

# Getting Started: Beginning Programming Knowledge For Your Intermec Computer

## Contents

Introduction - (2)

Operating Systems - (2)

Programming Languages and Development Environment - (3)

- Programming Languages
- Microsoft SDKs
- Active Sync
- Device Emulator
- Intermec Developer Library (IDL)
- Handheld Considerations
  - Battery Life
  - Memory
  - Pocket Internet Explorer
  - Screen Size and GUI

Help, Support and Assistance - (6)

- Developer Support
- Customer Support and Knowledge Central (KC)
- Intermec Application Software Group

Additional Tools and Utilities - (7)

- Intermec Connection Manager - Radio Tool
- iBrowse - Lockdown Internet Browser
- iLaunch - Lockdown Application Launcher
- SmartSystems - Intermec Management Tool
- TE2000 - Intermec Terminal Emulation
- WWAN Toolkit - WWAN Programming Toolkit
- WWAN Communicator - WWAN Communication Tool

## Introduction

Times have changed and so have the life cycles, operating systems, radios, features and more with regards to mobile computing. To further that point, it needs to be stressed that your handheld is not a desktop computer, and as such, is not programmed exactly the same way. Yes, the basic programming principles are the same, but there are many areas to take into account: memory, battery life, usability, and screen size, to but name a few. This paper discusses the options you have for application development and the common questions associated with the development process. Each section is an overview of an individual topic. To get more information on each topic please note the additional resources under each section.

## Operating Systems

With the exception of the CK1, all of Intermec's handhelds run either a version of the Windows CE or Pocket PC (Windows Mobile) Operating Systems. What many do not realize is that Windows CE manages every version of the Pocket PC Operating System. The CE Operating System (OS) is a scaled down version of Windows 2000. By scaling down the CE OS you get what is referred to as Pocket PC.

Just like others you see on the market today regarding software, the actual versions of these operating systems seem to "upgrade" every year to year and a half. In 2002 we saw the introduction of CE 3.0 and later Pocket PC 2002. A year later came CE 4.2 and then its corresponding Pocket PC 2003 (Windows Mobile 2003). With the writing of this paper the current operating systems are CE 5.0 and Pocket PC 5.0 (Windows Mobile 5.0). These latest operating systems are currently installed on the CK6x and CN30 series and on the 700 Color expected in mid-2006.

It is important to note that although Intermec's tools now function the same across all new handhelds and operating systems, the same cannot be said of all of Microsoft's and third party tools. If you are writing an application using Microsoft or third party APIs, it is important to check which operating systems they are supported on. Remember, Pocket PC is not Windows CE and vice versa. If you are unsure about a utility, application or API, do some research and ask what operating systems are supported. Furthermore, this will help you in the testing process to make sure everything you are running or writing is supported on your handheld(s).

	700 Mono	700 Color	CK3x	CK6x	CN30	CN2x	CK1	CV60
PPC 2000	X							
PPC 2002	X	X						
PPC 2003		X				X		
PPC 2005		TBD		X	X			
CE 3.0			X					
CE 4.2			X			X		X
CE 5.0			TBD	X				
Linux							X	
XP Pro								X
E XP								X

\*Note: this is a general reference table showing Intermec's entire series of handhelds; please check with your sales team as to what options are available on individual handhelds in a series.

### Additional Resources:

- \* [Microsoft Mobility Homepage](#): great start for anything regarding MS and your handheld.
- \* [Getting Started with Windows Mobile 5.0](#): white papers, downloads, tools and more!
- \* [Mobile PC and Tablet PC Development Center](#): great start for all using these (CVxx).
- \* [Mobile Developers Community](#): blogs, white papers, web talks, and more!

## Programming Languages and Development Environment

### Programming Languages

Now that you know what operating system is installed on your handheld, you are ready to set up your development environment. Before we delve into the tools to using for the current handhelds and 5.0 versions of the operating system, it is important to mention what was used in the past.

With the 700 Color Series we were introduced to Pocket PC 2002 (PPC) and later 2003. Pocket PC 2002 supports a programming tool called Embedded Visual Tools 3.0. This tool includes embedded versions of C++ and VB (eC++ and eVB). Pocket PC 2003 supports its successor, Embedded Visual Tools 4.0. Unlike 3.0 the only supported language is eC++, however, older eVB applications will run on this OS after installing the runtime cab file from Microsoft. With the introduction of PPC 2003, Microsoft provided limited support for eVB. In mid-2005 Microsoft made the announcement that eVB would not be supported beyond the already released PPC 2003.

Additional Resources:

- \* [Embedded Visual Tools 3.0 Download](#) – General MS Download Location.
- \* [Embedded Visual Tools 4.0 Download](#) – General MS Download Location.
- \* [Visual Studio 2003.net 2003 Main Web site](#) – General Web site Location.
- \* [Visual Studio 2005 Developer Center](#)– General Web site Location.
- \* [Java Technology](#) – Sun’s Home for Java.
- \* [Microsoft E-Learning for Visual Studio 2005](#) – Additional Resources.
- \* [Code Snippets in Visual Studio 2005](#) – Additional Resource from dotnetjunkies.
- \* [Visual Studio 2005](#) – Additional Resource from C# Corner.

### Microsoft SDKs

Knowing your handhelds operating system and the toolset you want to program with, how do you build an application for the handheld/operating system? If you are going to build an application

	PPC 2000	PPC 2002	PPC 2003	PPC 2005	CE 3.0	CE 4.2	CE 5.0	Linux
Embedded Tools 3.0	X	X			X			
Embedded Tools 4.0			X			X	X	
Visual Studio 2003	X	X	X		X	X	X	
Visual Studio 2005			X	X			X	
Java	X	X	X	X	X	X	X	
MCL		X	X	TBD		X	TBD	X

\*Note: this is a general table relating operating systems to their supported programming languages. MCL is also offered in a DOS version and the CK1 Linux device accepts general C programming. This table details the coding languages in which development can be fully utilized with the corresponding operating systems. Many of the previous versions will run on the newer operating systems, however, functionality such as new coding and debugging is not. Visual Studio also requires its corresponding .net cf version installed; if not already a part of the installed OS.

Both of these operating systems support another toolset called Visual Studio.NET 2003. This introduced two “new” managed coding languages: VB.net and C#. In addition to the languages previously mentioned, there is full Java support from a third party JVM (Java Virtual Manager) installed onto the device. With the possible confusion of these old tool sets and constant changes, many ask where they start with the new breed of handheld computers.

In October 2005, the world was finally introduced to one common programming toolset: Visual Studio 2005. Visual Studio 2005 is the programming tool you need to be programming with to have full development support in CE and Pocket PC 2005. Visual Studio 2005 provides you full support for C++, VB.net, and C#. Java may still be coded by installing a third party JVM as you have done in the past.

to bridge processors and operating systems you will need to compile for each of them. Aside from providing you the operating system target, the SDK will also provide you with additional samples and documentation from Microsoft. You will need to download and install the SDK of any operating system you wish to target. As of today, for Intermec handheld computers, you will compile for two processor types: ArmV4 (700 Color, CN2b) and ArmV4i (CN2a, CK6x, and CN30).

#### Additional Resources:

- \* [SDK for PPC 2005 Based Pocket PCs](#) – Covers all languages supported.
- \* [SDK for PPC 2003 Based Pocket PCs](#) – Covers all languages supported.
- \* [SDK for PPC 2002 Based Pocket PCs](#) – Covers all languages supported.
- \* [SDK for PPC 2000 Based Pocket PCs](#) – Covers all languages supported.
- \* [Win CE 3.0 API Reference](#) – docs for all interfaces, functions, structures, and other elements.
- \* [SDK for Win CE 4.2 Based Pocket PCs](#) – Covers all languages supported.
- \* [SDK for Win CE 5.0 Based Pocket PCs](#) – Covers all languages supported.

#### Active Sync

Assumption: Your development environment is coming along; you now know your OS, have a programming environment, and can compile an application for your handheld. However, since none of us are perfect and development requires much trial and error, you will likely be deploying your application to your handheld to test. The best and most commonly used method for this is via a program by Microsoft called Active Sync. Active Sync acts as the gateway between your Windows-based PC and Pocket PC based device, enabling the transfer of information from your computer to your device. This connection is typically made via a USB or Serial Cable from your dock. Pocket PC 5.0 requires the Active Sync version to be 4.1 or higher and can be downloaded here from the [Microsoft Web site](#).

#### Additional Resources:

- \* [Active Sync Troubleshooting Guide](#) – Tools and overview from Pocket PC FAQ

#### Device Emulator

Microsoft Visual Studio (2003 and 2005) comes equipped with a device emulator program that aids in your testing if a device is not available. For the most part, an emulator allows you to test on a “virtual device”. It is important to note that an emulator is not a substitute for true device testing. Although an emulator is a great test and demonstration environment for general operating system and Microsoft programming, it is not a viable test alternative for Intermec specific APIs and hardware functionality. All testing of Intermec specific APIs and hardware must be done on an actual device as it is not supported in the emulator. It is also important to be sure that your handheld is running the latest operating system to ensure testing is done with the latest fixes and drivers. Please contact a support option below to find information on the latest operating system for your device, and how to obtain a copy if needed.

#### Additional Resources:

- \* [Localized PPC 5.0 Emulator Images](#) – test in all available PPC 5.0 PC localizations.
- \* [Windows CE 5.0 Device Emulator](#) - contains the emulator technologies featured CE 5.0.
- \* [Windows Mobile 5.0 Device Emulator](#) - contains the emulator technologies featured WM 5.0.

#### Intermec Developer Library

The Intermec Developer Library (IDL) is a free Intermec provided developer toolset for all programmers that code for Intermec devices. In the past there were individual SDKs for each handheld.

This meant that an application had to be changed and recompiled for each different handheld you chose to pilot with and then deploy. The IDL is a one-stop-shop of resource kits that works across all of our Intermec handhelds from the 700 Mono onward. The resource kits are contained in one easy to download package from the [IDL Web site](#).

With the IDL you are able to use one common API set that will work across all of our handhelds and their operating systems. When a new device or piece of hardware (radio, peripheral, etc) is introduced, Intermec takes the time to update the APIs to save you time and money. The IDL offers these resource kits for your programming needs: Antares Migration, Bluetooth, Communications, Data Collection, Device, Mobile Gadgets, Printing, and RFID.

Each resource kit will contain the needed deliverables for each API, as well as documentation and samples. To help make the programming environment complete, each resource kit will install directly into the desktop Visual Studio environment.

Aside from the main resource kits there are many other important areas of the IDL:

- Software Showcase: Intermec and third party tools, programs, and utilities for your devices.
- Beta Zone: Intermec area that allows testing with beta software before it is officially released. You must fill out the form and agree to the beta agreement.
- RSS Feeds and Update Notification: Sign up to always be aware of the latest updates and offerings of the IDL.

#### Additional Resources:

- \* [IDL Web site](#) – Resource Kits, Beta Zone, Software Showcase and More!

#### Handheld Considerations

##### Battery Life

Battery life is perhaps the most important and confusing area for developers. Desktop developers have always had to build applications around hardware: memory, processor, video card... but never a battery. How your battery is managed by your application will severely be impacted by your backlight, suspend time, and radio power management. Each of these settings individually affects the handheld's overall battery life and are inter-related. Following is a discussion of each in a bit more detail.

**Backlight:** The backlight is one of the largest power draws on the handheld and depending on suspend settings it can be used continuously. By default your handheld is set to 100% backlight, and will remain “on” depending on the handheld's suspend settings. If looking to optimize your application for battery life, the backlight is the first setting to work with.

Generally, if your users are working in an artificial light setting, 50% provides a good brightness, and could potentially save hours of battery life with one setting alteration. Those working in natural light, particularly bright natural light, will need their backlights set to either 75 or 100%. Working with the users to test potential settings is highly advised. In the past, some programmers have opted to allow 2 settings that the users could toggle via an onscreen button (one brighter setting, another lower). This not only allows a good user experience, but also provides enhanced battery management above an overall 100% default.

Suspend Time: Allowing the handheld to suspend when not in use is the best thing a programmer can do to extend battery life. By default this setting is 1 min, however, via direct registry keys you can set this as low as a few seconds. A recommended setting is between 30 seconds to 1 minute depending on your usage scenario and could save hours of battery life. However, not every customer situation allows you to suspend the terminal. There are many customers running a handheld continually throughout the day whose scenario cannot allow suspend enabled. In cases such as this, optimization of the radio(s) is most important.

Radio Power Management: Contrary to popular belief, each radio does not function the same when it comes to power management and having a radio “on” does not draw much power in the long run. With regard to radios, “on” is not the same as on and actively transmitting or looking for a connection.

The 802.11 radio does not remain powered during hibernation/suspend and does not drain power at this point. This does mean that on each suspend/hibernation the handheld will need to re-authenticate with the server upon wake-up. The driver for the 802.11 radio is loaded by default upon boot and puts the radio in an “on” state. This simply means that the radio is ready should a connection need to be made. In this state the radio is not drawing much power and will decrease connection times as it is already loaded and ready to authenticate when ready. However, once the radio is on and actively connected to an access point, the larger power draw begins. The 802.11 radio will have its highest power draw when it is actively searching for a connection/access point and is sending or receiving data. Therefore, limiting connections and pings based on usage needs and requirements is a good battery management technique.

The WWAN radios function differently in that they remain on and ready for a connection throughout suspend or hibernation. Once the WWAN radio is turned on, it will remain on until turned off via software. Much like the 802.11 radio, being on and ready for a connection does not draw much power, and decreases the connection time to a network. Both the WWAN and 802.11 radios require setup connection parameters in order to try to establish connection. Once these parameters are entered into the handheld, they are not lost unless a handheld is cold-booted or they are removed via software. Having each radio parameter loaded when the radios are on and available for a connection will speed up connection times to their respective networks. In the case of a seamless roaming application between networks, parameter loading allows more efficient switching between radio types. It is important to note that the radios are not able to remain connected at the same time; this would also severely impact battery life.

## **Memory**

Built-in storage (main memory) comes in two versions: ROM and RAM. When working with a handheld device it is important to understand both types of memory and how they can impact an application. It is also important to note the amounts of each of these pieces of memory and how much is used out of the box. These amounts will vary by handheld model, specifications, and operating system.

ROM (flash memory – non volatile): ROM is short for read-only memory. This refers to a memory chip that permanently stores instructions and data without requiring power. ROM will always contain the main files to run your handheld computer, cold or clean booting (depending on your handheld) will return your handheld to

these base settings. Most of the contents of ROM are written onto the chip at the time of manufacturing and can not be changed or altered without a “reflash.” The ROM of Intermec terminals come built with the operating system and Intermec peripheral drivers/programs directly installed. In the case of some devices (32mb ROM) this will take up your entire ROM chip, however, in larger devices (64mb ROM) there is some area left over for additional use. In order to alter any Intermec content on the device, you must replace the contents of its ROM; accomplished by flashing the operating system.

In the past, the ROM size of a handheld was not an overly important feature; in typical computer fashion, there was generally enough to handle anything you “should” want to do. However, unlike your desktop you cannot just go out and buy a larger hard drive for your programs. ROM size is important to take into account if you are looking at potentially upgrading the operating system in your device at some time (if available). Each year handheld computers have become more common and popular, and each year the market is asking more and more from handheld devices. In order to meet the growing demand for greater performance, the operating systems have become larger to accommodate these new features.

RAM (main memory – volatile): RAM is short for random access memory. This is a grouping of memory chips that hold programs and data files and function as an overall workspace. RAM is different from ROM in that it is volatile. This means that should your handheld lose all power or be cold-booted, you will lose all of its contents. Advantages of RAM are that it can be modified much easier, and its contents are more easily accessible and processed faster. However, due to the possible loss of data from RAM, a storage card is your best option for storing long term data as it is non-volatile. Keeping files that are in current use for an application, or portions of a database that are accessed frequently are excellent uses of RAM. Do not forget the volatility of RAM and the need to flush any important data (database entries) to a non-volatile form of memory (storage card) frequently.

Secure Digital Cards (SDMMC Disk – flash memory – non volatile): Secure Digital cards are lightweight, postage stamp-sized cards that use flash memory. These cards are now commonly used across many handheld devices and all new Intermec devices. SD cards present several main advantages as they can be purchased in the memory size needed for each customer situation and are non-volatile. These cards should be used to store long term program files for later reinstallation and/or cold boot persistence including program files, installation requirements, databases, and more.

## **Pocket Internet Explorer**

Pocket Internet Explorer is a full-featured pocket browser for your handheld devices. This browser comes loaded by default with Pocket PC (Windows Mobile) and does many similar operations as the full desktop browser. However, because it is smaller and consumes less resources, it will not behave with the same rigor as a desktop browser.

Pocket Internet Explorer was first introduced as a part of Windows CE 1.0 in 1996. It is important to note that PIE does not derive from Internet Explorer code and was written from scratch to be as lightweight as possible. PIE was initially developed around the time of Internet Explorer 3.0 and was a significantly scaled back version of this counterpart. Although both the desktop and pocket versions of the browsers have come a long way, it is vital to keep in mind that the PIE is not the desktop Internet Explorer.

If you are building a web based application for your handheld devices, it is important to handle the testing aspect of your application on the handheld side and not utilizing a full desktop browser. If you are purchasing a third party web based application for your mobile devices, it is important to check that it is supported on the exact operating system intended to run on the devices.

Currently supported on PPC 2003 (and higher) are: XHTML basic, Cascading Style Sheets (mobile profile) and URL Mapper with UI, HTML 4.0, Jscript v5.5, WAP 2.0 (WAP 1.2.1 + XHTML), WTLS, GDI+ Imaging, Animated GIFS and IPv6.

Additional Resources:

- \* [Internet Explorer for Pocket PC](#) – HTML and Object Model Reference
- \* [Web Browsers FAQ](#) – Compares Pocket IE 1.x – 4.0
- \* [PIE Sections](#) – Pocket PC Developer Network
- \* [PIE User's Guide](#) – HPC: Factor

### Screen Size and GUI

Many who start developing applications on a handheld think it will be the same as for a desktop, and because of this often design and build with this belief in mind. The interface of the mobile user is extremely different to that of the desktop user; so much so that testing, development, and demo use should actually be done on the device and not the desktop.

In typical desktop design you would have detailed menus on the left or right side of the screen with the user choosing from these options. What many fail to take into account is how a handheld is used; by right and left handed people with a stylus in their hand. If you create a menu on the left or right side of the screen you run the risk of the user covering the entire rest of the screen upon making this one menu selection if they hold the device with the opposite hand of that in your assumption. However, if you set your menu(s) on the bottom of the screen you are keeping the entire top of the screen open to view while the user navigates through the screens. This point becomes very obvious when you stop to think about it along with the ergonomics of your design and the handheld you are working with, but many first-time developers do not take this into consideration.

Colors and input methods are other important areas to take into account. Your desktop is set in an artificial light environment and more than likely with an improved video card, monitor, and processing environment than the handheld you are programming for. It is important when making color choices for a mobile application that you take your handheld out in natural light and conditions similar to your users so that all situations are evaluated and options weighed. Some customers have offered two lighting or color choices in their applications depending on the user environment to provide the best situation from a viewing sense. Data can be entered by using the embedded keyboard (sip), the physical keyboard (different layouts available), and/or the scanner. Each of these methods has their strengths and weaknesses and the environment that the device will be used in will impact the option(s) chosen for implementation.

It is important when designing a mobile application that the users are taken into account in every possible situation, and that the application is tested in real life situations.

## Help, Support and Assistance

### Developer Support

In a perfect world, all programmers would understand exactly how the handhelds, radios, and peripherals work and how to program for them. Everyone knows this is not the case and programming applications is not always an easy task. In order to provide our customers and VARs with the best experience possible, Intermec has a Developer Support group to help with programming questions and needs. To contact Developer Support, please use their e-mail alias: [dev\\_supt@intermec.com](mailto:dev_supt@intermec.com). It is important to include as much information about the issue(s) as possible so that they are able to get the questions/problems addressed as soon as possible. Another great stop for tips, tricks, white papers, operating systems, FAQ's, and more is the Developer Support Web site: <http://www.intermec.com/developersupport>.

Supported Products: 700 series, CK6x series, CN30, CK1, CN2x series

Additional Support: IDL, iBrowse, Connection Manager, iLaunch, MCL, WWAN Toolkit

### Customer Support and Knowledge Central (KC)

Intermec's excellent support organization is a great resource for hardware and general handheld questions.. The most common way of getting support is via the toll-free number: 1-800-755-5505 (United States 5am – 5pm Pacific Time). However, the entire organization uses a support and answer knowledge base called Knowledge Central (KC). Not only is KC a location to post and track your questions/issues, but it also houses a large knowledge base to first check any of your questions. KC accounts are free and can be setup directly on the Web site. Notifications on cases and answer updates are sent via e-mail.

### Intermec Application Software Group

Combining system studies, consultation, project management, application development, and integrated quality testing, Intermec's Software Services team is uniquely qualified to design, develop and implement a comprehensive solution to meet your company's needs. Whether automating for the first time or upgrading your current system, Intermec's proven methodologies will ensure a successful deployment.

Intermec Software Services offers substantial experience in deploying mobile and wireless computing solutions across a variety of industries. Our people, products, and processes can complement your own internal resources to ensure successful implementation.

We develop complete custom software packages built with rapid software development tools such as Intermec's Mobile Framework to get the job done. Our software solutions are developed by teams experienced in writing applications for industrial hardware platforms and take into consideration factors such as limited processing power and battery life to develop intuitive, easy-to-use applications for mobile workers with little computing experience.

It's up to you to choose just how much or how little assistance you need from us. In any case, our specially trained project managers, computing professionals and system integrators are available to coordinate and facilitate the successful implementation of your custom application.

## **Additional Tools and Utilities**

\*Note: A demonstration copy of all tools can be found inside the [Software Showcase](#).

### **Intermec Connection Manager**

The Intermec Connection Manager Utility is a utility designed to run in the background and is used in conjunction with a businesses application. It monitors available connection options, and selects the connection based on predefined priorities (Ethernet>802.11b>WWAN). It provides seamless switching between mediums so that the business application simply looks for an IP connection to host. There are configurable options through the user interface or also available programmatically.

### **iBrowse**

iBrowse is a locked-down application for the Intermec 700 Series, CK30, CK31, CK60, CN2, CN30, CV60 terminals and SF51 scanners. "Locked-down" means that the user does not have the ability to exit, key in their own URLs, or use the back button. This allows the web pages to completely define where the user is to go. iBrowse also prevents the onscreen keyboard from popping up every time an edit box gets focus. The keyboard is still available if needed, but does not pop up automatically when an edit box gets focus. You can enter a password to exit the iBrowse application, allowing an administrator to modify any necessary system settings.

iBrowse has different compatibility modes dependent on the device and operating system on which iBrowse is running. All Pocket PC terminals, Windows Mobile (Pocket PC) terminals, and the CK30 standard version are Pocket Internet Explorer (PIE) compatible. All other Windows CE devices (with the exception of the CK30 standard CE image) support the Internet Explorer 6-like functionality within iBrowse.

### **iLaunch**

iLaunch is a locked-down application for the Intermec 700 Series, CK30, CK31, CK60, CN2A, CN30, and CV60 terminals that enables the user to launch other applications via a configurable set of buttons. The label on each button and the application that button launches is configurable via an INI file. The administrator can configure up to ten programs to appear on the menu screen.

In addition to providing a common user interface to access the various applications made available to them, iLaunch also locks down all applications that start while iLaunch is running. If Pocket Internet Explorer (PIE) is one of the programs in the INI file, when the user runs PIE, it comes up and looks normal, but the user cannot bring up the Start menu when tapping in the upper-left corner of the screen. When the user taps the X in the upper-right corner, PIE is minimized and the user is back at the iLaunch screen.

### **SmartSystems**

SmartSystems Foundation is a software platform that provides a single, integrated environment for hands-free deployment and management of devices located anywhere in the enterprise – on-site or remote. Foundation works with the Intermec components of your system including mobile computers, RFID readers, printer, barcode scanners and software to make the most of limited IT resources, and lower your total cost of ownership for data collection equipment.

Intermec devices are packed with powerful features. Capabilities such as wireless communications, built-in 802.1x security, Windows Mobile, Bluetooth and full data collection support are available when you take a device out of the box. Foundation lets you view and configure all of the settings contained in a device from one convenient location for quick set-up, allows you to create a backup copy of those settings to easily pass down to any similar device, and provides easy access to the wide range of features integrated into Intermec equipment.

### **TE2000**

TE2000 will add Native Terminal Emulation functionality to your Intermec 700 Color, CK30, CV60, and Antares terminals. With TE2000 you can standardize on one terminal emulation platform with extended commands and multiple emulations to choose from (3270, 5250, and VT/ANSI). Intermec's TE2000 provides extended functionality beyond emulations. Some of the advanced command options include: selectable commands, fallback controllers, protected and unprotected fields on the screen, auto-login scripts and the ability to remap any key to any specified string.

### **WWAN Toolkit**

The WWAN Toolkit is a software product developed by Intermec specifically for developers and integrators of Intermec handheld products utilizing Wide Area Wireless (WWAN) radios. This comprehensive library enables partners to easily and quickly interface WWAN functionality into their applications without having to go through the additional expense of costly and time consuming development work of their own. The WWAN Toolkit is a software solution for a variety of markets including manufacturing, warehousing, transportation/logistics, DSD and more. The WWAN Toolkit application has the ability to initiate and receive voice, data and fax calls, control radio power and network attachments and report carrier and signal strength.

### **WWAN Communicator**

The WWAN Communicator is an application that lets you use your GSM/GPRS or CDMA/1xRTT radio-equipped 76x Color Mobile Computer like a cell phone. It is based on Intermec's WWAN Toolkit, which also includes Communicator source code. For more information; please contact your Intermec sales representative.

**North America**

**Corporate Headquarters**

6001 36th Avenue West  
Everett, Washington 98203  
Phone: (425) 348-2600  
Fax: (425) 355-9551

**South America & Mexico  
Headquarters Office**

Newport Beach, California  
Phone: (949) 955-0785  
Fax: (949) 756-8782

**Europe/Middle East &**

**Africa Headquarters Office**

Reading, United Kingdom  
Phone: +44 118 923 0800  
Fax: +44 118 923 0801

**Asia Pacific**

Headquarters Office  
Singapore  
Phone: +65 6303 2100  
Fax: +65 6303 2199

**Internet**

[www.intermec.com](http://www.intermec.com)  
Worldwide Locations:  
[www.intermec.com/locations](http://www.intermec.com/locations)

**Sales**

Toll Free NA: (800) 934-3163  
Toll in NA: (425) 348-2726  
Freephone ROW: 00 800 4488 8844  
Toll ROW: +44 134 435 0296

**OEM Sales**

Phone: (425) 348-2762

**Media Sales**

Phone: (513) 874-5882

**Customer Service and Support**

Toll Free NA: (800) 755-5505  
Toll in NA: (425) 356-1799

Copyright © 2007 Intermec Technologies Corporation. All rights reserved.  
Intermec is a registered trademark of Intermec Technologies Corporation. All other  
trademarks are the property of their respective owners. Printed in the U.S.A.  
611743-01A 03/07



In a continuing effort to improve our products, Intermec Technologies Corporation  
reserves the right to change specifications and features without prior notice.