ResetReason;// most recent reset type
DWORD ObjectStoreState;// state of object store
} HAL_RESET_INFO, * PHAL_RESET_INFO;

// Reset reason types
#define HAL_RESET_TYPE_UNKNOWN0
#define HAL_RESET_REASON_HARDWARE1// cold
#define HAL_RESET_REASON_SOFTWARE2// suspend
#define HAL_RESET_REASON_WATCHDOG4
#define HAL_RESET_BATT_FAULT8// power fail
#define HAL_RESET_VDD_FAULT16// warm boot

// Object store state flags
#define HAL_OBJECT_STORE_STATE_UNKNOWN0
#define HAL_OBJECT_STORE_STATE_CLEAR1
```

IOCTL_HAL_GET_BOOT_DEVICE

This IOCTL code allows software to check which device CE booted from.

Usage

#include "oemioctl.h"

Syntax

```
BOOL KernelIoControl( IOCTL_HAL_GET_BOOT_DEVICE,LPVOID
lpInBuf,DWORD nInBufSize,LPVOID lpOutBuf,DWORD
nOutBufSize,LPDWORD lpBytesReturned );
```

Parameters

lpInBuf	Should be set to NULL.
lpInBufSize	Should be set to zero.
lpOutBuf	Must point to a buffer large enough to hold a DWORD (4 bytes) with the boot device. The following boot devices are supported: #define HAL_BOOT_DEVICE_UNKNOWN0 #define HAL_BOOT_DEVICE_ROM_XIP 1 #define HAL_BOOT_DEVICE_ROM 2 #define HAL_BOOT_DEVICE_PCMCIA_ATA 3 #define HAL_BOOT_DEVICE_PCMCIA_LINEAR 4 #define HAL_BOOT_DEVICE_IDE_ATA 5 #define HAL_BOOT_DEVICE_IDE_ATAPI 6
nOutBufSize	The size of <i>lpOutBuf</i> in bytes (4).
lpBytesReturned	The number of bytes returned by the function.

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the extended error value.

IOCTL_HAL_REBOOT



Note: Using this is no longer recommended, use IOCTL_HAL_WARMBOOT (page 73) or IOCTL_HAL_COLDBOOT (page 74). This is supported for backward-compatibility but its use can lead to difficulties.

Causes the system to perform a warm-boot. The object store is retained.

Usage

#include "oemioctl.h"

Syntax

```
BOOL KernelIoControl( IOCTL_HAL_REBOOT, LPVOID
lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD
nOutBufSize, LPDWORD lpBytesReturned );
```

Parameters

lpInBuf	Should be set to NULL.
lpOutBuf	Should be NULL.
lpInBufSize	Should be set to zero.
nOutBufSize	Should be zero.

Return Values

None.

IOCTL_PROCESSOR_INFORMATION

Returns processor information.

Usage

#include "pkfuncs.h"

Syntax

```
BOOL KernelIoControl( IOCTL_PROCESSOR_INFORMATION, LPVOID
lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD
nOutBufSize, LPDWORD lpBytesReturned );
```

Parameters

<code>lpInBuf</code>	Should be set to NULL.
<code>nInBufSize</code>	Should be set to zero.
<code>lpOutBuf</code>	Should be a pointer to the <code>PROCESSOR_INFO</code> structure. Its structure stores information describing the CPU more descriptively.
<pre>typedef __PROCESSOR_INFO { WORD wVersion; // Set to value 1 WCHAR szProcessorCore[40]; // "ARM\0" WORD wCoreRevision; // 4 WCHAR szProcessorName[40]; // "PXA255\0" WORD wProcessorRevision; // 0 WCHAR szCatalogNumber[100]; // 0 WCHAR szVendor[100]; // "Intel Corporation\0" DWORD dwInstructionSet; // 0 DWORD dwClockSpeed; // 400 }</pre>	
<code>nOutBufSize</code>	Should be set to <code>sizeof(PROCESSOR_INFO)</code> in bytes.
<code>lpBytesReturned</code>	Returns <code>sizeof(PROCESSOR_INFO)</code> ;

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. `GetLastError()` may be used to get the extended error value.

IOCTL_GET_CPU_ID

Returns Xscale processor ID.

Usage

#include "oemioctl.h"

Syntax

```
BOOL KernelIoControl( IOCTL_GET_CPU_ID, LPVOID lpInBuf,
DWORD nInBufSize, LPVOID lpOutBuf, DWORD nOutBufSize, LPDWORD
lpBytesReturned );
```

Parameters

<code>lpInBuf</code>	Should point to a <code>CPUIInfo</code> structure defined in <code>OEMIOCTL.H</code> .
<code>lpInBufSize</code>	Should be <code>sizeof(CPUIInfo)</code> .
<code>lpOutBuf</code>	Should be NULL.
<code>nOutBufSize</code>	Should be set to 0.
<code>lpBytesReturned</code>	Returns <code>sizeof(PROCESSOR_INFO)</code> ;

Return Values

Returns TRUE if function succeeds. Returns FALSE if the function fails. `GetLastError()` may be used to get the extended error value.

Networking APIs

The API provided by Intermec Technologies exposes a limited set of routines that allows a programmer to access and affect the 802.11b/g network interface card from within their application. The routines provided also reads/writes values to the CE registry that pertain to the 802.11b/g radio driver. By using the provided functions, a programmer can alter the 802.11b/g parameters of Network Name (SSID), WEP keys, infrastructure modes, radio channel, and power management modes. A programmer can also retrieve network connect status and signal strength indication from the RF network card.

The API is contained within the 80211API.DLL file that should be present in any load with the 802.11b/g networking installed.

80211API.DLL	This file is an Intermec authored file that provides the programmer with a set of API calls to configure or monitor status of the 802.11b/g network.
80211PM.DLL	This handles profile management for radio configurable values.
URODDSVX.EXE	This handles radio configuration and security authentication based on a selected profile. There is a user interface to this service that provides status of the supplicant as well as status of the 80211b/g authentication process.
ZNICZIO.DLL	A replacement for NDISUIO.DLL that supports the Funk Supplicant.

The Profile Manager supports up to four radio configuration profiles. These profiles are the same as those set by the Wireless Network control panel applet that runs on the Windows CE unit. You can configure different 802.11b/g profiles and switch between them using the 802.11 API. See “[ConfigureProfile\(\)](#)” on page 92 for more information.

Basic Connect/Disconnect Functions

Below are functions available for the 751G when enabled with the 802.11b/g radio module.

RadioConnect()

Connects to the available radio. Use this function if you plan on using a lot of API calls that talk directly to the radio. Note that the 802.11b/g radio must be enabled via NDISTRAY before you can connect to it.

Syntax	UINT RadioConnect ();
Parameters	None.
Return Values	ERROR_SUCCESS when successful, otherwise ERR_CONNECT_FAILED
Remarks	Call this function before you call any other function found within this API. It hunts out and connects to the 802.11b/g radio available on the system. Check extended error codes if it returns anything for information.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_RadioConnect)(); #else UINT RadioConnect(); #endif</pre>

RadioDisconnect()

Call this function when done using the 802.11 API to clean up a connection from a previous `RadioConnect()` call. If you do not call this function, you may leave memory allocated.

Syntax	<code>UINT RadioDisconnect();</code>
Parameters	None.
Return Values	<code>ERROR_SUCCESS</code> when successful, otherwise <code>ERR_CONNECT_FAILED</code>
Remarks	None.
Definitions	<code>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_RadioDisconnect)(); #else UINT RadioDisconnect(); #endif</code>

RadioDisassociate()

Call this function to have the 802.11b/g radio disassociate from the current service set. The radio then enters an “off” mode until it is woken again by setting the Service Set Identifier (SSID). Also, the NDIS driver generates an NDIS media disconnect event.

Syntax	<code>UINT RadioDisassociate();</code>
Parameters	None
Return Values	<code>ERROR_SUCCESS</code> when successful, otherwise <code>ERR_CONNECT_FAILED</code> .
Remarks	None
Definitions	<code>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_RadioDisassociate)(); #else UINT RadioDisassociate(); #endif</code>

Query Information Functions**GetAssociationStatus()**

Call this to obtain the radio’s current association status with a service set.

Syntax	<code>UINT GetAssociationStatus(ULONG &);</code>
Parameters	<code>NDIS_RADIO_ASSOCIATED</code> Indicates the radio is associated with an access point <code>NDIS_RADIO_SCANNING</code> Indicates radio is looking for an access point with which to associate
Return Values	<code>ERROR_SUCCESS</code> when successful, <code>ERR_QUERY_FAILED</code> when the query failed, or <code>ERR_CONNECT_FAILED</code> if a connection with the radio failed.
Remarks	Data is only valid if the function returns <code>ERROR_SUCCESS</code> . Also, if <code>ERROR_SUCCESS</code> is returned, your <code>ULONG</code> reference is populated by one of the parameters listed above.
Definitions	<code>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_GetAssociationStatus)(ULONG &); #else UINT GetAssociationStatus(ULONG &); #endif</code>

GetAuthenticationMode()

Call this function to obtain the radio's current authentication mode.

Syntax	<code>UINT GetAuthenticationMode(ULONG &);</code>	
Parameters	<code>NDIS_RADIO_AUTH_MODE_OPEN</code>	802.11b/g Open Authentication. Indicates that the radio is using an open system.
	<code>NDIS_RADIO_AUTH_MODE_SHARED</code>	802.11b/g Shared Authentication. Indicates that the radio is using a shared key.
	<code>NDIS_RADIO_AUTH_MODE_AUTO</code>	Auto switch between Open/Shared. Indicates automatic detection is used when available.
	<code>NDIS_RADIO_AUTH_MODE_ERROR</code>	Defined as error value. Indicates the authentication mode was not determined at this time or is unknown.
	<code>NDIS_RADIO_AUTH_MODE_WPA</code>	WPA Authentication
	<code>NDIS_RADIO_AUTH_MODE_WPA_PSK</code>	WPA Preshared Key Authentication
	<code>NDIS_RADIO_AUTH_MODE_WPA_NONE</code>	WPA None
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.	
Remarks	Data is only valid if ERROR_SUCCESS is returned. Also, if ERROR_SUCCESS is returned, your USHORT reference is populated with one of the parameters listed above.	
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_GetAuthenticationMode)(ULONG &); #else UINT GetAuthenticationMode(ULONG &); #endif</pre>	

GetBSSID()

Call this function to get the current MAC address (BSSID) of the service set. In ESS mode, this is the MAC address of the access point the radio is associated with. In IBSS mode, this is a randomly generated MAC address, and serves as the ID for the IBSS.

Syntax	<code>UINT GetBSSID(TCHAR *);</code>	
Parameters	Pointer to a character array, which is populated with the current BSSID after a successful call.	
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.	
Remarks	If ERROR_SUCCESS is returned, your TCHAR array is populated with the BSSID of the current service set: xx-xx-xx-xx-xx-xx	
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_GetBSSID)(TCHAR *); #else UINT GetBSSID(TCHAR *); #endif</pre>	

GetDiversity()

Call this function to get the current diversity setting of your 802.11b/g radio. This function uses an optional NDIS5.1 OID to query the radio, which a large number of 802.11b/g devices do not support. This function may be inaccurate.

Syntax	<code>UINT GetDiversity(USHORT *);</code>
Parameters	<p><code>ANT_PRIMARY</code> The primary antenna is selected.</p> <p><code>ANT_SECONDARY</code> The secondary antenna is selected.</p> <p><code>ANT_DIVERSITY</code> The radio is in diversity mode, and uses both antennas</p>
Return Values	<code>ERROR_SUCCESS</code> when successful, <code>ERR_QUERY_FAILED</code> when the query failed, or <code>ERR_CONNECT_FAILED</code> if a connection with the radio failed.
Remarks	If <code>ERROR_SUCCESS</code> is returned, your <code>USHORT</code> reference is populated with one of the parameters listed above.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_GetDiversity)(USHORT *); #else UINT GetDiversity(USHORT *); #endif</pre>

GetLinkSpeed()

Call this function to get the current link speed of the 802.11b/g radio.

Syntax	<code>UINT GetLinkSpeed(int &);</code>
Parameters	This function accepts an <code>int</code> reference, and your <code>int</code> is populated with the current link speed, in Mbps, rounded to the nearest whole integer, for example: 1, 2, 5, 11, etc.
Return Values	<code>ERROR_SUCCESS</code> when successful, <code>ERR_QUERY_FAILED</code> when the query failed, or <code>ERR_CONNECT_FAILED</code> if a connection with the radio failed.
Remarks	Data returned is valid if <code>ERROR_SUCCESS</code> is returned.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_GetLinkSpeed)(int &); #else UINT GetLinkSpeed(int &); #endif</pre>

GetMac()

Call this function to get the MAC address of the 802.11b/g radio. *Call RadioConnect() before calling this function for this function to work properly.*

Syntax	UINT GetMac (TCHAR *);
Parameters	Pointer to a character array, which is populated with the MAC address after a successful call.
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.
Remarks	If ERROR_SUCCESS is returned, your TCHAR array is populated with the formatted MAC address of the adapter, as follows: xx-xx-xx-xx-xx-xx
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_GetMac)(TCHAR *); #else UINT GetMac(TCHAR *); #endif</pre>

GetNetworkMode()

Call this function to get the current Network Mode (SSID) for the 802.11b/g radio.

Syntax	UINT GetNetworkMode (ULONG &);												
Parameters	<table border="0"> <tr> <td>NDIS_NET_MODE_IBSS</td> <td>802.11b/g Ad-Hoc Mode.</td> </tr> <tr> <td>NDIS_NET_MODE_ESS</td> <td>802.11b/g Infrastructure Mode.</td> </tr> <tr> <td>NDIS_NET_MODE_UNKNOWN</td> <td>Anything Else/Unknown Error</td> </tr> <tr> <td>NDIS_NET_AUTO_UNKNOWN</td> <td>Automatic Selection. <i>Use of this option is not supported or recommended.</i></td> </tr> <tr> <td>NDIS_NET_TYPE_OFDM_5G</td> <td>5 Gigahertz 54 Mbps</td> </tr> <tr> <td>NDIS_NET_TYPE_OFDM_2_4G</td> <td>802.11b/g 2.4 Gigahertz</td> </tr> </table>	NDIS_NET_MODE_IBSS	802.11b/g Ad-Hoc Mode.	NDIS_NET_MODE_ESS	802.11b/g Infrastructure Mode.	NDIS_NET_MODE_UNKNOWN	Anything Else/Unknown Error	NDIS_NET_AUTO_UNKNOWN	Automatic Selection. <i>Use of this option is not supported or recommended.</i>	NDIS_NET_TYPE_OFDM_5G	5 Gigahertz 54 Mbps	NDIS_NET_TYPE_OFDM_2_4G	802.11b/g 2.4 Gigahertz
NDIS_NET_MODE_IBSS	802.11b/g Ad-Hoc Mode.												
NDIS_NET_MODE_ESS	802.11b/g Infrastructure Mode.												
NDIS_NET_MODE_UNKNOWN	Anything Else/Unknown Error												
NDIS_NET_AUTO_UNKNOWN	Automatic Selection. <i>Use of this option is not supported or recommended.</i>												
NDIS_NET_TYPE_OFDM_5G	5 Gigahertz 54 Mbps												
NDIS_NET_TYPE_OFDM_2_4G	802.11b/g 2.4 Gigahertz												
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.												
Remarks	If ERROR_SUCCESS is returned, your ULONG reference is populated with one of the parameters listed above.												
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_GetNetworkMode)(ULONG &); #else UINT GetNetworkMode(ULONG &); #endif</pre>												

GetNetworkType()

Call this function to get the current network type of the radio. Do not confuse this with GetNetworkMode().

Syntax	<code>UINT GetNetworkType(ULONG &);</code>
Parameters	<p><code>NDIS_NET_TYPE_FH</code> Indicates this is a frequency hopping radio.</p> <p><code>NDIS_NET_TYPE_DS</code> Indicates that this is a direct sequence radio.</p> <p><code>NDIS_NET_TYPE_UNDEFINED</code> Indicates this radio type is unknown or undefined.</p>
Return Values	<code>ERROR_SUCCESS</code> when successful, <code>ERR_QUERY_FAILED</code> when the query failed, or <code>ERR_CONNECT_FAILED</code> if a connection with the radio failed.
Remarks	If <code>ERROR_SUCCESS</code> is returned, your <code>ULONG</code> reference is populated with one of the parameters listed above.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_GetNetworkType)(ULONG &); #else UINT GetNetworkType(ULONG &); #endif</pre>

GetSSID()

Call this function to get the desired SSID of the 802.11b/g radio. *Call RadioConnect() before calling this function for this function to work properly.*

Syntax	<code>UINT GetSSID(TCHAR *);</code>
Parameters	Pointer to a character array, which is populated with the current SSID when successful.
Return Values	<code>ERROR_SUCCESS</code> when successful, <code>ERR_QUERY_FAILED</code> when the query failed, or <code>ERR_CONNECT_FAILED</code> if a connection with the radio failed.
Remarks	If <code>ERROR_SUCCESS</code> is returned, your <code>TCHAR</code> array is populated with the desired SSID.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_GetSSID)(TCHAR *); #else UINT GetSSID(TCHAR *); #endif</pre>

GetPowerMode()

Call this function to get the current power savings mode of the radio. *Note: Do not use Automatic Switching mode at this time.*

Syntax	<code>UINT GetPowerMode(ULONG &);</code>	
Parameters	<code>NDIS_RADIO_POWER_MODE_CAM</code>	Continuous Access Mode (<i>ie: always on</i>).
	<code>NDIS_RADIO_POWER_MODE_PSP</code>	Power Saving Mode.
	<code>NDIS_RADIO_POWER_UNKNOWN</code>	Unknown power mode.
	<code>NDIS_RADIO_POWER_AUTO</code>	Auto.
	<code>NDIS_RADIO_POWER_MODE_FAST_PSP</code>	Fast PSP, good savings, fast
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.	
Remarks	If ERROR_SUCCESS is returned, the ULONG reference is populated with a parameter listed above.	
Definitions	<pre>#ifndef DYNAMIC_LOADING typedef UINT (*PFN_GetPowerMode)(ULONG &); #else UINT GetPowerMode(ULONG &); #endif</pre>	

GetRSSI()

Call this function to get the current RSSI (Radio Signal Strength Indicator), in Dbm.

Syntax	<code>UINT GetRSSI(ULONG &);</code>	
Parameters	References a ULONG populated with current RSSI after a successful call.	
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.	
Remarks	If ERROR_SUCCESS is returned, your ULONG reference contains the RSSI. Valid RSSI range is from -100 Dbm to -30 Dbm.	
Definitions	<pre>#ifndef DYNAMIC_LOADING typedef UINT (*PFN_GetRSSI)(ULONG &); #else UINT GetRSSI(ULONG &); #endif</pre>	

GetTXPower()

Call this function to get the current transmit power of the radio.

Syntax	<code>UINT GetTXPower(ULONG &);</code>	
Parameters	<code>NDIS_POWER_LEVEL_63</code>	63 mW
	<code>NDIS_POWER_LEVEL_30</code>	30 mW
	<code>NDIS_POWER_LEVEL_15</code>	15 mW
	<code>NDIS_POWER_LEVEL_5</code>	5 mW
	<code>NDIS_POWER_LEVEL_1</code>	1 mW
	<code>NDIS_POWER_LEVEL_UNKNOWN</code>	Unknown Value or Error.
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.	
Remarks	If ERROR_SUCCESS is returned, your ULONG reference is populated with the TX power in milliwatts (mW). Valid ranges are from 5 mW to 100 mW.	
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_GetTXPower)(ULONG &); #else UINT GetTXPower(ULONG &); #endif</pre>	

GetWepStatus()

Call this to get the current state of the radio's WEP and encryption levels.

Syntax	<code>UINT GetWepStatus(ULONG &);</code>	
Parameters	<code>NDIS_ENCRYPTION_1_ENABLED</code>	WEP is enabled; TKIP and AES are not enabled, and a transmit key may or may not be available. (<i>same as NDIS_RADIO_WEP_ENABLED</i>)
	<code>NDIS_ENCRYPTION_DISABLED</code>	Indicates AES, TKIP, WEP are disabled, transmit key available. (<i>Same as NDIS_RADIO_WEP_DISABLED</i>)
	<code>NDIS_ENCRYPTION_NOT_SUPPORTED</code>	Indicates WEP, TKIP, AES not supported. (<i>Same as NDIS_RADIO_WEP_NOT_SUPPORTED</i>)
	<code>NDIS_ENCRYPTION_1_KEY_ABSENT</code>	Indicates AES, TKIP, WEP disabled, transmit key not available. (<i>Same as NDIS_RADIO_WEP_ABSENT</i>)
	<code>NDIS_ENCRYPTION_2_ENABLED</code>	Indicates that TKIP and WEP are enabled; AES is not enabled, and a transmit key is available.
	<code>NDIS_ENCRYPTION_2_KEY_ABSENT</code>	Indicates no transmit keys available for use by TKIP or WEP, TKIP, WEP enabled; AES is not enabled.
	<code>NDIS_ENCRYPTION_3_ENABLED</code>	Indicates that AES, TKIP, and WEP are enabled, and a transmit key is available.
	<code>NDIS_ENCRYPTION_3_KEY_ABSENT</code>	Indicates no transmit keys available for use by AES, TKIP, WEP; and AES, TKIP, WEP enabled.
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.	
Remarks	If ERROR_SUCCESS is returned, your ULONG reference is populated with one of the parameters listed above.	
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_GetWepStatus)(ULONG &); #else UINT GetWepStatus(ULONG &); #endif</pre>	

GetRadioIpAddress()

Call this function to obtain a formatted string indicating whether DHCP is enabled, and what is the current adapters IP address.

Syntax	<code>UINT GetRadioIpAddress(TCHAR *);</code>
Parameters	Pointer to a character array that contains the formatted string of the IP address and static/DHCP information.
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.
Remarks	If ERROR_SUCCESS is returned, your TCHAR array contains a string formatted as follows: IP: DHCP Enabled\nxxx.xxx.xxx.xxx\n or IP: DHCP Disabled\nxxx.xxx.xxx.xxx\n
Definitions	<code>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_GetRadioIpAddress)(TCHAR *); #else UINT GetRadioIpAddress(TCHAR *); #endif</code>

GetCCXStatus()

Call this to get information about the current CCX status of the adapter.

Syntax	<code>UINT GetCCXStatus(ULONG &);</code>
Parameters	NDIS_NETWORK_EAP_MODE_OFF Disable EAP mode. NDIS_NETWORK_EAP_MODE_ON Enable EAP mode.
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.
Remarks	If ERROR_SUCCESS is returned, your ULONG reference is populated with one of parameters listed above.
Definitions	<code>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_GetCCXStatus)(ULONG &); #else UINT GetCCXStatus(ULONG &); #endif</code>

Set Information Functions

AddWep()

Call this function to add a WEP key to the radio. Call this function multiple times when adding more than one WEP key. Save the “default” key for last. For example, when adding four keys, and the second key is the default transmit key, add keys 1, 3 and 4 *before* you add key 2. **Note:** *Add the default transmit key last.*

Syntax	UINT AddWep (ULONG, BOOL, TCHAR *);
Parameters	ULONG Specifies the key index to be set. Valid values are 0–3. BOOL When set to TRUE, specifies that this key is the default transmit key. TCHAR Pointer to a character array that specifies the key data in either HEX (length of 10 or 26) or ASCII (length of 5 or 13). This string must be null-terminated.
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.
Remarks	When adding WEP keys to the radio, turn off encryption before you add the keys, then turn encryption back on afterwards. Also, be sure to add the TRANSMIT KEY last.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_AddWep)(ULONG, BOOL, TCHAR *); #else UINT AddWep(ULONG, BOOL, TCHAR *); #endif</pre>

EnableWep()

Enables or disables WEP encryption on the radio (TRUE/FALSE).

Syntax	UINT EnableWep (BOOL);
Parameters	Set BOOL to TRUE to enable WEP encryption, or FALSE to disable WEP encryption.
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.
Remarks	Call this function with TRUE as the parameter to enable WEP encryption. Call this function with the FALSE parameter to disable WEP encryption. This is an alias for EncryptionStatus(). See the following: EnableWEP(TRUE) = EncryptionStatus(NDIS_ENCRYPTION_1_ENABLED) EnableWEP(FALSE) = EncryptionStatus(NDIS_ENCRYPTION_DISABLED)
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_EnableWep)(BOOL); #else UINT EnableWep(BOOL); #endif</pre>

EncryptionStatus()

Call this function to set the desired encryption status.

Syntax	UINT EncryptionStatus (UINT <i>mode</i>);	
Parameters	NDIS_ENCRYPTION_1_ENABLED	WEP is enabled; TKIP and AES are not enabled, and a transmit key may or may not be available. (<i>same as NDIS_RADIO_WEP_ENABLED</i>)
	NDIS_ENCRYPTION_DISABLED	Indicates that AES, TKIP, and WEP are disabled, and a transmit key is available. (<i>Same as NDIS_RADIO_WEP_DISABLED</i>)
	NDIS_ENCRYPTION_NOT_SUPPORTED	Indicates that encryption (WEP, TKIP, and AES) is not supported. (<i>Same as NDIS_RADIO_WEP_NOT_SUPPORTED</i>)
	NDIS_ENCRYPTION_1_KEY_ABSENT	Indicates that AES, TKIP, and WEP are disabled, and a transmit key is not available. (<i>Same as NDIS_RADIO_WEP_ABSENT</i>)
	NDIS_ENCRYPTION_2_ENABLED	Indicates that TKIP and WEP are enabled; AES is not enabled, and a transmit key is available.
	NDIS_ENCRYPTION_2_KEY_ABSENT	Indicates that there are no transmit keys available for use by TKIP or WEP, TKIP and WEP are enabled; and AES is not enabled.
	NDIS_ENCRYPTION_3_ENABLED	Indicates that AES, TKIP, and WEP are enabled, and a transmit key is available.
	NDIS_ENCRYPTION_3_KEY_ABSENT	Indicates that there are no transmit keys available for use by AES, TKIP, or WEP, and AES, TKIP, and WEP are enabled.
	Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.
Remarks	None.	
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_EncryptionStatus)(UINT mode); #else UINT EncryptionStatus(UINT mode); #endif</pre>	

RemoveWep()

Call this with a key index of 0-3 to remove the WEP key at that index.

Syntax	UINT RemoveWep (ULONG);	
Parameters	ULONG value that specifies the key index to set. Valid values are 0–3.	
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.	
Remarks	On disassociation with all BSSIDs of the current service set, WEP key is removed by the adapter.	
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_RemoveWEP)(ULONG); #else UINT RemoveWep(ULONG); #endif</pre>	

SetAuthenticationMode()

Call this function to set the desired authentication mode.

Syntax	<code>UINT SetAuthenticationMode(ULONG);</code>	
Parameters	<code>NDIS_RADIO_AUTH_MODE_OPEN</code>	802.11b/g Open Authentication. Indicates that the radio is using an open system.
	<code>NDIS_RADIO_AUTH_MODE_SHARED</code>	802.11b/g Shared Authentication. Indicates that the radio is using a shared key.
	<code>NDIS_RADIO_AUTH_MODE_AUTO</code>	Auto switch between Open/Shared. Indicates automatic detection is used when available.
	<code>NDIS_RADIO_AUTH_MODE_ERROR</code>	Defined as error value. Indicates the authentication mode was not determined at this time or is unknown.
	<code>NDIS_RADIO_AUTH_MODE_WPA</code>	WPA Authentication
	<code>NDIS_RADIO_AUTH_MODE_WPA_PSK</code>	WPA Preshared Key Authentication
	<code>NDIS_RADIO_AUTH_MODE_WPA_NONE</code>	WPA None
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.	
Remarks	None.	
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_SetAuthenticationMode)(ULONG); #else UINT SetAuthenticationMode(ULONG); #endif</pre>	

SetChannel()

This function is currently not implemented. Ad-hoc networks automatically select a channel or use the already existing channel.

Syntax	<code>UINT SetChannel(USHORT);</code>	
Parameters	USHORT value that should populate with the desired channel (1-14).	
Return Values	None.	
Remarks	None.	
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_SetChannel)(USHORT); #else UINT SetChannel(USHORT); #endif</pre>	

SetNetworkMode()

Call this function to set the desired Network Mode.

Syntax	UINT SetNetworkMode (ULONG);	
Parameters	NDIS_NET_MODE_IBSS	802.11b/g Ad-Hoc Mode.
	NDIS_NET_MODE_ESS	802.11b/g Infrastructure Mode.
	NDIS_NET_MODE_UNKNOWN	Anything Else/Unknown Error
	NDIS_NET_AUTO_UNKNOWN	Automatic Selection. <i>Using this is not supported or recommended</i>
	NDIS_NET_TYPE_OFDM_5G	5 Gigahertz 54 Mbps
	NDIS_NET_TYPE_OFDM_2_4G	802.11b/g 2.4 Gigahertz
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.	
Remarks	None.	
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_SetNetworkMode)(ULONG); #else UINT SetNetworkMode(ULONG); #endif</pre>	

SetPowerMode()

Call this function to set the desired power mode.

Syntax	UINT SetPowerMode (ULONG mode);	
Parameters	NDIS_RADIO_POWER_MODE_CAM	Continuous Access Mode (<i>ie: always on</i>).
	NDIS_RADIO_POWER_MODE_PSP	Power Saving Mode.
	NDIS_RADIO_POWER_UNKNOWN	Unknown power mode.
	NDIS_RADIO_POWER_AUTO	Auto.
	NDIS_RADIO_POWER_MODE_FAST_PSP	Fast PSP, good savings, fast
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.	
Remarks	None.	
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_SetPowerMode)(ULONG mode); #else UINT SetPowerMode(ULONG mode); #endif</pre>	

SetSSID()

Call this function with a pointer to a null-terminated TCHAR array containing the desired SSID to set the desired SSID of the adapter.

Syntax	UINT SetSSID (TCHAR *);
Parameters	Pointer to a character array that contains the desired SSID. This should be null-terminated.
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.
Remarks	If an “ANY” network is desired, pass in _T(“ANY”).
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_SetSSID)(TCHAR *); #else UINT SetSSID(TCHAR *); #endif</pre>

SetCCXStatus()

Call this function to set the desired CCX / Network EAP status.

Syntax	UINT SetCCXStatus (ULONG);
Parameters	NDIS_NETWORK_EAP_MODE_OFF Disable Network EAP / CCX NDIS_NETWORK_EAP_MODE_ON Enable Network EAP / CCX
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.
Remarks	None.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_SetCCXStatus)(ULONG); #else UINT SetCCXStatus(ULONG); #endif</pre>

SetMixedCellMode()

Call this function to set the desired mixed cell mode.

Syntax	UINT SetMixedCellMode (ULONG);
Parameters	NDIS_MIXED_CELL_OFF Disable Mixed Cell NDIS_MIXED_CELL_ON Enable Mixed Cell
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.
Remarks	None.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_SetMixedCellMode)(ULONG); #else UINT SetMixedCellMode(ULONG); #endif</pre>

Helper Functions

ConfigureProfile()

If using the Intermec 802.11b/g Profile Management system, you can program the API to configure the radio to a specific profile by passing the profile name.

Syntax	<code>UINT ConfigureProfile(TCHAR *);</code>
Parameters	Pointer to a character array that contains the profile name. This should be null-terminated.
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.
Remarks	Call this function with a pointer to a null-terminated TCHAR array that contains the name of the profile you wish to configure. This function reads profile data from the profile manager, sets that profile as the default active profile, and configures the radio appropriately. If needed, the supplicant and any other related services are automatically started and stopped.
Definitions	<code>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_ConfigureProfile)(TCHAR *); #else UINT ConfigureProfile(TCHAR *); #endif</code>

EnableSuppLogging()

Call this function to set the desired supplicant logging mode.

Syntax	<code>UINT EnableSuppLogging(ULONG);</code>				
Parameters	<table border="0"> <tr> <td><code>NDIS_SUPP_LOGGING_ON</code></td> <td>Supplicant Logging Enabled</td> </tr> <tr> <td><code>NDIS_SUPP_LOGGING_OFF</code></td> <td>Supplicant Logging Disabled</td> </tr> </table>	<code>NDIS_SUPP_LOGGING_ON</code>	Supplicant Logging Enabled	<code>NDIS_SUPP_LOGGING_OFF</code>	Supplicant Logging Disabled
<code>NDIS_SUPP_LOGGING_ON</code>	Supplicant Logging Enabled				
<code>NDIS_SUPP_LOGGING_OFF</code>	Supplicant Logging Disabled				
Return Values	ERROR_SUCCESS when successful.				
Remarks	None.				
Definitions	<code>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_EnableSuppLogging)(ULONG); #else UINT EnableSuppLogging(ULONG); #endif</code>				

EnableZeroConfig()

This enables or disables the Wireless Zero Configuration Wizard from Microsoft. After calling this function, a warm-boot is required for the change to take effect. *Note: Enabling this function effectively disables all the SET commands in this API.*

Syntax	UINT EnableZeroConfig (USHORT);
Parameters	TRUE Enable Wireless Zero Config FALSE Disable Wireless Zero Config
Return Values	ERROR_SUCCESS when successful, ERR_ZERO_CONFIG_CHANGE_FAILED when the query failed.
Remarks	Call this function to set the desired Zero Config status.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_EnableZeroConfig)(USHORT); #else UINT EnableZeroConfig(USHORT); #endif</pre>

GetCurrentDriverName()

Call this function to populate the TCHAR array with the driver name.

Syntax	UINT GetCurrentDriverName (TCHAR *);
Parameters	Pointer to a TCHAR array which contains the name of the driver when successful.
Return Values	ERROR_SUCCESS when successful.
Remarks	This function is called with a pointer to a TCHAR array that is large enough to hold the name of the driver PLUS the null terminator.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_GetCurrentDriverName)(TCHAR *); #else UINT GetCurrentDriverName(TCHAR *); #endif</pre>

isDHCPEnabled()

Call this to determine whether DHCP is enabled on the current adapter.

Syntax	UINT isDHCPEnabled ();
Parameters	None.
Return Values	TRUE if DHCP is enabled, FALSE if it is not.
Remarks	None.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_isDHCPEnabled)(); #else UINT isDHCPEnabled(); #endif</pre>

isOrinoco()

Call this function to determine whether the current radio is an ORiNOCO, Lucent, or WaveLAN radio.

Syntax	UINT isOrinoco ();
Parameters	None.
Return Values	TRUE if this is an ORiNOCO radio, and FALSE if it is not.
Remarks	None.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_isOrinoco)(); #else UINT isOrinoco(); #endif</pre>

isSupplicantRunning()

Call this function to determine whether the security supplicant is running.

Syntax	UINT isSupplicantRunning ();
Parameters	None.
Return Values	TRUE if the security supplicant is running, FALSE if it is not running.
Remarks	None.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_isSupplicantRunning)(); #else UINT isSupplicantRunning(); #endif</pre>

isZeroConfigEnabled()

Call this function to determine whether Zero Config is currently enabled.

Syntax	UINT isZeroConfigEnabled ();
Parameters	None.
Return Values	TRUE if ZeroConfig is enabled, and FALSE if it is disabled.
Remarks	None.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_isZeroConfigEnabled)(); #else UINT isZeroConfigEnabled(); #endif</pre>

RenewDHCP()

Call this to force a DHCP renewal on the current network adapter.

Syntax	UINT RenewDHCP() ;
Parameters	None.
Return Values	ERROR_SUCCESS when successful.
Remarks	You should not have to call this function on Microsoft Windows CE 4.2 .NET and later devices.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_RenewDHCP)(); #else UINT RenewDHCP(); #endif</pre>

ResetRadioToSystemSave()

Call this function to force the radio to reset to the last desired active profile.

Syntax	UINT ResetRadioToSystemSave() ;
Parameters	None.
Return Values	ERROR_SUCCESS when successful.
Remarks	None.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_ResetRadioToSystemSave)(); #else UINT ResetRadioToSystemSave(); #endif</pre>

StartScanList()

If a scan list is configured on the system, this causes the API to begin the process of scanning for an available network. This call can take quite a while to process (*depending upon the length of the scan list and how long it takes to find a valid network*), you may wish to call it from a separate thread.

Syntax	UINT StartScanList() ;
Parameters	None.
Return Values	ERROR_SUCCESS when successful.
Remarks	Call this function to start the scan list functionality of the system.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_StartScanList)(); #else UINT StartScanList(); #endif</pre>

StartSupplicant()

Call this to start the supplicant service if it is installed on the system.

Syntax	UINT StartSupplicant ();
Parameters	None.
Return Values	ERROR_SUCCESS when successful.
Remarks	None.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_StartSupplicant)(); #else UINT StartSupplicant(); #endif</pre>

StopSupplicant()

Call this function to stop the supplicant service.

Syntax	UINT StopSupplicant ();
Parameters	None.
Return Values	ERROR_SUCCESS when successful.
Remarks	None.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_StopSupplicant)(); #else UINT StopSupplicant(); #endif</pre>

SwitchPacketDriver()

Call this function to switch between available packet drivers on the system.

Syntax	UINT SwitchPacketDriver (USHORT);
Parameters	INTERMEC_PACKET_DRIVER Intermec Packet Driver (ZNICZIO) NDISUIO_PACKET_DRIVER Microsoft Packet Driver (NDISUIO)
Return Values	ERROR_SUCCESS when successful.
Remarks	After switching to a new packet driver, perform a warm boot for changes to take effect.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_SwitchPacketDriver)(USHORT); #else UINT SwitchPacketDriver(USHORT); #endif</pre>

Deprecated Functions

The following functions are deprecated. While these are not removed from the API, these are no longer supported. Their parameters are no longer applicable and the return value for all of these functions is: `ERR_FUNCTION_DEPRECATED`.

Function	Syntax
GetRTSThreshold(Deprecated)	UINT GetRTSThreshold (USHORT &);
GetMedia(Deprecated)	UINT GetMedia (ULONG &);
GetMedium(Deprecated)	UINT GetMedium (ULONG &);
GetNicStats(Deprecated)	UINT GetNicStats (NDIS_802_11_STATISTICS &);
SetRTSThreshold(Deprecated)	UINT SetRTSThreshold (USHORT &);
SetTXRate(Deprecated)	UINT SetTXRate (UCHAR);
EncryptWepKeyForRegistry(Deprecated)	UINT EncryptWepKeyForRegistry (TCHAR * szDest, TCHAR * szSource);
SetDiversity(Deprecated)	UINT SetDiversity (USHORT);

Notifications

Use the following information to programmatically control the vibrator, to write an application to turn on the vibrator when a message is received via the WLAN radio link, and turn it off when the user presses a key.

Vibrator support is in the NLED driver as a false LED. The vibrator is LED 5 and identified with an CycleAdjust of -1. The vibrate option is available in the notifications panel when the vibrator is in the system.

Regarding an applications interface to Nled.dll, LEDs must be available for applications to use through the CoreDll.dll file. To use the LED functions, declare these as extern "C" as follows:

```
extern "C" BOOL WINAPI NLEDGetDeviceInfo(UINT nInfoId, void *pOutput);
extern "C" BOOL WINAPI NLEDSetDevice( UINT nDeviceId, void *pInput);
```

The LEDs are enumerated for access through the data structures associated with these APIs:

Notification LED	0
Radio On LED	1
Alpha Lock LED	2
Scanner LED	3
Low Battery	4
Vibrator	5

NLEDGetDeviceInfo

Usage

```
#include "nled.h"
```

Syntax

```
BOOL NLEDGetDeviceInfo ( UINT nInfoId, void *pOutput );
```

Parameters

nInfoId	Integer specifying the information to return. These values are defined:								
NLED_COUNT_INFO	<i>pOutput</i> buffer specifies the number of LEDs on the device.								
NLED_SUPPORTS_INFO_ID	<i>pOutput</i> buffer specifies information about the capabilities supported by the LED.								
NLED_SETTINGS_INFO_ID	<i>pOutput</i> buffer has information about LED current settings.								
pOutput	Pointer to the buffer to which the information is returned. The buffer points to various structure types defined in "nled.h", depending on the value of <i>nId</i> , as detailed in the following table:								
	<table border="1"> <thead> <tr> <th>Value of <i>nID</i></th> <th>Structure in <i>pOutput</i></th> </tr> </thead> <tbody> <tr> <td>LED_COUNT_INFO</td> <td>NLED_COUNT_INFO</td> </tr> <tr> <td>NLED_SUPPORTS_INFO</td> <td>NLED_SUPPORTS_INFO</td> </tr> <tr> <td>NLED_SETTINGS_INFO</td> <td>NLED_SETTINGS_INFO</td> </tr> </tbody> </table>	Value of <i>nID</i>	Structure in <i>pOutput</i>	LED_COUNT_INFO	NLED_COUNT_INFO	NLED_SUPPORTS_INFO	NLED_SUPPORTS_INFO	NLED_SETTINGS_INFO	NLED_SETTINGS_INFO
Value of <i>nID</i>	Structure in <i>pOutput</i>								
LED_COUNT_INFO	NLED_COUNT_INFO								
NLED_SUPPORTS_INFO	NLED_SUPPORTS_INFO								
NLED_SETTINGS_INFO	NLED_SETTINGS_INFO								

NLEDSetDevice

Usage

```
#include "nled.h"
```

Syntax

```
BOOL NLEDSetDevice ( UINT nDeviceId, void *pInput );
```

Parameters

nDeviceId	Integer specifying the device identification. The following is defined:
NLED_SETTINGS_INFO_ID	Contains information about the desired LED settings.
pInput	Pointer to the buffer that contains the NLED_SETTINGS_INFO structure.

Reboot Functions

There are several methods, via Kernel I/O Control functions, that an application program can use to force the 751G to reboot.

You can also see the Device Resource Kit in the IDL for information about these functions. The IDL is available as a download from the Intermec web site at www.intermec.com/idl. Contact your Intermec representative for information.

- **IOCTL_HAL_REBOOT**
This performs a warm-boot ([page 76](#)). *See note on the next page.*

- **IOCTL_HAL_COLDBOOT**
Forces a cold reboot. This resets the 751G, reloads Windows CE as if a power-up was performed, and discards the contents of the Windows CE RAM-based object store (page 74).
- **IOCTL_HAL_WARMBOOT**
Performs a system warm-boot, preserving the object store (page 73).



Note: Using IOCTL_HAL_REBOOT is no longer recommended, use either IOCTL_HAL_WARMBOOT or IOCTL_HAL_COLDBOOT. IOCTL_HAL_REBOOT is still supported for backward-compatibility but its use can lead to difficulties.

Reprogramming the 751G Keypad



Note: Use caution when remapping the keypad. Improper remapping may render the keypad unusable. Data within the 751G could also be lost, should any problems occur.

Applications have the ability to remap keys on the 751G keypad. This allows applications to enable keys that would otherwise not be available, such as the [F1] function key. Also, to disable keys that should not be available, such as the alpha key because no alpha entry is required. Use care when attempting to remap the keypad because improper remapping may render the keypad unusable. Cold-booting the device loads the default keymap, thus correcting this situation.

Note that remapping the keys in this way affects the key mapping for the entire system, not just for the application that does the remapping.

There are three “planes” supported for the 751G keypad. Keys used in more than one shift plane must be described in each plane.

- The unshifted plane contains values from the keypad when not pressed with other keys, such as pressing [1] to enter a “1.”
- The orange plane contains values from the keypad when a key is simultaneously pressed with the [orange] key, such as pressing [orange] [1] to enter a “Send” command.
- The alpha plane contains values from the keypad when the keypad is placed in alpha mode by pressing the green [Alpha] key, such as pressing [Alpha] [1] to enter a “Caps” command.

Key Values

Key values for each plane are stored in the registry. All units ship with a default key mapping already loaded in the registry. Applications that wish to change the default mapping need to read the appropriate key from the registry into an array of Words, modify the values required and then write the updated values back into the registry. The registry access can be done with standard Microsoft API calls, such as RegOpenKeyEx(), RegQueryValueEx(), and RegSetValueEx().

For the 751G keypad, these registry keys contain the plane mappings:

- The unshifted plane mapping is in the registry at:
HKEY_LOCAL_MACHINE\HARDWARE\DEVICEMAP\KEYBD\Vkey
- The orange plane mapping is in the registry at:
HKEY_LOCAL_MACHINE\HARDWARE\DEVICEMAP\KEYBD\VkeyGold
- The alpha plane mapping is in the registry at:
HKEY_LOCAL_MACHINE\HARDWARE\DEVICEMAP\KEYBD\VkeyAlpha

How Key Values Are Stored in Registry

To know which fields to update in the registry, you must know what Scan Codes are assigned to each physical key (see the table starting on the next page). The Scan Code is used at the lowest level of the system to let the keypad driver know which physical key was pressed. The keypad driver takes that scan code and looks it up in a table (a copy of the one stored in the registry) to determine which values to pass on to the operating system.

Each registry key is just an array that describes to the keypad driver what value needs to be passed for each physical key. The key values are indexed by the scan code, this is a zero-based index. For example in the unshifted plane, the [4] key has a scan code of 0x06. This means that the seventh word under the “Vkey” registry key has the value for the [4] key. Taking a sample of the “Vkey” registry key shows the following values:

```
00,00,0B,05,02,03,C1,07,04,03,BE,00,34,00,00,00,. . .
```

The value is 34,00. The values are in reverse byte order because that is the way the processor handles data. When writing an application, nothing needs to be done to swap the bytes, as this happens automatically when the data is read into a byte value. This is something to be aware of when looking at the registry. Knowing this, we can see that the value that the keypad driver passes to the system is a hex 34. Looking that up on an UNICODE character chart, we see that it maps to a “4”. If you wanted the key, labeled “4”, to output the letter “A” instead, you would need to change the seventh word to “41” (the hexadecimal representation of “A” from the UNICODE chart), then put the key back into the registry.

If you wish to disable a certain key, remap its scan code to 0x00.



Note: Do not remap scan codes 0x01, 0x41, 0x42, 0x43, 0x44. Remapping these could render the 751G unusable until a cold-boot is performed.

Change Notification

Changing registry keys alone does not immediately change key mappings. To notify the keypad driver the registry was updated, use the CreateEvent() API to signal the “ITC_KEYBOARD_CHANGE” named event.

Advanced Keypad Remapping

It is possible to map multiple key presses or named system events to a button - changing what buttons fire the scanner, controls the volume, and suspend/resume the device. These options can cut down the number of keys to press or remapping which key behaves like the action key.

Scan Codes

At the lowest driver level, the 751G keypad identifies keys as scan codes sent via the keypad microcontroller, and cannot be changed without modifying the keypad firmware.

Scan Codes

Press this Key	Meaning	ScanCode
	Reserved	0x00
[I/O]	I/O button	0x01
	Scanner Handle Trigger	0x02
	Scanner Left	0x03
	Scanner Right	0x04
[4]	4/GHI/A2	0x06
	None	0x07
[left arrow]	Left arrow/Back Tab	0x08
	None	0x09
[BkSp]	BkSp// (forward slash)	0x0A
[orange]	Orange plane	0x0B
	None	0x0C
[Esc]	Esc/- (minus sign)	0x0D
[down arrow]	Down arrow/Volume decrease	0x0E
[1]	1/Caps/Send	0x0F
[7]	7/PQRS/PgUp	0x10
[Alpha]	Alpha (green) plane	0x11
	None	0x12
[up arrow]	Up arrow/Volume increase	0x13
[right arrow]	Right arrow/Tab	0x14
[2]	2/ABC/End	0x15
[8]	8/TUV/* (asterisk)	0x16
[0]	0/Win	0x17
[5]	5/JKL/A3	0x18
	None	0x19
[Action]	Action/+ (plus symbol)	0x1A
[3]	3/DEF/backlight	0x1B
[9]	9/WXYZ/PgDn	0x1C
[Enter]	Enter/@ (at symbol)	0x1D
[6]	6/MNO/A4	0x1E

Scan Codes (continued)

Press this Key	Meaning	ScanCode
	None	0x1F–0x40
Battery LED	Charge Detect	0x41
Left LED	LCD frontlight	0x42
LED above [Esc]	Ambient light	0x42
	Threshold crossed	0x42
	Headset detected	0x43
	Keypad Backlight	0x44
LED above [Esc]	Ambient Light	0x44
	Threshold Crossed	0x44

Sample View of Registry Keys

See the registry on your device for the latest 751G key mappings.

```
[HKEY_LOCAL_MACHINE\HARDWARE\DEVICEMAP\KEYBD]
"ResumeMask"=dword:7
"Vkey"=hex:00,00,0B,05,02,03,C1,07,04,03,BE,00,34,00,00,00,\
25,00,00,00,08,00,03,02,00,00,1B,00,28,00,31,00,\
37,00,01,02,00,00,26,00,27,00,32,00,38,00,30,00,\
35,00,00,00,01,03,33,00,39,00,0D,00,36,00,00,00,\
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
00,00,07,05,01,05,03,05,02,05
"VkeyGold"=hex:00,00,0B,05,02,03,C1,07,04,03,BE,00,34,00,00,00,\
09,01,00,00,BF,00,03,02,00,00,BD,00,75,00,72,00,\
21,00,01,02,00,00,76,00,09,00,73,00,38,01,5B,00,\
35,00,00,00,BB,01,09,05,22,00,32,01,36,00,00,00,\
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
00,00,07,05,01,05,03,05,02,05
"VkeyAlpha"=hex:00,00,0B,05,02,03,C1,07,04,03,BE,00,47,00,00,00,\
25,00,00,00,08,00,03,02,00,00,1B,00,28,00,02,02,\
50,00,01,02,00,00,26,00,27,00,41,00,54,00,20,00,\
4A,00,00,00,01,03,44,00,57,00,0D,00,4D,00,00,00,\
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
00,00,07,05,01,05,03,05,02,05
```



4 Maintaining the Computer

Use this chapter to update your system software, solve problems you may encounter and perform routine maintenance.

Updating the System Software

When you upgrade your 751G, you are updating the operating system (OS) and the Persistent Storage Manager (PSM) files.

The PSM files are stored in the Flash File Store region, and deliver Intermec value-added functionality such as data collection, unit configuration and diagnostics, the Intermec wireless security suite, and the SmartSystems™ Foundation.

You need to download the latest upgrade files from the Intermec web site to your desktop PC. Contact your Intermec representative for web browser information, then do the following:

- 1 Start your web browser and go to the Intermec web site at the location given by your representative.
- 2 Click the applicable link, fill out the appropriate information, then click **Submit Information**. Click **OK** to continue.
- 3 Click the link and download the .zip file to your PC.
- 4 Follow the steps in one of the next sections:
 - If using a storage card to upgrade the 751G, see below.
 - If using the SmartSystems Console to upgrade the 751G, see [“Using the SmartSystems Console to Upgrade the Computer” on page 105](#).

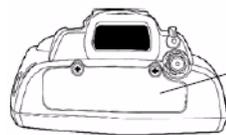
Using a Storage Card to Upgrade the Computer

To use a Secure Digital card to upgrade the 751G, you need a Secure Digital card reader and a Secure Digital card formatted as FAT.



Note: The 751G currently supports Delkin Secure Digital cards only. Intermec cannot guarantee that other Secure Digital cards will work with the 751G.

- 1 Locate the storage card access door at the top of the 751G, remove its two screws, remove the door, then remove the storage card. See the *Model 751G Mobile Computer Quick Start Guide* (P/N 962-054-093) for more information.

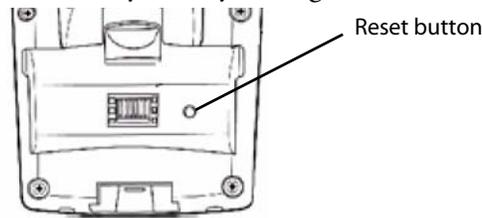


Storage Media Access Door

This shows the top of the 751G. Note the keypad is to the bottom.

- 2 Place the storage card in your desktop PC card reader, then copy all required upgrade files to the storage card.
- 3 Remove the storage card from your card reader, reinstall it in the 751G.
- 4 Press the **Reset** button inside the battery compartment to perform a cold-boot. Do not use force or a sharp object when pressing the **Reset**

button, or you may damage the **Reset** button.



This shows the back of the 751G inside the battery compartment.

- 5 Return the 751G to DC power, such as installing it into a dock connected to external power.
- 6 When the Bootloader Menu shows “complete,” remove the storage card, then press the **Reset** button again to load the new operating system.



Note: The upgrade will fail if the 751G is not connected to external power. For help, see “[Accessories](#)” on page 18.

- 7 When the 751G finishes booting, insert the battery, then close the battery door.

You may use the 751G. You have reset the 751G to its default configuration. You need to set the date and time and to set its network communications parameters to reestablish communications with the other devices in the wireless network.

Using the SmartSystems Console to Upgrade the Computer

You can use the SmartSystems Console to upgrade the operating system on your 751G. The console is part of SmartSystems Foundation and is available from the Intermec web site via the Intermec Developer Library (IDL). Before you can upgrade your computer, you need:

- the SmartSystems Foundation. To download SmartSystems Foundation, go to www.intermec.com/idl and open the Device Management page.
 - the device upgrade .exe file. This file is available from the Intermec web site at www.intermec.com. Go to **Service & Support > Downloads**. Make sure the file you select is for your language.
- 1 Install SmartSystems Foundation on your PC and open the SmartSystems Console.
 - 2 Ensure the SmartSystems Console and 751Gs are on the same subnet.
 - 3 Make sure your 751Gs are either in a communications dock or charging dock, such as the AD14, or that power management is disabled.
 - 4 Download the device upgrade .exe file to your desktop PC.

- 5 Double-click the .exe file on your desktop PC. An InstallShield application starts and walks you through the process of extracting the upgrade files in the default location.



Note: Do not change the default location where InstallShield extracts the files. The SmartSystems Console requires that the files be in this location.

- 6 From the SmartSystems Console, locate the device upgrade to install.
- 7 Drag-and-drop the device upgrade onto each 751G you want to upgrade. The SmartSystems Console will tell you that it is installing the upgrade on your 751G.

Once the upgrade is done downloading to your 751G, your 751G replaces the operating system and then automatically performs a cold-boot. Progress messages do appear on the 751G screen.



The SmartSystems Console will show your 751G as offline (*note the red stop sign*) until the 751G reboots and reconnects to the system.

Troubleshooting Your Computer

Before sending the 751G in for service, save its data and configuration. Intermec is responsible only for the keypad and hardware features to match the original configuration when doing repairs or replacements.

Problems While Operating the Computer

Problem	Solution
You press the power switch to turn on the 751G and nothing happens.	Make sure the backlight is on. Make sure you have a charged 751G Battery installed correctly. For help, see “Installing and Charging the Battery” on page 5 . The battery may be discharged. Replace the battery with a spare charged battery, or charge the battery. Perform a warm-boot or press the reset button in the battery cavity.
The Battery status LED is on.	If the battery status LED is a steady green, the battery is more than 95% charged and unit is on a charger. If the battery status LED is blinking red, then the battery is low. If the battery status LED is a steady red, the main battery is on charge.
The 751G appears to be locked up and you cannot enter data.	Press the power switch to turn off the 751G, then press the power switch again to turn on the 751G. Press and hold the power switch ten seconds to warm-boot the 751G. Try reloading the firmware. See “Updating the System Software” on page 104 . If the 751G does not boot or reset, contact your Intermec representative for help.

Problems While Configuring the Computer

Problem	Solution
You scan a configuration command, such as Beeper Volume, and you hear three low beeps.	If you are working in the Intermec Settings applet, you cannot scan configuration commands. Exit the applet to scan configuration commands.
You scan or enter an option for the Scanner Model configuration command and you hear three low beeps.	You may have scanned or entered a Scanner Model command that does not apply to the type of scanner that you have installed. Try scanning or entering the Scanner Model command again and select an option for the type of device you are using.
You cannot type a character on the keypad or you can only type uppercase or lowercase letters.	You may have locked a modifier key on the keypad. Check the 751G toolbar to see if it contains an icon with a locked symbol. Press the necessary key sequence to unlock the key. See “Keypad” on page 10 .

Problems with Wireless Connectivity

Problem	Solution
When you turn on the 751G after it was suspended for a while (10-15 minutes or longer), it can no longer send or receive messages over the network.	Host may have deactivated or lost current terminal emulation session. In a TCP/IP direct connect network, turn off the “KeepAlive” message from host to maintain the TCP session while a 751G is suspended.
The No Network Connection icon appears on the toolbar. The 751G is not communicating with the access point.	The 751G is not connected to access point. Ensure access point is turned on and operating. Move closer to access point to reestablish communications. Ensure the 751G is configured correctly for network. 751G radio parameters must match all access point values. If you have an 802.11b/g radio and its radio initialization process failed, reset the 751G (see “Resetting Your Computer” on page 13). If No Network Connection icon still appears, you may have a defective radio card. For help, contact your local Intermec representative.
The 751G is connected to the Intermec Application Server or host computer and you move to a new site to collect data. The Network Connection icon was visible, but is now replaced with the No Network Connection icon.	Move closer to an access point or to a different location to reestablish communications until the Network Connection icon appears. Any data you collected while out of range is transmitted over the network.
The Network Connection icon is in the toolbar, but you cannot establish a terminal emulation session with the host computer.	There may be a problem with the host computer, with the connection between the Intermec Application Server and the host computer, or with the connection between the access point and the host computer. Check with network administrator to make sure the host is running and allowing users to login to the system.
The Network Connection icon is in the toolbar, but the host computer is not receiving any data from the 751G.	In a UDP Plus network, there may be a problem with the connection between the Intermec Application Server and the host computer. Check with network administrator or see the user’s manual for the Intermec Application Server. In a TCP/IP network, there may be a problem with the connection between the access point and the host computer. Check with network administrator or use your access point user’s manual.

Problems While Configuring 802.1x Security

If you have trouble configuring the computer for 802.1x security, check these problems and possible solutions.

Problem	Solution
The 751G indicates that it is authenticated, but it does not communicate with the host.	Make sure the 751G IP address, host IP address, subnet mask, default router are configured for network.
The 751G does not appear to be authenticating and a network connection icon does not appear on the toolbar.	The 751G may not be communicating with access point. Make sure the 751G network name matches access point network name (SSID). 802.1x security network may not be active. Ensure the server software is properly loaded and configured on server PC. See server software documentation for help.
A network connection icon appears in the toolbar, but then disappears.	The 751G may not be communicating with the intended access point. Make sure the 751G network name matches the access point network name. Default network name is "INTERMEC." Access point may not be communicating with server. Ensure the access point is turned on, properly configured, and has 802.1x security enabled.
The 751G indicates it is not authenticated.	User Name and Password parameters on the 751G must match the user name and password on authentication server. You may need to reenter the password on both the 751G, authentication server. On your authentication server, the user and group are allowed and the group policy is allowed to log into the server. For help, see the documentation that shipped with your authentication server software. IP address and secret key for access point must match the IP address and secret key on authentication server. You may need to reenter the IP address and secret key on both your access point and authentication server. Authentication server software is running on server PC
You are setting up multiple access points in a network, with different SSIDs, and the connection fails.	The 751G does not save WEP key values when changing the SSID. Reenter the WEP key value after changing the SSID, select Apply Network Settings from the 802.11 Radio menu. You should now be able to connect to the different access points.
You receive a message saying "The server certificate has expired or your system date is incorrect" after you perform a clean-boot on the 751G.	Date and time are not saved when a clean-boot is performed. Reenter the date and time, then select Apply Network Settings from the 802.11 Radio menu.

Problems While Scanning Bar Codes

Problem	Solution
You cannot see a red beam of light from the scanner when you press the Scan button and aim the scanner at a bar code label.	You may be too far away from the bar code label. Try moving closer to the bar code label and scan it again. For information, see " Scanning Bar Codes " on page 13. You may be scanning the bar code label "straight on." Change the scanning angle and try again. Move within two feet of a wall to test the effective scan of the scanner.
When you release the Scan button or handle trigger, the Good Read light does not turn off.	The Good Read light will remain on if you configure the 751G to use continuous/edge triggering. If you configure the 751G for level triggering and the Good Read light remains on, there may be a problem. Press the Scan button or pull the trigger again without scanning a bar code label. If the light is still on, contact your local Intermec representative.

Problems While Scanning Bar Codes (continued)

Problem	Solution
The input device attached to the 751G does not work well or read bar code labels very quickly.	Set the Scanner Model command to the specific attached input device. Check enabled bar code symbologies and enable only the symbologies being used.
The scanner will not read the bar code label.	Aim the scanner beam to cross entire bar code label in one pass. Vary the scanning angle. Check the quality of the bar code label, Scan a bar code label that you know will scan. Compare the two bar code labels to see if the bar code quality is too low. You may need to replace the label that you cannot scan. Ensure the bar code symbology is enabled. Use the Intermec Settings applet to check the symbologies. Expand Data Collection > Symbologies beneath devices listed (scanner, virtual wedge) to check and enable symbologies, then scan the bar code label again. Ensure the 751G application is expecting input from a bar code. You may need to type this information instead.
The scanner does not read the bar code labels quickly, or the scanning beam seems to be faint or obscured.	The scanner window may be dirty. Clean the window with a solution of ammonia and water. Wipe dry. Do not allow abrasive material to touch the window.
You scan a valid bar code label to enter data for your application. The data decoded by the scan module does not match the data encoded in the bar code label.	The 751G may have decoded the bar code label in a symbology other than the label's actual symbology. Try scanning the bar code label again. Make sure you scan the entire label.

Cleaning the Scanner

To keep the 751G in good working order, you may need to clean the EA11 scanner window.

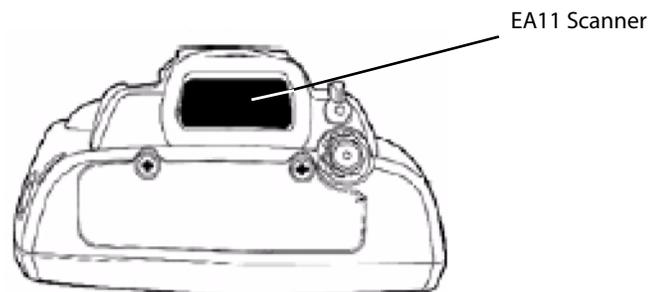
Clean the scanner window as often as needed for the environment in which you are using the 751G. To clean the 751G, use a solution of ammonia and water.



Caution

There are no user-serviceable parts inside the 751G. Opening the unit will void the warranty and may cause damage to the internal components.

Press the power switch to turn off the 751G. Dip a clean cloth towel in the ammonia solution and wring out the excess. Wipe off the scanner window, wipe dry. Do not allow abrasive material to touch this surface.



751G with EA11 Scanner. Note that this is the top of the unit.

5 Network Support

The 751G Color Mobile Computer automatically installs the appropriate software for radio use when the unit is powered on. It provides wireless connectivity via the Wireless Local Area Network (WLAN) using an 802.11b/g radio option that provides up to 11 Mb/sec throughput.



Note: Desktop icons and applet icons are shown to the left. Any place that **Start** is mentioned, tap the following Windows icon in the bottom, left corner of your desktop.



802.11b/g Communications

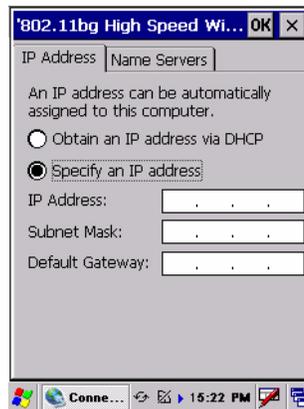


Note: The Network Selection APIs change the network adapter configuration programmatically. See “Networking APIs” on page 78 for the APIs.

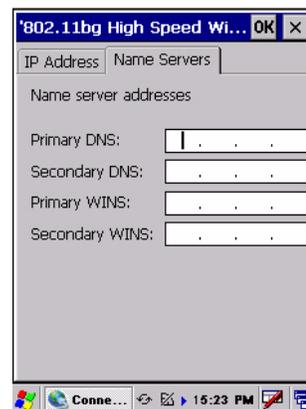
To configure the radio



- 1 Tap **Start > Settings > Network and Dial-up Connections**.
- 2 Double-tap the applicable radio connection icon to access and configure its properties.



IP Address



Name Servers

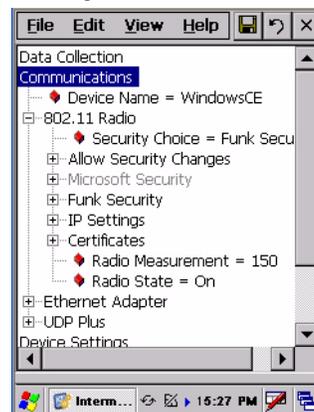
These two screens are for the Samsung radio using the SWLD26C1 connection icon shown on the left. Should your 751G have the Wistron radio installed, you can expect to see a BCMCF1 connection icon.

To configure the 802.11b/g network adapter power through Intermec Settings



Intermec Settings

- 1 Select **Start > Settings > Control Panel**, then double-tap the **Intermec Settings** icon.
- 2 Tap (+) to expand **Communications > 802.11 Radio**.
- 3 Configure your radio settings, then tap **File > Save**.



Remote Access (Modems)

You can set up connections to the Internet and corporate network at work to browse the Internet or intranet, send and receive e-mail, and synchronize information using ActiveSync. Connections are made via wireless networks.

Your 751G has two groups of connection settings: My ISP and My Work Network. Use My ISP settings to connect to the Internet. Use My Work Network settings to connect to any private network.

- **My ISP:** Once connected, you can send and receive e-mail messages by using Messaging and view web pages by using Internet Explorer Mobile. The communication software for creating an ISP connection is already installed on your 751G. Your service provider provides the software needed to install other services, such as paging and fax services. If this is the method you want to use, see the following paragraph.
- **My Work Network:** Connect to the network at your company or organization where you work. Once connected, you can send and receive e-mail messages by using Messaging, view web pages by using Internet Explorer Mobile, and synchronize with your desktop. If this is the method you want to use, see [“Connecting to Work” on page 116](#).

Connecting to an Internet Service Provider

You can connect to your ISP, and use the connection to view web or WAP pages. Get an ISP dial-up access telephone number, a user name, and a password from your ISP. Some ISPs require information in front of the user name, such as MSN/username.

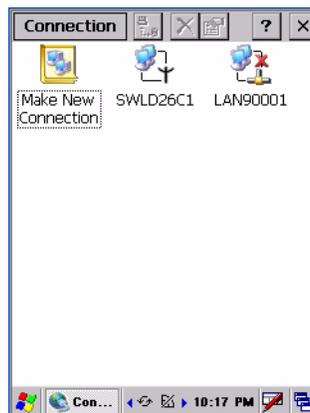


To view additional information for any screen in the wizard or while changing settings, tap the **Help** icon in the upper-right corner.



Make New Connection

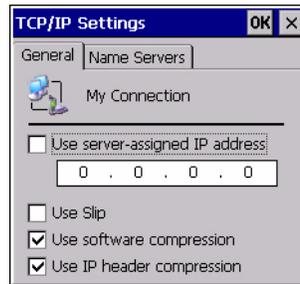
- 1 Tap **Start > Settings > Network and Dial-up Connections**, then double-tap the **Make New Connection** icon.



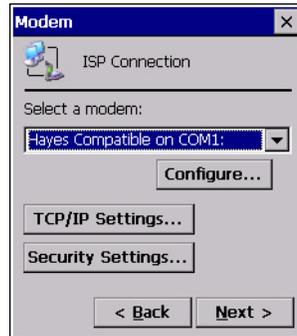
- 2 Enter a name for the connection, such as “ISP Connection,” tap **Next**.



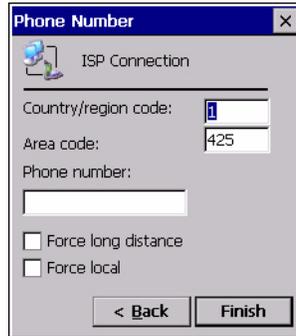
- 3 You should not need to change any advanced settings. Most ISPs now use a dynamically-assigned address. If the ISP you are connecting to does not use a dynamically-assigned address, tap **TCP/IP Settings**, clear (uncheck) **User server-assigned IP address**, then enter the IP address. Tap **OK** to return to the Modem page.



- 4 Select “Hayes Compatible on COM1” from the **Select a modem** drop-down, then tap **Next** to continue.



- 5 Enter the access phone number, then tap **Finish** to return to the Connection page.



- 6 Double-tap the new connection, then enter the user name, password, and domain (if provided by an ISP or your network administrator).



- 7 Tap **Dial Properties**, then specify your current location from the drop-down list.

Specify your current phone type. If your phone type is pulse dialing, tap **Pulse dialing**. If your type is tone dialing (as most phone lines are), then tap **Tone dialing**. These selections apply to all connections.

Tap **OK** to return to the Settings page.



- 8 To start the connection, use the Internet Explorer to visit web and WAP pages. For more information, see [“Internet Explorer” on page 33](#).

Connecting to Work

If you have access to a network at work, you can view intranet pages, synchronize your 751G, and possibly access the Internet. Your network administrator may also give you Virtual Private Network (VPN) settings.

A VPN connection helps you to securely connect to servers, such as a corporate network, via the Internet. Ask your network administrator for a user name, password, domain name, TCP/IP settings, and host name or IP address of the VPN server.

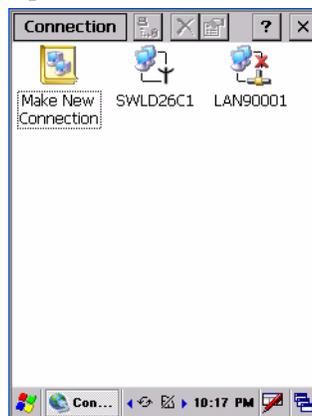


To view additional information for any screen in the wizard or while changing settings, tap the **Help** icon in the upper-right corner.



Make New Connection

- 1 Tap **Start > Settings > Network and Dial-up Connections**, then double-tap the **Make New Connection** icon.



- 2 Enter a name for the connection, such as “VPN Connection.” Select **Virtual Private Network (PPTP)**, then tap **Next** to continue.

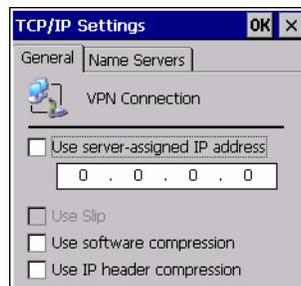


- 3 Enter the **Host name or IP address** field.



- 4 You should not need to change any TCP/IP settings. Instances where you are to change TCP/IP settings include the following. To change these settings, tap **TCP/IP Settings** on the Connection page. Otherwise, tap **Finish**.

- The server to which you are connecting does not use dynamically-assigned addresses, and you need to enter your TCP/IP settings.
- You need to change server DNS or WINS settings.



- 5 Insert the necessary equipment into the device, and double-tap the new **VPN Connection** icon to connect to the host.
- 6 Use a desired program to automatically begin connecting. For example, switch to the Internet Explorer and browse to a web or WAP page.

Ending a Connection

To disconnect, do one of the following:

- When connected via cable or cradle, detach your 751G.
- When connected via a wireless network, switch off the connection.

Configuring Security

Use the next sections to understand how to configure each type of security on your wireless 751G.

Loading Certificates



If you choose to use Transport Layer Security (TLS) with WPA or 802.1x security, you need to have a unique client certificate on the 751G and a trusted root certificate authority (CA) certificate. If you choose to use PEAP, you need to load a root CA certificate. You can use a third-party CA to issue unique client certificates and a root certificate.

If your CA is on your WLAN, select **Start > Settings > Control Panel**, double-tap the **Certificates** icon, then tap **View** to see certificate details. To remove a certificate, press and hold a certificate, then select **Delete**.



Wireless Networks

Your wireless adapter (network interface card) connects to wireless networks of two types: infrastructure networks and ad-hoc networks.

- Infrastructure networks get you in your corporate network and Internet. Using the 802.11b/g infrastructure mode, the 751G establishes a wireless connection to an access point, linking you to the rest of the network.
- Ad-hoc networks are private networks shared between two or more clients, even with no access point.

Each wireless network is assigned a name (or Service Set Identifier - SSID) to allow multiple networks to exist in the same area without infringement.

Intermec recommends using security measures with wireless networks to prevent unauthorized access to your network and to ensure your privacy of transmitted data. Authentication (cryptographically protected) by both the network and the user, transmitted data, and encryption are required elements for secure networks. Schemes are available to implement the features.

Encryption

AES (Advanced Encryption Standard)	A block cipher, a type of symmetric key cipher that uses groups of bits of a fixed length - called blocks. A symmetric key cipher is a cipher using the same key for both encryption and decryption. As implemented for wireless, this is also known as CCMP, which implements AES as TKIP and WEP are implementations of RC4.
CKIP (Cisco Key Integrity Protocol)	This is Cisco's version of the TKIP protocol, compatible with Cisco Aironet products.
TKIP (Temporal Key Integrity Protocol)	This protocol is part of the IEEE 802.11i encryption standard for wireless LANs, which provides per-packet key mixing, a message integrity check and a re-keying mechanism, thus overcoming most of the weak points of WEP. This encryption is more difficult to crack than the standard WEP. Weak points of WEP include: No Initiation Vector (IV) reuse protection, weak keys, no protection against message replay, no detection of message tampering, and no key updates.
WEP (Wired Equivalent Privacy) encryption	With preconfigured WEP, both the client 751G and access point are assigned the same key, which can encrypt all data between the two devices. WEP keys also authenticate the 751G to the access point - unless the 751G can prove it knows the WEP key, it is not allowed onto the network. WEP keys are only needed if they are expected by your clients. There are two types available: 64-bit (5-character strings, 12345) (default) and 128-bit (13-character strings, 1234567890123). Enter these as either ASCII (12345) or Hex (0x3132333435).

Key Management Protocols

WPA (Wi-Fi Protected Access)	This is an enhanced version of WEP that does not rely on a static, shared key. It encompasses a number of security enhancements over WEP, including improved data encryption via TKIP and 802.11b/g authentication with EAP. WiFi Alliance security standard is designed to work with existing 802.11 products and to offer forward compatibility with 802.11i.
WPA2 (Wi-Fi Protected Access)	Second generation of WPA security. Like WPA, WPA2 provides enterprise and home Wi-Fi users with a high level of assurance that their data remains protected and that only authorized users can access their wireless networks. WPA2 is based on the final IEEE 802.11i amendment to the 802.11 standard ratified in June 2004. WPA2 uses the Advanced Encryption Standard (AES) for data encryption and is eligible for FIPS (Federal Information Processing Standards) 140-2 compliance.

Authentication

EAP (Extensible Authentication Protocol)	802.11b/g uses this protocol to perform authentication. This is not necessarily an authentication mechanism, but is a common framework for transporting actual authentication protocols. Intermec provides a number of EAP protocols for you to choose the best for your network.
EAP-FAST (Flexible Authentication via Secure Tunneling)	A publicly accessible IEEE 802.1X EAP type developed by Cisco Systems. It is available as an IETF informational draft. An 802.1X EAP type that does not require digital certificates, supports a variety of user and password database types, supports password expiration and change, and is flexible, easy to deploy, and easy to manage.
LEAP (Lightweight Extensible Authentication Protocol)	Also known as Cisco-Wireless EAP, provides username/password based authentication between a wireless client and a RADIUS server. In the 802.1x framework, traffic cannot pass through a wireless network access point until it successfully authenticates itself.
EAP-PEAP (Protected Extensible Authentication Protocol)	Performs secure authentication against Windows domains and directory services. It is comparable to EAP-TTLS both in its method of operation and its security, though not as flexible. This does not support the range of inside-the-tunnel authentication methods supported by EAP-TTLS. Microsoft and Cisco both support this protocol.

Authentication (continued)

EAP-TLS (Transport Layer Security)	Based on the TLS (Transport Layer Security) protocol widely used to secure web sites. This requires both the user and authentication server have certificates for mutual authentication. While cryptically strong, this requires corporations that deploy this to maintain a certificate infrastructure for all users.
EAP-TTLS (Tunneled Transport Layer Security)	This protocol provides authentication like EAP-TLS (see page 120) but does not require certificates for every user. Instead, authentication servers are issued certificates. User authentication is done using a password or other credentials that are transported in a securely encrypted “tunnel” established using server certificates. EAP-TTLS works by creating a secure, encrypted tunnel through which you present your credentials to the authentication server. Thus, inside EAP-TTLS there is another <i>inner authentication protocol</i> that you must configure via Additional TTLS Settings.

The 751G provides three types of security for your wireless network: Wi-Fi Protected Access 2 (WPA2/802.11i), WPA, and WEP. 802.1x should be referred to as an authentication method used for WPA and WPA2. Another authentication method for WPA and WPA2 would be the Pre-Shared Key (PSK).

Choosing Between Microsoft and Funk Security

Before you can implement a security solution on the 751G, you need to choose between Microsoft and Funk security:

- By default, Funk security is enabled. It provides everything you get with Microsoft security plus the addition of Cisco Compatible Extensions features. It also provides additional authentication types like EAP-TTLS, LEAP, and EAP-FAST.
- Microsoft security, with its Microsoft Zero Config feature, is also available. To switch to Microsoft security, go to “[Configuring Microsoft Security](#)” on [page 126](#) to start.



Note: Your security choice does not depend on your authentication server. For example, you can choose Funk security if you use Microsoft Active Directory® to issue certificates.

Configuring Funk Security

You can define up to four profiles for your Funk Odyssey security. Different profiles let your 751G communicate in different networks without having to change all of your security settings. For example, you can set up one profile for the manufacturing floor and one for the warehouse.



Intermec
Settings

- 1 Select **Start > Settings > Control Panel**, then double-tap the **Intermec Settings** icon.
- 2 Tap (+) to expand **Communications > 802.11 Radio > Funk Security**.

- 3 Select an active profile, then configure its security settings.



Using WPA Security

Wi-Fi Protected Access (WPA) is a strongly enhanced, interoperable Wi-Fi security that addresses many of the vulnerabilities of Wired Equivalent Privacy (WEP). Instead of WEP, WPA uses Temporal Key Integrity Protocol (TKIP) for its data encryption method. Currently, WPA satisfies IEEE 802.11i standards.

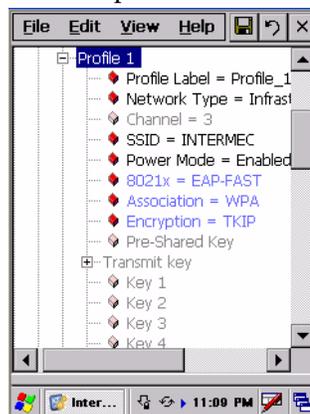
WPA runs in Enterprise (802.1x) mode or PSK mode:

- In Enterprise mode, WPA provides user authentication using 802.1x and the Extensible Authentication Protocol (EAP). That is, an authentication server (such as a RADIUS server) must authenticate each device before the device can communicate with the wireless network.
- In PSK mode, WPA provides user authentication using a shared key between the authenticator and the 751G. WPA-PSK is a good solution for small offices or home offices that do not want to use an authentication server.

To use WPA security, you need an access point with an 802.11b/g radio that supports WPA.

Configuring WPA Security With Funk Security

Use this procedure to set WPA security with Funk security.



- 1 Make sure you have configured the communications and radio parameters on your 751G and that Funk is your security choice.
- 2 Open Intermec Settings. Tap (+) to expand **Communications > 802.11 Radio > Funk Security > Profile X** with “X” being “1” through “4.”
- 3 For **Association**, select “WPA” and press **Enter**.
- 4 For 8021x, select “PEAP,” “TLS,” “TTLS,” “LEAP,” or “EAP-FAST” and press **Enter**.

If you select “TTLS” or “PEAP:”

- a Select **User Name**, type your user name, then press **Enter**.
- b Select **User Password**, type a user password, then press **Enter**.
- c For **Validate Server Certificate**, select “Yes,” then press **Enter**. *Note that you must have the date on the 751G set correctly when you enable Validate Server Certificate.*
- d You must enter a **User Name** and **Subject Name**. You can also enter a **Server 1 Common name** or **Server 2 Common name** if you want to increase your level of security.

If you select “TLS:”

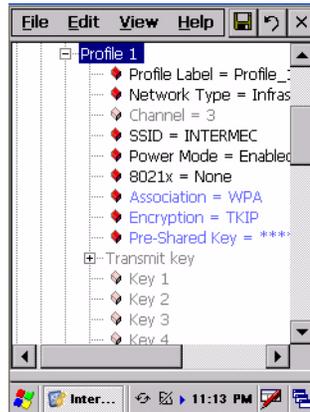
- a Load a user and root certificate on your 751G. For help, see [“Loading Certificates” on page 118](#).
- b For **Validate Server Certificate**, select “Yes,” then press **Enter**. *Note that you must have the date on the 751G set correctly when you enable Validate Server Certificate.*
- c You must enter a **User Name** and **Subject Name**. You can also enter a **Server 1 Common name** or **Server 2 Common name** if you want to increase your level of security.

If you select “LEAP” or “EAP-FAST:”

- a Select **User Name**, type your user name, then press **Enter**.
- b Select **User Password**, type a user password, then press **Enter**.

Configuring WPA-PSK Security With Funk Security

Use this procedure to set WPA-PSK security on your 751G with Funk security.



- 1 Make sure you have configured the communications and radio parameters on your 751G and that Funk is your security choice.
- 2 Open InterMec Settings. Tap (+) to expand **Communications > 802.11 Radio > Funk Security > Profile X** with “X” being “1” through “4.”
- 3 For **Association**, select “WPA” and press **Enter**.
- 4 For **802.1x**, select “None” and press **Enter**.
- 5 For **Pre-Shared Key**, enter the pre-shared key or the passphrase.

The pre-shared key must be a value of 32 hex pairs preceded by 0x for a total of 66 characters. The value must match the key value on the access point. The passphrase must be from 8 to 63 characters. After you enter a passphrase, the 751G internally converts it to a pre-shared key. This value must match the passphrase on the authenticator.

- 6 Exit the InterMec Settings applet.

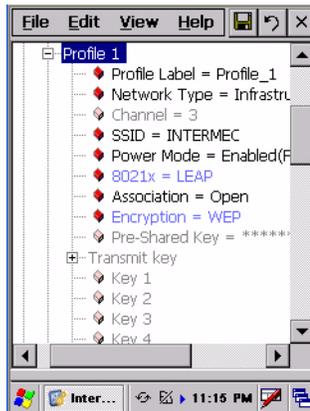
Using 802.1x Authentication

802.1x authentication provides centralized user authentication using an authentication server, authenticators (access points), and supplicants. These components communicate using an EAP authentication type, such as TLS (Transport Layer Security) or PEAP (Protected Extensible Authentication Protocol). 802.1x security provides data encryption using dynamic WEP key management. To use 802.1x security, you need:

- An access point with an 802.11b/g radio.
- A 751G with an 802.11b/g radio and the 802.1x/WPA security option.

Configuring 802.1x Security With Funk Security

This sets 802.1x security on your 751G with Funk security.



- 1 Make sure you have configured the communications and radio parameters on your 751G and that Funk is your security choice.
- 2 Open Intermec Settings. Tap (+) to expand **Communications > 802.11 Radio > Funk Security > Profile X** with “X” being “1” through “4.”
- 3 For **Association**, select “Open” and press **Enter**. When working with Cisco Aironet access points, you can select “Network-EAP.”
- 4 For **Encryption**, select “WEP” and press **Enter**.
- 5 For **8021x**, select “PEAP,” “TLS,” “TTLS,” “LEAP,” or “EAP-FAST” and press **Enter**.

If you select “TTLS” or “PEAP:”

- a Select **User Name**, type your user name, then press **Enter**.
- b Select **User Password**, type a user password, then press **Enter**.
- c For **Validate Server Certificate**, select “Yes,” then press **Enter**. *Note that you must have the date on the 751G set correctly when you enable Validate Server Certificate.*
- d Enter a **User Name** and **Subject Name**. You can also enter a **Server 1 Common name** or **Server 2 Common name** to increase security.

If you select “TLS:”

- a Load a user and root certificate on your 751G ([page 118](#)).
- b For **Validate Server Certificate**, select “Yes,” then press **Enter**. *Note that you must have the date on the 751G set correctly when you enable Validate Server Certificate.*
- c You must enter a **User Name** and **Subject Name**. You can also enter a **Server 1 Common name** or **Server 2 Common name** if you want to increase your level of security.

If you select “LEAP” or “EAP-FAST:”

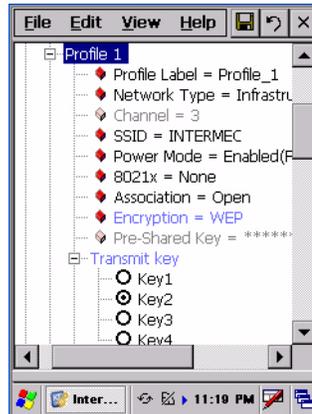
Select **User Name**, then type your user name. press **Enter**, select **User Password**, type a user password, then press **Enter**.

Using Static WEP Security

The 751G uses the Wired Equivalent Privacy (WEP) protocol to add security to your wireless network based on the 802.11b/g standard. To use WEP security, you need an access point with an 802.11b/g radio.

Configuring Static WEP Security With Funk Security

Use this procedure to set Static WEP security with Funk security.



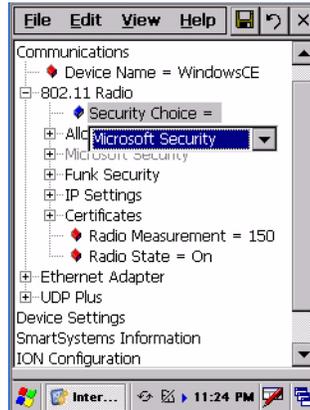
- 1 Make sure you have configured the communications and radio parameters on your 751G and that Funk is your security choice.
- 2 Open Intermec Settings. Tap (+) to expand **Communications** > **802.11 Radio** > **Funk Security** > **Profile X** with “X” being “1” through “4.”
- 3 For **Association**, select “Open” and press **Enter**.
- 4 For **Encryption**, select “WEP” and press **Enter**.
- 5 For **8021x**, select “None” and press **Enter**.
- 7 For **Transmit key**, select which WEP key to use for encryption of transmitted data.
- 8 Define a value for each key, up to four. Enter an ASCII key or a hex key either 5 or 13 bytes long based on the radio capability. Set a 5-byte value for 64-bit WEP or a 13-byte value for 128-bit WEP. Precede hex keys with 0x and make sure the keys use 5 or 13 hex pairs.

Configuring Microsoft Security

The default security setting is Funk. If you want to use Microsoft security, you need to select it as your security choice.



- 1 Select **Start > Settings > Control Panel**, then double-tap the **Intermec Settings** icon. Tap (+) to expand **Communications > 802.11 Radio > Security Choice**. Select “Microsoft Security” from the drop-down list, then press **Enter**.



- 2 Tap **Yes** or press **Esc** to clear the alert box, save your settings, then perform a clean-boot on the 751G.

SmartSystems™ Foundation

Use the SmartSystems Foundation (www.intermec.com/SmartSystems) to configure and manage your network. You can also contact your Intermec representative for support.

This tool, available as a free download from Intermec, includes a management console that provides a default method to configure and manage Intermec devices “out-of-the-box,” without the purchase of additional software licenses. This is for anyone who must configure and deploy multiple devices or manage multiple licenses.

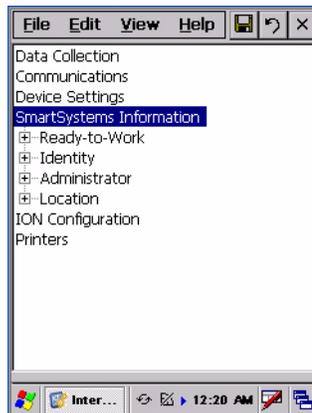
Use the Intermec Settings applet to do device configuration settings within the SmartSystems Foundation. Information about the Intermec Settings applet is in the *Intermec Computer Command Reference Manual* (P/N 073529) available online at www.intermec.com.

Information about the SmartSystems Foundation is available as an online help within the SmartSystems Console application. Select **SmartSystems > Help** in the console to access the manual.



Intermec
Settings

Tap **Start > Settings > the System tab > the Intermec Settings icon**, then tap to expand the **SmartSystems Information** option.





Symbols

__resetmeplease__.txt, 62

Numerics

1D area imager reading distances, 42

2D area imager reading distances, 42

2D Imager
about, 40

802.11b/g
API, 78
configuration profiles, 78
network adapter power, 112

802.11b/g information
radios, 112

802.1x authentication
Funk, 123

802.1x security
troubleshooting, 108

80211API.dll, 78

80211PM.dll, 78

A

Accessory list, 18

ActiveSync
installing applications, 43

ActiveSync See Microsoft ActiveSync

Adding programs
Microsoft ActiveSync, 28
to the Start menu, 29
via Microsoft ActiveSync, 29
via Windows Explorer, 29
Windows CE .NET, 27

AddWep(), 87

Adjusting settings
Windows CE .NET, 27

Advanced Encryption Standard, 119

AES (Advanced Encryption Standard), 119

Alpha plane on keypad, 99

Ambient lighting, 2

ANT_DIVERSITY
GetDiversity(), 81

ANT_PRIMARY
GetDiversity(), 81

ANT_SECONDARY
GetDiversity(), 81

APIs
802.11b/g, 78

Applets
backlight, 2
intemec settings
beeper volume, 9, 41, 127
intermec settings
802.11b/g, 112
funk security, 120

smartsystems, 9, 41, 127
system properties, 26
utilities, 7

ASCII
printing, 39
printing to a port
port print method, 39
raw text to printer, 39

AutoCab
command line syntax, 46

AutoUser.dat, 44

Avalanche, 45

B

Backlight applet
ambient light sensor, 2
Backlight control panel applet
keypad, 10

Bar codes
troubleshooting, 108

Basic connect/disconnect functions, 78

Battery
ambient lighting, 2
capacity, 19
low battery conditions, 4
specifications, 19
status, 4

Beeper
softening the volume, 9
volume
turning it on, 8

Browsing the Internet
Internet Explorer, 33

Build information
CE.NET, 14
PSM, 14

C

CAB files
after the extraction, 62
creating, 53
INF files, 53
with CAB Wizard, 65
installation functions
SETUP.DLL, 62

Cabinet Wizard
creating CAB files, 65
troubleshooting, 66
using the application, 53

CabWiz.ddf, 66

CabWiz.exe, 53, 66

Capacitor, internal super, 4

Card support
radios, 17

- Secure Digital cards, 16
 - pull tabs, 16
- Cisco Key Integrity Protocol, 119
- CKIP (Cisco Key Integrity Protocol), 119
- ClassID field values
 - VN_CLASS_ASIC, 69
 - VN_CLASS_BOOTSTRAP, 69
 - VN_CLASS_KBD, 69
- Cleaning the scanner window and screen, 109
- Cold boot
 - IOCTL_HAL_COLDBOOT, 74
- Cold boot, performing, 13
- COM port
 - configuration, 41
 - wedge settings, 41
- COM1 port, 39
- Comm port wedge
 - settings, 41
- Command line syntax
 - AutoCab, 46
- Communications options, 111
- CompactFlash card slot, 16
- Computer shutdown, 4
- Configuration parameters, 39
- ConfigureProfile(), 92
- Configuring security, 118
- Configuring the Computer
 - troubleshooting, 107
- Connecting to
 - an ISP, 113
 - work, 116
- Connections
 - ending, 117
 - to an ISP, 113
 - via modem, 113
 - to work, 116
 - via modem
 - to an ISP, 113
- Conserving battery power, 2
- Contents in SDMMC Disk folder, 17
- Control panel applets
 - backlight, 10
 - intemec settings
 - beeper volume, 8
 - power
 - battery status, 4
- CoreDll.dll, 97
- CreateEvent(), 100
- Creating
 - a modem connection
 - to an ISP, 113
 - CAB files, 53
 - with CAB Wizard, 65
 - INF files, 53
- D**
- Deprecated functions, 97
- Desktop screen
 - Windows CE .NET, 23
- DeviceId.h, 71
- Display specifications, 18
- DllRegisterServer, 56
- DllUnregisterServer, 56
- DRAM
 - low battery shutdown, 5
- E**
- EAP (Extensible Authentication Protocol), 119
- EAP-FAST, 119, 120
- EasySet
 - scan bar code labels, 39
- EnableSuppLogging(), 92
- EnableWep(), 87
- EnableZeroConfig(), 93
- EncryptionStatus(), 88
- EncryptWepKeyForRegistry(Deprecated), 97
- Ending a connection, 117
- Environmental specifications, 18
- Epson Escape Sequences, 39
- ERROR_INSUFFICIENT_BUFFER
 - IOCTL_HAL_ITC_READ_PARM, 68
 - IOCTL_HAL_ITC_WRITE_SYSPARM, 71
- ERROR_INVALID_PARAMETER
 - IOCTL_HAL_ITC_READ_PARM, 68
 - IOCTL_HAL_ITC_WRITE_SYSPARM, 71
- Expansion slot specifications, 18
- Extensible Authentication Protocol, 119
- F**
- FAST (Flexible Authentication via Secure Tunneling), 119, 120
- Files
 - accessing on the Secure Digital card, 17
- Flash File Store
 - packaging an application, 47
- Flexible Authentication via Secure Tunneling (FAST), 119, 120
- Funk security, 120
 - 802.1x, 123
 - selecting a profile, 120
 - static WEP, 125
 - WPA, 121
- G**
- GDI approach, 39
- GetAssociationStatus(), 79
- GetAuthenticationMode(), 80
- GetBSSID(), 80

Index

- GetCCXStatus(), 86
- GetCurrentDriverName(), 93
- GetDiversity(), 81
- GetLinkSpeed(), 81
- GetMac(), 82
- GetMedia(Deprecated), 97
- GetMedium(Deprecated), 97
- GetNetworkMode(), 82
- GetNetworkType(), 83
- GetNicStats(Deprecated), 97
- GetPowerMode(), 84
- GetRadioIpAddress(), 86
- GetRSSI(), 84
- GetRTSThreshold(Deprecated), 97
- GetSSID(), 83
- Getting connected
 - ISP, 113
 - to an ISP, 113
 - creating a modem connection, 113
 - to work, 116
 - Windows Mobile, 113
- GetTXPower(), 85
- GetWepStatus(), 85
- H**
- HAL
 - version of Pocket PC
 - IOCTL_HAL_GET_BOOTLOADER_VERSION_INFO, 73
 - IOCTL_HAL_GET_OAL_VERINFO, 72
- Headset jack, 4
- Helper functions, 92
- Hirose docking connector
 - headset jack, 4
 - microphone, 3
- I**
- ID field values
 - IOCTL_HAL_ITC_READ_PARM
 - ITC_NVPARM_80211_INSTALLED, 70
 - ITC_NVPARM_80211_RADIO_TYPE, 70
 - ITC_NVPARM_ANTENNA_DIVERSITY, 69
 - ITC_NVPARM_BLUETOOTH_INSTALLED, 70
 - ITC_NVPARM_CONTRAST, 69
 - ITC_NVPARM_DISPLAY_TYPE, 68
 - ITC_NVPARM_ECN, 69
 - ITC_NVPARM_EDBG_SUBNET, 69
 - ITC_NVPARM_EDG_IP, 68
 - ITC_NVPARM_ETHERNET_ID, 68
 - ITC_NVPARM_INTERMEC_DATA_COLLECTION_HW, 69
 - ITC_NVPARM_INTERMEC_DATA_COLLECTION_SW, 69
 - ITC_NVPARM_INTERMEC_SOFTWARE_CONTENT, 69
 - ITC_NVPARM_LAN9000_INSTALLED, 70
 - ITC_NVPARM_MANF_DATE, 68
 - ITC_NVPARM_MCODE, 69
 - ITC_NVPARM_RTC_RESTORE, 69
 - ITC_NVPARM_SERIAL_NUM, 68
 - ITC_NVPARM_SERIAL2_INSTALLED, 70
 - ITC_NVPARM_SERVICE_DATE, 68
 - ITC_NVPARM_SIM_PROTECT_HW_INSTALLED, 70
 - ITC_NVPARM_SIM_PROTECT_SW_INSTALLED, 70
 - ITC_NVPARM_VERSION_NUMBER, 69
 - ITC_NVPARM_VIBRATE_INSTALLED, 70
 - ITC_NVPARM_WAN_FREQUENCY, 69
 - ITC_NVPARM_WAN_INSTALLED, 69
 - ITC_NVPARM_WAN_RADIO_TYPE, 69
 - ITC_NVPARM_WAN_RI, 69
 - IOCTL_HAL_ITC_WRITE_SYSPARM
 - ITC_DOCK_SWITCH, 71
 - ITC_WAKEUP_MASK, 71
 - ITC_AMBIENT_FRONTLIGHT, 71
 - ITC_AMBIENT_KEYBOARD, 71
 - ITC_REGISTRY_SAVE_ENABLE, 71
- IDLs
 - data collection, 9
 - device, 67, 98
 - smartsystems foundation, 105
 - URL, 15
- Imager
 - beeper volume
 - turning it off, 9
 - turning it on, 8
 - configuration parameters, 39
- INF files
 - creating, 53
- Input panel
 - keyboard, 24
 - Pocket Word, 31
 - selecting typed text, 25
 - transcriber, 25
 - Windows CE .NET, 23
- Installation functions
 - Setup.dll, 62

- Installing applications
 - Avalanche, 45
 - SmartSystems, 45
 - using a storage card, 44
 - using Secure Digital cards, 44
 - with ActiveSync, 43
- Intermec Developer Library, 9
- Intermec part numbers, 18
- Intermec settings
 - 802.11b/g, 112
 - beeper volume, 8, 127
- Intermec Settings applet
 - Funk security, 120
- Intermec settings applet
 - smartsystems, 9, 41, 127
- INTERMEC_PACKET_DRIVER
 - SwitchPacketDriver(), 96
- Internal card slots, 16
- Internal scanners
 - reading distances
 - 2D area imager, 42
 - EA11, 42
 - specifications, 18
- Internet Explorer
 - browsing the Internet, 33
 - software build version
 - CE.NET, 14
 - PSM, 14
- Internet Explorer Mobile
 - getting connected, 113
- IOCTL_GET_CPU_ID, 77
- IOCTL_HAL_COLDBOOT, 74, 99
- IOCTL_HAL_GET_BOOT_DEVICE, 75
- IOCTL_HAL_GET_BOOTLOADER_VERINFO, 73
- IOCTL_HAL_GET_DEVICE_INFO, 67
- IOCTL_HAL_GET_DEVICEID, 71
- IOCTL_HAL_GET_OAL_VERINFO, 72
- IOCTL_HAL_GET_RESET_INFO, 74
- IOCTL_HAL_ITC_READ_PARM, 68
- IOCTL_HAL_ITC_WRITE_SYSPARM, 70
- IOCTL_HAL_REBOOT, 76, 98
- IOCTL_HAL_WARMBOOT, 73, 99
- IOCTL_PROCESSOR_INFORMATION, 76
- isDHCPEnabled(), 93
- isOrinoco(), 94
- ISP
 - connecting to via Windows Mobile, 113
 - creating
 - a modem connection, 113
 - Internet Explorer, 33
 - Windows Mobile, 113
- isSupplicantRunning(), 94
- isZeroConfigEnabled(), 94
- ITC_DOCK_SWITCH, 71
- ITC_WAKEUP_MASK, 71
- ITC_AMBIENT_FRONTLIGHT, 71
- ITC_AMBIENT_KEYBOARD, 71
- ITC_DEVID_80211RADIO_INTEL_2011B, 70
- ITC_DEVID_80211RADIO_MAX values
 - ITC_DEVID_80211RADIO_INTEL_2011B, 70
 - ITC_DEVID_80211RADIO_NONE, 70
- ITC_DEVID_80211RADIO_NONE, 70
- ITC_DEVID_INTERMEC_EVIO, 69
- ITC_DEVID_INTERMEC2D_IMAGER, 69
- ITC_DEVID_OEM2D_IMAGER, 69
- ITC_DEVID_SCANHW_MAX values
 - ITC_DEVID_INTERMEC_EVIO, 69
 - ITC_DEVID_INTERMEC2D_IMAGER, 69
 - ITC_DEVID_OEM2D_IMAGER, 69
 - ITC_DEVID_SCANHW_NONE, 69
 - ITC_DEVID_SE900_LASER, 69
 - ITC_DEVID_SE900HS_LASER, 69
- ITC_DEVID_SCANHW_NONE, 69
- ITC_DEVID_SE900_LASER, 69
- ITC_DEVID_SE900HS_LASER, 69
- ITC_DEVID_WANRADIO_NONE, 69
- ITC_DEVID_WANRADIO_SIEMENS_MC45, 69
- ITC_DEVID_WANRADIO_SIEMENS_MC46, 69
- ITC_DEVID_WANRADIO_SIERRA_SB555, 69
- ITC_DEVID_WANRADIO_XIRCOM_GEM3503, 69
- ITC_KEYBOARD_CHANGE
 - CreateEvent(), 100
- ITC_NVPARAM_80211_INSTALLED, 70
- ITC_NVPARAM_80211_RADIO_TYPE, 70
- ITC_NVPARAM_ANTENNA_DIVERSITY, 69
- ITC_NVPARAM_BLUETOOTH_INSTALLED, 70
- ITC_NVPARAM_CONTRAST, 69
- ITC_NVPARAM_DISPLAY_TYPE, 68
- ITC_NVPARAM_ECN, 69
- ITC_NVPARAM_EDBG_SUBNET, 69
- ITC_NVPARAM_EDG_IP, 68
- ITC_NVPARAM_ETHERNET_ID, 68
- ITC_NVPARAM_INTERMEC_DATACOLLECTION_HW, 69
- ITC_NVPARAM_INTERMEC_DATACOLLECTION_SW, 69
- ITC_NVPARAM_INTERMEC_SOFTWARE_CONTENT, 69
- ITC_NVPARAM_LAN9000_INSTALLED, 70

Index

ITC_NVPARAM_MANF_DATE, 68
ITC_NVPARAM_MCODE, 69
ITC_NVPARAM_RTC_RESTORE, 69
ITC_NVPARAM_SERIAL_NUM, 68
ITC_NVPARAM_SERIAL2_INSTALLED, 70
ITC_NVPARAM_SERVICE_DATE, 68
ITC_NVPARAM_SIM_PROTECT_HW_INSTALLED, 70
ITC_NVPARAM_SIM_PROTECT_SW_INSTALLED, 70
ITC_NVPARAM_VERSION_NUMBER, 69
ITC_NVPARAM_VIBRATE_INSTALLED, 70
ITC_NVPARAM_WAN_FREQUENCY, 69
ITC_NVPARAM_WAN_INSTALLED, 69
ITC_NVPARAM_WAN_RADIO_TYPE, 69
ITC_NVPARAM_WAN_RI, 69
ITC_REGISTRY_SAVE_ENABLE, 71

K

KernelIoControl
 IOCTL_GET_CPU_ID, 77
 IOCTL_HAL_COLDBOOT, 74, 99
 IOCTL_HAL_GET_BOOT_DEVICE, 75
 IOCTL_HAL_GET_BOOTLOADER_VERINFO, 73
 IOCTL_HAL_GET_DEVICE_INFO, 67
 IOCTL_HAL_GET_DEVICEID, 71
 IOCTL_HAL_GET_OAL_VERINFO, 72
 IOCTL_HAL_GET_RESET_INFO, 74
 IOCTL_HAL_ITC_READ_PARM, 68
 IOCTL_HAL_ITC_WRITE_SYSPARM, 70
 IOCTL_HAL_REBOOT, 76, 98
 IOCTL_HAL_WARMBOOT, 73, 99
 IOCTL_PROCESSOR_INFORMATION, 76
KernelIoControl(), 67
Key sequences
 alpha (green) keys, 11
 orange keys, 10
Keyboard
 Windows CE .NET input panel, 24
Keyboard See Keypad
Keypad
 advanced remapping, 101
 alpha (green) key sequences, 11
 backlight control panel applet, 10
 change notification, 100
 driver registry settings, 100
 orange key sequences, 10
 planes, 99

registry settings
 alpha plane, 100
 gold plane, 100
 unshifted plane, 100
sample registry keys, 102
scan codes, 101
specifications, 19

L

LED status, 11
Line printing, 39
Loading certificates, 118
lpBytesReturned
 IOCTL_GET_CPU_ID, 77
 IOCTL_HAL_GET_BOOT_DEVICE, 75
 IOCTL_HAL_GET_BOOTLOADER_VERINFO, 73
 IOCTL_HAL_GET_DEVICE_INFO, 67
 IOCTL_HAL_GET_DEVICEID, 72
 IOCTL_HAL_GET_OAL_VERINFO, 72
 IOCTL_HAL_GET_RESET_INFO, 74
 IOCTL_HAL_ITC_READ_PARM, 68
 IOCTL_HAL_ITC_WRITE_SYSPARM, 70
 IOCTL_PROCESSOR_INFORMATION, 77
lpInBuf
 IOCTL_GET_CPU_ID, 77
 IOCTL_HAL_COLDBOOT, 74
 IOCTL_HAL_GET_BOOT_DEVICE, 75
 IOCTL_HAL_GET_BOOTLOADER_VERINFO, 73
 IOCTL_HAL_GET_DEVICE_INFO, 67
 IOCTL_HAL_GET_DEVICEID, 72
 IOCTL_HAL_GET_OAL_VERINFO, 72
 IOCTL_HAL_GET_RESET_INFO, 74
 IOCTL_HAL_ITC_READ_PARM, 68
 IOCTL_HAL_ITC_WRITE_SYSPARM, 70
 IOCTL_HAL_REBOOT, 76
 IOCTL_HAL_WARMBOOT, 74
 IOCTL_PROCESSOR_INFORMATION, 77
lpInBufSize
 IOCTL_GET_CPU_ID, 77
 IOCTL_HAL_COLDBOOT, 74
 IOCTL_HAL_GET_BOOT_DEVICE, 75
 IOCTL_HAL_GET_DEVICE_INFO, 67
 IOCTL_HAL_GET_DEVICEID, 72
 IOCTL_HAL_GET_OAL_VERINFO, 72
 IOCTL_HAL_GET_RESET_INFO, 74
 IOCTL_HAL_REBOOT, 76
 IOCTL_HAL_WARMBOOT, 74

- lpOutBuf
 - IOCTL_GET_CPU_ID, 77
 - IOCTL_HAL_COLDBOOT, 74
 - IOCTL_HAL_GET_BOOT_DEVICE, 75
 - IOCTL_HAL_GET_BOOTLOADER_VERI
NFO, 73
 - IOCTL_HAL_GET_DEVICE_INFO, 67
 - IOCTL_HAL_GET_DEVICEID, 72
 - IOCTL_HAL_GET_OAL_VERINFO, 72
 - IOCTL_HAL_GET_RESET_INFO, 74
 - IOCTL_HAL_ITC_READ_PARM, 68
 - IOCTL_HAL_ITC_WRITE_SYSPARM, 70
 - IOCTL_HAL_REBOOT, 76
 - IOCTL_HAL_WARMBOOT, 74
 - IOCTL_PROCESSOR_INFORMATION, 77
- M**
- MakeCab.exe, 66
- Memory and storage
 - specifications, 19
- Messaging
 - getting connected, 113
- Microphone, 3
- Microprocessor
 - specifications, 19
- Microsoft ActiveSync
 - adding programs, 28
 - adding programs to Start menu, 29
 - URL, 30
- Microsoft security, 120
- Microsoft WordPad, 30
- Modems
 - creating a connection
to an ISP, 113
- N**
- nDeviceId
 - NLEDGetDeviceInfo, 98
- NDIS_ENCRYPTION_1_ENABLED
 - EncryptionStatus(), 88
 - GetWepStatus(), 85
- NDIS_ENCRYPTION_1_KEY_ABSENT
 - EncryptionStatus(), 88
 - GetWepStatus(), 85
- NDIS_ENCRYPTION_2_ENABLED
 - EncryptionStatus(), 88
 - GetWepStatus(), 85
- NDIS_ENCRYPTION_2_KEY_ABSENT
 - EncryptionStatus(), 88
 - GetWepStatus(), 85
- NDIS_ENCRYPTION_3_ENABLED
 - EncryptionStatus(), 88
 - GetWepStatus(), 85
- NDIS_ENCRYPTION_3_KEY_ABSENT
 - EncryptionStatus(), 88
 - GetWepStatus(), 85
- NDIS_ENCRYPTION_DISABLED
 - EncryptionStatus(), 88
 - GetWepStatus(), 85
- NDIS_ENCRYPTION_NOT_SUPPORTED
 - EncryptionStatus(), 88
 - GetWepStatus(), 85
- NDIS_MIXED_CELL_OFF
 - SetMixedCellMode(), 91
- NDIS_MIXED_CELL_ON
 - SetMixedCellMode(), 91
- NDIS_NET_AUTO_UNKNOWN
 - GetNetworkMode(), 82
 - SetNetworkMode(), 90
- NDIS_NET_MODE_ESS
 - GetNetworkMode(), 82
 - SetNetworkMode(), 90
- NDIS_NET_MODE_IBSS
 - GetNetworkMode(), 82
 - SetNetworkMode(), 90
- NDIS_NET_MODE_UNKNOWN
 - GetNetworkMode(), 82
 - SetNetworkMode(), 90
- NDIS_NET_TYPE_DS
 - GetNetworkType(), 83
- NDIS_NET_TYPE_FH
 - GetNetworkType(), 83
- NDIS_NET_TYPE_OFDM_2_4G
 - GetNetworkMode(), 82
 - SetNetworkMode(), 90
- NDIS_NET_TYPE_OFDM_5G
 - GetNetworkMode(), 82
 - SetNetworkMode(), 90
- NDIS_NET_TYPE_UNDEFINED
 - GetNetworkType(), 83
- NDIS_NETWORK_EAP_MODE_OFF
 - GetCCXStatus(), 86
 - SetCCXStatus(), 91
- NDIS_NETWORK_EAP_MODE_ON
 - GetCCXStatus(), 86
 - SetCCXStatus(), 91
- NDIS_POWER_LEVEL_1
 - GetTXPower(), 85
- NDIS_POWER_LEVEL_15
 - GetTXPower(), 85
- NDIS_POWER_LEVEL_30
 - GetTXPower(), 85
- NDIS_POWER_LEVEL_5
 - GetTXPower(), 85
- NDIS_POWER_LEVEL_63
 - GetTXPower(), 85

Index

- NDIS_POWER_LEVEL_UNKNOWN
 - GetTXPower(), 85
- NDIS_RADIO_ASSOCIATED
 - GetAssociationStatus(), 79
- NDIS_RADIO_AUTH_MODE_AUTO
 - GetAuthenticationMode(), 80
 - SetAuthenticationMode(), 89
- NDIS_RADIO_AUTH_MODE_ERROR
 - GetAuthenticationMode(), 80
 - SetAuthenticationMode(), 89
- NDIS_RADIO_AUTH_MODE_OPEN
 - GetAuthenticationMode(), 80
 - SetAuthenticationMode(), 89
- NDIS_RADIO_AUTH_MODE_SHARED
 - GetAuthenticationMode(), 80
 - SetAuthenticationMode(), 89
- NDIS_RADIO_AUTH_MODE_WPA
 - GetAuthenticationMode(), 80
 - SetAuthenticationMode(), 89
- NDIS_RADIO_AUTH_MODE_WPA_NONE
 - GetAuthenticationMode(), 80
 - SetAuthenticationMode(), 89
- NDIS_RADIO_AUTH_MODE_WPA_PSK
 - GetAuthenticationMode(), 80
 - SetAuthenticationMode(), 89
- NDIS_RADIO_POWER_AUTO
 - GetPowerMode(), 84
 - SetPowerMode(), 90
- NDIS_RADIO_POWER_MODE_CAM
 - GetPowerMode(), 84
 - SetPowerMode(), 90
- NDIS_RADIO_POWER_MODE_FAST_PSP
 - GetPowerMode(), 84
 - SetPowerMode(), 90
- NDIS_RADIO_POWER_MODE_PSP
 - GetPowerMode(), 84
 - SetPowerMode(), 90
- NDIS_RADIO_POWER_UNKNOWN
 - GetPowerMode(), 84
 - SetPowerMode(), 90
- NDIS_RADIO_SCANNING
 - GetAssociationStatus(), 79
- NDIS_SUPP_LOGGING_OFF
 - EnableSuppLogging(), 92
- NDIS_SUPP_LOGGING_ON
 - EnableSuppLogging(), 92
- NDISUIO.DLL, 78
- NDISUIO_PACKET_DRIVER
 - SwitchPacketDriver(), 96
- nInBufSize
 - IOCTL_HAL_GET_BOOTLOADER_VERI
NFO, 73
 - IOCTL_HAL_ITC_READ_PARM, 68
- IOCTL_HAL_ITC_WRITE_SYSPARM, 70
- IOCTL_PROCESSOR_INFORMATION, 77
- nInfoId
 - NLEDGetDeviceInfo, 98
- NLED driver
 - vibrator, 97
- NLED.H, 98
 - NLEDGetDeviceInfo, 98
 - NLEDSetDevice, 98
- NLED_COUNT_INFO
 - NLEDGetDeviceInfo, 98
- NLED_SETTINGS_INFO_ID
 - NLEDGetDeviceInfo, 98
 - NLEDSetDevice, 98
- NLED_SUPPORTS_INFO_ID
 - NLEDGetDeviceInfo, 98
- NLEDGetDeviceInfo, 98
- NLEDSetDevice, 98
- nOutBufSize
 - IOCTL_GET_CPU_ID, 77
 - IOCTL_HAL_COLDBOOT, 74
 - IOCTL_HAL_GET_BOOT_DEVICE, 75
 - IOCTL_HAL_GET_BOOTLOADER_VER
INFO, 73
 - IOCTL_HAL_GET_DEVICE_INFO, 67
 - IOCTL_HAL_GET_DEVICEID, 72
 - IOCTL_HAL_GET_OAL_VERINFO, 72
 - IOCTL_HAL_GET_RESET_INFO, 74
 - IOCTL_HAL_ITC_READ_PARM, 68
 - IOCTL_HAL_ITC_WRITE_SYSPARM, 70
 - IOCTL_HAL_REBOOT, 76
 - IOCTL_HAL_WARMBOOT, 74
 - IOCTL_PROCESSOR_INFORMATION, 77
- O**
- Object Store
 - packaging an application, 47
- Object store
 - IOCTL_HAL_COLDBOOT, 74
 - IOCTL_HAL_WARMBOOT, 73
- OEMIOCTL.H
 - IOCTL_GET_CPU_ID, 77
 - IOCTL_HAL_COLDBOOT, 74
 - IOCTL_HAL_GET_BOOT_DEVICE, 75
 - IOCTL_HAL_GET_BOOTLOADER_VER
INFO, 73
 - IOCTL_HAL_GET_OAL_VERINFO, 72
 - IOCTL_HAL_GET_RESET_INFO, 74
 - IOCTL_HAL_ITC_READ_PARM, 68
 - IOCTL_HAL_ITC_WRITE_SYSPARM, 70
 - IOCTL_HAL_REBOOT, 76
 - IOCTL_HAL_WARMBOOT, 73
- Oldstyle device ID, 71

- Operating system
 - specifications, 19
- Operating temperature, 18
- Operating the Computer
 - troubleshooting, 106
- Orange plane on keypad, 99
- OSVERSIONINFO.dwBuildNumber, 55
- OSVERSIONINFO.dwVersionMajor, 55
- OSVERSIONINFO.dwVersionMinor, 55
- P**
- Packaging an application
 - Flash File Store, 47
 - Object Store, 47
 - Persistent Storage Manager, 47
 - Secure Digital storage cards, 47
- Page format printing, 39
- Patent information, vii
- Performing a cold boot, 13
- Physical dimensions
 - specifications, 19
- Physical specifications, 18
- pInput
 - NLEDSetsDevice, 98
- PKFUNCS.H
 - IOCTL_HAL_GET_DEVICEID, 71
 - IOCTL_PROCESSOR_INFORMATION, 76
- Planes, keypad, 99
- Pocket PC
 - IOCTL_HAL_GET_BOOTLOADER_VERINFO, 73
 - IOCTL_HAL_GET_OAL_VERINFO, 72
- Pocket Word
 - synchronizing, 32
 - typing mode, 31
 - writing mode, 32
- pOutput
 - NLEDGetDeviceInfo, 98
- Power applet
 - battery status, 4
- Printer support, 39
- Printer, Zebra PT403 portable, 39
- Processor information
 - IOCTL_PROCESSOR_INFORMATION, 76
- Programs, adding or removing
 - Windows CE .NET, 27
- PSM
 - determining build version, 14
 - packaging an application, 47
- Q**
- Query information functions, 79
- R**
- RadioConnect(), 78
- RadioDisassociate(), 79
- RadioDisconnect(), 79
- Radios
 - card support, 17
- Reader commands, 39
- Reading distances
 - 2D area imager, 42
 - EA11, 42
- Reboot methods
 - IOCTL_HAL_COLDBOOT, 99
 - IOCTL_HAL_REBOOT, 98
 - IOCTL_HAL_WARMBOOT, 99
- Recharging time, 19
- Recovery CD
 - RegFlushKey() API, 66
 - updating the system software, 104
- RegFlushKey(), 66
- Registry
 - keypad remapping, 100
 - sample view of key mapping, 102
 - save location
 - IOCTL_HAL_ITC_WRITE_SYSPARM, 70
- Registry settings
 - keypad driver, 100
 - keypad planes
 - alpha, 100
 - gold, 100
 - unshifted, 100
- Registry storage
 - enabling, 7
- RegOpenKeyEx(), 99
- RegQueryValueEx(), 99
- RegSetValueEx(), 99
- Regulatory approvals
 - specifications, 19
- RemoveWep(), 88
- Removing programs
 - Windows CE .NET, 27, 29
- RenewDHCP(), 95
- Reset button, 13
- ResetRadioToSystemSave(), 95
- Resource kits
 - data collection, 9
 - device, 67, 98
 - smartsystems foundation, 105
 - URL, 15
- RPM.EXE, 56
- RPMCE212.INI, 56
- S**
- Samsung radio
 - 802.11b/g information, 112
- SanDisk SD cards, 15

Index

- Scan codes
 - keypad, 101
 - Scanner
 - beeper volume
 - turning it on, 8
 - specifications, 18
 - Scanning bar codes
 - troubleshooting, 108
 - Secure Digital card slot, 16
 - Secure Digital cards
 - accessing files, 17
 - installing applications, 44
 - packaging an application, 47
 - pull tabs, 16
 - removing, 17
 - specifications, 18
 - upgrading computer, 104
 - Security
 - choosing between Funk and Microsoft, 120
 - configuring, 118
 - loading certificates, 118
 - wireless network, 118
 - Set information functions, 87
 - SetAuthenticationMode(), 89
 - SetCCXStatus(), 91
 - SetChannel(), 89
 - SetDiversity(Deprecated), 97
 - SetMixedCellMode(), 91
 - SetNetworkMode(), 90
 - SetPowerMode(), 90
 - SetRTSThreshold(Deprecated), 97
 - SetSSID(), 91
 - Settings applets
 - intermec settings
 - funk security, 120
 - SetTXRate(Deprecated), 97
 - Setup.dll, 56, 62
 - DllMain, 62
 - installation functions, 62
 - SHFullScreen(), 66, 67
 - SIM card slot, 16
 - SIM cards
 - protection hardware, 70
 - software installed, 70
 - SmartSystems, 9, 41, 45, 127
 - upgrading computer, 105
 - Software versions
 - CE.NET build, 14
 - PSM builds, 14
 - SourceDisksNames.MIPS, 56
 - SourceDisksNames.SH3, 56
 - Speaker, 3
 - Specifications, 18
 - battery, 19
 - display, 18
 - environmental, 18
 - expansion slots, 18
 - integrated scanner options, 18
 - integrated wireless, 18
 - keypad options, 19
 - memory and storage, 19
 - microprocessor, 19
 - operating system, 19
 - physical dimensions, 19
 - regulatory approvals, 19
 - standard communications, 19
 - Standard communications
 - specifications, 19
 - Start Menu
 - adding programs, 29
 - via Microsoft ActiveSync, 29
 - via Windows Explorer, 29
 - StartScanList(), 95
 - StartSupplicant(), 96
 - Static WEP security
 - Funk, 125
 - Status icons
 - Windows CE .NET, 23
 - StopSupplicant(), 96
 - Storage media, 15
 - specifications, 18
 - Storage temperature, 18
 - SwitchPacketDriver(), 96
 - Synchronize system time, 36
 - Synchronizing
 - Pocket Word, 32
 - System Properties applet
 - active programs, 26
 - System time, 36
 - SYSTEMINFO.dwProcessorType, 54
- ## T
- Tahoma.ttf, 56
 - Temperatures
 - battery, 19
 - specifications, 18
 - Temporal Key Integrity Protocol, 119
 - Tethered scanner
 - settings, 41
 - Time server, 36
 - TKIP (Temporal Key Integrity Protocol), 119
 - Tools CD
 - CAB files, 44

- Troubleshooting, 106
 - 802.1x security, 108
 - bar code scanning, 108
 - CAB Wizard, 66
 - configuration, 107
 - operation, 106
 - wireless connectivity, 107
- Typing mode
 - Pocket Word, 31
- Typing on the screen
 - Pocket Word, 31
- U**
- Unshifted plane on keypad
 - regular keypad, 99
- Updating
 - bootloader, 43
- URLs
 - full screen display, 67
 - Microsoft support, 22
 - Windows CE .NET support, 22
- URODDSV.CEXE, 78
- Utilities applet
 - registry save, 7
- UUID, 71
- V**
- Vibrator
 - programming, 97
- Viewing mobile favorites and channels
 - Internet Explorer, 33
- VN_CLASS_ASIC, 69
- VN_CLASS_BOOTSTRAP, 69
- VN_CLASS_KBD, 69
- W**
- WAN radio IDs
 - ITC_DEVID_WANRADIO_NONE, 69
 - ITC_DEVID_WANRADIO_SIEMENS_MC
45, 69
 - ITC_DEVID_WANRADIO_SIEMENS_MC
46, 69
 - ITC_DEVID_WANRADIO_SIERRA_SB555,
69
- WAP pages, 33
 - connecting to an ISP, 113
- Warm boot
 - IOCTL_HAL_REBOOT, 76
 - IOCTL_HAL_WARMBOOT, 73
- Wavelink Avalanche, 45
- WCStart.ini, 56
- Web pages, 33
 - connecting to an ISP, 113
- WEP (Wired Equivalent Privacy) encryption, 119
- Wi-Fi Protected Access, 119, 121
- Windows CE .NET
 - basic skills, 22
 - Desktop screen, 23
 - notifications, 24
 - programs, 23
 - Start menu, 23
 - support URLs, 22
 - task bar, 23
 - where to find information, 22
- Windows Explorer
 - adding programs to Start menu, 29
 - Windows CE .NET, 26
- Windows Mobile
 - getting connected, 113
- Wired Equivalent Privacy, 119, 125
- Wireless connectivity
 - troubleshooting, 107
- Wireless network
 - security, 118
 - specifications, 18
- Wistron radio
 - 802.11b/g information, 112
- WordPad, 30
 - creating a document, 31
- Work
 - getting connected, 116
- WPA (Wi-Fi Protected Access), 119
- WPA security
 - Funk, 121
- WPA2 (Wi-Fi Protected Access), 119
- Writing mode
 - Pocket Word, 32
- Writing on the screen
 - Pocket Word, 32
- X**
- Xscale processor ID
 - IOCTL_GET_CPU_ID, 77
- Z**
- Zebra PT403 portable printer, 39
- ZNICZIO.DLL, 78



Corporate Headquarters
6001 36th Avenue West
Everett, Washington 98203
U.S.A.

tel 425.348.2600

fax 425.355.9551

www.intermec.com

751G Color Mobile Computer User's Manual



P/N: 961-054-036C